AntiLeakBench: Preventing Data Contamination by Automatically Constructing Benchmarks with Updated Real-World Knowledge

Xiaobao Wu[1,2*] **Liangming Pan**[5] **Yuxi Xie**[3,2*] **Ruiwen Zhou**[4,2*] **Shuai Zhao**[1]
Yubo Ma[1] **Mingzhe Du**[1,3] **Rui Mao**[1] **Anh Tuan Luu**[1] **William Yang Wang**[2]
[1]Nanyang Technological University [2]University of California, Santa Barbara
[3]National University of Singapore [4]Shanghai Jiao Tong University [5] University of Arizona
xiaobao002@e.ntu.edu.sg william@cs.ucsb.edu

Abstract

Data contamination hinders fair LLM evaluation by introducing test data into newer models' training sets. Existing studies solve this challenge by updating benchmarks with newly collected data. However, they fail to guarantee contamination-free evaluation as the newly collected data may contain pre-existing knowledge, and their benchmark updates rely on intensive human labor. To address these issues, we in this paper propose AntiLeakBench, an automated anti-leakage benchmarking framework. Instead of simply using newly collected data, we construct samples with explicitly new knowledge absent from LLMs' training sets, which thus ensures strictly contamination-free evaluation. We further design a fully automated workflow to build and update our benchmark without human labor. This significantly reduces the cost of benchmark maintenance to accommodate emerging LLMs. Through extensive experiments, we highlight that data contamination likely exists before LLMs' cutoff time and demonstrate that AntiLeakBench effectively overcomes this challenge. [1]

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated prominent capabilities in multiple fields (Radford et al., 2019; Touvron et al., 2023). To thoroughly assess these capabilities, various benchmarks have been developed, such as MMLU (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021). They are typically static and publicly accessible, serving as standardized tools to evaluate LLMs' performance. Unfortunately, the static nature of these benchmarks presents a significant challenge: **Data Contamination**, where their test data may end up in newer LLMs' training sets. This issue can inflate model performance

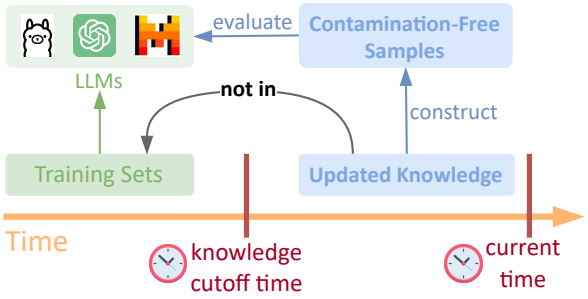

Figure 1: Illustration of AntiLeakBench. It constructs contamination-free samples based on the knowledge updated after LLMs' cutoff time, which thus are not in LLMs' training sets.

and thus undermine the reliability and validity of these benchmarks (Golchin and Surdeanu, 2023b; Roberts et al., 2023; Deng et al., 2024; Dong et al., 2024; Jiang et al., 2024). To avoid data contamination, recent studies dynamically update the benchmarks by collecting new data released after LLM's knowledge cutoff time (Kiela et al., 2021; Potts et al., 2021; Kasai et al., 2023; Jain et al., 2024). For instance, LiveBench (White et al., 2024) collects newly released questions from math exams and code platforms like LeetCode.

However, despite their prevalence, we emphasize these benchmarks encounter two limitations: **(i) Weak guarantee for contamination-free evaluation.** They ignore to verify whether the newly collected data contain truly new knowledge (Roberts et al., 2023; White et al., 2024; Jain et al., 2024). For example, exam questions or coding problems from LeetCode may be later republished or referenced. As a result, the knowledge of these data may overlap with LLMs' training, which potentially leads to data contamination. **(ii) High dependency on human labor for maintenance.** Their benchmark updates often require intensive human labor, such as annotating the collected data (Gu et al., 2024; Xu et al., 2024). In consequence, this significantly hinders their frequent maintenance,

[*]Work done during visiting at UCSB.
[1]Our code and data are available at https://github.com/bobxwu/AntiLeakBench.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18403–18419

especially in light of the rapid emergence of new LLMs. For instance, RealTimeQA (Kasai et al., 2023) and KoLA (Yu et al., 2023) have barely been updated recently. Together these limitations undermine the reliability and practicality of existing benchmarks for contamination-free evaluation.

To address these limitations, we in this paper propose **AntiLeakBench**, an automated anti-leakage benchmarking framework to prevent data contamination. As illustrated in Figure 1, rather than directly collecting newly released data as previously, we identify new real-world knowledge updated after LLM's cutoff time. Then we construct question-answering samples querying these updated knowledge, accompanied by their corresponding real-world supporting documents. This ensures the updated knowledge is absent from LLMs' training sets, and thus the constructed samples on them are **strictly contamination-free**.

Furthermore, we design **a fully automated workflow** to build and update AntiLeakBench. It eliminates the need for human labor, enabling the benchmark to be seamlessly updated to accommodate emerging LLMs. As such, this significantly reduces the maintenance cost of our benchmark, enhancing its practicality and scalability.

We evaluate a series of LLMs on our AntiLeakBench with samples before and after the cutoff time. We observe that their performance commonly drops after the cutoff time. This trend highlights the likely data contamination in LLM evaluation. The experimental results additionally manifest the effectiveness of AntiLeakBench for contamination-free evaluation. The contributions of this paper can be concluded as follows:

- We propose AntiLeakBench, an automated anti-leakage benchmarking framework, ensuring contamination-free evaluation by constructing test samples with updated real-world knowledge.
- We propose an automated building workflow that automatically builds and updates AntiLeakBench without human labor, enabling to easily accommodate emerging new LLMs.
- We conduct extensive experiments involving various LLMs on multiple tasks and demonstrate the effectiveness of AntiLeakBench for contamination-free evaluation.

2 Related Work

Data Contamination Many benchmarks have been widely used to assess the impressive capabilities of LLMs across various tasks, like question-answering, reading comprehension, and math reasoning (Pan et al., 2023, 2024; Zhou et al., 2024; Wu, 2025; Zhao et al., 2025). Notable examples include ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), BIG-bench (Srivastava et al., 2022), and GSM8K (Cobbe et al., 2021). Although they have played a significant role, their static nature may cause the data contamination issue (Magar and Schwartz, 2022; Yang et al., 2023; Golchin and Surdeanu, 2023a; Jacovi et al., 2023; Li, 2023; Oren et al., 2023; Sainz et al., 2023; Ni et al., 2024). Due to this, some work discloses several benchmarks are gradually less effective (Schaeffer, 2023; Zhou et al., 2023). For example, evidence shows some LLMs have overfitted to the GSM8K, compromising its validity (Zhang et al., 2024).

Contamination-Free Evaluation To achieve contamination-free evaluation, recent studies dynamically update benchmarks by collecting new data (Zhu et al., 2023; Thrush et al., 2022; Qian et al., 2024; Srivastava et al., 2024). For instance, RealTimeQA (Kasai et al., 2023) periodically collects multi-choice quizzes from newspapers. Similarly, LiveCodeBench (Jain et al., 2024) frequently crawls programming questions from code platforms like LeetCode. LiveBench (White et al., 2024) extends them by covering more domains, such as math competitions, research papers, and news articles. They cannot guarantee contamination-free evaluation since they rely solely on the newly released data and also need intensive human labor for construction (Liska et al., 2022; Mousavi et al., 2024). Recent ADU (Ying et al., 2024) updates existing benchmarks by paraphrasing data through LLMs, but it may risk introducing mistakes and biases into evaluation. Different from these studies, our AntiLeakBench constructs samples with newly updated real-world knowledge to ensure contamination-free evaluation. It also introduces a fully automated building workflow without human labor. These differences make AntiLeakBench a more reliable and practical benchmarking framework for consistent contamination-free evaluation.

3 AntiLeakBench

In this section, we present how to automatically build AntiLeakBench with updated knowledge. Figure 2 illustrates the building workflow. We

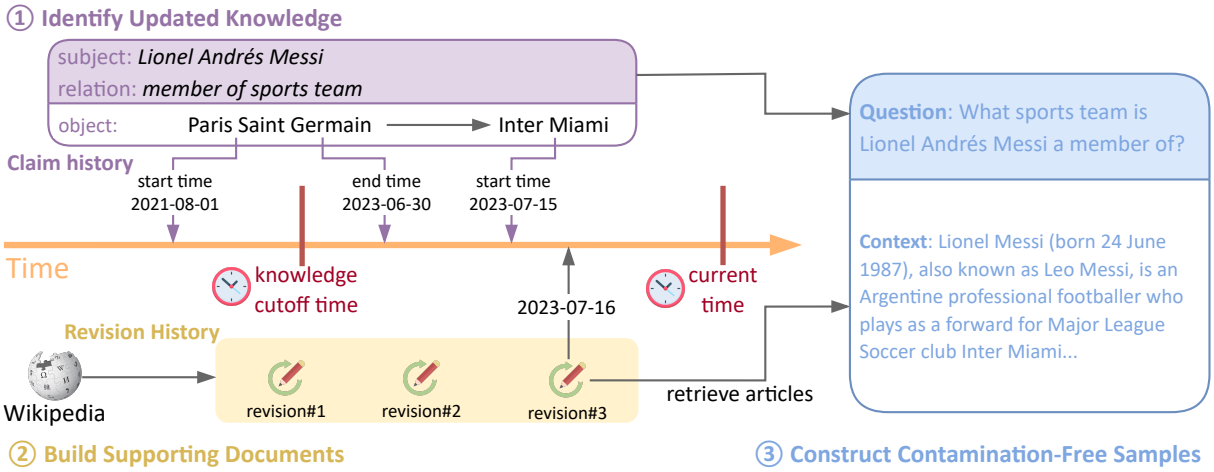

Figure 2: Illustration of the automated benchmark building workflow without human labor. After data preparation, it includes three main steps: (1) Identify updated knowledge after the cutoff time; (2) Build supporting documents; (3) Construct contamination-free samples (Figure 3 exemplifies how to construct multi-hop samples).

mainly focus on question-answering tasks regarding the updated knowledge for evaluation. This is because they allow precise control over the knowledge being evaluated, thus ensuring contamination-free evaluation. In contrast, using other tasks like summarization and code generation poses two challenges: (**i**) Identifying the specific knowledge embedded in them is inherently complex, making strict contamination-free unfeasible. (**ii**) Their intricate context structures make it difficult to automatically synthesize new high-quality samples without introducing errors and inconsistencies.

3.1 Preparing Data

We begin by preparing the data to build the benchmarks. For the knowledge source, we leverage Wikidata (Vrandečić and Krötzsch, 2014), a widely used knowledge base that is frequently updated by contributor communities. Wikidata provides an extensive repository of real-world factual claims involving numerous entities. Each claim is represented as a triplet *(subject, relation, object)*, such as *(Lionel Andrés Messi, member of sports team, Paris Saint Germain)*.

We sample a subset of relations associated with physical entities, such as *member of sports team*; we exclude the less meaningful relations, for example, ones about virtual entities, like *geometry coordinates*, similar to Zhong et al. (2023); Wu et al. (2024f) (See details in Appendix A). For each claim, we extract two qualifiers from Wikidata: *start time* and *end time*, which specifies when the claim starts and ends, *e.g.,* the timeline of a football player in a team. We use these prepared data to build and update our AntiLeakBench.

3.2 Identifying Updated Knowledge

We then identify updated knowledge based on the prepared data. Considering t_1 as the knowledge cutoff time of an LLM and t_2 as the current time, our goal is to find the updated knowledge that occurs after t_1 and before t_2. For this purpose, we figure out the history of claims. Specifically, we group all the claims by their subject and relation and sort the claims in each group chronologically based on their start time. If a new claim emerges after the cutoff time t_1 in the history, *i.e.,* the object changes, we identify the new claim as the updated knowledge. For instance, in Figure 2 the object of (*Lionel Andrés Messi*; *member of sports team*) shifts after the cutoff time: *Paris Saint Germain* → *Inter Miami*. From this shift, we extract updated knowledge with the new claim: *(Lionel Andrés Messi; member of sports team; Inter Miami)*. We emphasize that the LLM is unaware of this knowledge because it occurs after its cutoff time.

One may wonder what if the object changes twice and eventually reverts to its original one, like a player returning to his previous team. To exclude this case, we additionally confirm that the new claim is different from the old one and then consider it as updated knowledge.

3.3 Building Supporting Documents

To provide context for the updated knowledge, we build supporting documents from real-world sources. While LLMs could generate such documents, this may introduce mistakes or biases as

LLMs often hallucinate or misinterpret information (White et al., 2024). To maintain accuracy and reliability, we rely on the well-maintained and widely trusted Wikipedia as the source of supporting documents. This choice is further justified since the updates of Wikidata commonly follow the updates of Wikipedia (Vrandečić and Krötzsch, 2014).

In detail, we denote the identified updated knowledge as a claim (s_1, r_1, o_1) and retrieve the Wikipedia page revision history of either its subject s_1 or object o_1, determined by its relation, *e.g.,* subject *Lionel Andrés Messi* for relation *member of sports team* in Figure 2. Then we find the revision made after the start time of the updated knowledge. With this revision, we retrieve its corresponding article from Wikipedia and check if the summary of the article contains both the subject and object (or their aliases). If true, we consider this article as a supporting document for the updated knowledge (We show their validity in Sec. 3.7). For example, Figure 2 illustrates a Wikipedia article indicating *Lionel Andrés Messi* is a member of *Inter Miami*. Here the supporting document is revised after LLMs' cutoff time, so it is also nonexistent in their training sets.

3.4 Constructing Contamination-Free Samples

Now we construct test samples querying the above updated knowledge (s_1, r_1, o_1), with its supporting document as context, denoted as D. Since the knowledge and the supporting document are both absent from LLMs' training sets, the constructed test samples for them are strictly contamination-free. This is different from previous studies: they overlook verifying if the inherent knowledge of their samples does not exist in LLM's training sets.

Tasks We mainly focus on question-answering tasks, including both single-hop and multi-hop questions. Following the common practice (Yang et al., 2018; Kočiskỳ et al., 2018; Bai et al., 2024), each sample is denoted as (Q, C, A) with question Q, context C, and expected answer A.

- **Single-Hop**. In this task, we directly ask what is the answer to the updated knowledge by providing the supporting document. We first consider **Single-Hop Gold**: Question Q is formulated with predefined templates based on the relation in the updated knowledge, Context C only includes the supporting document D, and Answer A is the object o_1 and its aliases. For instance, in Figure 2 the question is *What sports team is Lionel Andrés Messi a member of?*; the context is *Lionel Andrés Messi*'s Wikipedia article; the answer is the name and aliases of his sports team.

  To enhance the difficulty, we further introduce a new task: **Single-Hop** N_d: Question Q remains the same, and we augment Context C with N_d distracting documents. Here these distracting documents are randomly sampled from other samples' supporting documents which do not contain the subject s_1 and object o_1 of the updated knowledge. As such, this task further assesses the long-context capability of LLMs, specifically locating relevant information within distractors.

- **Multi-Hop**. Moreover, we construct multi-hop questions for harder reasoning. We first design **Multi-Hop Gold**. Specifically, starting with the updated knowledge (s_1, r_1, o_1), we build a chain of H connecting claims: $((s_1, r_1, o_1), (s_2, r_2, o_2), \ldots, (s_H, r_H, o_H))$. Here the object of the i-th triplet serves as the subject of the $(i{+}1)$-th triplet, *i.e.,* $o_i = s_{i+1}$. As such, Question Q is formulated across this chain, Context C includes the supporting document of each knowledge in this chain, and Answer A is the last object o_H. For instance, Figure 2 shows a multi-hop question *Who is the coach of the sports team that Lionel Andrés Messi is a member of?*; the context is *Lionel Andrés Messi*'s

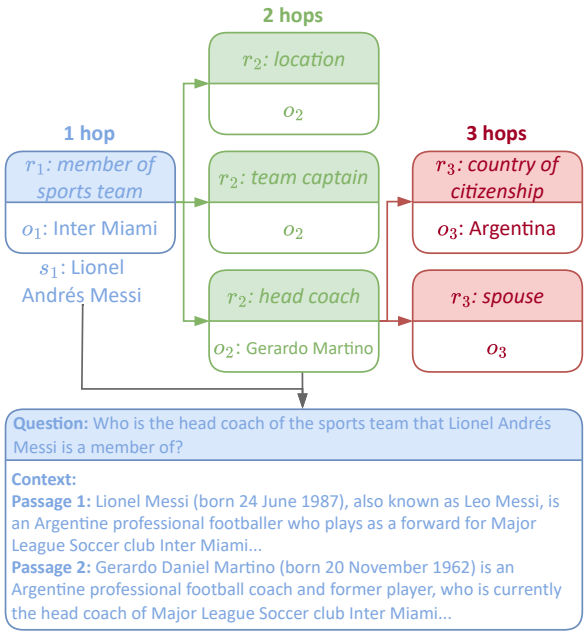

Figure 3: Illustration of constructing multi-hop samples by finding the consequent relations of previous objects.

Benchmark	Strictly Contamination-Free	Automated	Multilingual	Data Source
Realtime QA(Kasai et al., 2023)	✗	✗	✗	Real world
LiveBench(White et al., 2024)	✗	✗	✗	Real world
ADU(Ying et al., 2024)	✗	✓	✗	LLM generation
AntiLeakBench	✓	✓	✓	Real world

Table 1: Comparisons between AntiLeakBench and other benchmarking frameworks.

and *Gerardo Martino*'s Wikipedia articles; the answer is the last object *Gerardo Martino*.

Similarly, we also consider **Multi-Hop** N_d: the context additionally covers N_d randomly sampled supporting documents of other samples as the distracting documents. This task assesses the multi-hop reasoning ability of LLMs, requiring them to connect and integrate information across a long context with distractors.

Question Formats We consider two common question formats.

- **Generation.** Question Q solely is the question as aforementioned.
- **Multi-Choice.** In this format, we additionally prompt LLMs with 4 options and ask them to select one. We design the 4 options as follows: (**a**) **Correct option**, *i.e.,* the correct answer to the question. (**b**) **Unknown option**, represented as a string "Unknown". (**c**) **Outdated option**. The old answer before the cutoff time, *e.g., Paris Saint Germain*) in Figure 2. (**d**) **Noise option**. A randomly sampled, unrelated answer from other samples. For multi-hop questions, we ignore finding outdated options and instead provide two noise options to accelerate the building workflow.

The above introduces the automated workflow to build our AntiLeakBench. Table 2 shows an example of Single-Hop Gold, and more examples are in Appendix D.

3.5 Benchmark Maintenance

Our AntiLeakBench supports easy maintenance. We only need to download the latest Wikidata dump and then execute our automated workflow to update benchmarks. The whole process requires no human labor, and hence we can effortlessly maintain the benchmarks for newer LLMs.

3.6 Multilingual Benchmarks

Moreover, AntiLeakBench features multilingual evaluation. It can seamlessly produce samples in various languages via our automated workflow with the multilingual nature of both Wikidata and Wikipedia. This enables us to evaluate LLMs in various linguistic contexts. See more details in Appendix A.

Attributes	Examples
question (generation)	What sports team is Lionel Andrés Messi a member of?
answer (generation)	Inter Miami CF Inter Miami Club Internacional de Fútbol Miami
question (multi-choice)	What sports team is Lionel Andrés Messi a member of? A. Inter Miami CF B. Paris Saint-Germain F.C. C. Prime Minister of Romania D. Unknown.
answer (multi-choice)	A
subject	Lionel Messi Lionel Andres Messi Lionel Andrés Messi
pid	P54 (member of sports team)
object	Inter Miami CF Inter Miami Club Internacional de Fútbol Miami
object_old	Paris Saint-Germain F.C. Paris Saint-Germain Football Club Paris Saint-Germain FC
context	Lionel Andrés Messi (; born 24 June 1987), also known as Leo Messi, is an Argentine professional footballer who plays as a forward for Major League Soccer club Inter Miami...

Table 2: An example from AntiLeakBench.

Quality Metrics	Single-Hop Gold	Multi-Hop Gold
Context Accuracy	97.3	98.7
Answer Accuracy	96.7	97.3

Table 3: Data quality by human verification.

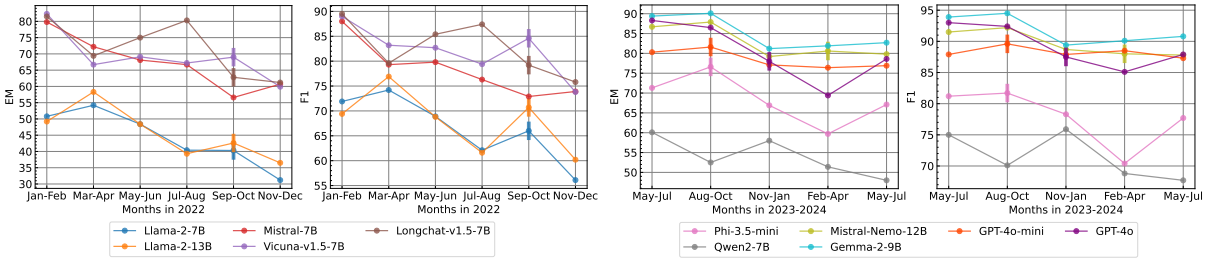

Figure 4: EM and F1 performance at each time interval. Marker | denotes LLM's cutoff time.

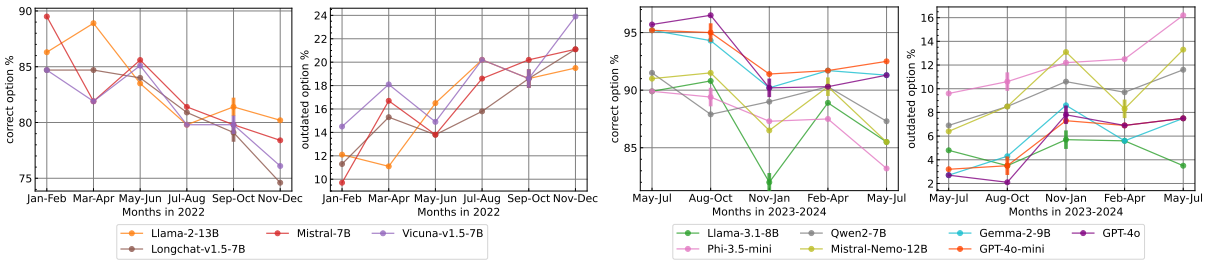

Figure 5: Correct and outdated option proportions at each time interval. Marker | denotes LLM's cutoff time.

3.7 Data Quality Verification

To verify the data quality of produced samples in AntiLeakBench, we conduct human verifications. We ask human annotators to evaluate the accuracy of both answers and contexts in the samples. See Appendix B for the verification procedure and agreement analysis. Table 3 shows that AntiLeakBench achieves a high standard of data quality with question and context accuracy over 96%.

3.8 Comparisons with Existing Benchmarks

Table 1 compares our AntiLeakBench with other benchmarks. We highlight its vital advantages: (**i**) **Strictly contamination-free**. AntiLeakBench ensures that the constructed samples must cover updated knowledge absent from LLMs' training sets, which guarantees strictly contamination-free evaluation. (**ii**) **Automated workflow**. Our fully automated building workflow avoids human labor, which significantly reduces the cost of maintaining the benchmark for newer LLMs. In consequence, this fortifies its adaptivity and long-term applicability. (**iii**) **Multilingual**. Our method supports the construction of multilingual samples. This enables us to extensively assess LLMs' capabilities across different languages. (**iv**) **Real-world data**. We build test samples from real-world data sources like Wikipedia, rather than LLM-generated content. This grounds the benchmark in practical and authentic data. With these advantages, AntiLeakBench serves as a reliable testbed for evaluating LLMs in a strictly contamination-free environment.

4 Experiment

In this section, we experiment with different LLMs on our AntiLeakBench to show its effectiveness and investigate data contamination.

4.1 Experiment Setup

Large Language Models We experiment with the following 12 common language models: Llama-2-7B (Touvron et al., 2023), Llama-2-13B (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023), Vicuna-v1.5-7B (Chiang et al., 2023), LongChat-v1.5-7B (Li* et al., 2023), Llama-3.1-8B (Dubey et al., 2024), Phi-3.5-mini (Abdin et al., 2024), Qwen-2-7B (Yang et al., 2024), Mistral-Nemo-12B (Jiang et al., 2023), Gemma-2-9B (Team, 2024). We also consider proprietary models: GPT-4o-mini and GPT-4o (Achiam et al., 2023). Table 8 summarizes their release and knowledge cutoff time. We use the prompts in appendix C following Bai et al. (2024), which **explicitly ask LLMs to use the provided context to answer questions.**

Constructing Test Samples To investigate the impact of data contamination, we on purpose construct two kinds of test samples: (**i**) **Pre-cutoff samples**, containing knowledge before the cutoff time. These samples may already exist in the LLMs' training sets. (**ii**) **Post-cutoff samples**, containing knowledge updated after the cutoff time. These samples are absent from LLMs' training sets.

Language Models	Single-Hop								Multi-Hop								Avg	
	Gold		N_d=3		N_d=5		N_d=7		Gold		N_d=3		N_d=5		N_d=7			
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Llama-2-7B	40.6	63.5	16.8	41.2	11.6	30.9	9.4	24.5	33.6	50.2	**19.4**	**32.2**	**15.8**	**28.1**	**12.2**	**22.7**	19.9	36.7
Llama-2-13B	42.7	65.3	14.0	40.6	9.4	30.6	7.0	24.0	13.3	34.6	4.1	21.5	2.7	17.8	2.3	15.2	11.9	31.2
Mistral-7B	65.4	77.2	27.8	41.3	16.7	27.3	7.3	15.3	21.4	27.9	11.5	17.2	8.1	14.3	6.5	11.1	20.6	29.0
Vicuna-v1.5-7B	66.8	79.9	39.1	60.4	25.8	48.3	15.3	39.1	26.0	43.5	11.1	22.9	8.1	19.5	5.4	15.7	24.7	41.2
Longchat-v1.5-7B	**75.5**	**84.5**	**58.2**	**72.8**	**47.6**	**65.5**	**37.0**	**56.3**	**38.8**	**51.4**	17.6	30.6	12.0	25.8	4.7	3.9	**36.4**	**48.9**
Llama-3.1-8B	19.2	66.2	21.4	59.4	18.1	53.5	14.2	45.7	24.4	50.2	11.7	33.0	9.4	27.5	6.8	21.9	15.6	44.7
Phi-3.5-mini	69.0	78.7	34.0	40.5	26.5	33.7	15.2	22.2	45.4	59.7	20.8	29.5	14.9	21.1	9.8	14.4	29.4	37.5
Qwen-2-7B	54.8	72.4	15.5	38.5	9.8	26.6	7.2	21.2	35.9	48.3	23.7	33.4	18.1	26.1	13.6	20.1	22.3	35.8
Mistral-Nemo-12B	82.7	89.7	75.6	83.8	66.3	75.1	51.8	62.2	57.7	67.3	39.1	47.7	33.8	41.4	24.0	29.0	53.9	62.0
Gemma-2-9B	**85.0**	**91.6**	80.2	86.2	68.8	75.2	55.4	61.2	**82.7**	**86.4**	63.0	68.3	55.8	61.2	49.0	53.5	67.5	73.0
GPT-4o-mini	78.5	88.1	80.3	89.2	79.1	88.1	79.2	88.5	68.8	83.1	60.5	75.3	57.1	73.1	54.2	70.6	69.7	82.0
GPT-4o	81.2	89.5	**84.1**	**90.8**	**83.5**	**90.3**	**84.8**	**91.4**	71.5	85.9	**71.9**	**86.1**	**70.2**	**84.8**	**70.2**	**84.8**	**77.2**	**87.9**

Table 4: EM (Exact Match) and F1 results in the **generation** format on AntiLeakBench. Gold means only gold documents; N_d is the number of distracting documents. The best is in **bold**.

To construct these samples, we establish two time periods according to the LLMs' cutoff time summarized in Table 8: (**i**) From 2022-01-01 to 2023-01-01, divided into 2-month intervals. This targets the first 5 models since their knowledge cutoff time mostly falls around Sep 2022, such as Vicuna-v1.5-7B. (**ii**) From 2023-05-01 to 2024-08-01, divided into 3-month intervals. Similarly, this is tailored for the last 7 models whose cutoff time mostly falls around the end of 2023, *e.g.,* Llama-3.1-8B and GPT-4o. Here we divide these periods into shorter intervals because (i) this can ensure each interval contains a sufficient number of samples, enabling statistically meaningful analysis, and (ii) this provides detailed granular insights into how contamination risks and model performance evolve over time. We report the statistics of the produced samples of our benchmark in Tables 6 and 7. See more building details in Appendix A.

Evaluation Metrics For the generation format (Sec. 3.4), we use **EM** (Extract Matching) and token-based **F1** scores, following the standard practice in question answering (Rajpurkar et al., 2016). For the multi-choice format, we use **Acc** (Accuracy) and **F1** scores following Kasai et al. (2023).

4.2 Data Contamination Analysis

We first analyze the impact of data contamination by looking into the performance trends of LLMs. Figure 4 presents the trends of EM and F1 scores under the Single-Hop Gold task in the generation format (Llama-3.1-8B is excluded due to its low EM; See Sec. 4.3). We observe a general performance decline for most LLMs after their knowledge cutoff time, although all samples are constructed in the same way. For instance, Vicuna-v1.5-7B's EM and F1 start dropping near its cutoff time Sep 2022; Similarly, GPT-4o-mini remains stable until its cutoff time Oct 2023 and decreases thereafter. According to this decline, we conclude two key findings:

(1) Pre-cutoff samples come with data contamination, which inflates LLMs' performance. Pre-cutoff samples may overlap with LLMs' training sets. This allows LLMs to correctly handle these samples based on their prior knowledge rather than actually understanding the provided context. As a result, evaluating models solely on pre-cutoff samples can overestimate their capabilities.

(2) Contamination-free post-cutoff samples are more challenging and can more accurately assess LLMs. Post-cutoff samples are free from contamination since they include knowledge updated after LLMs' cutoff time. To deal with these samples, LLMs must demonstrate true comprehension and reasoning over the given context and questions, as they cannot rely on prior knowledge alone. As such, evaluating with post-cutoff samples more accurately reflects LLMs' abilities.

Interestingly, we notice that some LLMs experience performance drops even before the cutoff time. This is probably because their training data cover less knowledge close to the cutoff time. Knowledge sources such as news articles and Wikipedia pages usually become widely documented only after initial events occur.

Language Models	Single-Hop								Multi-Hop								Avg	
	Gold		N_d=3		N_d=5		N_d=7		Gold		N_d=3		N_d=5		N_d=7			
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
Llama-2-7B	41.7	30.7	3.7	5.6	3.5	5.3	2.8	5.4	18.7	30.9	6.8	9.9	5.6	8.1	3.6	6.9	10.8	12.9
Llama-2-13B	**82.1**	**82.2**	73.7	73.6	60.1	59.9	51.7	51.3	**97.5**	**97.5**	**88.5**	**88.5**	**82.8**	**83.1**	75.2	75.2	76.5	76.4
Mistral-7B	81.8	81.8	65.9	65.8	58.3	58.2	52.3	52.3	88.7	88.6	77.2	77.2	72.7	72.8	67.7	67.2	70.6	70.5
Vicuna-v1.5-7B	80.1	80.0	**75.6**	**75.4**	**73.1**	**72.9**	**69.6**	**69.4**	96.8	96.9	84.0	84.2	82.6	83.0	**77.0**	**77.2**	**79.8**	**79.9**
Longchat-v1.5-7B	79.6	79.7	68.5	68.8	65.1	51.8	62.3	61.2	93.2	93.4	76.7	78.0	70.4	71.5	66.6	68.0	72.8	71.6
Llama-3.1-8B	86.7	90.4	62.2	74.0	48.9	62.9	37.8	52.9	70.5	81.4	50.7	64.8	40.9	56.2	30.8	44.9	53.6	65.9
Phi-3.5-mini	87.4	87.5	85.6	85.8	84.7	85.4	79.6	82.5	96.5	97.0	85.3	86.2	78.0	80.3	68.6	72.3	83.2	84.6
Qwen-2-7B	89.1	39.7	83.0	27.9	78.2	24.6	77.0	78.5	97.6	98.3	94.5	54.2	92.4	46.4	91.5	91.7	87.9	57.7
Mistral-Nemo-12B	88.5	71.1	88.8	71.8	84.7	70.2	77.8	83.8	91.1	94.6	77.1	68.4	69.9	64.0	43.1	58.7	77.6	72.8
Gemma-2-9B	92.4	92.4	86.7	86.5	76.9	61.6	69.4	69.3	97.1	97.1	88.3	88.3	81.8	65.4	77.4	77.4	83.8	79.8
GPT-4o-mini	**93.2**	**93.2**	**93.8**	**93.8**	93.3	93.3	93.5	93.5	**98.5**	**98.5**	**96.4**	**96.4**	**95.4**	**95.4**	93.5	93.5	**94.7**	**94.7**
GPT-4o	92.8	92.8	93.5	93.5	**94.0**	**94.0**	**94.0**	**94.0**	97.9	97.9	95.8	95.8	**95.4**	**95.4**	**93.9**	**93.9**	94.7	94.7

Table 5: Acc and F1 results in the **multi-choice** format on AntiLeakBench. Gold means only gold documents; N_d is the number of distracting documents. The best is in **bold**.

Besides, Figure 5 plots the proportions of correct and outdated options selected under the Single-Hop Gold task in the multi-choice format (Llama-2-7B is excluded due to its low performance; See Sec. 4.3). Notably LLMs increasingly favor outdated options over correct ones (See option types in Sec. 3.4). For example, Mistral-Nemo-7B's proportion of selecting correct options decreases from 91.0% to 85.5%, while the proportion of outdated options rises from 6.4% to 13.3%. These results again validate that pre-cutoff samples, due to data contamination, could inflate the performance.

In summary, the above results demonstrate the effectiveness of our AntiLeakBench, which effectively ensures contamination-free evaluation and serves as a more reliable assessment of LLMs.

4.3 Overall Performance Analysis

Next we analyze the overall performance of LLMs. Tables 4 and 5 summarize the average results over all samples across tasks in the generation and multi-choice format, respectively. According to these results, we have the following observations.

(1) AntiLeakBench poses a significant challenge for LLMs. Table 4 shows that most models score EM and F1 below 50, indicating a substantial gap between their capabilities and the benchmark's requirements. Only two models, GPT-4o-mini and GPT-4o, reach EM and F1 scores around 70 and 80. These results highlight the challenging nature of our AntiLeakBench for evaluating LLMs.

(2) Proprietary models lead in performance. Tables 4 and 5 reveal that proprietary models surpass open-source ones by a large margin. For instance, GPT-4o achieves the highest average EM and F1 scores, 77.2 and 87.9, respectively, while the runner-up open-source model only has 53.9 and 62.0. Additionally, GPT-4o's performance remains relatively stable despite the increasing number of distracting documents. This substantial margin may be attributed to their longer max context length, superior model architectures, and larger parameter sizes.

4.4 Difficulty Analysis

We further analyze the difficulty concerning tasks and question formats based on Tables 4 and 5. We summarize the key findings as follows.

(1) Multi-choice format is significantly easier. LLMs generally perform better in the multi-choice format compared to the generation format. For example, GPT-4o-mini achieves Acc and F1 scores over 90, surpassing its performance in the generation format. This is because the multi-choice format simplifies the tasks by providing explicit answer options, enabling LLMs to identify correct answers with partial comprehension.

(2) Longer contexts bring about higher difficulty. LLMs' performance gradually decreases along with more distracting documents, such as from N_d=3 to 7 in Table 4. The reason is that long contexts distract LLMs from locating relevant information. This underscores the necessity of LLMs' stronger ability to handle long contexts.

(3) Multi-hop tasks are more challenging. Compared to single-hop ones, LLMs' performance be-

comes lower on multi-hop tasks. For example, in Table 4 the EM score of Longchat-v1.5-7B decreases from 75.5 on the Single-Hop Gold to 38.8 on the Multi-Hop Gold; GPT-4o-mini decreases from 78.5 to 68.8. These multi-hop tasks require LLMs to decompose complex questions and reason across long and interconnected contexts.

5 Conclusion

In this paper, we propose AntiLeakBench, an automated anti-leakage benchmarking framework. Unlike solely collecting newly released data as before, it constructs test samples based on identified updated real-world knowledge, ensuring strictly contamination-free evaluation. It also introduces a fully automated building workflow without human labor. This enables frequent and efficient benchmark updates to accommodate emerging LLMs, greatly simplifying maintenance. These advantages establish AntiLeakBench as an ideal testbed for contamination-free evaluation, providing accurate and fair assessments for LLMs.

Limitations

Our benchmarking framework can ensure strictly contamination-free evaluation with a fully automated building workflow, but we believe there are some limitations to be explored as future work:

- To implement the automated building workflow, we mainly consider the question-answering tasks for evaluation. Although we have devised different task difficulty levels, more diverse tasks can be explored in the future, which will more extensively evaluate LLMs in a contamination-free manner. We think the challenges lie in how to identify the knowledge embedded in these tasks and how to automatically construct new high-quality samples.
- Our benchmarking framework uses Wikidata and Wikipedia as data sources. While these sources are extensive and frequently updated by their contributor communities, they may contain incorrect information in rare cases. As discussed in Sec. 3.7, most produced samples are correct as verified by humans.

Ethics Statement

In this paper, we build and maintain our AntiLeakBench with Wikidata and Wikipedia as the data sources. We acknowledge that Wikidata and Wikipedia may contain inaccurate information in a few cases as they are extremely abundant and rely on human labor for maintenance. Besides, our work focuses on real-world commonsense knowledge by sampling relations with physical entities (See Appendix A). This can avoid harmful information in most situations.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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. 2024. 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.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8698–8711.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shahriar Golchin and Mihai Surdeanu. 2023a. Data contamination quiz: A tool to detect and estimate contamination in large language models. *arXiv preprint arXiv:2311.06233*.

Shahriar Golchin and Mihai Surdeanu. 2023b. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, et al. 2024. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18099–18107.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5084.

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. *arXiv preprint arXiv:2403.07974*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059*.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime qa: what's the answer right now? In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 49025–49043.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Dacheng Li*, Rulin Shao*, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How long can open-source llms truly promise on context length?

Yucheng Li. 2023. An open source data contamination report for llama series models. *arXiv preprint arXiv:2310.17589*.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165.

Seyed Mahed Mousavi, Simone Alghisi, and Giuseppe Riccardi. 2024. Dyknow:dynamically verifying time-sensitive factual knowledge in llms. *arXiv preprint arXiv:2404.08700*.

Shiwen Ni, Xiangtao Kong, Chengming Li, Xiping Hu, Ruifeng Xu, Jia Zhu, and Min Yang. 2024. Training on the benchmark is not all you need. *arXiv preprint arXiv:2409.01790*.

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*.

Fengjun Pan, Xiaobao Wu, Zongrui Li, and Anh Tuan Luu. 2024. Are LLMs good zero-shot fallacy classifiers? In *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, pages 14338–14364, Miami, Florida, USA. Association for Computational Linguistics.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. Dynasent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. 2023. To the cutoff... and beyond? a longitudinal perspective on llm data contamination. In *The Twelfth International Conference on Learning Representations*.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need. *arXiv preprint arXiv:2309.08632*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Saurabh Srivastava, Anto PV, Shashank Menon, Ajay Sukumar, Alan Philipose, Stevin Prince, Sooraj Thomas, et al. 2024. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *arXiv preprint arXiv:2402.19450*.

Gemma Team. 2024. Gemma. *blog*.

Tristan Thrush, Kushal Tirumala, Anmol Gupta, Max Bartolo, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams, and Douwe Kiela. 2022. Dynatask: A framework for creating dynamic ai benchmark tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 174–181.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, et al. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*.

Xiaobao Wu. 2025. Sailing ai by the stars: A survey of learning from rewards in post-training and test-time scaling of large language models. *arXiv preprint arXiv:2505.02686*.

Xiaobao Wu, Xinshuai Dong, Thong Nguyen, Chaoqun Liu, Liang-Ming Pan, and Anh Tuan Luu. 2023. Infoctm: A mutual information maximization perspective of cross-lingual topic modeling. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pages 13763–13771.

Xiaobao Wu, Xinshuai Dong, Liangming Pan, Thong Nguyen, and Anh Tuan Luu. 2024a. Modeling dynamic topics in chain-free fashion by evolution-tracking contrastive learning and unassociated word exclusion. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3088–3105, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xiaobao Wu, Chunping Li, Yan Zhu, and Yishu Miao. 2020. Short text topic modeling with topic distribution quantization and negative sampling decoder. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1772–1782, Online.

Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. 2022. Mitigating data sparsity for short text topic modeling by topic-semantic contrastive learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2748–2760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024b. A survey on neural topic models: Methods, applications, and challenges. *Artificial Intelligence Review*.

Xiaobao Wu, Thong Thanh Nguyen, Delvin Ce Zhang, William Yang Wang, and Anh Tuan Luu. 2024c. FASTopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xiaobao Wu, Fengjun Pan, and Anh Tuan Luu. 2024d. Towards the TopMost: A topic modeling system toolkit. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 31–41, Bangkok, Thailand. Association for Computational Linguistics.

Xiaobao Wu, Fengjun Pan, Thong Nguyen, Yichao Feng, Chaoqun Liu, Cong-Duy Nguyen, and Anh Tuan Luu. 2024e. On the affinity, rationality, and diversity of hierarchical topic modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024f. AKEW: Assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15118–15133, Miami, Florida, USA. Association for Computational Linguistics.

Zhikun Xu, Yinghui Li, Ruixue Ding, Xinyu Wang, Boli Chen, Yong Jiang, Xiaodong Deng, Jianxin Ma, Hai-Tao Zheng, Wenlian Lu, et al. 2024. Let llms take on the latest challenges! a chinese dynamic question answering benchmark. *arXiv preprint arXiv:2402.19248*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

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*. Association for Computational Linguistics.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. 2024. Automating dataset updates towards reliable and timely evaluation of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. Conference on Neural Information Processing Systems, NeurIPS.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023. Kola: Carefully benchmarking world knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, et al. 2024. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Anh Tuan Luu. 2025. A survey of recent backdoor attacks and defenses in large language models. *Transactions on Machine Learning Research (TMLR)*.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

Ruiwen Zhou, Wenyue Hua, Liangming Pan, Sitao Cheng, Xiaobao Wu, En Yu, and William Yang Wang. 2024. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. *arXiv preprint arXiv:2412.08972*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023. Dyval: Graph-informed dynamic evaluation of large language models. *arXiv e-prints*, pages arXiv–2309.

Time period	Single-Hop				Multi-Hop			
	Gold	N_d=3	N_d=5	N_d=7	Gold	N_d=3	N_d=5	N_d=7
2022-01-01 to 2023-01-01	1090	1089	1088	1088	443	443	443	443
2023-05-01 to 2024-08-01	819	818	818	818	941	939	939	939

Table 6: Sample sizes in the constructed AntiLeakBench in the experiments.

Time period	Single-Hop				Multi-Hop			
	Gold	N_d=3	N_d=5	N_d=7	Gold	N_d=3	N_d=5	N_d=7
2022-01-01 to 2023-01-01	5998	23163	33867	46033	24646	40611	50846	61761
2023-05-01 to 2024-08-01	7210	27501	40800	54451	25505	43926	53898	66957

Table 7: Average word counts of samples in the constructed AntiLeakBench in the experiments.

A Building Workflow Details

We use the Wikidata dump released on 2024-08-05 as our data source. To sample relations from Wikidata, we browse the relation list from Wikidata (Vrandečić and Krötzsch, 2014) [2] and manually select common relations associated with physical entities and exclude the less meaningful ones, such as those related to virtual entries (*e.g.,* geographic coordinates and IMDB ID). The sampled relations mainly focus on common topics, such as sports, politics, and entertainment (Wu et al., 2020, 2022, 2023, 2024a,e,d,b,c). We predefine the question templates for each sampled relation, for instance, *What sports team is a member of?* for the relation *member of sports team*. For more details, see the configuration files in our code. We follow simple-wikidata-db [3] and extract the claims of sampled relations, qualifiers, aliases, and Wikipedia titles from the dump. To find updated knowledge, we combine the claims and their *start time* and *end time* qualifiers and then sort them by *start time*. We check if the object changes after the preset cutoff time and identify updated knowledge if it changes, for instance, Messi's sports team changed in Figure 2.

Given an entity, we employ the MediaWiki APIs [4] and use its extracted Wikipedia title to retrieve its Wikipedia article and revision history. Note that our workflow supports building multilingual benchmarks. We only need to prepare the configuration files in another language and then execute the workflow by specifying that language. The workflow automatically retrieves corresponding Wikipedia articles in that language.

The statistics of produced benchmarks are reported in Tables 6 and 7.

[2] https://www.wikidata.org/wiki/Wikidata:Database_reports/List_of_properties/all

[3] https://github.com/neelguha/simple-wikidata-db

[4] https://www.mediawiki.org/wiki/MediaWiki

Model	Release time	Knowledge cutoff time
Llama-2-7B	2023-07	2022-09
Llama-2-13B	2023-07	2022-09
Mistral-7B	2023-09	2022*
Vicuna-v1.5-7B	2023-07	2022-09
Longchat-v1.5-7B	2023-07	2022-09
Llama-3.1-8B	2024-07	2023-12
Phi-3.5-mini	2024-08	2023-10
Qwen-2-7B	2024-06	2023*
Mistral-Nemo-12B	2024-07	2024-04
Gemma-2-9B	2024-08	2024-06*
GPT-4o-mini	2024-07	2023-10
GPT-4o	2024-07	2023-12

Table 8: Release time and knowledge cutoff time of LLMs. * means estimated time since some LLMs do not disclose their cutoff time.

Quality Metrics	Single-Hop Gold		Multi-Hop Gold	
	Raw Agreement	Gwet's AC1	Raw Agreement	Gwet's AC1
Context Accuracy	0.98	0.99	0.97	0.98
Answer Accuracy	0.97	0.98	0.96	0.97

Table 9: Annotation agreement analysis.

B Data Quality Verification

We recruit three annotators (graduate students) to verify the data quality of our benchmark. Specifically, we provide 100 samples of Single-Hop Gold and 100 samples of Multi-Hop Gold; then we ask annotators to determine (i) if the provided context

can sufficiently answer the question; (ii) if the given answer is accurate and can appropriately address the question given the context. Table 3 summarizes the average accuracy scores concerning Single-Hop Gold and Multi-Hop Gold.

Table 9 reports the inter-annotator agreement analysis. We mainly use two agreement coefficients: Raw Agreement and Gwet's AC1. This is because the annotations are extremely unbalanced (most of them are positive); other coefficients like Krippendorff's Alpha and Fleiss' Coefficient may be inappropriate (Gwet, 2008). It shows that the annotations maintain high agreement.

C Prompts

Following Bai et al. (2024), we use the following prompts for generation and multi-choice question formats mentioned in Sec. 3.4.

```
You are given an article and a question. Answer the
question based on the given article as concisely as
you can, using a single phrase or sentence if
possible. Do not provide any explanation.

Article: <context>

Question: <question>

Answer:
```

```
You are given an article, a question, and four
options. Select one option to answer the question
based on the given article. Only give the option
(A, B, C, or D), and do not output any other words.

Article: <context>

Question: <question>
<options>

Answer:
```

D Examples of AntiLeakBench

We list some examples in the AntiLeakBench.

Examples of Single-Hop Gold


```
[
    {
         "question": "What sports team is Duncan Cowan Ferguson a coach of?",
         "answer": [
             "Inverness Caledonian Thistle F.C.",
             "Inverness Caledonian Thistle Football Club",
             "Inverness Caledonian Thistle FC",
             "Inverness Caledonian Thistle",
             "Inverness",
             "ICTFC",
             "Caley Thistle"
         ],
         "context": "Duncan Cowan Ferguson (born 27 December 1971) is a Scottish football coach and former
player who is the manager of Scottish Championship club Inverness Caledonian Thistle..."
    },
    {
        "question": "What is the position held by Leo Docherty?",
        "answer": [
            "Minister of State for the Armed Forces"
        ],
        "context": "Leo Docherty (born 4 October 1976) is a British politician serving as Minister of State
for the Armed Forces since 26 March 2024..."
    },
    {
        "question": "What sports team is Keiren Westwood a member of?",
        "answer": [
            "Queens Park Rangers F.C.",
            "Queens Park Rangers Football Club",
            "Queens Park Rangers FC",
            "Queens Park Rangers",
            "The Hoops",
            "The Rs",
            "QPRFC"
        ],
        "context": "Keiren Westwood (born 23 October 1984) is a professional footballer who plays as a
goalkeeper for Queens Park Rangers. Born in England, he plays international football for the Republic of
Ireland..."
    }
]
```

Examples of Multi-Hop Gold


```
[
    {
        "question": "What is the headquarter of the sports team that Kevin Luckassen is a member of?",
        "answer": [
            "Arad",
            "Arad, Romania"
        ],
        "context": [
            "Passage 1: Kevin Luckassen (born 27 July 1993) is a Dutch professional footballer who plays as
a forward for Romanian Liga I club UTA Arad...",
            "Passage 2: Asociatia Fotbal Club UTA Arad (), commonly known as UTA Arad or simply UTA (Uzina
Textila Arad (\"Textiles Factory of Arad\")), is a Romanian professional football club based in the city of
Arad, Romania, which competes in the Liga I..."
        ]
    },
    {
        "question": "Who is the spouse of the officeholder of President of Finland?",
        "answer": [
            "Suzanne Innes-Stubb",
            "Suzanne Innes",
            "Suzanne Stubb",
            "Suzanne Elizabeth Innes-Stubb",
            "Suzanne Elizabeth Innes"
        ],
        "context": [
            "Passage 1: Cai-Goran Alexander Stubb (born 1 April 1968) is the 13th and current President of
Finland, having won the 2024 presidential election. He previously served as Prime Minister of Finland from
2014 to 2015...",
            "Passage 2: Suzanne Elizabeth Innes-Stubb (born 25 January 1970) is a British-Finnish attorney
and the wife of Alexander Stubb, President of Finland. She was the first person of overseas origin to
become the spouse of the President of Finland..."
        ]
    },
    {
        "question": "What is the country of citizenship of the head coach of the sports team that Marquez
Reshard Valdes-Scantling is a member of?",
        "answer": [
            "United States of America",
            "United States",
            "American",
            "Americans"
        ],
        "context": [
            "Passage1: Marquez Reshard Valdes-Scantling (born October 10, 1994) is an American football
wide receiver for the Kansas City Chiefs of the National Football League (NFL). He played college football
at NC State and South Florida, and was drafted by the Packers in the fifth round of the 2018 NFL Draft.",
            "Passage 2: Andrew Walter Reid (born March 19, 1958) is an American football coach who is the
head coach for the Kansas City Chiefs of the National Football League (NFL). Reid was previously the head
coach of the Philadelphia Eagles, a position he held from 1999 to 2012. From 2001 to 2012, he was also the
Eagles' executive vice president of football operations, making him the team's general manager. He is the
only NFL coach to win 100 games and appear in four consecutive conference championships with two different
franchises."
        ]
    }
]
```