A Survey on Data Contamination for Large Language Models

Anonymous ACL submission

Abstract

Recent advancements in Large Language Models (LLMs) have demonstrated significant progress in various areas, such as text generation and code synthesis. However, the reliability of performance evaluation has come under scrutiny due to data contamination—the unintended overlap between training and test datasets. This overlap has the potential to artificially inflate model performance, as LLMs are typically trained on extensive datasets scraped from publicly available sources. These datasets often inadvertently overlap with the benchmarks used for evaluation, leading to an overestimation of the models' true generalization capabilities. In this paper, we first examine the definition and impacts of data contamination. Secondly, we review methods for contamination-free evaluation, focusing on three strategies: data updating-based methods, data rewriting-based methods, and prevention-based methods. Specifically, we highlight dynamic benchmarks and LLM-driven evaluation methods. Finally, we categorize contamination detecting methods based on model information dependency: white-Box, gray-Box, and black-Box detection approaches. Our survey highlights the requirements for more rigorous evaluation protocols and proposes future directions for addressing data contamination challenges.

1 Introduction

Recent breakthroughs in Large Language Models (LLMs) have demonstrated remarkable capabilities in text generation, code synthesis, and mathematical reasoning (Zhao et al., 2023; OpenAI et al., 2024; DeepSeek-AI et al., 2025). However, the reliability of LLM evaluation is increasingly questioned due to data contamination-the unintended overlap between training and test data sets (Balloccu et al., 2024; Chang et al., 2024). This is especially problematic as LLMs use large web-scraped datasets that are prone to overlap with testing benchmarks. LLMs are known to memorize portions of their training data, and under certain prompts, they can reproduce this data verbatim (Carlini et al., 2022). As highlighted by (Sainz et al., 2023), a critical consequence of data contamination is that scientific studies relying on contaminated LLMs may produce erroneous conclusions, potentially invalidating valid hypotheses. Furthermore, Ippolito et al. (2023) demonstrated that contaminated models can inadvertently align with copyright-protected content post hoc, posing significant challenges for the responsible development of LLMs. To underscore the importance of addressing data contamination in both the development and evaluation of LLMs, we present a comprehensive review of data contamination issues in this paper.

In section 2, we define data contamination as the inclusion of data from the testing set during the pre-training phase, which artificially inflates model performance. Recent studies extend this definition along two dimensions: phase-based contamination in LLMs' lifecycle and benchmark-based contamination in LLMs' evaluation. For phase-based analysis, contamination mechanisms include pre-training phase leakage, fine-tuning biases, cross-modal leakage (Yao et al., 2024), and indirect human interactions (Palavalli et al., 2024). Meanwhile, benchmark-based contamination operates at two granularities: instance-level contamination, and dataset-level contamination. Dataset-level contamination is categorized by severity into simple rewriting, label leakage, text leakage, and dual text-label leakage. The impacts are discussed in the following four areas: collecting evidence, factors discussion, non-contamination scenarios, and quantifying contamination.

In section 3, we discuss how to achieve contamination-free evaluation. For static benchmarks, current research focuses on three key contamination-free strategies: automatically up-

dating datasets using the most recent data, rewriting existing data, and implementing proactive risk prevention mechanisms. Meanwhile, dynamic evaluation frameworks(Zhu et al., 2024a; Lei et al., 2024; Zhang et al., 2024e; Ying et al., 2024) generate test samples using techniques like combinatorial optimization, graph-based reasoning, and controlled randomization, creating an evolving evaluation system. Additionally, the LLM-as-a-Evaluator paradigm(Bai et al., 2024) turns LLMs into meta-evaluators, enabling intelligent assessments independent of static benchmarks.

In section 4, we explore methodologies for detecting data contamination in LLMs. We categorize data contamination detection approaches into three distinct paradigms: white-box detection, which relies on full access to model architectures or training data to achieve high precision, employing techniques such as N-gram overlap (Brown et al., 2020) or embedding similarity (Reimers, 2019); gray-box detection, which leverages partial model information, such as token probabilities, to identify contamination; and black-box detection, which operates without access to internal model details, relying instead on heuristic rules (the details are outlined in Appendix B). Together, these approaches illustrate the evolving and multifaceted landscape of data contamination detection methods, each offering unique advantages and challenges.

The organization of this paper is as follows, as shown in figure 1. In Section 2, we discuss existing work on the definition and impacts of data contamination. Section 3 summarizes current methods for constructing contamination-free datasets and dynamic evaluation approaches. Section 4 discusses how to detect data contamination. Finally, in Section 5, we present several significant future challenges in this area.

2 What is Data Contamination

2.1 Definition

In recent years, a growing body of research has emerged to address the issue of data contamination in LLMs. However, the field lacks a unified definition or standardized methodology to comprehensively summarize data contamination. Brown et al. (2020) was among the first to highlight pre-training data contamination, employing an N-gram diagnostic method to demonstrate how contamination artificially inflates model performance. Hartmann et al. (2023) further explored the connection between LLM memorization and data contamination, noting that both phenomena involve the regurgitation of pre-training data. Schwarzschild et al. (2024) proposed that strings can be considered memorized if they can be reproduced using a shorter prompt, while Karamolegkou et al. (2023) investigated verbatim memorization, particularly in the context of copyrighted materials. Building on these foundational studies, our research extends the framework into two significant directions: (1) examining vulnerabilities across the entire lifecycle of LLMs, including pre-training, fine-tuning, and post-deployment contamination, and (2) addressing risks to benchmark integrity, such as data manipulation and potential label leakage.

2.1.1 Phase-based Contamination

For phase-based contamination, recent research has identified stage-specific contamination risks throughout the lifecycle of LLMs: pre-training (where test data may leak into training corpora), fine-tuning (where models are unintentionally exposed to evaluation data), and post-deployment (where models absorb biases from real-world interactions). Sainz et al. (2023) systematically mapped contamination pathways across these critical phases, while Balloccu et al. (2024) introduced the concept of indirect data contamination, highlighting how human interactions during LLM training can inadvertently introduce biases, even in the absence of explicit test data inclusion. Furthermore, multimodal large language models (MLLMs) face heightened contamination challenges due to the integration of diverse data modalities (Yin et al., 2023). Song et al. (2024) proposed a bimodal taxonomy, distinguishing between unimodal contamination and cross-modal contamination, and developed traceability frameworks tailored specifically for MLLMs.

2.1.2 Benchmark-based Contamination

For benchmark-based contamination, prior research has generally approached the issue from two primary perspectives. The first focuses on whether labels are leaked or whether samples are rewritten, while the second categorizes contamination at either the instance level or the dataset level. Yang et al. (2023) considered even simple rewording—such as synonym substitution or translation—as a form of contamination. Yao et al. (2024) further revealed cross-language contamination through the detection of option rewriting. In

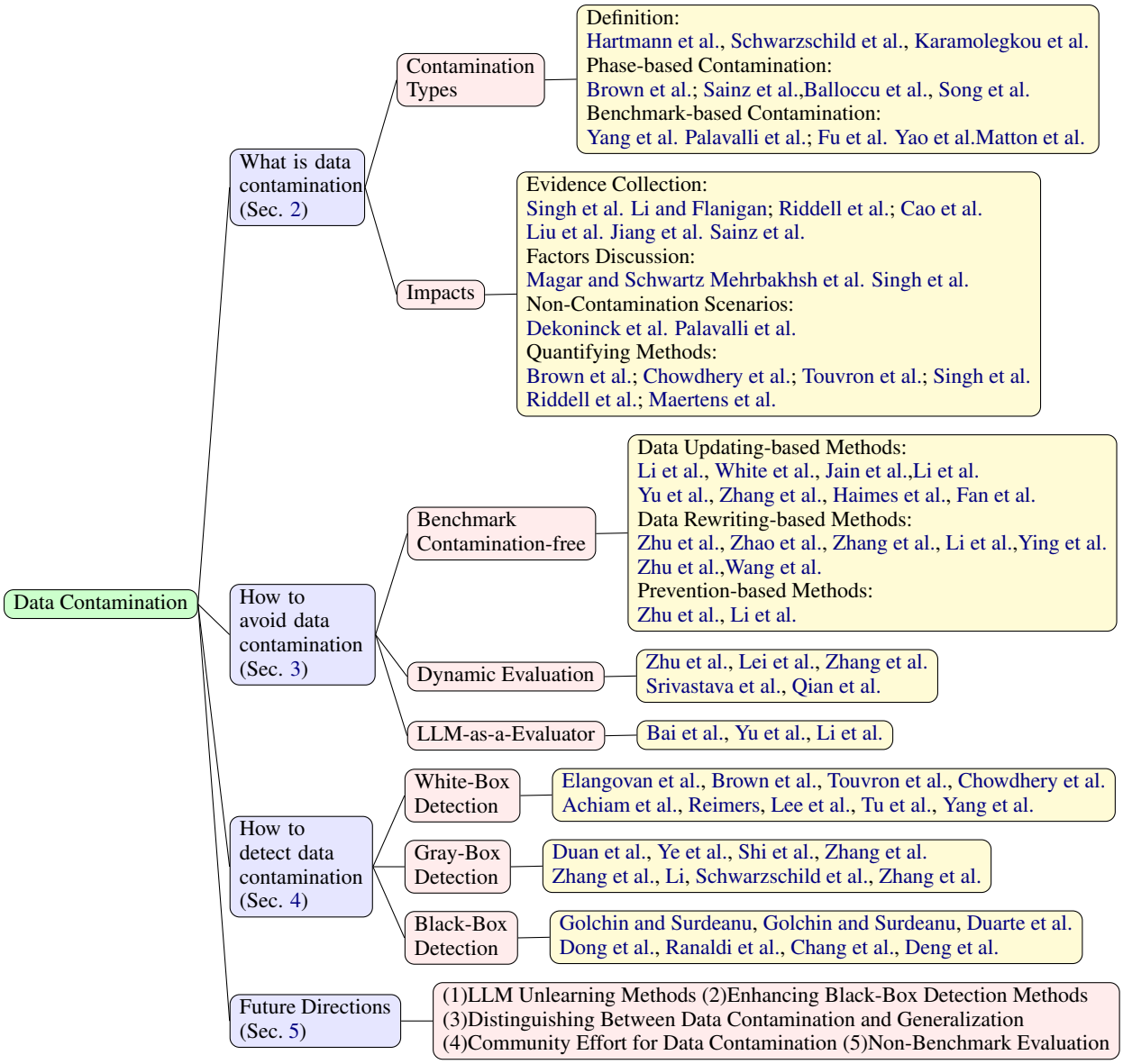

Figure 1: Structure of this paper

code generation tasks, Palavalli et al. (2024) established a systematic taxonomy, categorizing contamination into dataset-level (e.g., test data leakage or mixing) and instance-level (e.g., output masking, input/output rewriting, or augmentation). Expanding on this, Matton et al. (2024) identified three distinct sources of contamination and proposed the LBPP benchmark as a countermeasure. Additionally, Fu et al. (2024) provided formal mathematical definitions for contamination at both the instance and dataset levels, defining instance-level contamination through membership inference attacks and dataset-level contamination as either full or partial contamination.

2.2 Impacts

Data contamination critically undermines evaluation reliability and research validity. As (Sainz et al., 2023) demonstrated, benchmark overfitting can artificially inflate model performance and compromise scientific conclusions in NLP studies. (Singh et al., 2024) identified two principal analysis approaches: causal analysis through controlled retraining experiments, and post-hoc contamination inference via performance pattern examination without model retraining.

2.2.1 Evidence Collection

Initial contamination investigation focuses on temporal data analysis and adversarial detection methods. Li and Flanigan (2024) proposed evaluating

models on pre/post-training datasets with membership inference attacks, revealing contamination effects on zero/few-shot performance. Riddell et al. (2024) demonstrated performance inflation on seen HumanEval/MBPP samples, while Cao et al. (2024) validated contamination mitigation strategy through using the most recent benchmarks. Jiang et al. (2024) differentiated between text contamination (input samples) and true contamination (input-output pairs). Liu et al. (2024) exposed Chinese LLMs' superficial knowledge despite broad training exposure. Sainz et al. (2023) highlighted that current evidence on contamination remains fragmented across publications and informal channels, suggesting that the prevalence of contamination may be significantly underestimated.

2.2.2 Factors Discussion

In this section, we discuss some factors influencing contamination. Magar and Schwartz (2022) found that exploitation of contaminated data is influenced by factors like model size, learning rate, and the position of contaminated data, suggesting that memorization does not always lead to exploitation. Mehrbakhsh et al. (2024) designed GPT-4-generated templates to investigate how the complexity of test instances influences the contamination in Llama-2 7B, aiming to better understand how varying levels of difficulty and diversity in the templates can influence the model's performance. Singh et al. (2024) proposed a new contamination evaluation protocol, ConTAM, to explore how data contamination affects the evaluation results of LLMs, and provided a method to quantify the impact of contamination.

2.2.3 Non-Contamination Scenarios

In this section, we explore non-contamination scenarios, where the overlap between training and testing data does not lead to performance improvement. Dekoninck et al. (2024) established a causal relationship between model performance improvement and data contamination, explicitly defining cases where such overlap exists but does not enhance performance as non-contamination. Furthermore, Palavalli et al. (2024) clarified several phenomena that improve performance on downstream tasks without being influenced by contamination. These include language understanding, prior task understanding, and transductive learning. These phenomena enhance empirical results while preserving the integrity of both the task and the model, distinguishing them from contamination-related performance gains.

2.2.4 Quantifying contamination

Contamination scoring mechanisms classify evaluation samples through threshold-based indices. We have summarized some common contamination detection methods in table 1. For instance, Brown et al. (2020) used N-grams to evaluate contamination by checking whether each token in the tested sample appears in an n-gram from the pre-training corpus. Chowdhery et al. (2023) calculated the contamination score based on the proportion of contaminated n-grams. In contrast, Touvron et al. (2023) introduced a method to align extensions between the testing samples and pre-training corpus, allowing mismatches in certain token positions using a "skip_budget" hyperparameter. Singh et al. (2024) further extended this method, focusing on the longest contaminated token span rather than all potential matches. Riddell et al. (2024) employed the Dolos toolkit (Maertens et al., 2022) to measure semantic similarity by converting programs into abstract syntax trees (ASTs) and performing k-gram matching.

3 How to Avoid Data Contamination

This section discusses methods to avoid data contamination in evaluation. First, to reduce risks, benchmarks are often constructed following three strategies: Data updating-based methods, Data rewriting-based methods, and prevention-based methods. Second, Dynamic evaluation generates adaptive samples using techniques like algorithmic composition, graph structures, randomization, and reasoning graphs, ensuring controlled complexity and diversity. Finally, LLM-as-a-evaluator eliminates contamination risks, making it a key for contamination-free evaluation.

3.1 Benchmark Contamination-free Strategies

Contamination-free benchmarking strategies ensure datasets stay up-to-date, preventing models from using outdated data. Rewriting construction combines human efforts like manual labeling with LLM-assisted techniques such as rephrasing to avoid contamination. Preventive measures involve technical defenses like encryption, access control, and de-contamination during inference to guarantee the reliability and fairness of LLM evaluation.

3.1.1 Data Updating-based Methods

Using the most recent data is intuitive for constructing contamination-free benchmarks, and some studies have proposed automatically collecting recent data to build questions. LatestEval proposed an automated pipeline to dynamically generate contamination-free test sets from recent materials (Li et al., 2024d). White et al. (2024) introduced LiveBench, a dynamically updated benchmark that integrates tasks across math, coding, and reasoning with automated scoring to mitigate data contamination and evaluation biases. Similarly, Jain et al. (2024) developed LiveCodeBench, a code-specific benchmark that expands beyond HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) by assessing self-repair and prediction abilities while ensuring periodic updates. To evaluate LLMs' world knowledge, Yu et al. (2023) introduced the KoLA benchmark, which combines stable knowledge sources (e.g., Wikipedia) with recent data to balance evaluation fairness and contamination prevention. Zhang et al. (2024d) introduced PatentMIA, crawling Chinese patent data from Google Patents. This dataset contains 5,000 patents with a publication date after March 1, 2024, and 5,000 patents published before January 1, 2023. Haimes et al. (2024) proposed to use retro-holdout datasets to detect public benchmark influence on model training and measure discrepancies between benchmark results and real-world performance. Fan et al. (2024) introduced NPHardEval4V-a dynamically updated benchmark to assess reasoning capabilities of MLLMs. In code evaluation, EvoCodeBench is proposed to dynamically align with real-world code repositories to guarantee fair evaluation.

3.1.2 Data Rewriting-based Methods

This type of methods use data augmentation to remove contamination from benchmarks, with LLMs' superior rephrasing and verifying capabilities. Human intervention is also integrated into the rewriting process to create novel data with a distribution similar to the original data. Zhu et al. (2024d) proposed Clean-Eval to purify contaminated benchmarks by paraphrasing and back-translating data into semantically equivalent but lexically distinct forms. Zhao et al. (2024) proposed the MMLU-CF dataset, which is constructed by collecting diverse questions, cleaning data, sampling difficulty reasonably, checking data integrity with LLMs, and applying rewriting methods such as rephrasing questions and shuffling options to ensure the dataset remains contamination-free. Zhang et al. (2024a) provided GSM1k, employing manual labeling, three-tier quality control, and leak prevention design to avoid data contamination. Through meticulous human construction, GSM1k achieves high similarity to GSM8k(Cobbe et al., 2021) in style, difficulty, and human solve rates, while maintaining complete content independence. Meanwhile, LLMs can serve as assistants for rewriting or generating questions. CLEVA is generated by non-repetitive sampling for each evaluation round. Each test sample is further enhanced with multiple data rewriting strategies before being used to assess LLMs, significantly mitigating the risk of data contamination (Li et al., 2023). Ying et al. (2024) updated benchmarks with two strategies: style-preserving mimicry with LLMs and cognitive-level expansion using Bloom's taxonomy. Similarly, Zhu et al. (2024b) proposed Multi-Principle Assessment (MPA), which utilizes LLM-based agents to automatically transform existing questions into new ones. Wang et al. (2024) introduced a multi-agent framework to implement self-evolving benchmarks, which dynamically mutates question contexts and structures to update benchmarks.

3.1.3 Prevention-based Methods

Preventive measures focus on safeguarding test data integrity through technical and procedural controls. Core strategies include encrypting public test data with public-key cryptography, enforcing strict access permissions, and prohibiting derivative data creation. Zhu et al. (2024c) introduced Inference-Time Decontamination (ITD), a novel technique that identifies and rewrites potentially memorized responses during model inference. Li et al. (2024c) introduced C^2LEVA, a comprehensive bilingual benchmark with systematic contamination prevention mechanisms, which implements proactive measures such as test data rotation and enhanced encryption.

3.2 Dynamic Evaluation

Dynamic approaches address data contamination by leveraging adaptive assessment frameworks. Zhu et al. (2024a) introduced DYVAL, a graph-based system that generates evaluation samples through algorithmic composition, constraint application, and functional descriptions. Its directed acyclic graph (DAG) architecture facilitates multi-step reasoning tasks with precisely controlled complexity. Lei et al. (2024) developed S3EVAL, a

framework for SQL evaluation that utilizes randomized table-query pairs. This synthetic approach allows for customizable task lengths and difficulty levels, while systematically assessing long-context reasoning capabilities. Zhang et al. (2024e) proposed the DARG method, which dynamically generates evaluation samples with adjustable complexity and diversity using adaptive reasoning graphs. Srivastava et al. (2024) introduced functionalization, a technique that transforms static question-answer pairs into parameterized code, enabling the generation of infinite test variants. Qian et al. (2024) further extended dynamic evaluation by perturbing key variables in questions, allowing for the dynamic generation of datasets with controlled variations.

3.3 LLM-as-a-Evaluator

Next-generation evaluation leverages LLMs themselves as assessment tools. LLMs are no longer just "artisans" of content generation; they have become "judges" of content quality. They can serve the roles of scoring, ranking, and selection. Bai et al. (2024) presented the "LM-as-Examiner" framework, generating questions and evaluating responses through reference-free analysis. Yu et al. (2024) deployed LLMs as "Interactors" in structured multi-turn dialogues that probe model capabilities while minimizing contamination risks. Li et al. (2024b) proposed TreeEval-a benchmark-free system where LLMs generate hierarchical question trees. This adaptive approach adjusts difficulty based on model performance, creating unique assessment paths that prevent data contamination.

4 How to Detect Data Contamination

The definition of data contamination detection refers to the process of determining, through a specific methodology, whether a given text or dataset has been included in the training corpus of a particular model. As LLMs continue to advance, data contamination detection has emerged as a critical challenge in model evaluation. Here, we categorize detection approaches into three paradigms based on the level of access to model information: white-box, gray-box, and black-box methods. This taxonomy highlights an evolving detection landscape. White-box methods, which leverage full access to model architectures or training data, offer high precision but are often limited in applicability. Gray-box approaches, which utilize partial model information, strike a balance between practicality and effectiveness. Black-box technologies rely on heuristic assumptions (detailed in Appendix B.2) and operate without access to internal model details. As specialized detection methods continue to emerge, the research community is increasingly recognizing the importance of data contamination to distort evaluation outcomes. To support this growing awareness, we provide detailed descriptions of several contamination detection tools in Appendix C.

4.1 White-Box Detection

White-box methods directly utilize model internals or training data to detect data contamination. When pre-training corpora are accessible, content overlap with evaluation datasets can be explicitly measured (Elangovan et al., 2021). Prominent LLMs including LLaMA2 (Touvron et al., 2023), PaLM (Chowdhery et al., 2023), and GPT-4 (Achiam et al., 2023) all emphasize the necessity of detecting pre-training/evaluation overlaps. The n-gram overlap method, prioritized for its computational efficiency and simplicity, has become a standard tool for detecting contamination. Comparative implementations of these n-gram based overlap detection strategies are systematically summarized in Table 1.

Embeddings similarity compares texts via cosine similarity of their embeddings, capturing semantic relationships beyond lexical variations (Reimers, 2019). Lee et al. (2023) used a similarity exclusion method based on embeddings, reducing dataset redundancy and filtering out duplicate data to ensure clean training data. To address sophisticated contamination forms, (Yang et al., 2023) introduced a hybrid approach combining embedding similarity search with GPT-4 powered semantic analysis. This detects paraphrased samples, enabling proactive benchmark decontamination.

For known model weights, Tu et al. (2024) proposed DICE to identify in-distribution contamination during fine-tuning by analyzing layer-specific activation patterns. This method trains contamination classifiers on sensitive intermediate layers, demonstrating a strong correlation between detection signals and performance inflation across multiple LLMs.

4.2 Gray-Box Detection

Gray-box approaches in membership inference attacks (MIAs) leverage partial model information

Model	Author	Method	Token-Level
GPT3	(Brown et al., 2020)	n-gram (n=13)	×
Palm	(Chowdhery et al., 2023)	n-gram	✓
Llama2	(Touvron et al., 2023)	extended n-gram	✓
GPT4	(OpenAI et al., 2024)	n-gram (n=50)	×
Phi-4	(Abdin et al., 2024)	hybrid n-gram	×

Table 1: N-gram method used for contamination detection, Token-level refers to the standard for measuring contamination scores using tokens.

such as token probabilities to distinguish training data from non-members. Duan et al. (2024) systematically investigated the underwhelming MIA performance on LLMs, identifying three primary contributing factors: the massive scale of training datasets that complicates memorization patterns, the limited number of training iterations that reduce model overfitting, and the inherently fuzzy decision boundaries between member and non-member samples. To address these shortcomings, the MIN-K% method established token-based effective methods using outlier token probabilities for pretraining data detection (Shi et al., 2024). Zhang et al. (2024c) subsequently proposed Min-K%++, theoretically grounding detection in local probability maxima identification. Zhang et al. (2024d) proposed DC-PDD to employ corpus frequency divergences to reduce false positives. Ye et al. (2024) introduced PAC, an MIA method that calculates polarization distances through input perturbations. Zhang et al. (2024b) developed PaCoST, which statistically compares model confidence on original test items versus distributionally-similar counterparts, to reveal widespread contamination across open-source models.

Alternative gray-box strategies, including perplexity-based memorization detection (Li, 2023a) and the adversarial compression ratio (ACR) metric (Schwarzschild et al., 2024), quantify memorization through input-output token efficiency.

4.3 Black-Box Detection

Black-box methods operate without access to model internals, training corpus, and are often accompanied by limitations in computational resources. Specifically, these methods heavily rely on certain assumptions shown in Appendix B.

Golchin and Surdeanu (2023a) introduced a multiple-choice question framework in which each question presents an original instance alongside three perturbed versions (where words are replaced with contextually relevant synonyms) and one invalid option. If the LLM consistently selects the original instance, this behavior may indicate the presence of data contamination. Building on this, Golchin and Surdeanu (2023b) proposed a guided instruction-based detection method, which effectively identifies contamination in datasets through instance completion and heuristic evaluation.

Duarte et al. (2024) developed DE-COP, a copyright detection framework that employs verbal versus paraphrased multiple-choice probing. Using benchmarks such as BookTection and arXivTection, DE-COP reveals temporal patterns in the training data of commercial LLMs. Similarly, Deng et al. (2023) proposed TS-Guessing, a protocol designed to test a model's ability to reconstruct masked elements of test data. This approach uncovers subtle contamination in major benchmarks. Further advancing this line of research, Dong et al. (2024) introduced CDD to identify contamination by analyzing the peakedness of output distributions. When paired with the TED mitigation technique, the CDD approach effectively addresses both explicit and implicit forms of contamination while preserving the validity of model evaluations.

As highlighted by (Ranaldi et al., 2024), the Text-to-SQL task with GPT-3.5 involves data contamination, where the model is tasked with reconstructing masked column names using the table name, the remaining column names, and contextual information. Similarly, Chang et al. (2023) introduced a challenging cloze task and employed data archaeology to examine the memorization of passages from 571 novels by using LLMs.

5 Future Directions

5.1 LLM Unlearning Methods

Unlearning techniques offer the potential to mitigate LLM privacy risks by erasing specific data elements. Future research should explore integrat-

Method	Authors	Assumption	Certain Tasks
DCQ	(Golchin and Surdeanu, 2023a)	Verbatim-Memorization	×
Guided Instruction	(Golchin and Surdeanu, 2023b)	Verbatim-Memorization	×
DE-COP	(Duarte et al., 2024)	Verbatim-Memorization	×
CDD	(Dong et al., 2024)	Output Distribution	×
TS-Guessing	(Deng et al., 2023)	Verbatim-Memorization	×
ATD	(Ranaldi et al., 2024)	Verbatim-Memorization	✓
Data Archaeology	(Chang et al., 2023)	Verbatim-Memorization	✓

Table 2: Black-box contamination detection methods, details of the assumptions underlying these approaches can be found in Appendix B.2.

ing contamination mitigation through targeted unlearning mechanisms that remove biases or leaked information from certain sources. This emerging field shows promise and fundamental challenges. For instance, Shumailov et al. (2024) claimed such data erasure may be fundamentally unachievable in current architecture.

5.2 Enhancing Black-box Detection Methods

Black-box detection methods require more attention as most LLMs are black-box models. Some existing contamination detection methods heavily rely on heuristic rules. Fu et al. (2024) categorized the assumptions of multiple detection methods and their validation status, demonstrating that some assumptions may be invalidated in multiple scenarios. In other words, the stability of the assumptions underlying current data contamination detection approaches remains uncertain. Given that black-box methods may have broad applicability, more research into their reliability and effectiveness is essential.

5.3 Distinguishing Between Data Contamination and Generalization

The ambiguity between contamination and generalization remains unresolved. A core paradox lies in why in-distribution (ID) data contamination can not be interpreted as an alternative of LLMs' generalization capability, given the intrinsic overlap between memorization and generalization in LLMs (Zhang et al., 2021). Despite growing attention to state-of-the-art LLMs, the community lacks a standard definition for distinctions between contamination and generalization.

5.4 Community Effort for Data Contamination

Previously, the community has made some efforts to collect evidence of contamination, as shown in appendix A. Furthermore, the data contamination prevention paradox manifests as an inverse relationship between protective efficacy and benchmark availability. While enhanced safeguards reduce contamination risks, they simultaneously constrain the usability of existing benchmarks through stringent data isolation. As a result, dynamic evaluation should become the mainstream approach, and such strategies should be embraced as a community consensus.

5.5 Non-Benchmark Evaluation

LLM-as-a-judge approaches (Section 3.3) confront reliability challenges from persistent model biases. Current implementations often yield assessment inconsistencies that diverge from human judgment standards. Future directions should prioritize developing adversarial testing frameworks and hybrid evaluation frameworks to bridge the alignment gap between automated scoring and human values.

6 Conclusion

Our paper examines three fundamental perspectives in data contamination research: (1) defining data contamination through the lenses of phases and benchmarks; (2) exploring methodologies for conducting contamination-free evaluations, with a particular focus on dynamic evaluation and LLM-based assessment techniques; and (3) investigating methods for detecting data contamination, offering a comprehensive analysis of existing techniques and their limitations. This work serves as both an introductory guide for researchers new to the field and a roadmap that underscores data contamination as a critical challenge in LLM evaluation. Furthermore, we provide actionable recommendations for enhancing contamination-aware evaluation systems, aiming to foster more robust and reliable LLM development practices.

7 Limitations

While we extensively cover various forms of data contamination, it is possible that new contamination mechanisms or models may not be fully captured in our analysis. Additionally, our focus is primarily on data contamination within the context of LLMs, and we may not have fully incorporated previous research on data contamination in other areas of machine learning.

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, and 8 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, and 1 others. 2024. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 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 (Volume 1: Long Papers)*, pages 67–93.

Sebastian Bordt, Harsha Nori, and Rich Caruana. 2023. Elephants never forget: Testing language models for memorization of tabular data. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. 2024. Elephants never forget: Memorization and learning of tabular data in large language models. In *Conference on Language Modeling (COLM)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *NeurIPS*, 33:1877–1901.

Jialun Cao, Wuqi Zhang, and Shing-Chi Cheung. 2024. Concerned with data contamination? assessing countermeasures in code language model. *arXiv preprint arXiv:2403.16898*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. 2024. Constat: Performance-based contamination detection in large language models. *arXiv preprint arXiv:2405.16281*.

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*.

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.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. De-cop: Detecting copyrighted content in language models training data. *arXiv preprint arXiv:2402.09910*.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1325–1335, Online. Association for Computational Linguistics.

Lizhou Fan, Wenyue Hua, Xiang Li, Kaijie Zhu, Mingyu Jin, Lingyao Li, Haoyang Ling, Jinkui Chi, Jindong Wang, Xin Ma, and Yongfeng Zhang. 2024. Nphardeval4v: A dynamic reasoning benchmark of multimodal large language models. *arXiv preprint arXiv:2403.01777*.

Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. 2024. Does data contamination detection work (well) for llms? a survey and evaluation on detection assumptions. *arXiv preprint arXiv:2410.18966*.

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*.

Jacob Haimes, Cenny Wenner, Kunvar Thaman, Vassil Tashev, Clement Neo, Esben Kran, and Jason Schreiber. 2024. Benchmark inflation: Revealing llm performance gaps using retro-holdouts. *arXiv preprint arXiv:2410.09247*.

V Hartmann, A Suri, V Bindschaedler, D Evans, S Tople, and R West. 2023. Sok: memorization in general-purpose large language models. arxiv. *Preprint posted online on October*, 24.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia. Association for Computational Linguistics.

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*.

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*.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, Singapore. Association for Computational Linguistics.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Fangyu Lei, Qian Liu, Yiming Huang, Shizhu He, Jun Zhao, and Kang Liu. 2024. S3Eval: A synthetic, scalable, systematic evaluation suite for large language model. 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 1259–1286.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18471–18480.

Jia Li, Ge Li, Xuanming Zhang, Yihong Dong, and Zhi Jin. 2024a. Evocodebench: An evolving code generation benchmark aligned with real-world code repositories. *arXiv preprint arXiv:2404.00599*.

Xiang Li, Yunshi Lan, and Chao Yang. 2024b. Treeeval: Benchmark-free evaluation of large language models through tree planning. *arXiv preprint arXiv:2402.13125*.

Yanyang Li. 2024. Awesome data contamination. https://github.com/lyy1994/awesome-data-contamination.

Yanyang Li, Tin Long Wong, Cheung To Hung, Jianqiao Zhao, Duo Zheng, Ka Wai Liu, Michael R. Lyu, and Liwei Wang. 2024c. C^2leva: Toward comprehensive and contamination-free language model evaluation. *arXiv preprint arXiv:2412.04947*.

Yanyang Li, Jianqiao Zhao, Duo Zheng, Zi-Yuan Hu, Zhi Chen, Xiaohui Su, Yongfeng Huang, Shijia Huang, Dahua Lin, Michael Lyu, and Liwei Wang. 2023. CLEVA: Chinese language models EVAluation platform. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 186–217.

Yucheng Li. 2023a. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation. *arXiv preprint arXiv:2309.10677*.

Yucheng Li. 2023b. An open source data contamination report for large language models. *arXiv preprint arXiv:2310.17589*.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024d. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. *arXiv preprint arXiv:2312.12343*.

Chuang Liu, Renren Jin, Mark Steedman, and Deyi Xiong. 2024. Evaluating Chinese large language models on discipline knowledge acquisition via memorization and robustness assessment. In *Proceedings of the 1st Workshop on Data Contamination (CONDA)*, pages 1–12, Bangkok, Thailand. Association for Computational Linguistics.

Rien Maertens, Charlotte Van Petegem, Niko Strijbol, Toon Baeyens, Arne Carla Jacobs, Peter Dawyndt, and Bart Mesuere. 2022. Dolos: Language-agnostic plagiarism detection in source code. *Journal of Computer Assisted Learning*, 38(4):1046–1061.

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.

Alexandre Matton, Tom Sherborne, Dennis Aumiller, Elena Tommasone, Milad Alizadeh, Jingyi He, Raymond Ma, Maxime Voisin, Ellen Gilsenan-McMahon, and Matthias Gallé. 2024. On leakage of code generation evaluation datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13215–13223.

Behzad Mehrbakhsh, Dario Garigliotti, Fernando Martínez-Plumed, and Jose Hernandez-Orallo. 2024. Confounders in instance variation for the analysis of data contamination. In *Proceedings of the 1st Workshop on Data Contamination (CONDA)*, pages 13–21.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Medha Palavalli, Amanda Bertsch, and Matthew Gormley. 2024. A taxonomy for data contamination in large language models. In *Proceedings of the 1st Workshop on Data Contamination (CONDA)*, pages 22–40, Bangkok, Thailand. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2406.17681*.

Federico Ranaldi, Elena Sofia Ruzzetti, Dario Onorati, Leonardo Ranaldi, Cristina Giannone, Andrea Favalli, Raniero Romagnoli, and Fabio Massimo Zanzotto. 2024. Investigating the impact of data contamination of large language models in text-to-SQL translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13909–13920, Bangkok, Thailand. Association for Computational Linguistics.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2024. How much are llms contaminated? a comprehensive survey and the llmsanitize library. *arXiv preprint arXiv:2404.00699*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Martin Riddell, Ansong Ni, and Arman Cohan. 2024. Quantifying contamination in evaluating code generation capabilities of language models. *arXiv preprint arXiv:2403.04811*.

Oscar Sainz, Jon 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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. 2024. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. 2024. Ununlearning: Unlearning is not

sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*.

Aaditya K Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *arXiv preprint arXiv:2411.03923*.

Dingjie Song, Sicheng Lai, Shunian Chen, Lichao Sun, and Benyou Wang. 2024. Both text and images leaked! a systematic analysis of multimodal llm data contamination. *arXiv preprint arXiv:2411.03823*.

Saurabh Srivastava, Anto PV, Shashank Menon, Ajay Sukumar, Alan Philipose, Stevin Prince, Sooraj Thomas, and 1 others. 2024. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *arXiv preprint arXiv:2402.19450*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shangqing Tu, Kejian Zhu, Yushi Bai, Zijun Yao, Lei Hou, and Juanzi Li. 2024. Dice: Detecting in-distribution contamination in llm's fine-tuning phase for math reasoning. *arXiv preprint arXiv:2406.04197*.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Zhongyu Wei, and Xuanjing Huang. 2024. Benchmark self-evolving: A multi-agent framework for dynamic llm evaluation. *arXiv preprint arXiv:2402.11443*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*.

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*.

Feng Yao, Yufan Zhuang, Zihao Sun, Sunan Xu, Animesh Kumar, and Jingbo Shang. 2024. Data contamination can cross language barriers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17864–17875.

Wentao Ye, Jiaqi Hu, Liyao Li, Haobo Wang, Gang Chen, and Junbo Zhao. 2024. Data contamination calibration for black-box LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10845–10861.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and YAN Shuicheng. 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*.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, and 1 others. 2023. Kola: Carefully benchmarking world knowledge of large language models. *arXiv preprint arXiv:2306.09296*.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. KIEval: A knowledge-grounded interactive evaluation framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5967–5985.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, and 1 others. 2024a. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Huixuan Zhang, Yun Lin, and Xiaojun Wan. 2024b. 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.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2024c. Min-k*arXiv preprint arXiv:2404.02936*.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024d. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274.

Zhehao Zhang, Jiaao Chen, and Diyi Yang. 2024e. Darg: Dynamic evaluation of large language models via adaptive reasoning graph. *arXiv preprint arXiv:2406.17271*.

Qihao Zhao, Yangyu Huang, Tengchao Lv, Lei Cui, Qinzheng Sun, Shaoguang Mao, Xin Zhang, Ying Xin, Qiufeng Yin, Scarlett Li, and 1 others. 2024. Mmlu-cf: A contamination-free multi-task language understanding benchmark. *arXiv preprint arXiv:2412.15194*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023.

A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024a. Dyval: Dynamic evaluation of large language models for reasoning tasks. *arXiv preprint arXiv:2309.17167*.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024b. Dynamic evaluation of large language models by meta probing agents. *arXiv preprint arXiv:2402.14865*.

Qin Zhu, Qingyuan Cheng, Runyu Peng, Xiaonan Li, Tengxiao Liu, Ru Peng, Xipeng Qiu, and Xuanjing Huang. 2024c. Inference-time decontamination: Reusing leaked benchmarks for large language model evaluation. *arXiv preprint arXiv:2406.13990*.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yun-Ze Song, Jiao Yueyang, Yumeng Zhang, Hanxu Hu, Yiran Wei, Rui Wang, and Hongyuan Lu. 2024d. CLEAN–EVAL: Clean evaluation on contaminated large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 835–847.

A Data Contamination Evidence Collection Efforts

Several initiatives are currently collecting evidence on data contamination. Below are key platforms and resources involved in this effort:

- **The Language Model Contamination Index (LM Contamination Index)**: This is a database used to track and record evidence of language model contamination. For more information, visit: https://hitz-zentroa.github.io/lm-contamination/.
- **CONDA-Workshop Data Contamination Database**: This is a community-driven project focused on the centralized collection of data contamination evidence. The goal is to help the community understand the extent of the problem and assist researchers in avoiding previous mistakes. Detailed information can be found at: https://huggingface.co/spaces/CONDA-Workshop/Data-Contamination-Database.

B Definition of Assumptions

B.1 Verbatim Memorization

In the context of LLMs, verbatim memorization (Carlini et al., 2021, 2022) refers to the phenomenon where a model recalls exact sequences of text, often from the data it has been trained on. This occurs when a model has seen a specific passage or piece of information during its training process and is able to reproduce it exactly when prompted. Verbatim memorization can lead to issues of data contamination, where the model unintentionally outputs copyrighted or sensitive material verbatim, causing concerns regarding privacy, intellectual property, and validity in analytical tasks.

B.2 Black-Box Method Assumption

Golchin and Surdeanu (2023a) has assumed that when a model has memorized instances from the original dataset, it will prefer selecting options containing the original instance over semantically similar perturbations. Additionally, LLMs may exhibit positional biases, where certain positions in multiple-choice options are more likely to be chosen, leading to potential overestimation or underestimation of contamination levels.

Golchin and Surdeanu (2023b) gave the assumption that by providing a "guided instruction" with dataset name, partition information, and part of the reference instance, LLMs can generate the complete version of the data instance. This allows for calculating overlap between generated completions and reference instances, helping to infer whether the dataset partition is contaminated.

Duarte et al. (2024) assumed that LLMs may memorize specific copyrighted content, such as books or academic papers, during training. When encountering similar content, they can distinguish whether they've seen it before. DE-COP exploits this by designing multiple-choice questions to test if the model can accurately identify original copyrighted content from paraphrased versions. Additionally, model selection biases can affect copyright detection results, and DE-COP introduces a calibration method to minimize such biases.

In (Dong et al., 2024), it is assumed that contaminated training data significantly affects the output distribution of large language models. Specifically, when trained on contaminated data, the model's output distribution becomes more peaked, causing it to produce more consistent outputs on contaminated data, favoring outputs strongly correlated with the training data.

Deng et al. (2023) assumed that if an LLM can accurately guess missing parts of a test set, such as keywords or answer options, without external assistance, it suggests that the model has encountered the corresponding benchmark data during training. This indicates memorization-based contamination.

The TS-Guessing protocol tests whether the model has memorized benchmark data by having it guess hidden information.

Ranaldi et al. (2024) assumed that data contamination can be detected solely by analyzing the inputs and outputs of LLMs. For example, unusually high accuracy on tasks from datasets like Spider indicates that the model may have been exposed to this dataset during training, leading to memorization rather than genuine understanding. Additionally, data contamination may lead to inflated performance on zero-shot tasks when the model encounters potentially contaminated data during training.

Chang et al. (2023) assumed that LLMs may memorize portions of text from their training data, especially when evaluation datasets contain known texts. This memorization can lead to inflated performance on tasks such as code generation. Moreover, data repetition on the web—through search engines and open datasets—encourages memorization, which improves accuracy on tasks involving familiar content.

B.3 Memorization and Data Contamination

Instance-level contamination (Fu et al., 2024) does not always lead to verbatim memorization. Utilizing instance generation (Carlini et al., 2022; Karamolegkou et al., 2023), demonstrates that verbatim memorization requires repeated exposures to this instance x during training. Indeed, future research on contamination should place more emphasis on LLMs' memorization.

C Data Contamination Detector

Li (2023b) present Contamination Detector to check whether test examples appear on the internet via Bing search and Common Crawl index. The tool is available at: https://github.com/liyucheng09/Contamination_Detector.

Ravaut et al. (2024) presented an open-source library for contamination detection in NLP datasets and LLMs. The library combines multiple methods for contamination detection and is available at: https://github.com/liyucheng09/Contamination_Detector.

Overlapy is a Python package developed to evaluate textual overlap (N-Grams) between two volumes of text. This tool can be accessed at: https://github.com/nlx-group/overlapy.

Yao et al. (2024) introduced Deep Contam, a method that detects cross-lingual contamination, which inflates LLMs' benchmark performance while evading existing detection methods. An effective detection method is provided in the repository, accessible at: https://github.com/ShangDataLab/Deep-Contam.

Tu et al. (2024) discussed the detection of in-distribution data contamination using LLM's internal state. The tool is available at: https://github.com/THU-KEG/DICE.

Bordt et al. (2023, 2024) presented Tabmemcheck, an open-source Python library designed to test language models for memorization of tabular datasets. The package includes four different tests for verbatim memorization of a tabular dataset (header test, row completion test, feature completion test, first token test). It also provides additional heuristics to test what an LLM knows about a tabular dataset, such as feature names test, feature values test, dataset name test, and sampling. The package can be found at: https://github.com/interpretml/LLM-Tabular-Memorization-Checker.

Yang et al. (2023) provided a package that includes the LLM decontaminator, which quantifies a dataset's rephrased samples relative to a benchmark. Based on the detection results, the contamination of rephrased samples in the dataset can be estimated and removed from the training set. This tool is available at: https://github.com/lm-sys/llm-decontaminator.