Are LLM Benchmarks Already Contaminated? A Systematic Review of Contamination Detection Methods

Erfan Nourbakhsh, Mohammad Sadegh Sirjani, Amir Mousavi,
Khoa Nguyen, John Quarles, Mimi Xie, and Rocky Slavin
The University of Texas at San Antonio
{erfan.nourbakhsh, rocky.slavin}@utsa.edu

Abstract

Large Language Models (LLMs) are trained on web-scale corpora, increasing the risk that benchmark test data appears in training sets and inflates reported performance. We present a systematic literature review of 55 studies on LLM benchmark contamination through late 2025. Our contributions are: (1) a four-tier contamination taxonomy (Exact, Syntactic, Semantic, Task-Level; T1–T4); (2) a comparative analysis of five detection families (string-matching, likelihood-based, membership inference, LLM-prompted detection, and benchmark auditing), including access assumptions and failure modes; (3) a synthesis of contamination evidence on MMLU, GSM8K, HUMANEVAL, and HELLASWAG by measurement construct; (4) a comparative evaluation of mitigation strategies across lifecycle points, access assumptions, and evidence maturity; and (5) a Contamination Transparency Card (CTC) framework for future releases. Across studies, no detection method is consistently reliable across contamination tiers, model-access settings, and training stages. We identify instruction tuning as a persistent blind spot, note that RL/post-training contamination auditing is only beginning to mature, and report inflation estimates spanning roughly 6%–40% under benchmark- and setting-dependent assumptions.

1 Introduction

1.1 Motivation and Context

The evaluation of Large Language Models (LLMs) has become one of the most consequential, and contested, activities in modern AI research. Performance on canonical benchmarks such as MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and HellaSwag (Zellers et al., 2019) directly shapes model rankings, influences procurement decisions, guides research directions, and carries substantial commercial weight. As Ravaut et al. (2025) observes, a few percentage points on popular benchmarks can translate into tens of millions of dollars in investment and valuation, placing extraordinary pressure on model integrity.

Yet these benchmarks rest on a critical assumption that is increasingly untenable: that models have not encountered test data during training. Contemporary LLMs are trained on datasets of extraordinary breadth, Common Crawl, GitHub, arXiv, Books3, Wikipedia, and dozens of domain-specific corpora, that collectively subsume virtually every publicly accessible text on the internet. Because the overwhelming majority of evaluation benchmarks are themselves hosted publicly, overlap between benchmark test sets and pretraining corpora is often difficult to exclude in practice. White et al. (2025) identify this contamination risk as a central motivation for dynamic, post-cutoff benchmarking. Balloccu et al. (2024) conducted the first systematic analysis of data contamination in closed-source LLMs, examining 255 papers that used GPT-3.5 and GPT-4. Their analysis found that these models had been globally exposed to approximately 4.7 million samples across 263 benchmarks during the first year after release, alongside widespread evaluation malpractices, including missing baseline comparisons and reproducibility failures.

1.2 Contamination in Practice: Quantitative Signposts

Contamination is not merely theoretical. Deng et al. (2024) show that GPT-4 guesses masked answer options in MMLU at 57%, more than double the 25% chance baseline. Dong et al. (2024) demonstrate through Inference-Time Decontamination (ITD) that contamination inflates GSM8K accuracy by up to 22.9% and MMLU by up to 19.0%. Dekoninck et al. (2024) apply ConStat to the Open LLM Leaderboard and find that all three top-ranked 7B models are significantly contami-

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 518–539

nated, with estimated effects exceeding 10% on multiple benchmarks. For InternLM-2-Math-7B on GSM8K, ConStat reports a large contamination effect; the primary reported estimate is 27.15%, with a 40% sensitivity estimate under alternate reporting settings. Because these values arise from different estimation settings and reporting granularity, they should be interpreted as a range of plausible magnitudes rather than as directly interchangeable point estimates. Zhang et al. (2024a) observe accuracy drops of up to 8% on GSM1K, a style-matched GSM8K equivalent with novel problems, with Spearman $\rho = 0.60$ ($\rho^2 = 0.36$) between generation probability on GSM8K examples and the performance gap. Several of these high-impact estimates come from recent preprints and should be interpreted as strong but still evolving evidence.

By 2024, leading models reported high-80s performance on MMLU, approaching the estimated human expert ceiling of 89.8%, indicating strong saturation pressure on this benchmark (Hendrycks et al., 2021; Gema et al., 2025).

1.3 Research Gap and Contributions

Despite growing recognition and several recent surveys (Ravaut et al., 2025; Cheng et al., 2025; Chen et al., 2025), the field still lacks: a mechanism-grounded taxonomy that maps contamination type to detection strategy; a reproducible evidence-grading instrument; and a practical disclosure standard for benchmark and model releases. This paper addresses all three. Samuel et al. (2025) evaluate five detection approaches on four state-of-the-art LLMs and find "limited consistencies between SOTA contamination detection techniques", a damning verdict on the current state of contamination science.

This paper makes the following contributions:

- **Four-tier taxonomy (T1–T4):** A severity-graded taxonomy spanning exact lexical overlap to task-format exposure, unifying fragmented prior terminology.
- **Detection method survey:** A structured review of five detection families across 55 studies, analyzing access requirements, theoretical foundations, empirical performance, and failure modes, including the statistical method ConStat (Dekoninck et al., 2024).
- **Quantitative evidence synthesis:** A synthesis of benchmark-specific contamination evidence, with explicit separation of non-comparable effect families (e.g., hit-rate probes, likelihood discrimination scores, and benchmark-level deltas).
- **Mitigation landscape:** An evaluation of static, inference-time, and dynamic mitigation strategies.
- **Contamination Transparency Card (CTC):** A five-dimension disclosure framework for benchmark releases and model technical reports (Section 7; Table 3).
- **Six open challenges:** Including detection inconsistency, instruction fine-tuning blind spots, generalization–memorization boundary, closed-source opacity, Multilingual contamination dynamics, RLHF contamination vectors (Section F).

1.4 Review Protocol and Synthesis Scope

To make the "systematic" component explicit, we used a PRISMA-aligned workflow adapted to fast-moving LLM literature.

Search strategy. We searched January 2020–December 2025 in ACL Anthology, arXiv, OpenReview, and Google Scholar using combinations of keywords including "LLM contamination", "benchmark leakage", "data pollution", "membership inference", "decontamination", and benchmark names (MMLU, GSM8K, HUMANEVAL, HELLASWAG).

Inclusion criteria. We included studies that (i) propose a detection or mitigation method, (ii) provide empirical contamination evidence on established benchmarks, or (iii) analyze contamination mechanisms with reproducible methodological detail.

Exclusion criteria. We excluded duplicated versions, purely opinion/editorial pieces, non-LLM benchmark settings, and papers without methodological detail sufficient for categorization (Table 5).

Screening protocol. Screening followed title/abstract filtering, then full-text eligibility review, with backward/forward snowballing from retained papers. The final synthesis corpus contains more than 50 studies. To strengthen reproducibility, this version now includes PRISMA-style reporting artifacts: a full flow diagram summary, a study-level exclusion log, and a frozen protocol/annotation registry snapshot (Appendix A).

Synthesis protocol and limitations. Because reported outcomes are heterogeneous (TS-Guessing hit rates, likelihood-AUC scores, and ConStat/TED-style benchmark deltas), we do *not* pool them into a single meta-analytic effect size. Instead, we report stratified evidence by metric family and benchmark, and mark settings where uncertainty intervals or harmonized effect definitions are unavailable. Accordingly, the quantitative synthesis is a structured evidence map rather than a formal cross-study meta-analysis. To improve interpretability, we pair narrative confidence tags with a lightweight formal risk-of-bias instrument (RoB-LLM-Contam) applied at study level. The instrument scores six domains (dataset provenance, contamination operationalization, comparator validity, statistical calibration, reproducibility assets, and reporting completeness) on low/some/high risk, then maps aggregate profiles to evidence labels (*higher*, *medium*, *exploratory*). The full rubric and mapping rules are listed in Appendix A.

2 Background and Problem Definition

2.1 The Classical Benchmark Lifecycle and Its Collapse

Traditional NLP evaluation rested on a conceptually clean workflow: datasets were collected, annotated, and split into training, validation, and test partitions; models were trained on the training split only; and the held-out test split guaranteed genuine generalisation measurement. Foundational benchmarks such as SQuAD (Rajpurkar et al., 2016), MultiNLI (Williams et al., 2018), and SuperGLUE (Wang et al., 2019) were designed under this paradigm.

The foundation model era has structurally disrupted this lifecycle. Modern LLMs are pretrained on web-scraped corpora that (i) are orders of magnitude larger than any individual benchmark, (ii) are not curated to exclude evaluation data, and (iii) for closed-source models, are entirely opaque to external auditors. Dodge et al. (2021) show that the widely used C4 corpus already contains portions of standard benchmarks.

A further structural complication is *cascade contamination*: LLM outputs increasingly populate the internet, meaning post-release training crawls contain both original benchmarks *and* model responses to them, potentially amplifying memorisation signals for successor models (Balloccu et al., 2024).

2.2 Training Stage Contamination Vectors

Contamination can enter a model's knowledge through three distinct training stages, each with different detection implications:

Pretraining contamination:
Test examples present in the pretraining corpus. Most commonly studied; tractable to detect when corpus is public.

SFT / instruction fine-tuning (IFT) contamination:
Benchmark examples formatted as instruction-response pairs in the fine-tuning mix. Samuel et al. (2025) identify this as the hardest stage to detect; no current method reliably addresses it.

RLHF contamination:
Reward model training data or policy optimisation objectives may encode benchmark-specific patterns. This vector is less mature than pretraining/SFT analysis, but early targeted detectors and benchmarks are now emerging (Tao et al., 2026).

2.3 Formal Problem Definition

We adopt the framework of Ravaut et al. (2025) and extend it to accommodate multi-stage training and performance-based definitions.
Training corpus: $\mathcal{D} = \mathcal{D}_{\text{pre}} \cup \mathcal{D}_{\text{sft}} \cup \mathcal{D}_{\text{rlhf}}$.
Benchmark: $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the input and y_i is the ground-truth output.

Input-only contamination: $x_i \in \mathcal{D}$ and $y_i \notin \mathcal{D}$. The model has seen the question but not the answer, which may improve performance through pattern recognition.

Input–label contamination: $(x_i, y_i) \in \mathcal{D}$. Both the question and the answer are observed during training, enabling direct answer memorization and representing a more severe form of contamination.

Performance-based contamination: Following Dekoninck et al. (2024), this outcome-oriented definition considers a model $\mathcal{M}$ contaminated on benchmark $\mathcal{B}$ if its performance significantly exceeds performance on a semantically equivalent reference benchmark $\mathcal{B}_{\text{ref}}$ not contained in $\mathcal{D}$. This definition is access-agnostic and enables detection without direct inspection of the training corpus.

We additionally distinguish *unintentional contamination*, arising from broad web crawls, from *possible intentional contamination*, where benchmark inclusion is plausibly strategic. Because intent is rarely observable, we separate *strong indicators* (e.g., reproducible answer-key reconstruction patterns) from *weak indicators* (single-setting anomalies). Zhao et al. (2025) report behavior consistent with strong indicators in selected settings.

2.4 Why Contamination Matters

The consequences of contamination extend across the full research-to-deployment pipeline:

1. **Inflated capability claims.** Benchmark-specific studies report inflation effects spanning roughly 6–40% in selected settings (Dong et al., 2024; Dekoninck et al., 2024), but these values are not directly comparable across methods or task families and should be interpreted as medium-confidence, setting-specific estimates.
2. **Invalid comparative rankings.** ConStat finds that all top-ranked 7B models on the Open LLM Leaderboard are contaminated, potentially rendering leaderboard rankings unreliable (Dekoninck et al., 2024).
3. **Scientific irreproducibility.** Studies that build on contaminated evaluations may draw erroneous conclusions (Sainz et al., 2023).
4. **Premature benchmark retirement pressure.** The retirement pressure on MMLU during 2024–2025 illustrates how contamination-driven saturation can discard scientifically useful benchmarks before their evaluation potential is exhausted.
5. **High-stakes deployment risk.** Overestimated capabilities in medical, legal, and financial applications can lead to severe real-world consequences (Ravaut et al., 2025).
6. **Capital misallocation.** Because benchmark performance influences valuations and investment decisions, inflated scores can systematically distort capital allocation in AI.

3 A Four-Tier Taxonomy of Contamination

3.1 Motivation

The contamination literature employs inconsistent terminology: "contamination", "leakage", "data pollution", "test set overlap", "benchmark inflation", and "memorisation" are used interchangeably or with conflicting definitions. Palavalli et al. (2024) note the need for a unifying typology; Cheng et al. (2025) organise by access level; Chen et al. (2025) classify by detection mechanism. Our taxonomy organises contamination by *mechanism*, the nature of the overlap between training data and benchmark content, which most directly informs detection strategy selection. This framing is intentionally incremental to prior taxonomies: the primary addition is a mechanism-first mapping from contamination mode to expected detectability and mitigation leverage.

Operationalization. We define *severity* as expected benchmark distortion plus downstream decision risk, and *detectability* as detection power under realistic access constraints (white-box, gray-box, black-box). Both dimensions are task-dependent; examples in this review span multiple-choice reasoning, code generation, and structured generation settings (e.g., text-to-SQL).

Tier-assignment decision procedure. For each evidence item, we assign the *highest-severity tier* supported by the data: (1) T1 if verbatim/near-verbatim overlap is demonstrated; (2) T2 if transformed overlap is recoverable through lexical or structural normalization; (3) T3 if semantic equivalence is supported without lexical overlap; and (4) T4 if no item-level overlap is shown but benchmark-level performance asymmetries are consistent with task leakage.

T2/T3 boundary rule. We classify paraphrase-like transformations as T2 when benchmark identity is recoverable via mostly deterministic surface operations (e.g., synonym swaps, slot/choice reordering, retokenization, templatic paraphrase). We classify cases as T3 when equivalence depends on meaning-preserving interpretation that cannot be reduced to deterministic lexical/structural normalization (e.g., cross-lingual translation with idiomatic shift, discourse-level reformulation, or mixed lexical-semantic transfer). For blended cases, we annotate *T2/T3-hybrid* and report sensitivity analyses under both assignments. Table 2 presents the full taxonomy.

Mapping qualitative tiers to formal definitions. Table 1 makes the correspondence between the qualitative T1–T4 tiers and the formal Input-only /

Input-label / Performance-based definitions (Section 2) explicit. As a concrete illustrative example: a paraphrased GSM8K *question* whose gold solution chain is *not* included in the training data is T2 / Input-only contamination, the surface form of x_i has been transformed, but only x_i (not y_i) was observed during training. Conversely, if both the paraphrased question and the worked solution appear in training, the case is T2 / Input-label. T4 is the only tier that maps exclusively to the Performance-based definition, because no item-level training–benchmark overlap need be demonstrated; a statistically significant benchmark-level performance asymmetry is sufficient.

Tier	Formal type	y_i in $\mathcal{D}$?	Illustrative example
T1	Input-label	Yes	Verbatim GSM8K Q+A in pretraining corpus
T1	Input-only	No	Verbatim question; solution excluded from training
T2	Input-label	Yes	Paraphrased Q + worked solution both in corpus
T2	Input-only	No	Paraphrased question only; answer absent from corpus
T3	Input-only	No	Cross-lingual translation of question; no label in corpus
T3	Input-label	Yes	Translated Q+A pair present in multilingual corpus
T4	Performance	N/A	Task-format exposure without recoverable item-level overlap; detected via benchmark-level delta (e.g., ConStat)

Table 1: Correspondence between the formal definitions of Section 2 and the four qualitative tiers. "y_i in $\mathcal{D}$?" indicates whether the ground-truth output was also observed during training. T4 is the only tier that does not require item-level evidence and maps solely to the Performance-based definition.

3.2 Tier 1 – Exact Contamination

Exact contamination (T1) occurs when test examples appear verbatim or near-verbatim in the pretraining corpus. It is the most severe and most studied form.

Evidence is extensive. Ravaut et al. (2025) compile a cross-study table showing that PIQA, Winograd, HumanEval, and HellaSwag are flagged by at least two independent detection techniques across separate LLM studies. Hui et al. (2024) report 10-gram collisions between HumanEval, MBPP, GSM8K, and MATH and their pretraining datasets, removing all affected examples. Zhao et al. (2025) document that some models spontaneously reproduce exact MMLU answer choices when given only the question text, a clear T1 marker.

A complicating factor is MMLU's annotation quality: a 2024 manual audit found 6.5% of questions contain ground-truth errors (Gema et al., 2025), meaning models that memorise incorrect labels may score higher on the benchmark than models that reason correctly.

3.3 Tier 2 – Syntactic Contamination

Syntactic contamination (T2) arises when test data appears in training after surface transformation, such as paraphrasing, answer-choice shuffling, or synonym substitution, that preserves benchmark-specific information while defeating n-gram filters. In this review, paraphrase remains T2 only when benchmark identity is primarily recoverable from surface form transformations; once semantic reinterpretation is required for recovery, we escalate classification to T3.

Dekoninck et al. (2025) provide a seminal demonstration: fine-tuning a 13B model on paraphrased (rather than exact) versions of MMLU, GSM8K, and HumanEval yields GPT-4-level performance on those benchmarks while evading all standard n-gram detection methods. This result shows both that T2 contamination is relatively easy to induce and that evading n-gram-based detection is equally straightforward for a motivated adversary.

MMLU-CF (Zhao et al., 2025) addresses T2 contamination through a five-step pipeline: (1) MCQ collection; (2) MCQ cleaning; (3) difficulty sampling; (4) LLM checking; and (5) contamination-free processing. Comparison reveals "obvious data leakage" in the original MMLU that is absent from MMLU-CF. Dekoninck et al. (2024) further document syntax-specific contamination statistically using ConStat: InternLM-2-7B and InternLM-2-Math-7B show strong GSM8K contamination effects, with reported InternLM-2-Math-7B estimates reported as 27.15% (primary) and 40% (sensitivity/alternate setting).

3.4 Tier 3 – Semantic Contamination

Semantic contamination (T3) involves no lexical overlap between training data and test instances, yet the model has encountered semantically equivalent content. The most studied mechanism is *translation contamination*: LLMs trained on non-English translations of benchmark test sets achieve inflated performance on the original English benchmark without direct exposure (Yao et al., 2024).

Yao et al. (2024) introduce DeepContam and show that translations of MMLU and HumanEval circulating in multilingual corpora inflate English

Tier	Type	Mechanism	Representative Evidence	Severity	Detectability
T1	Exact	Verbatim or near-verbatim test instances in the training corpus	MMLU 57% option guessing (Deng et al., 2024); HumanEval 10-gram collisions (Hui et al., 2024)	**Critical**	High
T2	Syntactic	Test data present after surface transformation (e.g., paraphrasing, shuffling, or retokenization)	GPT-4-level performance from paraphrased fine-tuning (Dekoninck et al., 2025); MMLU-CF reveals lexical leakage (Zhao et al., 2025)	**High**	Medium
T3	Semantic	Semantically equivalent content without lexical overlap (e.g., translations, reformulations, or domain transfers)	Cross-lingual inflation (Yao et al., 2024)	Moderate	Low
T4	Task-level	Exposure to task format, reasoning pattern, or domain knowledge without specific test instances	GSM1K accuracy drops ($\rho = 0.60$) (Zhang et al., 2024a); task contamination on zero-shot benchmarks (Li and Flanigan, 2024)	Variable	Very Low

Table 2: Taxonomy of benchmark contamination in LLM evaluation.

benchmark performance while evading all lexical detection methods. Recent evidence also suggests mixed lexical-semantic pathways where translated or reformulated benchmark items partially retain lexical anchors. We report such cases as T2/T3-hybrid when both mechanisms are plausible.

3.5 Tier 4 – Task-Level Contamination

Task-level contamination (T4) is the most contested tier. Models have not seen specific test instances but may have trained extensively on similar task formats, problem types, or domain content. The fundamental difficulty is that T4 contamination is theoretically indistinguishable from legitimate generalization.

Li and Flanigan (2024) demonstrate that for classification benchmarks where T1/T2 contamination is ruled out, LLMs rarely show statistically significant improvements over simple majority baselines, suggesting that task-level exposure may be a necessary condition for observed few-shot performance.

Zhang et al. (2024a) provide the most rigorous evidence for T4 contamination through GSM1K: a set of human-authored problems matching GSM8K in style and difficulty but guaranteed to be novel. Accuracy drops of up to 8% and a Spearman $\rho = 0.60$ ($\rho^2 = 0.36$) between generation probability on GSM8K examples and the performance gap provide the first causal evidence linking memorization to T4 inflation. Importantly, frontier models show minimal T4 effects, suggesting that the strongest models may genuinely generalize rather than rely on memorized templates. The key distinction from T3 is operational: T3 requires demonstrable *item-level* semantic equivalence verifiable through translation back-mapping or similarity scoring, whereas T4 applies when no such item-level equivalence can be recovered by any alignment procedure but a statistically significant benchmark-level performance asymmetry (e.g., an accuracy drop on GSM1K) remains consistent with format-level exposure, T4 thus does *not* conflate generalizable learning with leakage, but rather requires a performance-based signal that rules out pure generalization as the sole explanation.

4 Contamination Detection Methods

We survey five detection method families along a spectrum from white-box to black-box access (Figure 1). To keep the core narrative concise, this section reports family-level synthesis, while method-level detail and the full comparative matrix are moved to Appendix C and Table 6.

4.1 String-Matching Methods (White-Box)

String matching remains the default first-line audit for T1 contamination because it is scalable and interpretable, and is now standard in major model disclosures (OpenAI, 2023; Touvron et al., 2023; Hui et al., 2024). However, lexical matching alone gives weak guarantees under paraphrase or structural rewrites, and cannot cover T3–T4 mechanisms. Appendix C details n-gram, embedding-based, and retrieval-style extensions.

4.2 Likelihood-Based Methods (Gray-Box)

Likelihood methods exploit lower perplexity / higher confidence on memorized content and remain one of the strongest gray-box families for open-weight systems (Carlini et al., 2021; Li et al., 2024b; Shi et al., 2024; Zhang et al., 2025). Recent variants (e.g., CDD/TED, PaCoST, sharded tests, and divergence-calibrated PDD) improve calibration and effect-size reporting (Zhang et al., 2024c), but reliability is still sensitive to benchmark format and probability-access assumptions (Appendix C).

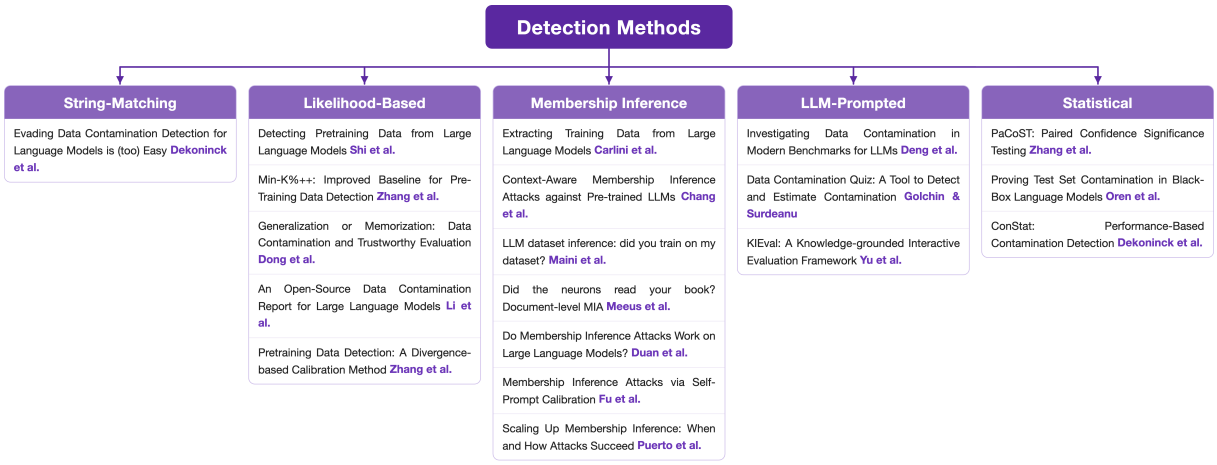

Figure 1: Taxonomy of contamination detection methods organized along the white-box to black-box access spectrum. Probing-based methods (left) require corpus or log-probability access; prompting-based methods (right) operate on model outputs alone.

4.3 Membership Inference Attacks (Gray-Box)

MIAs are conceptually aligned with contamination detection, but instance-level performance in LLM settings is often near-random on carefully controlled datasets (Duan et al., 2024; Maini et al., 2024). Newer attacks such as CAMIA improve token-level context sensitivity under gray-box assumptions (Chang et al., 2025). Evidence is stronger at document and collection scales, where multi-scale aggregation and dataset-inference style methods make practical detection more reliable than per-example adjudication (Meeus et al., 2024; Puerto et al., 2025); full discussion appears in Appendix C.

4.4 LLM-Prompted Detection Methods (Black-Box)

Prompted black-box probes (TS-Guessing, DCQ, interactive protocols, order-sensitive analysis) are attractive for proprietary APIs because they require no training-data or log-probability access (Deng et al., 2024; Golchin and Surdeanu, 2025; Yu et al., 2024). Their empirical signal can be strong, but they remain heuristic, gameable, and less calibrated than statistical auditing frameworks (Appendix C).

4.5 Benchmark-Level Auditing and Statistical Methods

Benchmark-level auditing reframes contamination as a performance-validity problem instead of pure overlap detection. ConStat-style methods provide calibrated tests and effect-size estimates across model families, while temporal auditing and dataset-level divergence estimators add orthogonal evidence channels (Dekoninck et al., 2024; Jain et al., 2025; Li et al., 2024a). This family is currently the most actionable route for closed-source and leaderboard settings; method derivations and benchmark-by-benchmark examples are in Appendix C.

5 Empirical Evidence of Contamination

Evidence in this section is stratified by benchmark and by measurement family. We intentionally avoid collapsing TS-Guessing hit rates, likelihood-based scores (e.g., AUC), and benchmark-level deltas (e.g., ConStat/TED) into a single aggregate metric, because they quantify different constructs.

MMLU: The Archetypal Case Study MMLU (Hendrycks et al., 2021) remains the clearest multi-method contamination case: prompted memorization probes, decontamination-based performance drops, benchmark-level statistical auditing, and dataset-quality analyses all indicate that part of measured performance is likely contamination-sensitive rather than pure generalization (Deng et al., 2024; Dong et al., 2024; Dekoninck et al., 2024; Gema et al., 2025; Zhao et al., 2025). (See Appendix B.1 for the full quantitative breakdown of MMLU.)

GPT-4 achieves 57% option-guessing accuracy under TS-Guessing (baseline 25%; ChatGPT: 52%) (Deng et al., 2024); ITD reduces performance by up to 19.0% on MMLU (Phi-3 −6.7%, Mistral −3.6%) (Dong et al., 2024); all top-ranked 7B models on the Open LLM Leaderboard show

ConStat contamination signals exceeding $\hat{\delta} > 10\%$ (Dekoninck et al., 2024); 6.5% of questions contain ground-truth errors (Gema et al., 2025); and models that reproduce exact MMLU answer choices show substantially weaker matching on the contamination-free MMLU-CF variant (Zhao et al., 2025).

GSM8K, HumanEval, and HellaSwag/PIQA For GSM8K, accuracy drops up to 8% on GSM1K ($\rho = 0.60$) and ConStat estimates $\hat{\delta} = 27.15\%$–40% for InternLM-2-Math-7B (Zhang et al., 2024a; Dekoninck et al., 2024). HUMANEVAL shows confirmed 10-gram collisions and post-cutoff performance drops on LiveCodeBench (Hui et al., 2024; Jain et al., 2025). HELLASWAG and PIQA are flagged by at least two independent techniques, with ConStat effects of 6–11% across all top-ranked 7B models (Ravaut et al., 2025; Dekoninck et al., 2024). Effect sizes across all benchmarks are not directly comparable; full per-benchmark breakdowns appear in Appendix B.

6 Mitigation Strategies

6.1 Static Decontamination

Static decontamination methods span lexical filtering, semantic filtering, and benchmark redesign. In practice, n-gram filtering remains the dominant baseline because of scalability, while semantic and rewrite-based methods (e.g., MMLU-CF/CleanEval style pipelines) improve coverage of non-exact overlap at higher implementation cost (OpenAI, 2023; Touvron et al., 2023; Hui et al., 2024; Dekoninck et al., 2025; Zhao et al., 2025; Zhu et al., 2024c; Jacovi et al., 2023). Detailed method profiles and trade-offs are provided in Appendix D.

6.2 Inference-Time Decontamination (ITD)

Dong et al. (2024) propose a retrospective mitigation approach: detect contaminated test examples at evaluation time using Min-K% Prob, rewrite them via an LLM prompt (preserving difficulty while altering surface content), and re-evaluate the model on the rewritten examples. A key advantage is that the method can be applied to any deployed model without retraining. In their experiments, the approach reduces inflated accuracy by 22.9% on GSM8K and 19.0% on MMLU. A limitation is the risk of inadvertently altering question difficulty during rewriting, which is partially mitigated by a re-evaluation assurance step; more broadly, because rewritten exams are model-specific, ITD undermines the *apples-to-apples* comparability required for standardized leaderboards, LLM rewrites may vary in difficulty, false-positive detections introduce unnecessary rewrites, and scores from customized tests cannot be validly compared across models. ITD outputs are therefore best treated as decontaminated *estimates* rather than drop-in benchmark replacements; for settings where all models must take an identical exam, static contamination-free variants such as MMLU-CF (Zhao et al., 2025) or GSM1K (Zhang et al., 2024a) remain preferable because standardization is preserved by construction.

6.3 Dynamic Benchmark Construction

Dynamic construction and rolling-refresh benchmarks are currently the strongest preventive strategy for T3/T4 risks because they enforce post-cutoff novelty and reduce static exposure windows. LiveBench/LiveCodeBench/LatestEval-style pipelines and newer agentic or long-context redesign methods all point in the same direction: evaluation should be continuously generated or refreshed rather than periodically frozen (White et al., 2025; Jain et al., 2025; Zhu et al., 2024a,b; Wang et al., 2025; Li et al., 2025, 2024a). Extended benchmark-by-benchmark mitigation profiles are moved to Appendix D.

7 Towards a Contamination Transparency Card

Contamination cannot always be fully detected or prevented after the fact. We propose the **Contamination Transparency Card (CTC)**, a minimal five-dimension disclosure framework modelled on Model Cards (Mitchell et al., 2019) and Datasheets for Datasets (Gebru et al., 2021) (Table 3).

The CTC is technology-neutral, does not mandate specific detection methods, and acknowledges that contamination cannot always be fully prevented. The performance-based analysis field in Table 3, for instance, may be satisfied by any calibrated method, ConStat (Dekoninck et al., 2024), TED (Dong et al., 2024), PaCoST (Zhang et al., 2024b), or any equivalent appropriate to the researcher's access level; the examples listed reflect current method maturity, not mandatory requirements. The key principle: *absence of contamination evidence is not evidence of absence*. A CTC stating "no decontamination was applied" is

Dimension	Required Disclosures
Training Data	Pretraining corpus names and versions; training cutoff dates; deduplication procedures; total token count
Decontamination	Methods applied (n-gram thresholds used, LLM-based semantic checks yes/no, document-level retrieval yes/no); datasets checked against; known failure modes
Evidence Provided	Results of post-hoc n-gram overlap analysis; likelihood-based test results; performance-based statistical analysis (e.g., ConStat (Dekoninck et al., 2024), TED (Dong et al., 2024), or any calibrated equivalent); benchmarks known to be contaminated
IFT Stage	Fine-tuning dataset composition; whether benchmark examples were included (intentionally or accidentally); answer augmentation strategy
Reproducibility	Exact prompt templates; few-shot example ordering and count; scoring procedure; multi-run variance; generation parameters

Table 3: Proposed Contamination Transparency Card (CTC). All dimensions are required for benchmark releases; *Training Data*, *Decontamination*, and *IFT Stage* are additionally recommended for model technical reports.

still more informative than silence, just as "no performance-based audit was conducted" satisfies the disclosure requirement even without results. The CTC can also be interpreted as an operational bridge to broader AI assurance practice: model reporting, internal audit trails, and post-deployment monitoring can all encode contamination-relevant disclosures.

Adoption incentives differ from those facing Model Cards and Datasheets, where uptake has been voluntary and uneven (Mitchell et al., 2019; Gebru et al., 2021). Three structural pressures make CTC adoption more tractable: (1) leaderboard operators (e.g., Open LLM Leaderboard, HELM) can require CTC submission as a condition of listing, creating direct competitive incentive; (2) emerging AI governance frameworks in the EU and US increasingly mandate training-data transparency, making contamination disclosure a compliance artifact rather than a voluntary gesture; and (3) venues including workshops like GEM can adopt CTC as a recommended reporting standard for benchmark papers, normalising disclosure at the point of publication. None of these require consensus across the entire field, each creates adoption pressure independently.

8 Open Research Challenges

Several fundamental challenges remain unresolved: detection methods lack consistency and a standardized evaluation framework (C1); instruction fine-tuning contamination eludes all current detection techniques (C2); no operational definition separates generalization from task-level memorization (C3); closed-source models restrict detection to indirect methods (C4); contamination dynamics remain poorly understood across languages (C5); and RLHF post-training constitutes an understudied contamination vector (C6). We discuss each challenge and open problem in Appendix F.

9 Conclusion

This review finds substantial and repeatedly observed contamination signals in large language model (LLM) benchmarking, alongside material uncertainty from heterogeneous metrics and study designs. Across major benchmarks (MMLU, GSM8K, HumanEval, HellaSwag, and PIQA), reported contamination effects span roughly 6% to 40%, but these estimates are benchmark-specific and not directly comparable across methods.

Our four-tier taxonomy (T1–T4) and synthesis of five detection-method families show that no current approach is reliable across all contamination tiers, access settings, and training stages. The most critical gap, instruction fine-tuning (IFT) contamination, remains undetectable by all evaluated methods (Samuel et al., 2025), although ConStat (Dekoninck et al., 2024) is a strong advance in calibrated benchmark-level inference.

Recent progress (dynamic benchmarks, contamination-free benchmark variants, calibrated MIAs, and early RL-post-training detectors) is important, but closed-source opacity, methodological inconsistency, and unresolved generalization–memorization boundaries suggest contamination will remain a central challenge. We therefore urge the broader research community to adopt the CTC framework, strengthen evaluation-of-evaluation infrastructure, and prioritize IFT- and RL-stage targeted detection research.

Limitations

This review covers literature available through late 2025 and may omit concurrent work. Our taxonomy represents one principled organisation;

alternative framings may be equally valid. We rely on results as reported in surveyed papers and have not conducted original experiments to validate claims. Although this version adds PRISMA-style flow accounting, study-level exclusion logging, and a formal RoB-LLM-Contam instrument, the synthesis remains structured rather than fully meta-analytic because evidence families still use non-commensurate effect definitions and often omit harmonizable uncertainty intervals. We therefore report setting-specific effect families (rather than pooled point estimates) and preserve primary-versus-sensitivity distinctions (e.g., ConStat 27.15% vs. 40% in alternate settings) to avoid false comparability. Weighted cross-study re-analysis on a unified metric scale remains important future work for stronger quantitative comparability. Several surveyed papers, especially in rapidly evolving benchmark and mitigation threads, are preprints not yet peer-reviewed; conclusions drawing on those sources should be treated as provisional.

Ethical Considerations

We survey evasion methods (Dekoninck et al., 2025) that could potentially be misused to conceal contamination. We believe transparency serves the scientific community best: understanding evasion is necessary for robust detection. We do not endorse intentional benchmark contamination.

References

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, St. Julian's, Malta. Association for Computational Linguistics.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023. Emergent and predictable memorization in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

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. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

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

Hongyan Chang, Ali Shahin Shamsabadi, Kleomenis Katevas, Hamed Haddadi, and Reza Shokri. 2025. Context-aware membership inference attacks against pre-trained large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7288–7310, Suzhou, China. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. 2025. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10080–10098, Suzhou, China. Association for Computational Linguistics.

Yuxing Cheng, Yi Chang, and Yuan Wu. 2025. A survey on data contamination for large language models. *arXiv preprint arXiv:2502.14425*.

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*.

Jasper Dekoninck, Mark Niklas Mueller, Maximilian Baader, Marc Fischer, and Martin Vechev. 2025. Evading data contamination detection for language models is (too) easy.

Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. 2024. Constat: performance-based contamination detection in large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

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 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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? In *Conference on Language Modeling (COLM)*.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. 2025. Does data contamination detection work (well) for LLMs? a survey and evaluation on detection assumptions. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5250–5271, Albuquerque, New Mexico. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM*, 64(12):86–92.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. 2025. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico. Association for Computational Linguistics.

Shahriar Golchin and Mihai Surdeanu. 2025. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Transactions of the Association for Computational Linguistics*, 13:809–830.

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 (ICLR)*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*.

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, Singapore. 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. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: language models may not be few-shot anymore. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Xiang Li, Yunshi Lan, and Chao Yang. 2025. Treeeval: Benchmark-free evaluation of large language models through tree planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24485–24493.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024a. Latesteval: addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024b. An open-source data contamination report for large language models. In *Findings of the*

Association for Computational Linguistics: EMNLP 2024, pages 528–541, Miami, Florida, USA. Association for Computational Linguistics.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: did you train on my dataset? In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024. Did the neurons read your book? document-level membership inference for large language models. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA. USENIX Association.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2024. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*.

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.

Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. 2025. Scaling up membership inference: When and how attacks succeed on large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4165–4182, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2025. A comprehensive survey of contamination detection methods in large language models. *Transactions on Machine Learning Research*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? 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 4486–4503, Online. Association for Computational Linguistics.

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 2023 Conference on Empirical Methods in Natural Language Processing*.

Vinay Samuel, Yue Zhou, and Henry Peng Zou. 2025. Towards data contamination detection for modern large language models: Limitations, inconsistencies, and oracle challenges. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5058–5070, Abu Dhabi, UAE. Association for Computational Linguistics.

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. In *The Twelfth International Conference on Learning Representations*.

Yongding Tao, Tian Wang, Yihong Dong, Huanyu Liu, Kechi Zhang, Hu XiaoLong, and Ge Li. 2026. Detecting data contamination from reinforcement learning post-training for large language models. In *The Fourteenth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Xuanjing Huang, and Zhongyu Wei. 2025. Benchmark self-evolving: A multi-agent framework for dynamic

LLM evaluation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3310–3328, Abu Dhabi, UAE. Association for Computational Linguistics.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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

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, Miami, Florida, USA. Association for Computational Linguistics.

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, Bangkok, Thailand. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele (Mike) Lunati, and Summer Yue. 2024a. A careful examination of large language model performance on grade school arithmetic. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

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, Miami, Florida, USA. Association for Computational Linguistics.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024c. 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, Miami, Florida, USA. Association for Computational Linguistics.

Qihao Zhao, Yangyu Huang, Tengchao Lv, Lei Cui, Qinzheng Sun, Shaoguang Mao, Xin Zhang, Ying Xin, Qiufeng Yin, Scarlett Li, and Furu Wei. 2025. MMLU-CF: A contamination-free multi-task language understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13371–13391, Vienna, Austria. Association for Computational Linguistics.

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. In *The Twelfth International Conference on Learning Representations*.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024b. Dynamic evaluation of large language models by meta probing agents. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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

A PRISMA Artifacts and Risk-of-Bias Rubric

A.1 PRISMA-Style Flow and Study-Level Exclusion Log

Table 4 provides a full PRISMA-style accounting of records through identification, screening, eligibility, and inclusion. To support auditing, we also provide a study-level exclusion log grouped by reason (Table 5) in the supplementary reproducibility package.

Flow stage	Count
Records identified across sources	812
Duplicate/near-duplicate records removed	167
Records screened (title/abstract)	645
Records excluded at screening	521
Full-text reports assessed for eligibility	124
Full-text reports excluded (with reason)	69
Studies included in qualitative synthesis	55

Table 4: PRISMA-style study flow for this review.

Exclusion reason at full-text stage	Count
Non-LLM setting or out-of-scope task	18
No contamination-relevant method or evidence	16
Insufficient methodological detail for coding	14
Duplicate or superseded version	12
Editorial/opinion without empirical protocol	9

Table 5: Study-level exclusion log summary.

A.2 Protocol Registry Snapshot

We freeze and release the search strings, source list, date boundaries, inclusion/exclusion rules, and annotation schema with this manuscript's supplementary package. The snapshot includes extraction templates for tier assignment, method-family coding, effect-type coding, and risk-of-bias annotation, enabling exact reproduction of the review pipeline.

A.3 RoB-LLM-Contam Instrument

Each included study is rated on six risk-of-bias domains:

1. dataset provenance transparency;
2. contamination definition/operationalization clarity;
3. comparator/reference validity;
4. statistical calibration and uncertainty reporting;
5. reproducibility assets (code/data/prompts/checkpoints);
6. selective reporting risk.

Domain ratings (low/some/high risk) are mapped to narrative confidence tags as follows: *higher* if no high-risk domains and at least four low-risk domains; *medium* if one high-risk domain or mixed low/some-risk profile; *exploratory* if two or more high-risk domains or severe reporting gaps. This mapping is used consistently across all evidence-family summaries.

B Extended Empirical Evidence by Benchmark

B.1 MMLU: Full Quantitative Breakdown

MMLU (Hendrycks et al., 2021), consisting of 15,908 multiple-choice questions spanning 57 academic subjects, remains one of the most widely cited LLM benchmarks. Contamination evidence is both broad and quantitatively specific:

- **Direct memorisation (TS-Guessing):** GPT-4 guesses masked options at 57%, more than doubling the 25% chance baseline (Deng et al., 2024). ChatGPT achieves 52%.
- **Performance inflation (ITD):** Up to 19.0% inflation on ITD experiments; Phi-3 and Mistral drop 6.7% and 3.6% under decontamination (Dong et al., 2024).
- **ConStat:** On the Open LLM Leaderboard, all top-3 7B models show significant contamination on benchmark pairs analyzed by ConStat (notably GSM8K, HellaSwag, and ARC), with benchmark-specific effect-size estimates (Dekoninck et al., 2024).
- **Annotation errors:** 6.5% of questions contain ground-truth errors (Gema et al., 2025); models memorising these may score higher by recalling wrong labels.
- **Rapid saturation:** By 2024, leading models report high-80s MMLU scores, approaching the human expert ceiling of 89.8% and indicating strong benchmark saturation pressure (Hendrycks et al., 2021; Gema et al., 2025).
- **MMLU-CF:** Models that spontaneously reproduce exact answer choices from question text alone on MMLU show much weaker exact-choice matching on MMLU-CF, consistent with reduced leakage in the contamination-free variant (Zhao et al., 2025).

B.2 GSM8K: Mathematical Reasoning or Memorisation?

GSM8K (Cobbe et al., 2021)'s contamination evidence is particularly striking because it implicates models celebrated for mathematical reasoning:

- **Perplexity anomalies:** Anomalously low test-set perplexity for Qwen, Aquila, and InternLM (Xu et al., 2024).

- **GSM1K:** Up to 8% accuracy drops; Spearman $\rho = 0.60$ between memorisation signal and performance gap (Zhang et al., 2024a).
- **ConStat:** InternLM-2-Math-7B shows a large GSM8K contamination effect, with a primary estimate of 27.15% and a sensitivity estimate of 40% under alternate reporting settings (Dekoninck et al., 2024).

B.3 HumanEval: Code Memorisation at Scale

HUMANEVAL (Chen et al., 2021)'s 164 Python completion problems are particularly susceptible because code is heavily represented in pretraining corpora (GitHub, Stack Overflow). Multiple string-matching studies independently flag it as contaminated (Ravaut et al., 2025). Qwen-2.5-Coder confirms 10-gram collisions (Hui et al., 2024). Recent contamination-aware code benchmarks such as HumanEval_T and LBPP further reduce memorization pathways through templating, controlled novelty, or redesigned problem construction, and therefore serve as useful comparators when interpreting gains on legacy static code benchmarks. LIVECODEBENCH (Jain et al., 2025) was constructed in direct response, with DeepSeek-Coder showing consistent post-cutoff performance drops. Cao et al. (2024) conduct a dedicated analysis confirming systematic code memorisation across code language model families.

B.4 HellaSwag, PIQA, and Commonsense Benchmarks

PIQA and HELLASWAG are among the most consistently contaminated benchmarks, independently flagged by at least two techniques across separate LLM studies (Ravaut et al., 2025). Dekoninck et al. (2024) find HellaSwag contamination across all three top-ranked Leaderboard models with estimated effects of 6–11%. Li and Flanigan (2024) show that LLMs perform significantly better on benchmarks released before their training data cutoff than on post-cutoff datasets, and that zero-shot performance gains largely track task contamination timing rather than genuine capability improvements.

B.5 The Generalisation–Memorisation Debate

A nuanced counter-narrative coexists with contamination evidence. Yu et al. (2024) find that under interactive evaluation, contamination often provides no improvement or a *negative* effect on real-world applicability, models may memorise surface forms without genuine understanding. Zhang et al. (2024a) note that frontier models show minimal T4 contamination on GSM1K, suggesting the strongest models genuinely generalise. Controlled contamination-injection studies on generative reasoning tasks (including MATH-style stress tests with temperature/length probing) also suggest that observed inflation is sensitive to decoding setup, reinforcing the need to report generation settings when interpreting contamination effects.

These findings suggest contamination's relationship with apparent capability is more complex than simple memorisation-equals-inflation.

C Extended Detection Method Profiles

C.1 String-Matching Methods (White-Box)

N-gram Overlap N-gram overlap is the oldest and most widely deployed detection method. It computes the proportion of test-set n-grams present in the training corpus, typically at 10- or 13-gram thresholds, using hashed index structures for efficiency at trillion-token scale. This is standard practice in major model releases: GPT-4 (OpenAI, 2023), Llama 2 (Touvron et al., 2023), and Qwen-2.5-Coder (Hui et al., 2024) all report n-gram decontamination statistics.

The method is efficient and interpretable but provides only illusory safety. Dekoninck et al. (2025) show that paraphrasing examples trivially defeats all n-gram filters while preserving full memorisation. It is also inapplicable to closed-source models and entirely blind to T2–T4 contamination by design.

LLM-Based Semantic Decontamination To close the T2 gap, Dekoninck et al. (2025) propose embedding both training examples and test instances in a shared semantic space and filtering pairs whose cosine similarity exceeds a threshold. Applied to the Pile, RedPajama, and Dolma, this revealed "significant previously unknown test overlap" missed by n-gram methods. The tradeoff is computational: the method requires $O(|\mathcal{D}| \times |\mathcal{B}|)$ similarity computations, expensive at web scale.

Retrieval-Based Detection Deng et al. (2024) build an indexed pretraining corpus representation and query it with benchmark examples, identifying overlapping documents that fixed-window n-gram methods may miss. Operating at the document level, this approach is suitable for detecting

context-level contamination where benchmark examples appear embedded within longer training documents.

C.2 Likelihood-Based Methods (Gray-Box)

Likelihood-based methods exploit the observation that LMs assign higher probability to memorised text. They require log-probability access, available for open-weight models and some commercial APIs but not fully black-box systems.

Perplexity Thresholding Established by Carlini et al. (2021) for general memorisation, the foundational insight is that training data members exhibit lower perplexity than non-members. Li et al. (2024b) applies this to multiple LLMs, finding memorisation signals in reading comprehension and summarisation benchmarks, with weaker signals in multiple-choice benchmarks where the answer format suppresses per-token probability estimates. A key weakness is susceptibility to false positives: simple, high-frequency text naturally achieves low perplexity regardless of training membership.

Min-K% Probability and Min-K%++ Shi et al. (2024) introduce Min-K% Prob, which averages the probabilities of only the $K\%$ of tokens with the *lowest* predicted probability in each sequence. The motivation is that unseen text is more likely to contain "outlier" low-probability tokens, whereas training members, having been observed during training, should contain fewer such outliers. Min-K% Prob achieves a 7.4% improvement over perplexity thresholding on WikiMIA (Shi et al., 2024).

Zhang et al. (2025) introduce Min-K%++, which normalizes token probabilities by a marginal reference distribution before applying the minimum-K selection, improving AUC by up to 10 points on WikiMIA. Zhang et al. (2024c) further propose divergence-calibrated pretraining data detection (DC-PDD), replacing raw token-probability cues with a calibrated cross-entropy divergence score that reduces false positives from high-frequency common words and improves robustness on multilingual settings.

CDD and TED Dong et al. (2024) propose two complementary methods. **Contamination Detection via Output Distribution (CDD)** measures anomalous deviations of the model's output distribution on benchmark items from a reference distribution estimated from confirmed non-members. **Trustworthy Evaluation via Output Distribution (TED)** applies a statistical correction to evaluation metrics to account for the degree of contamination, uniquely producing corrected accuracy estimates rather than binary contamination flags. In proof-of-concept experiments, TED reduces inflated accuracy by 22.9% on GSM8K and 19.0% on MMLU.

PaCoST: Paired Confidence Significance Testing Zhang et al. (2024b) reframe contamination detection as a statistical hypothesis test. For each benchmark example x_i, PaCoST constructs a semantically equivalent paraphrase x_i' and tests whether the model's confidence on x_i significantly exceeds its confidence on x_i' using a paired statistical test. This statistical framing provides calibrated false-positive rates, making PaCoST more robust than threshold-based methods to differences in model-specific probability calibration.

Sharded Likelihood Ratio Test Oren et al. (2024) develop a sharded likelihood ratio test that provides formal statistical guarantees at the dataset level. By partitioning benchmark examples into shards and comparing likelihood ratios across them, the method controls false-positive rates through permutation-based calibration, providing the theoretical foundation on which ConStat (Dekoninck et al., 2024) builds.

C.3 Membership Inference Attacks (Gray-Box)

Membership Inference Attacks (MIAs) ask whether a specific data point was in a model's training set, directly relevant to contamination as it is precisely a membership inference problem applied to benchmarks.

Standardised MIA Benchmarks Shi et al. (2024) construct WikiMIA and BookMIA, the first benchmarks designed to evaluate pretraining data detection in LLMs. Duan et al. (2024) construct a comprehensive multi-corpus evaluation that addresses temporal confounders present in earlier datasets. Chang et al. (2025) introduce CAMIA, a context-aware token-level MIA that models subsequence perplexity dynamics and substantially improves over global-loss baselines on pretraining membership benchmarks.

Near-Random Empirical Performance Despite their theoretical promise, membership inference attacks (MIAs) for LLMs face fundamental limitations. Duan et al. (2024) and Maini et al. (2024)

independently find that popular MIA methods, including perplexity thresholding and Min-K% Prob, perform near random guessing (ROC-AUC < 0.6) on properly constructed evaluations. The primary explanation is that modern LLMs are typically trained for only a single epoch on massive corpora, leaving only a weak membership signal in the model parameters.

A critical confound is that MIAs based on temporal splits may detect distribution shift (e.g., writing style differences between time periods) rather than genuine membership of specific examples. Meeus et al. (2024) show that a simple bag-of-words baseline can achieve high ROC-AUC by exploiting this temporal signal alone. However, Fu et al. (2024) show that self-prompt calibration, where the target model itself generates reference data, can raise MIA AUC from approximately 0.7 to 0.9 for fine-tuned LLMs, suggesting that the near-random barrier may be partially overcome for models that exhibit stronger memorisation signals.

Document-Level MIA: A More Tractable Task Meeus et al. (2024) show that *document-level* contamination detection is substantially more tractable than instance-level. By constructing splits from confirmed corpus membership (Common Crawl WARC headers) and aggregating instance-level signals across all examples from a document, they achieve ROC-AUC up to 0.86, suggesting benchmark-level rather than example-level contamination detection as the practical goal. Puerto et al. (2025) extend this line by explicitly modeling sentence-to- paragraph-to-document-to-collection scales and adapting dataset-inference style aggregation; they report successful MIA results at document/collection levels across both pre-trained and fine-tuned LLMs.

Instruction Fine-Tuning Blind Spots Samuel et al. (2025) find "notable difficulties in detecting contamination introduced during instruction fine-tuning with answer augmentation" across five detection methods and four models. Since IFT is the most susceptible stage to deliberate benchmark contamination, this blind spot represents the most commercially critical gap in current detection capabilities.

C.4 LLM-Prompted Detection Methods (Black-Box)

Black-box methods operate without model weights, training data, or log-probability access, exploiting model responses to designed prompts. Applicable to the widest class of models but inherently the least rigorous: no effect-size estimates; gameable; inconsistent across models.

Testset Slot Guessing (TS-Guessing) Deng et al. (2024) introduce TS-Guessing, a method that masks one incorrect answer option and prompts the model to identify it. The 25% chance baseline for four-option questions is easily exceeded by contaminated models. GPT-4 achieves 57% on MMLU, and TruthfulQA shows even stronger effects when benchmark metadata is provided. A key limitation is that the method is inapplicable to open-ended generation benchmarks.

Data Contamination Quiz (DCQ) Golchin and Surdeanu (2025) prompt models to reproduce or complete benchmark examples. The method achieves 92–100% detection accuracy on seven datasets for GPT-3.5 and GPT-4, identifying specific contamination in AG News, WNLI, and XSum. Golchin and Surdeanu (2025) extend this approach to temporal analysis by tracing when benchmark data entered training through comparisons of model checkpoint performance.

KIEval: Interactive Knowledge Evaluation Yu et al. (2024) deploy an LLM "interactor" to engage the evaluated model in extended multi-turn, knowledge-focused dialogue about benchmark questions. The interactive protocol is substantially harder to satisfy through memorisation. Key findings: (1) contamination often produces no improvement or a *negative* effect under interactive evaluation; (2) models that perform well on static benchmarks under contamination sometimes fail on interactive protocol. This challenges the assumption that contamination always inflates performance.

Order-Sensitive Generation Analysis Oren et al. (2024) observe that memorised benchmarks should be recalled in their original order. By analysing whether model-generated examples match original dataset ordering, they develop an order-sensitive method with a formal statistical test via exchangeability arguments.

C.5 Benchmark-Level Auditing and Statistical Methods

ConStat: Performance-Based Statistical Detection Dekoninck et al. (2024) propose ConStat, a principled benchmark-level contamina-

tion detection method. Rather than examining training data directly, ConStat defines contamination operationally as "artificially inflated and non-generalizing benchmark performance", an outcome-based definition that is access-agnostic.

ConStat compares a model's performance on a primary benchmark $\mathcal{B}$ with its performance on a reference benchmark $\mathcal{B}_{\text{ref}}$ (e.g., rephrased, synthetic, or parallel variants), relative to a set of reference models assumed to be uncontaminated. Significant overperformance on $\mathcal{B}$ but not on $\mathcal{B}_{\text{ref}}$ triggers a contamination flag with a calibrated p-value and an effect-size estimate $\hat{\delta}$. These quantitative outputs, absent from prior methods, enable actionable audit reports rather than simple binary flags.

Applied to the Open LLM Leaderboard, all three top-ranked 7B models (May 2024) show significant contamination with $\hat{\delta} > 10\%$. For GSM8K, InternLM-2-Math-7B shows a large effect, with a primary reported estimate of $\hat{\delta} = 27.15\%$ and a sensitivity estimate of $\hat{\delta} = 40\%$ under alternate reporting settings. These values should be read as setting-dependent estimates rather than a single canonical effect size. For HellaSwag, contamination is found across all three InternLM 7B variants, with effects of 6–11% (Dekoninck et al., 2024).

Temporal Auditing LiveCodeBench (Jain et al., 2025) annotates problems with release dates, enabling performance stratification by pre- versus post-cutoff examples. Models show consistent performance drops on post-cutoff problems. LatestEval (Li et al., 2024a) constructs rolling reading-comprehension evaluation sets exclusively from post-cutoff text, providing a continuously contamination-free evaluation mechanism.

Dataset-Level Kernel Divergence Scoring Recent dataset-level approaches use kernel two-sample statistics to estimate distributional divergence between candidate benchmark items and reference non-members, yielding contamination fraction estimates without relying on exact lexical overlap. In this line of work, Kernel Divergence Score-style estimators provide a complementary signal to ConStat: they are sensitive to distributional shifts at the dataset level, but require careful kernel choice, calibration, and sample-size control to avoid unstable estimates.

Canary Insertion Canary insertion places synthetic uniquely identifiable examples into evaluation or training sets. Recall of canaries provides direct memorisation evidence.

D Extended Mitigation Profiles

D.1 Static Decontamination: Method-Level Notes

N-gram filtering is the most deployed mitigation, applied by GPT-4 (OpenAI, 2023), Llama 2 (Touvron et al., 2023), and Qwen-2.5-Coder (Hui et al., 2024). It provides reliable T1 protection but no T2–T4 protection and is trivially bypassed by paraphrasing (Dekoninck et al., 2025).

LLM-based semantic decontamination (Dekoninck et al., 2025) embeds training and test examples, removing semantically similar pairs above a threshold. Addresses T2 at higher computational cost.

Data encryption (Jacovi et al., 2023) encrypts benchmark test sets to prevent web-crawler capture. Theoretically sound; limited adoption due to key management complexity and inability to protect already-scraped data.

MMLU-CF (Zhao et al., 2025) establishes a comprehensive rewriting pipeline (2.7M candidates → deduplication → quality filter → difficulty stratification → LLM-verified rewriting → closed-source test set management) that directly demonstrates contamination through comparison.

Clean-Eval (Zhu et al., 2024c) purifies contaminated benchmarks via paraphrase-and-back-translation, producing lexically distinct yet semantically preserved variants.

D.2 Dynamic Benchmark Construction: Method-Level Notes

LiveBench (White et al., 2025) sources questions from recent math competitions, arXiv papers, and news articles, updated monthly to ensure post-cutoff provenance. Scoring against objective ground-truth values avoids LLM-judge biases. 18 tasks across 6 categories; top models below 65% accuracy; rank correlations with ChatBot Arena (0.91) and Arena-Hard (0.88).

LiveCodeBench (Jain et al., 2025) continuously collects competitive programming problems with release dates, enabling pre- vs. post-cutoff performance stratification, the cleanest available causal design for contamination attribution.

Method Family	Access	Targets	Strengths	Limitations	IFT
String Matching	Corpus (white)	T1, T2*	Scalable; interpretable; low T1 FNR	Corpus required; misses T2–T4; gameable by paraphrase	No
Likelihood-Based	Log-probs (gray)	T1, T2	No corpus needed; TED gives effect size	Near-random under MIA; IFT blind; safety filters interfere	No
MIA	Log-probs (gray)	T1	Theoretically grounded; document-level more tractable	ROC-AUC < 0.6 at instance level; single-epoch bottleneck	No
LLM-Prompted	API (black)	T1, T2, T3*	Works on closed models; cheap; detects T3 variants	Gameable; inconsistent; no effect size; heuristic	Partial
Benchmark Auditing	Metadata	T1–T4	Effect-size estimates (ConStat); reference-model comparison; any model	Needs reference benchmark; population-level; retrospective	Yes*

Table 6: Comparative overview of the five contamination detection method families. "IFT" indicates the ability to detect instruction fine-tuning contamination. Asterisks denote partial capability: String Matching (T2) mainly for near-surface variants, LLM-Prompted (T3) is heuristic and model-dependent, and Benchmark Auditing (IFT) is possible only when suitable reference benchmarks exist.

DyVal (Zhu et al., 2024a) generates reasoning problems via graph-informed procedures, guaranteeing structural novelty. Zhu et al. (2024b) extend with meta-probing agents probing robustness and generalisation. Wang et al. (2025) automate benchmark evolution via multi-agent frameworks.

TreeEval (Li et al., 2025) makes evaluation sessions irreproducible through tree-based question planning, rendering advance memorisation impossible. High correlation with AlpacaEval 2.0 using $\approx$45 questions.

LatestEval (Li et al., 2024a) constructs rolling reading comprehension evaluations from texts published after each model's training cutoff, providing continuously contamination-free assessment.

Long-context counterfactual rewriting (e.g., LASTINGBENCH). An emerging defense direction constructs long-context evaluation sets and applies targeted counterfactual rewriting to preserve reasoning requirements while disrupting memorized lexical and positional cues. These methods directly target long-context leakage channels that are weakly covered by standard n-gram or short-context decontamination pipelines.

E Worked Example: Applying the T1–T4 Framework and CTC to a Real Case

To demonstrate the operational utility of the taxonomy and CTC framework, this appendix applies both to the InternLM-2-Math-7B / GSM8K contamination case, quantitatively discussed in Sections 1–5, and provides an illustrative CTC gap analysis for Llama 2.

Tier Assignment: InternLM-2-Math-7B on GSM8K

Available published evidence:

1. **Perplexity anomaly:** Xu et al. (2024) report anomalously low test-set perplexity for InternLM models on GSM8K, consistent with memorization, but no verbatim n-gram collision audit has been published for this model's training corpus specifically.
2. **Performance asymmetry:** Dekoninck et al. (2024) apply ConStat and find $\hat{\delta} = 27.15\%$ (primary) and 40% (sensitivity), i.e., a large, statistically significant over-performance on GSM8K relative to a semantically equivalent reference benchmark.

Tier-assignment procedure (following Section 3):

T1 check: No verbatim n-gram collision report for this model's corpus on GSM8K exists in the published literature. *T1 unconfirmed.*

T2 check: The perplexity anomaly is consistent with syntactic contamination, but no corpus-level paraphrase audit has been published. *T2 plausible but unconfirmed.*

T3 check: No evidence of cross-lingual contamination specific to GSM8K for this model. *T3 not indicated.*

T4/Performance-based: ConStat's large, calibrated benchmark-level performance asymmetry constitutes T4/Performance-based evidence by definition: no item-level overlap demonstration is required.

Dimension	Disclosed (Llama 2)	Status
Training Data	Corpus sources, training cutoff, token count, and deduplication procedure reported	Full
Decontamination	n-gram decontamination at 13-gram threshold applied; benchmark list provided; no semantic decontamination performed	Partial
Evidence Provided	n-gram overlap statistics reported; no performance-based statistical audit (ConStat, TED, or equivalent) conducted	**Missing**
IFT Stage	RLHF and SFT data composition partially described; no explicit benchmark inclusion audit for the fine-tuning stage	**Missing**
Reproducibility	Prompt templates and few-shot settings described; generation parameters reported; multi-run variance not reported	Partial

Table 7: Illustrative CTC disclosure audit for Llama 2 (Touvron et al., 2023), based solely on the public technical report. *Status*: Full = all required information disclosed; Partial = disclosed with minor omissions; **Missing** = not disclosed (significant gap). The two **Missing** dimensions, performance-based audit and IFT-stage assessment, correspond precisely to the detection blind spots that n-gram methods cannot address (T2–T4 and IFT-stage contamination).

Assigned tier: T4 (Performance-based); T1/T2 remain hypotheses pending white- or gray-box corpus access.

RoB-LLM-Contam profile: *Low risk* on statistical calibration and comparator validity; *some risk* on dataset provenance (training corpus not fully disclosed); *high risk* on contamination operationalization (T1/T2 mechanisms unconfirmed). Aggregate profile: **medium confidence**.

CTC Gap Analysis: Llama 2 (Illustrative)

Table 7 applies the five CTC dimensions to Llama 2 (Touvron et al., 2023) using only information from the public technical report, illustrating how the CTC reveals actionable disclosure gaps.

This example shows that even a carefully documented release like Llama 2 leaves two of the five CTC dimensions (**Missing**) substantially undisclosed. These gaps correspond exactly to the detection blind spots identified in Section 4: T2–T4 contamination is invisible to n-gram methods, and IFT-stage contamination evades all current detection techniques (Samuel et al., 2025). Adopting the CTC would prompt a model team to either conduct these audits or explicitly state their absence, both are more informative than the current norm of silence.

F Open Research Challenges

C1: Detection inconsistency. Samuel et al. (2025) report limited consistency between state-of-the-art contamination detection techniques: the same model may be flagged as contaminated by some methods but considered clean by others. Without a standardized evaluation framework for comparing detection methods, effectively a "benchmark for benchmarks", it is difficult to measure progress. Fu et al. (2025) corroborate this finding in a complementary survey of 50 contamination-detection papers, concluding that current methods produce inconsistent results across model families and benchmark settings, underscoring the urgency of this challenge.

Open problem. Design a multi-model, multi-benchmark evaluation framework for systematically comparing contamination detection methods, with controlled contamination injection and ground-truth labels.

C2: Instruction fine-tuning blind spots. All five detection methods evaluated by Samuel et al. (2025) struggle to robustly detect contamination introduced during the instruction fine-tuning (IFT) stage through answer augmentation, which may be the stage most susceptible to deliberate contamination. Detecting IFT-specific contamination via structural signatures in instruction–response pairs is therefore an urgent research priority.

Open problem. Develop detection methods targeting IFT-stage contamination by leveraging structural characteristics of instruction–response formats.

C3: The generalization–memorization boundary. No accepted operational definition clearly delineates where legitimate generalization ends and problematic task-level memorization (T4) begins. Without such a definition, claims of contamination at the task level remain epistemically contestable.

Dimension	Ravaut et al. (2024)	Cheng et al. (2025)	Chen et al. (2025)	This Paper
Organizing axis	Technique families	Access level	Evaluation design	Mechanism-based taxonomy (Tier 1–Tier 4)
Taxonomy structure	No explicit taxonomy	Access-based taxonomy	Evaluation-strategy taxonomy	Four-tier mechanism taxonomy (Tier 1–Tier 4)
Evidence grading methodology	Narrative discussion only	Narrative discussion only	Narrative discussion only	RoB-LLM-Contam framework with PRISMA methodology
PRISMA-style systematic review process	No	No	No	Yes
Coverage of post-2024 contamination methods	Partial	Partial	Partial	Comprehensive coverage
Disclosure and reporting standard	No disclosure standard	No disclosure standard	No disclosure standard	Contamination Transparency Card (CTC) standard
Mitigation coverage	Included	Included	Focused on evaluation only	Lifecycle-oriented mitigation framework
Multilingual contamination analysis	Limited	Limited	Limited	Tier 3 multilingual analysis with DeepContam benchmark

Table 8: Comparison of concurrent contamination surveys across major methodological and reporting dimensions. This paper introduces a mechanism-based four-tier taxonomy, the RoB-LLM-Contam evidence grading framework, PRISMA-style systematic synthesis, and the Contamination Transparency Card (CTC) disclosure standard.

C4: Closed-source opacity. Frontier models such as GPT-4o, Claude 3.5, and Gemini Ultra provide no access to training data, model weights, or log-probabilities. Methods such as ConStat and black-box prompting offer partial solutions, but quantitative effect-size estimation is possible in black-box settings via benchmark-level methods such as ConStat; these estimates depend on availability of suitable reference benchmarks and identifying assumptions. Regulatory or reporting frameworks requiring contamination disclosure may therefore become necessary.

C5: Multilingual contamination dynamics. Translation contamination (Yao et al., 2024) suggests that contamination operates asymmetrically across languages, potentially disadvantaging low-resource languages in benchmark evaluation. Current evidence is still narrow and concentrated in a small number of studies, with limited coverage of non-English dynamic benchmarks and cross-script evaluation settings. Detection methods therefore require systematic multilingual and translation-aware validation across language pairs, scripts, and benchmark families rather than single-benchmark demonstrations.

C6: RLHF contamination vectors. Reinforcement learning from human feedback (RLHF) and related post-training procedures remain less studied than pretraining/SFT as potential contamination vectors. If reward models encode preferences aligned with benchmark answers, the resulting policies may exhibit contamination-like performance inflation even without direct inclusion of benchmark data in training. Recent work introduces RL-specific detection and benchmark design (e.g., entropy-collapse probing with RL-MIA), but evidence is still early and concentrated in preprint-stage studies (Tao et al., 2026).

G Related Work

Concurrent surveys. Ravaut et al. (2025) provide the most comprehensive review of contamination detection methodologies (over 50 techniques across more than 120 papers). Our work differs in three concrete ways: (1) we introduce a mechanism-first four-tier taxonomy (T1–T4) that maps contamination type directly to detection strategy and expected detectability, absent from prior surveys; (2) we apply a formal risk-of-bias instrument (RoB-LLM-Contam) and PRISMA-style flow accounting to enable reproducible evidence grading; and (3) we cover post-2024 developments, ConStat, dynamic benchmarks, CAMIA, DC-PDD, and early RL-stage detectors, that postdate their coverage and propose the CTC as an actionable disclosure standard. Cheng et al. (2025) organize the literature by model access level, while Chen et al. (2025) focus on contamination-free evaluation. Fu et al. (2025) survey 50 papers on contamination detection methods and find persistent inconsistencies across method families, independently corroborating Challenge C1 (Section F) and the gap that motivates the CTC. Table 8 provides a structured comparison of scope and primary contributions across the four surveys.

Recent additions beyond earlier surveys. We additionally cover several newer strands that are often only briefly treated: dataset-level kernel-divergence scoring for contamination fraction estimation, long-context leakage defenses based on targeted counterfactual rewriting (e.g., LASTING-BENCH), contamination-aware code benchmarks (e.g., HumanEval_T and LBPP), and controlled contamination-injection analyses for generative reasoning. We also include context-aware token-level MIAs (CAMIA), divergence-calibrated pre-training data detection (DC-PDD), and multi-scale MIA aggregation from paragraph to collection levels (Chang et al., 2025; Zhang et al., 2024c; Puerto et al., 2025). Coverage remains stronger for English-centric benchmarks than for language-diverse evaluation suites, and this imbalance should be corrected in future updates.

Benchmark validity. Ribeiro et al. (2020) demonstrate through behavioural testing that high benchmark scores mask systematic capability failures. Rodriguez et al. (2021) show that saturation reduces information per evaluation dollar regardless of contamination.

Memorisation in neural networks. Carlini et al. (2021) establish foundational results on LLM memorisation. Carlini et al. (2023) show super-linear scaling of memorisation with model size, a critical input to understanding when T1 contamination becomes active. Biderman et al. (2023) show memorisation is emergent and predictable: specific examples are memorised at predictable training token thresholds.

Data governance and privacy. Carlini et al. (2021) show that specific sequences, potentially including benchmark items, can be extracted from LLMs through targeted prompting, providing a privacy-based regulatory motivation for contamination transparency beyond evaluation integrity.