Three Numbers and a Precondition: Auditing Multi-Source LLM Benchmarks for Source-Identity Leakage

Tian Xia
Microsoft
tianxia@microsoft.com

Sarp Aykent*
Microsoft
aykentsarp@gmail.com

Abstract

LLM benchmarks often pool rows from several sources (judges, retrieval configurations, training frameworks) and report one headline score. When label rates differ by source, a detector can earn that score by recognizing a row's source, not its quality, and such a score is no promise about a new source. Our audit reports three AUCs of one detector: given features plus source ($\mathrm{AUC}_{\mathrm{full}}$, the headline), features alone, and source alone. Their ratio $L = \mathrm{AUC}_{\mathrm{source}}/\mathrm{AUC}_{\mathrm{full}}$ shows how close the source alone comes to the headline. The precondition: L is read only above a default floor, $\mathrm{AUC}_{\mathrm{full}} \geq 0.65$, because two near-chance AUCs give a ratio near 1 although nothing leaks. On four RAGBench configurations (adherence rates 40.2% to 94.2%) the headline is 0.843 and the configuration label alone reaches 0.811 ($L = 0.96$), above every per-query feature. When five LLM judges grade the same hard-math answers, $L = 0.84$ looks similar, but the headline (0.594) fails the floor: judge identity predicts nothing. The floor, not L, separates the two panels. Six further audits with the floor and a 0.90 band on L frozen add the other outcomes, including a floor-passing panel whose source carries nothing.

1 Introduction

LLM evaluation often rests on per-row quality *features*: an LLM judge's grade of an answer (Li et al., 2026a; Stephan et al., 2025; Han et al., 2025), a metric's score for a response in retrieval-augmented generation (RAG) (Friel et al., 2024; Es et al., 2024), and the training log of a reinforcement-learning (RL) fine-tuning run. A benchmark often pools rows from several *sources* (judges, RAG configurations, training frameworks) and reports one headline score: how well a *detector* built on the features separates good rows from bad. The problem we study is that the detector can earn this score by recognizing the source instead. If one source holds mostly bad rows and the others mostly good ones, knowing where a row came from predicts its label. The cost falls on whoever relies on the features within a single source, such as their own RAG system: there the source is constant, ranks nothing, and the headline no longer says what to expect.

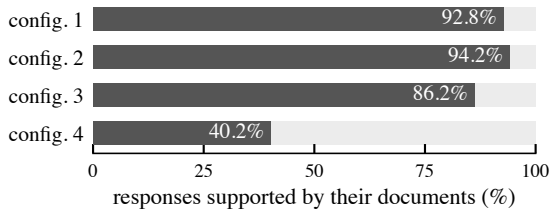


Figure 1: Why the source predicts the label. Adherence rate in the four pooled RAGBench configurations (500 rows each). Scoring each row by its configuration's rate alone reaches AUC 0.811; the best per-query score reaches 0.637 and the headline 0.843.

Take RAGBench (Friel et al., 2024), which publishes per-query scores for RAG responses in several configurations. We pool $2{,}000$ rows from four and predict the published adherence label (is the response supported by the retrieved documents?). The adherence rate is 92.8%, 94.2%, and 86.2% in three configurations and 40.2% in the fourth (Figure 1). Scoring each row by its configuration's rate alone reaches AUC 0.811. The best of the six released per-query scores reaches 0.637, and a simple detector that sees scores and configuration reaches 0.843. Taken alone, 0.843 suggests that the per-query scores track adherence. Yet a detector can nearly match it by answering "which configuration is this row from?" instead of "is this response supported by its documents?" This is a property of pooling, not a defect of RAGBench, which reports no such pooled score.

The confound is known from digital health, where diagnostic classifiers learn to recognize the

*Work done while at Microsoft; now at NVIDIA.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2484–2502
November 6-10, 2026 ©2026 Association for Computational Linguistics

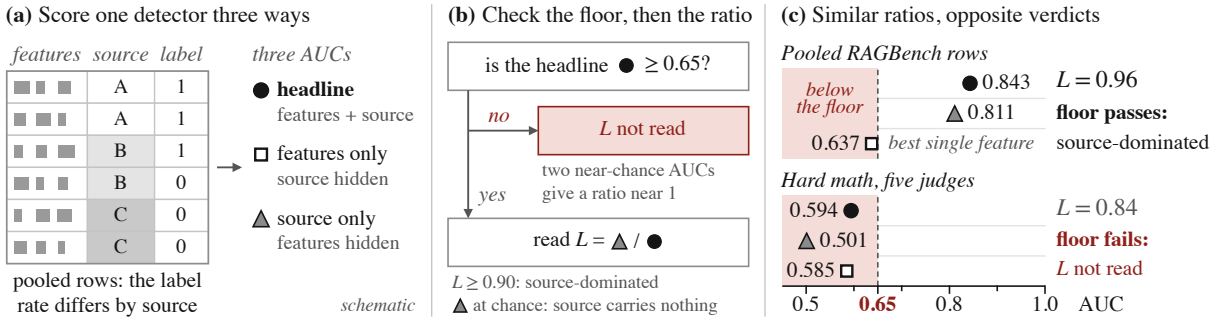


Figure 2: Three numbers and a precondition. (a) Schematic: one detector is scored three ways, given features plus source (the headline, circle), features only (square), and source only (triangle). (b) The ratio L of the source-only AUC to the headline is read only if the headline passes the floor. (c) Two panels with similar L get opposite verdicts, because only one passes the floor (on RAGBench the square is the best single feature).

participant instead of the disease (Saeb et al., 2017; Chaibub Neto et al., 2017). What LLM benchmarks lack is a way to report it. Existing audits ask whether test items reached training data, whether a judge favors certain models, or how far label rates shift across sources (§3). None reports, next to the headline, how close the source label alone comes to it.

The obvious number is the ratio of the source-only AUC to the headline. Alone it misleads, because a detector that knows nothing has AUC 0.5, not 0, and the ratio of two such AUCs is 1. When five LLM judges grade the same 200 hard-math answers, judge identity predicts nothing (AUC 0.501), yet the ratio is 0.84, not far from the 0.96 on RAGBench, because the headline is itself near chance (0.594).

We propose an audit of three numbers and a precondition (Figure 2). The numbers are the AUCs of one detector given features plus source ($\mathrm{AUC}_{\mathrm{full}}$, the headline), features alone ($\mathrm{AUC}_{\mathrm{features}}$), and source alone ($\mathrm{AUC}_{\mathrm{source}}$). The leakage ratio $L = \mathrm{AUC}_{\mathrm{source}}/\mathrm{AUC}_{\mathrm{full}}$ shows how close the source alone comes to the headline. The precondition is a floor: L is read only when $\mathrm{AUC}_{\mathrm{full}} \geq 0.65$. Above the floor, $L \geq 0.90$ is flagged as source-dominated. Both thresholds are defaults chosen on our own *panels*, the pooled row sets we audit. *Source-identity leakage* is an operational label with no causal content: the headline is recoverable from the source label alone, whatever the reason label rates differ (§2). Our contributions are:

1. **The floor, not L, separates the two panels.** Similar ratios ($L = 0.84$ for the judges, $L = 0.96$ on RAGBench) get opposite verdicts, because only one panel passes the floor (§5.1, §5.2).
2. **The report is three AUCs and one reading rule.** The AUCs, L (Eq. 1), and the bound $L \geq 0.5/\mathrm{AUC}_{\mathrm{full}}$, which shows why L cannot be read alone, map to three verdicts (§2, §4).
3. **With both thresholds frozen, six further audits produce each of the rule's outcomes.** They cover answer models, a 1,000-item judge panel, RL frameworks, and second feature sets (Table 1, §5.3). For RL fault panels we add a source-blind reference detector as a reference point, not a winner (§5.4).

The audit began in RL fine-tuning. We wanted to detect failing RL fine-tuning runs from training logs, and pooling a public fault-injection release with healthy public runs from other projects gave a panel in which the source lines up with fault status (§5.4). Section 2 defines the problem, §3 positions the audit, §4 gives the protocol, §5 applies it, and §6 states its limits. We propose no new detector and no new metric.

2 Problem Definition

The question. The audit asks three things of a pooled headline: how close does the source label alone come to it, how close do the features alone come to it, and is it strong enough for those comparisons to mean anything? It does not ask why label rates differ by source, whether any source is better or worse, or whether the benchmark is otherwise sound.

Setting. A multi-source benchmark, audited as a *panel*, is a set of n rows $\{(x_i, s_i, y_i)\}_{i=1}^{n}$. The features $x_i \in \mathbb{R}^d$ are the per-row quality measurements the benchmark publishes (per-query RAG

metrics, an LLM judge's grade, training-log statistics); the ones an auditor uses form the *feature set*. The *source* $s_i \in \{1, \ldots, S\}$ records what produced the row: which judge, which benchmark configuration, which training framework. The label $y_i \in \{0, 1\}$ is the quality outcome (correct, adherent, faulty). A *detector* is any function that scores a row for y, and the *headline* is the AUC of a detector over the pooled rows.

Three AUCs and their ratio. Build the same detector three times and change only what it sees (Figure 2). On two panel types the detector is a fixed score (§4.1). Given features and source, its AUC is the headline, $\mathrm{AUC}_{\text{full}}$. Given features only, its AUC is $\mathrm{AUC}_{\text{features}}$: what the advertised signal achieves alone. Given source only, its AUC is $\mathrm{AUC}_{\text{source}}$: each row is scored with its source's label rate. We call the three the *triad*. The subscript names the detector's inputs; on a given panel we name the source itself ($\mathrm{AUC}_{\text{judge}}$, $\mathrm{AUC}_{\text{config}}$, $\mathrm{AUC}_{\text{framework}}$, $\mathrm{AUC}_{\text{model}}$). The *leakage ratio* compares the source-only AUC with the headline:

$$L = \frac{\mathrm{AUC}_{\text{source}}}{\mathrm{AUC}_{\text{full}}}. \tag{1}$$

On RAGBench, $\mathrm{AUC}_{\text{full}} = 0.843$, $\mathrm{AUC}_{\text{source}} = 0.811$, and $L = 0.96$: the configuration label alone ranks rows almost as well as the full detector.

Why the ratio needs a precondition. An AUC is 0.5, not 0, when a detector knows nothing, and two things follow. First, L is not a share of signal. Scoring rows by their source's label rate never ranks worse than chance in-sample, so $\mathrm{AUC}_{\text{source}} \geq 0.5$ and $L \geq 0.5/\mathrm{AUC}_{\text{full}}$ even when the source is uninformative ($\mathrm{AUC}_{\text{source}}$ is the Mann-Whitney statistic of the per-source label rates; Appendix A). Second, if neither the features nor the source carries signal, both AUCs sit near 0.5 and $L \approx 1$ although nothing leaks. The 200-item judge panel of §5.1 is this case: $\mathrm{AUC}_{\text{judge}} = 0.501$ over $\mathrm{AUC}_{\text{full}} = 0.594$ gives $L = 0.84$, to two decimals the bound $0.5/0.594$. On RAGBench the bound is $0.5/0.843 = 0.59$, far below the observed 0.96. We therefore read L only when the headline passes the *floor*, $\mathrm{AUC}_{\text{full}} \geq 0.65$ (§4.2); below it the audit reports the triad and does not read L.

Terms.

Panel: the pooled rows of one audit.

Source: the label for what produced a row. On RAGBench it is a configuration.

Headline: the pooled AUC, $\mathrm{AUC}_{\text{full}}$, produced by the *full detector*.

Triad: the three AUCs: full, features-only, source-only.

Floor: the precondition $\mathrm{AUC}_{\text{full}} \geq 0.65$ for reading L; a panel *passes* or *fails* it.

Band: above the floor, $L \geq 0.90$ gives the verdict LEAKAGE-CONFIRMED (source-dominated). Both thresholds are defaults.

Verdict: the audit's outcome for a panel (§4.2); what an LLM judge outputs is a *grade*.

Leakage: the headline is recoverable from the source label alone, whatever the reason label rates differ (§5.2).

3 Related Work

Benchmark validity and LLM-evaluation audits. A headline can be inflated by something other than the advertised signal. Test items leak into pretraining data (Xu et al., 2024; Cheng et al., 2025; Chen et al., 2025), simple n-gram classifiers reach high scores on multiple-choice benchmarks without the tested capability (Pacchiardi et al., 2024), and accuracy drops when test sets are rebuilt (Recht et al., 2019). LLM judges favor related student models (Li et al., 2026a) and, on mathematical reasoning, the stronger candidate even when its answer is incorrect (Stephan et al., 2025); other judge biases (position, length, rubric order, shortcut) are studied by Gu et al. (2026), Li et al. (2026b), and Marioriyad et al. (2025), and Lee et al. (2026) propose statistical corrections. For RAG, Friel et al. (2024) and Es et al. (2024) provide per-query metrics, and Dietz et al. (2026) show that near-perfect LLM-judge scores are reachable when prompt templates or gold nuggets leak to the RAG system. These works target specific biases or metrics. Source-identity leakage is a separate problem: a benchmark may appear in no training corpus and carry no shortcut, yet its pooled headline can still be dominated by the source label.

Closest prior work: identity confounding. The contrast we use, scoring one detector with and without access to the source, comes from digital health, where the source is a study participant who contributes many records. Saeb et al. (2017) show that cross-validation with one participant's records in both training and test folds substantially underestimates the error of subject-wise evaluation, and Chaibub Neto et al. (2017, 2019) build permutation tests for how much of a classifier's perfor-

mance comes from recognizing participants. These works report error gaps or permutation p-values; none forms a ratio of AUCs or states how strong the pooled classifier must be before the contrast can be read. We add (i) a numeric precondition, $\mathrm{AUC}_{\text{full}} \geq 0.65$, where Chaibub Neto et al. (2017) state only that their tests "implicitly assume that the classifier is properly tuned"; (ii) three verdicts (§4.2) in place of a binary test outcome; (iii) the $L > 1$ regime as a sign of weak features, not leakage; (iv) application to LLM evaluation, where the source is a judge, configuration, or training framework; and (v) the observation that two panels with similar L receive opposite verdicts, so L cannot be reported alone.

Annotator models and shift statistics. A second line of work models annotator reliability and estimates each annotator's competence (Dawid and Skene, 1979; Whitehill et al., 2009; Hovy et al., 2013; Passonneau and Carpenter, 2014; Rodrigues and Pereira, 2018); disagreement on textual inference is often signal, not noise (Pavlick and Kwiatkowski, 2019). Such per-source competence models are a natural response once the triad shows a source-dominated headline (§6). Shift statistics such as KL divergence (Kullback and Leibler, 1951) and information value measure how far the label rate differs across sources, a form of prior-probability shift (Moreno-Torres et al., 2012). They separate a strongly shifted panel such as RAGBench from a crossed judge panel, where the shift is zero. They are computed from (s, y) alone, so they cannot say how close the source comes to a reported headline, or whether the full detector is strong enough for L to be read. The source-only AUC is a rank summary of the same label rates (Appendix A); the triad adds the comparison with the headline and the precondition, and we recommend reporting a shift statistic together with it, not instead of it. We keep AUC despite its classifier-dependent cost weighting (Hand, 2009); the protocol carries over to other ranking metrics (Appendix E.1).

4 The Audit Protocol

Section 2 defined the three AUCs and L. The protocol fixes how they are estimated (§4.1), when L may be read, and what is reported (§4.2). It needs only per-row features, a source label, and a binary quality label.

4.1 Estimating the three AUCs

Default: one detector, shared folds. The default procedure fits one detector under a shared stratified 5-fold split and reports out-of-fold AUCs for $\mathrm{AUC}_{\text{full}}$ and $\mathrm{AUC}_{\text{features}}$, so the two are directly comparable. We follow it, with Random Forests, on the RL, answer-model, MT-Bench, and MMLU panels. The source-only AUC needs no fitting: it scores each row by its source's empirical rate of $y = 1$ and is always computed in-sample. Every fold contains rows from every source, deliberately: this is the multi-source analogue of record-wise splits (Saeb et al., 2017), the setting in which per-source label rates can be learned. Because $\mathrm{AUC}_{\text{source}}$ is in-sample, L is biased upward when sources are many and rows per source are few (Appendix B).

Two panel types use a fixed score. No model is trained on RAGBench or the judge panels. The headline is the in-sample AUC of a fixed score that adds a per-source term to the feature value (on RAGBench, the row mean of the six scores), with equal weights on RAGBench and on the 200-item judge panel. The features-only entry is the AUC of the feature itself. RAGBench has six features that differ in scale and direction, so its features-only entry is the unweighted mean of the six single-feature AUCs, which is *not* a trained features-only detector. Because the constructions differ, we compare verdicts across panels, not third decimals, and auditors should state which construction they used.

4.2 Precondition: read L only above the floor

A ratio can only be read when its denominator can. We require the full detector to be clearly better than chance: L is read as leakage only if

$$\mathrm{AUC}_{\text{full}} \;\geq\; 0.65,$$

that is, at least 0.15 above chance. When a panel fails this floor, the audit reports the triad and does not read L: the full detector is near chance.

Why 0.65. The floor is an operating default chosen on our own panels, not a universal constant (derivation in Appendix A). Two considerations support it. First, it is far from chance at our panel sizes. Under the Hanley and McNeil (1982) null (a detector at chance), 0.65 sits about 3.6 standard errors above 0.5 on the 200-item judge panel that motivated it (counting items, not grades), and about 2.1 on the smallest panel in Table 1 (113 RL runs).

Panel	Source (levels)	n	AUC full	AUC features	AUC source	L	Floor	Verdict
Original analysis (in hand when the thresholds were chosen)								
RAGBench	configuration (4)	2,000	0.843	0.555*	0.811	**0.96**	pass	CONFIRMED
Hard math, 5 judges, 200 items	judge (5, crossed)	966	0.594	0.585	0.501	(0.84)	fail	TOO WEAK
MT-Bench human judgments	annotator, turn (130)	2,575	0.527	0.499	0.645	(1.22)	fail	TOO WEAK
MMLU, proxy label	subject (57)	14,042	0.509	0.495	0.556	(1.09)	fail	TOO WEAK
Audited afterwards with both thresholds frozen								
RL, two frameworks	framework label (3)	113	0.887	0.881	0.737	0.83	pass	none ($L < 0.90$)
RL, plus TRL	framework label (4)	134	0.911	0.884	0.838	**0.92**	pass	CONFIRMED
Hard math, 3 answer models	answer model (3)	3,456	0.871	0.786	0.741	0.85	pass	none ($L < 0.90$)
Hard math, 5 judges, 1,000 items	judge (5, crossed)	4,999	0.690	0.689	0.500	0.725	pass	RULED OUT
MT-Bench, second feature set	annotator, turn (130)	2,575	0.533	0.500	0.645	(1.21)	fail	TOO WEAK
MMLU, second feature set	subject (57)	5,000	0.520	0.500	0.578	(1.11)	fail	TOO WEAK

Table 1: All ten audits under one rule set (floor $\mathrm{AUC}_{\text{full}} \geq 0.65$; band $L \geq 0.90$). The ratio L is read only when the floor passes: four of the five floor-failing audits have L above 0.90. A parenthesized L is printed but not read; a bold L is at or above the band. Verdicts are those of §4.2 in short form (LEAKAGE-CONFIRMED, LEAKAGE-RULED-OUT, TOO-WEAK-TO-INTERPRET; *none*: the triad is reported with no verdict). A verdict describes the pooled headline, not a benchmark or a source; whether the detector needs the source is read from the features column. Here n counts the rows of each panel (for the judge panels, parsed grades on 200 and 1,000 items). *Unweighted mean of six single-feature AUCs, not a trained features-only detector (best single feature 0.637). Estimation differs by panel (§4.1), so compare verdicts, not third decimals. Row-level caveats are in Appendices B and C.

Second, verdicts do not depend on the exact value within a range: for every audit in Table 1 the point-estimate verdict is unchanged for any floor from 0.60 to 0.68. Just outside that range two panels move: the 200-item judge panel (0.594) would pass a floor of 0.55, and the 1,000-item judge panel (0.690) would fail a floor of 0.70 (under the stricter rule below, its interval, which starts at 0.663, already fails a floor of 0.67).

Near the floor, use the lower end of the interval. The 95% confidence interval (CI) on $\mathrm{AUC}_{\text{full}}$ spans about ± 0.02 at $n = 2{,}000$ rows and ± 0.06 at 200 items (Table 2), so an interval near 0.65 can contain the threshold. For such boundary cases we recommend a stricter form: L is read only if the *lower* end of the 95% interval on $\mathrm{AUC}_{\text{full}}$ exceeds 0.65. Reports use the point-estimate rule by default and add the CI-aware rule when the point estimate lies within one half-width of the threshold.

Above the floor: the band on L. With the floor passed, $L \geq 0.90$ gives the verdict LEAKAGE-CONFIRMED; lower values are reported as the triad without that verdict. This threshold, the *band*, is a reporting default.

At a floor of 0.65 its placement matters for three panels in Table 1 ($L = 0.83$, 0.85, and 0.92), so we print L next to every verdict; the floor, not the band, carries the diagnosis. Two more numbers complete the picture. The bound $0.5/\mathrm{AUC}_{\text{full}}$ (§2) is the part of an observed L that is there by construction. The features-only AUC tells whether the detector *needs* the source. When it is close to $\mathrm{AUC}_{\text{full}}$, the features carry the headline on their own, even if the source label is informative. When it is far below, the headline rests on the source. We do not read L against a permutation null: the one we computed (cf. Pauly et al., 2016), on one RL panel, shows direction only and does not transfer across panels (Appendix D.1).

$L > 1$ flags weak features. The ratio can exceed 1: the source label alone then outranks the detector that also sees the features. This happens when the features carry no quality signal. The full detector, scored out of fold, stays near chance; the source-only score, computed in-sample, fits label noise. A ratio above 1 is a statement about the features. In our panels it occurs only when the floor fails, so the floor catches it before L is read (Appendix B).

The audit in four steps.

1. Fix the inputs: state the source definition and the feature set, preferring features released with the benchmark.
2. Compute the triad and L.
3. Check the floor: if it fails (near the floor, by the CI-aware rule above), the verdict is TOO-WEAK-TO-INTERPRET and L is not read.
4. Otherwise read L: $L \geq 0.90$ gives LEAKAGE-CONFIRMED (source-dominated); $\mathrm{AUC}_{\text{source}}$

at chance, so that L sits at its bound, gives LEAKAGE-RULED-OUT (the source carries nothing); in between, the triad is reported with no verdict.

The report publishes the triad, L, the floor check, the source definition, the feature set, and the verdict (Figure 2). For the RAGBench panel it reads: source, configuration (4 levels); features, the six released per-query scores; $\mathrm{AUC}_{\mathrm{full}} = 0.843$ $[0.825, 0.860]$; best single feature 0.637; $\mathrm{AUC}_{\mathrm{source}} = 0.811$; $L = 0.96$; floor, pass; verdict, LEAKAGE-CONFIRMED. LEAKAGE-CONFIRMED says only that the source label alone ranks rows nearly as well as the full detector; whether the detector *needs* the source is read from $\mathrm{AUC}_{\mathrm{features}}$.

5 Case Studies: Similar L, Opposite Verdicts

This section makes one point: the floor, not L, decides what an audit can say. Two panels carry it (Table 2). The 200-item judge panel has $L = 0.84$ and fails the floor (§5.1); it comes first because it is the reason the floor exists. RAGBench has $L = 0.96$ and passes (§5.2). Six further audits, run with both thresholds frozen, add the rule's other outcomes (§5.3), and the RL panels show where the audit began (§5.4). Table 1 lists all ten audits under one rule set, four of them in hand when the two thresholds were chosen, and Figure 3 plots them.

5.1 The 200-item judge panel: the floor fails

The panel comprises 200 (problem, response, gold answer) items drawn from hard-math problem sets (AIME-24 and MATH-500 level 5), with gold answers re-extracted by regular expression (82 of the 200 responses, 41%, are correct). Each item receives a binary CORRECT/INCORRECT grade from each of five production LLM judges: Claude Opus 4.6, Claude Sonnet 4.6, Claude Haiku 4.5, GPT-4o, and one GPT-5-family reasoning model. After excluding the 3.4% of grades that failed to parse, the panel contains 966 grades. The grade is the only quality feature; the full detector combines it with the judge identity. This panel motivated the floor, so it illustrates the failure mode and does not test the threshold.

The floor fails under both rules. The headline $\mathrm{AUC}_{\mathrm{full}}$ is 0.594, with a 95% interval of $[0.534, 0.654]$ (Table 2). The point estimate is below 0.65, and so is the lower end of the interval that the CI-aware rule uses. Read alone, $L = 0.84$ would suggest judge-identity leakage; under either rule the verdict is TOO-WEAK-TO-INTERPRET: the panel supports no leakage statement.

	Hard math 5 judges, crossed	**RAGBench** 4 configurations
Label rate by source	equal by design	40.2% to 94.2%
$\mathrm{AUC}_{\mathrm{full}}$ (headline)	0.594	0.843
95% interval	[0.534, 0.654]	[0.825, 0.860]
$\mathrm{AUC}_{\mathrm{features}}$	0.585	0.555*
$\mathrm{AUC}_{\mathrm{source}}$	0.501	0.811
Leakage ratio L	0.84	0.96
Smallest possible L	0.84	0.59
Floor (point estimate)	fail	pass
Floor (CI lower end)	fail	pass
Verdict	TOO-WEAK-TO-INTERPRET	LEAKAGE-CONFIRMED

Table 2: The same audit on two panels. The ratios are similar and the verdicts opposite, because only RAGBench passes the floor ($\mathrm{AUC}_{\mathrm{full}} \geq 0.65$); on hard math L is, to two decimals, the smallest value it can take, $0.5/\mathrm{AUC}_{\mathrm{full}}$. Intervals: stratified bootstrap, $B = 1{,}000$ (200 items, 966 grades); analytic Hanley and McNeil (1982) ($n = 2{,}000$, base rate 0.783). *Mean of six single-feature AUCs (best 0.637), not a trained detector.

Why: weak grades and a crossed design. On this item set the grades carry little information: per-judge grade AUC spans 0.548 to 0.598 (Table 3, Appendix C), and the judges say CORRECT 77% to 93% of the time while 41% of answers are correct. These figures describe the item set, not the judges: on the 1,000-item panel of §5.3 the same judges reach accuracies of 0.755 to 0.768. The grade is the only feature, so $\mathrm{AUC}_{\mathrm{full}} = 0.594$ is barely above chance. The source-only $\mathrm{AUC}_{\mathrm{judge}} = 0.501$ sits at chance because of the design, not because the judges are weak. The design is *crossed*: every judge grades the same 200 items, so the true-label rate is the same for every judge. The small departure from 0.5 comes from the 3.4% of grades that failed to parse. The ratio $L = 0.84$ is therefore, to two decimals, the bound $0.5/0.594$: the panel carries no source signal, and L is high only because its denominator is near chance.

5.2 RAGBench: the floor passes and L is high

We pool four RAGBench configurations (Friel et al., 2024): HotpotQA, DelucionQA, EManual, and ExpertQA. Each has its own document corpus, and we draw 500 rows from each, $n = 2{,}000$ in total. The label is the published adherence indicator (response supported by retrieved documents),

with positive rate $1{,}567/2{,}000 = 78.3\%$. That rate differs sharply by configuration: HotpotQA 92.8%, DelucionQA 94.2%, EManual 86.2%, and ExpertQA 40.2%. One low-adherence configuration among three high-adherence ones is what the audit detects. The full detector adds the per-configuration adherence rate to the six per-query scores released with the benchmark: relevance, utilization, completeness, and a baseline LLM judge's adherence, context-relevance and utilization predictions (§4.1). The audit only re-analyzes published scores; RAGBench itself does not publish a pooled headline.

The floor passes and $L = 0.96$. The headline is $\mathrm{AUC}_{\mathrm{full}} = 0.843$, and the lower end of its 95% interval (0.825) is 0.175 above the floor (Table 2). The ratio can therefore be read, and $L = 0.96$ is above the band, so the verdict is LEAKAGE-CONFIRMED: the configuration label alone reaches AUC 0.811 against a headline of 0.843, although L could have been as low as 0.59.

The strongest single feature, `utilization_score`, reaches AUC 0.637; the other five range from 0.480 (`gpt3_context_relevance`) to 0.611 (`completeness`), and the unweighted mean of the six is 0.555. *Every* feature is weaker than the configuration-only score (0.811), which follows in closed form from the four adherence rates (Appendix A).

A detector given the configuration label can answer "is this row from ExpertQA?" instead of "is this response supported by its documents?" This is not a defect of RAGBench, which does what it advertises: per-query scores across heterogeneous configurations. The problem is a pooled headline over them, which would not measure what readers take it to measure.

The verdict does not depend on why the rates differ. The audit does not say *why* ExpertQA's adherence rate is 40.2%. Configuration identity may be a confounder (the adherence annotation behaves differently on that configuration) or a proxy for a real quality gap (responses there are less faithful to the retrieved documents). Our re-analysis cannot separate the two, and the verdict does not depend on the answer. Under either explanation the configuration label alone reaches AUC 0.811 while the best per-query feature reaches 0.637. A pooled headline of 0.843 therefore does not say how well these scores would rank responses inside one new configuration. If the gap is real, the accurate report

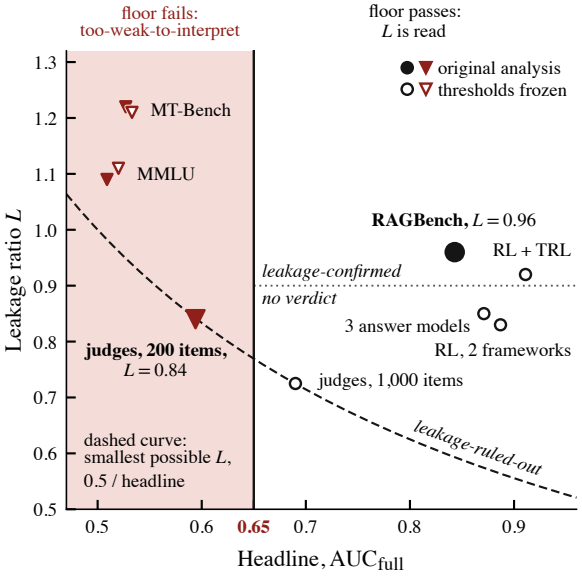


Figure 3: Leakage ratio L against the headline $\mathrm{AUC}_{\mathrm{full}}$ for the ten audits of Table 1. High L occurs on both sides of the floor: horizontal position, not height, decides whether L is read. Triangles fail the floor and circles pass it; hollow marks were audited with both thresholds frozen (for MT-Bench and MMLU, the second feature set). The dotted line is the band, $L \geq 0.90$. No audit can lie below the dashed curve, and the two crossed judge panels lie on it.

is one number per configuration, not a pooled headline. Nor is every floor-passing panel given that verdict: the $1{,}000$-item judge panel is LEAKAGE-RULED-OUT, and the answer-model panel, whose source is informative, gets no verdict (§5.3).

5.3 Six further audits with both thresholds frozen

The first two panels have similar leakage ratios and opposite verdicts, and the remedies differ: a floor failure calls for more informative judges or features, and LEAKAGE-CONFIRMED calls for per-source reporting or reweighting (Appendix E.2). Both thresholds were chosen with those panels in hand, so we froze them and ran six further audits (Table 1, lower block; Figure 3; Appendix C). Across all ten audits, high L values occur on both sides of the floor, so the floor check has to come before L is read.

Two are new LLM-evaluation panels. In the *answer-model panel* the source is the model that produced each response (3 models, $3{,}456$ responses to hard-math problems, with API and run-metadata features). The floor passes ($\mathrm{AUC}_{\mathrm{full}} = 0.871$), the features alone reach 0.786 (an upper bound on source-blind signal: token counts de-

pend on the tokenizer, and folds are row-level, so one problem can fall on both sides of a fold), and $L = 0.85$ sits below the band, so we report the triad with no verdict. Here both the source ($\mathrm{AUC}_{\mathrm{model}} = 0.741$) and the features are informative, and the audit reports both without choosing between them. In the 1,000-*item judge panel* the five judges of §5.1 grade 1,000 items from the same hard-math problem sets (4,999 parsed grades): $\mathrm{AUC}_{\mathrm{full}} = 0.690$ [0.663, 0.715], $\mathrm{AUC}_{\mathrm{judge}} = 0.500$, $L = 0.725$. The floor passes and judge identity carries nothing: the verdict is LEAKAGE-RULED-OUT.

Caveats on the 1,000-item judge panel. First, we had planned this scale-up expecting $\mathrm{AUC}_{\mathrm{full}}$ to stay below the floor; it did not. Second, the two item sets differ. This one is 62.2% correct against 41% at 200 items, and the judges are informative on it (per-judge accuracy 0.755 to 0.768). The comparison is therefore not like-for-like, and the change of floor verdict is not an effect of n alone; auditors should report how items were sampled next to the triad. Third, every judge grades every item, so $\mathrm{AUC}_{\mathrm{judge}} = 0.5$ in any crossed panel and the LEAKAGE-RULED-OUT verdict holds by design. It shows that a pooled headline cannot gain from judge identity under this design, not that individual judges are unbiased. The ratio then sits at its bound $0.5/\mathrm{AUC}_{\mathrm{full}}$, which is 0.725 here and 0.84 at 200 items: the value that looks high in §5.1 is, to two decimals, the smallest L could take there.

The last two audits repeat MT-Bench and MMLU under a second feature set written without access to the first (Appendix C.1); both still fail the floor. All four MT-Bench and MMLU audits have L above 1 (1.09 to 1.22): by the band alone they would count as source-dominated, which is the misreading the floor prevents. The remaining two frozen-threshold audits are the RL panels, which the next subsection describes (§5.4).

5.4 Where the audit began: RL training runs

To detect failing RL fine-tuning runs from their training logs, we pooled a fault-injection release for the OpenRLHF framework (Zhang et al., 2026) with healthy public runs from the verl framework (Yu et al., 2025; Luo et al., 2025). The source label then lines up with fault status: every faulty run comes from one release (Appendix D). Features can carry source clues too: the OpenRLHF runs have 20 to 40 logged steps, the verl runs 35 to 999.

On the unfiltered runs (117, with source labels taken from run names) all 96 faulty runs share one label, so the label restates fault status by construction. A logistic regression scores 1.000 with features and source and 0.907 with features only, and the source alone scores 1.000. Shuffling the source labels 1,000 times gives a ratio with mean 0.635 and maximum 0.756, against the observed 1.000 (Appendix D.1). This shows direction only, so we turn to panels in which the label is the framework.

The detector barely uses the framework label. On the two-framework panel (113 runs: 19 healthy and 94 faulty) the triad is $\mathrm{AUC}_{\mathrm{full}} = 0.887$, $\mathrm{AUC}_{\mathrm{features}} = 0.881$, $\mathrm{AUC}_{\mathrm{framework}} = 0.737$, $L = 0.83$: the floor passes and L sits below the band. Adding runs trained with a third framework, TRL (134 runs: 35 healthy and 99 faulty), gives 0.911, 0.884, 0.838, and $L = 0.92$, above the band. Under the rule of §4.2 the three-framework panel is LEAKAGE-CONFIRMED, and we report it that way. The two panels still tell the same story once all three AUCs are considered. The framework label alone is informative because fault labels are concentrated in one framework. Yet removing it from the trained detector moves the AUC only from 0.887 to 0.881 (0.911 to 0.884 with TRL). The triad separates what source identity *could* explain from what the detector actually uses. Because fault status and framework are nearly collinear in these panels, we treat the RL results as direction, not magnitude, and they are not evidence that any framework causes failures.

A source-blind reference detector. The triad does not say what a detector that never sees the source would score. We supply that reference on the single-framework RL panel (104 OpenRLHF runs: 10 healthy and 94 faulty). It is given in closed form: the squared Mahalanobis distance of a run's training-log features from the healthy runs, thresholded at the χ^2 95% point, with no parameter fit to fault labels (Mahalanobis-distance scoring is standard; Lee et al., 2018; Linderman et al., 2026). It reaches AUC 0.885, scored in-sample on the healthy runs that define it; a trained Random Forest on the same features reaches 0.81 under a different evaluation protocol (Table 5, Appendix D.1). Part of the 0.885 comes from run length alone, because the healthy runs share one step count. The detector is a reference point, not a winner: with held-out healthy runs from a framework-mixed pool its AUC is 0.55 to 0.60 against about 0.88 for a Random Forest. On separate healthy runs its χ^2 threshold

gives a held-out per-step false-alarm rate of 3.60% against a nominal 5% (Appendix D.2).

6 Discussion

What the audit tells a benchmark author. How close does the source alone come to the headline? The audit costs three AUCs and one threshold check, and it gives three answers. Below the floor, the detector is too weak and needs more informative features or judges. Above it with L high, the source alone nearly matches the headline, so the score is no promise about a single source: report per source. Above it with the source at chance, the headline borrows nothing from the source. In between, the triad is reported without a verdict. A bare ratio cannot tell the first from the second: it sends the 200-item judge panel to a source-level correction.

What was planned in advance. The 0.65 floor was formulated with the 200-item judge panel's result in hand, so that panel illustrates the failure mode and does not test the threshold; the contrast with RAGBench (Table 2) carries the main-body claim. Three expectations were written down before the corresponding runs, in an internal plan that was not deposited in a public registry, and all three were missed. MT-Bench and MMLU were planned as applications of the triad with target values of $\mathrm{AUC}_{\mathrm{full}}$ that neither reached; we report them in Appendix B as cases in which the floor fails in other benchmark genres, not as a calibration of the threshold. The $1{,}000$-item judge panel was planned to stay below the floor and passed it (§5.3).

What the audit does not claim. The protocol proposes no new detector (Mahalanobis-distance scoring is standard, Lee et al., 2018), no new AUC critique (Hand, 2009), and no new decomposition. The with-and-without-identity contrast comes from clinical machine learning (Saeb et al., 2017; Chaibub Neto et al., 2017). The audit does not remove source heterogeneity, since per-source label-rate differences are intrinsic to pooling, and it does not claim that any configuration, judge, model, or framework is better or worse, or that RAGBench is broken. A verdict describes a pooled headline, not any single source. Verdicts depend on the auditor's feature set. A LEAKAGE-CONFIRMED verdict is therefore a statement about the audited detector, features included: when the source-only AUC passes the floor on its own, as on RAGBench, a weaker feature set keeps the floor verdict and pushes L toward 1. The $L > 1$ regime is the built-in signal that the features have failed, and auditors should reuse feature sets published with a benchmark where they exist, as the RAGBench panel does, before engineering their own (Appendix C.1). Our panels suggest starting points where none is published: for judge panels, the grade itself; when the source is the responding model, API and run metadata, with tokenizer-dependent counts noted; for RL panels, training-log features, with run length checked as a possible source clue. Surface features of prompts and options carried no signal on MT-Bench or MMLU.

What to do when leakage is confirmed. When the verdict is LEAKAGE-CONFIRMED, the standard responses apply: per-source stratified reporting, base-rate reweighting, or per-source competence models (Whitehill et al., 2009; Hovy et al., 2013). The first two act on the source term by construction. Within a single source the source score is constant, so $\mathrm{AUC}_{\mathrm{source}} = 0.5$ and per-source reporting removes the source term without any modeling. Reweighting rows so that every source has the pooled label rate does the same, at a cost in variance. What the features achieve under either response has to be measured on the rows, which we do not report (Appendix E.2). The protocol itself is not specific to AUC or to binary labels: other ranking metrics fit the same structure once the threshold is recalibrated, and multi-class and regression variants are sketched, untested, in Appendix E.

7 Conclusion

A pooled benchmark score can be earned by recognizing the source. Three AUCs of one detector show how close the source alone comes to it, but only when the headline is clearly above chance. On RAGBench the configuration label alone reaches AUC 0.811 against the headline 0.843 ($L = 0.96$), so the headline does not say how well the per-query scores rank rows inside one configuration. On the 200-item judge panel a similar ratio ($L = 0.84$) is, to two decimals, the smallest value L can take over a near-chance headline, so the audit says only that the detector is too weak. The floor, not L, separates the two. Six audits with both thresholds frozen add the other outcomes. Both thresholds are defaults chosen on our own panels, so we ask benchmark authors for the numbers: three AUCs and a floor check next to the headline.

Limitations

The 0.65 floor and the $L \geq 0.90$ band are operating defaults chosen on our own panels. The six frozen-threshold audits apply them unchanged, but we selected and ran those audits ourselves on related data; one passes the floor narrowly ($\mathrm{AUC}_{\mathrm{full}} = 0.690$), the band decides the verdict of three panels, and no third party has audited a panel. Source coverage is small: five judges on two crossed item sets, where $\mathrm{AUC}_{\mathrm{judge}} = 0.5$ by design, three answer models, four RAGBench configurations, and RL panels whose faults are concentrated in one framework. The permutation null comes from one RL panel and shows direction only. Feature sets differ by panel and several were written by us, so floor verdicts may partly track feature provenance. Estimation is not uniform across panels (§4.1), and we report no row-level mitigation results (Appendix E.2).

Ethical Considerations

Human subjects. This paper proposes a methodological protocol for benchmark auditing; it does not introduce new datasets of human-subject data. The LLM-as-judge audit (§5.1) was run against publicly hosted hard-math problems with no personally identifiable information. The RAGBench audit (§5.2) only re-analyzes published scores, and the MT-Bench audit re-analyzes published human preference judgments. No human annotators were employed.

Societal impact: model selection and leaderboard interpretation. Headline numbers on multi-source benchmarks drive decisions made far from the paper that reports them: which judge model a team adopts, which RAG metric gates a release, which fault detector a training pipeline trusts, and how a leaderboard orders its entries. When a pooled headline is mostly recoverable from source identity, those decisions inherit a number that may not transfer to the single-source setting in which most practitioners deploy. The cost falls on downstream users who cannot rerun the benchmark. Publishing the triad next to the headline lets a reader discount it before acting; that is the benefit we intend.

The same verdicts can be over-read, and we see three risks. First, passing the floor, or a LEAKAGE-RULED-OUT verdict, is not a certificate of benchmark validity. That verdict says only that, under the auditor's source definition and feature set, source identity alone does not recover the headline. It is silent on test-set contamination, label noise, construct validity, shortcuts within a source, and any source variable the auditor did not encode. We therefore ask that audits be reported as numbers (the triad, L, the floor check, and the source definition and feature set used) and not as a pass badge on a leaderboard or data card.

Second, a LEAKAGE-CONFIRMED verdict describes a pooled headline; it is not a judgment of the benchmark's maintainers or of any single source. Per-source label rates can differ because the sources differ in real quality (§5.2), so the verdict should not be used to rank or disparage individual judges, configurations, models, or vendors.

Third, the 0.65 floor and the 0.90 band are operating defaults chosen on a small panel set (see Limitations). Hardening them into acceptance criteria for benchmark papers or leaderboard entries would turn a reporting aid into a gate it has not been validated to be. A floor failure in particular reflects the auditor's detector and features as much as the benchmark, and should prompt better features before it prompts dismissal of the benchmark.

Dual use. The audit protocol is symmetric. By telling reviewers which three numbers and one precondition to inspect, it tells benchmark authors which numbers a hostile construction must control. A determined adversary could inflate $\mathrm{AUC}_{\mathrm{full}}$ above 0.65 with one informative engineered feature while leaving $\mathrm{AUC}_{\mathrm{source}}$ high, so that the leakage ratio looks unremarkable even though source identity still dominates the headline.

We acknowledge this risk and argue the trade-off favors transparency. An author who publishes the audit commits to numbers that others can check, which rewards benchmarks built to pass it honestly, and reviewers will recognize the specific gaming patterns once they are documented. The alternative, in which only some auditors know what to compute, is worse.

Public-data provenance and responsible disclosure. All third-party data used in this paper are public: RAGBench, RFT-FaultBench, MT-Bench, MMLU, AIME-24, and MATH-500. The TRL runs, the three healthy runs of the false-alarm check, and the judge grades are our own. The healthy verl runs (Appendix D.1) were released through the originating projects' public channels; no private-project access was required. We have not audited

proprietary benchmarks; all case studies use public benchmarks.

Generative AI Usage Disclosure

Generative AI assistants were used for writing edits and code drafting. No scientific contributions originate from AI. All results were verified by the authors.

References

Elias Chaibub Neto, Abhishek Pratap, Thanneer M. Perumal, Meghasyam Tummalacherla, Brian M. Bot, Andrew D. Trister, Stephen H. Friend, Lara Mangravite, and Larsson Omberg. 2017. Learning disease vs participant signatures: A permutation test approach to detect identity confounding in machine learning diagnostic applications. *Preprint*, arXiv:1712.03120.

Elias Chaibub Neto, Abhishek Pratap, Thanneer M. Perumal, Meghasyam Tummalacherla, Phil Snyder, Brian M. Bot, Andrew D. Trister, Stephen H. Friend, Lara Mangravite, and Larsson Omberg. 2019. Detecting the impact of subject characteristics on machine learning-based diagnostic applications. *npj Digital Medicine*, 2(1):99.

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. *Preprint*, arXiv:2502.14425.

Alexander Philip Dawid and Allan M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28.

Laura Dietz, Bryan Li, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026. Insider knowledge: How much can RAG systems gain from evaluation secrets? In *Advances in Information Retrieval: 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, March 29–April 2, 2026, Proceedings, Part I*, volume 16483 of *Lecture Notes in Computer Science*, pages 582–597, Cham. Springer Nature Switzerland.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. RAGBench: Explainable benchmark for retrieval-augmented generation systems. *Preprint*, arXiv:2407.11005.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. A survey on LLM-as-a-judge. *The Innovation*, 7(6):101253.

Steve Han, Gilberto Titericz Junior, Tom Balough, and Wenfei Zhou. 2025. Judge's Verdict: A comprehensive analysis of LLM judge capability through human agreement. *Preprint*, arXiv:2510.09738.

David J. Hand. 2009. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123.

James A. Hanley and Barbara J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

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*.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Chungpa Lee, Thomas Zeng, Jongwon Jeong, Jy-yong Sohn, and Kangwook Lee. 2026. How to correctly report LLM-as-a-judge evaluations. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 65231–65252. PMLR.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7167–7177. Curran Associates, Inc.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2026a. Preference leakage: A contamination problem in LLM-as-a-judge. In *International Conference on Learning Representations*, pages 60486–60511.

Qingquan Li, Shaoyu Dou, Kailai Shao, Chao Chen, and Haixiang Hu. 2026b. Evaluating scoring bias in LLM-as-a-judge. In *Database Systems for Advanced Applications: 31st International Conference, DASFAA 2026, Jeju, South Korea, April 27–30, 2026, Proceedings, Part IV*, volume 16538 of *Lecture Notes in Computer Science*, pages 19–34, Singapore. Springer Nature Singapore.

Randolph W. Linderman, Noah Cowan, Yiran Chen, and Scott W. Linderman. 2026. A Bayesian nonparametric perspective on Mahalanobis distance for out of distribution detection. *Transactions on Machine Learning Research*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL. Notion blog post.

Arash Marioriyad, Mohammad Hossein Rohban, and Mahdieh Soleymani Baghshah. 2025. The silent judge: Unacknowledged shortcut bias in LLM-as-a-judge. In *NeurIPS 2025 Workshop: Reliable ML from Unreliable Data*.

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.

Lorenzo Pacchiardi, Marko Tešić, Lucy G. Cheke, and José Hernández-Orallo. 2024. Leaving the barn door open for Clever Hans: Simple features predict LLM benchmark answers. *Preprint*, arXiv:2410.11672.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

Markus Pauly, Thomas Asendorf, and Frank Konietschke. 2016. Permutation-based inference for the AUC: A unified approach for continuous and discontinuous data. *Biometrical Journal*, 58(6):1319–1337.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR.

Filipe Rodrigues and Francisco C. Pereira. 2018. Deep learning from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 1611–1618.

Sohrab Saeb, Luca Lonini, Arun Jayaraman, David C. Mohr, and Konrad P. Kording. 2017. The need to approximate the use-case in clinical machine learning. *GigaScience*, 6(5):gix019.

Andreas Stephan, Dawei Zhu, Matthias Aßenmacher, Xiaoyu Shen, and Benjamin Roth. 2025. From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 759–773, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, volume 22, pages 2035–2043. Curran Associates, Inc.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc.

Lingzhe Zhang, Tong Jia, Yunpeng Zhai, Liancheng Fang, Kening Zheng, Hongyi Liu, Xiaosong Huang, Philip S. Yu, and Ying Li. 2026. Towards robust LLM post-training: Automatic failure management for reinforcement fine-tuning. *Preprint*, arXiv:2605.04431.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Formal Notes: The Floor, the Bound, and the Source-Only AUC

These notes support §2 and §4.2.

The floor under the null. The 0.65 floor is motivated by the standard error of an AUC under the null model of Hanley and McNeil (1982). With true AUC $A = 0.5$ and base rate π, that standard error is $\mathrm{SE}_0(A = 0.5; n, \pi) = \sqrt{(n+1)/(12\, n_+\, n_-)}$, where $n_+ = \pi n$ and $n_- = (1-\pi)n$. For the 200-item judge panel ($\pi = 0.41$; we take the 200

items, not the 966 grades, as n, since grades on one item are not independent), $\mathrm{SE}_0 = 0.042$, so 0.65 sits $(0.65 - 0.5)/0.042 \approx 3.6$ standard errors above chance. For the two-framework RL panel ($n = 113$; 19 healthy and 94 faulty runs) the standard error under the null is 0.073, and the same threshold sits about 2.1 standard errors above chance. This is the least favorable case among the ten audits. Interpreting 0.65 as "clearly above chance at the panel's own n" is therefore well supported on the LLM panels and only indicative on the small RL panels.

Floor sweep. For every audit in Table 1 the point-estimate verdict is unchanged for any floor from 0.60 to 0.68. The floor-failing audits (0.594, 0.527, 0.509, and the two second-feature-set audits at 0.533 and 0.520) fail throughout that range, and the floor-passing ones (0.690 and above) pass throughout it. A floor of 0.55 would pass the 200-item judge panel, and a floor of 0.70 would fail the 1,000-item judge panel. The critique of Hand (2009), that AUC comparisons embed a classifier-dependent cost weighting, is a further reason not to interpret fine differences between near-chance AUCs.

Closed form of the source-only AUC, and the bound on L. The source-only AUC is itself a label-shift statistic with a closed form. Index the sources in decreasing order of their rate of $y = 1$, and let P_a and N_a be the numbers of positive and negative rows in source a, with totals P and N. Then

$$\mathrm{AUC}_{\text{source}} = \frac{\sum_{a<b} P_a N_b + \frac{1}{2}\sum_a P_a N_a}{PN}, \quad (2)$$

the Mann-Whitney statistic of the per-source label rate with within-source ties counted as one half. On RAGBench the four adherence rates of §5.2 (500 rows each) give 0.8106, the 0.811 of Table 2. Because the numerator never falls below $\frac{1}{2}PN$ under this ordering, $\mathrm{AUC}_{\text{source}} \geq 0.5$ and $L \geq 0.5/\mathrm{AUC}_{\text{full}}$. The smallest possible L is $0.5/0.594 = 0.84$ on the 200-item judge panel, where it is attained to two decimals, and $0.5/0.843 = 0.59$ on RAGBench, where the observed value is 0.96. When every source has the same label rate, as in a crossed design with no missing rows, all pairs across sources are ties and $\mathrm{AUC}_{\text{source}} = 0.5$ exactly. The source-label permutation null, used as a direction check only, is in Appendix D.1.

B Two Planned Panels That Fail the Floor (MT-Bench, MMLU)

This appendix reports two LLM-evaluation panels (MT-Bench, MMLU) that we had planned in advance as applications of the triad, with target values written down before the runs. Both missed their planned targets and fail the floor, and we report them as additional floor-failing cases and not as demonstrations of the triad. The reframing was made after the results were known and is acknowledged as such. Both panels have $L > 1$: the auditor's per-row feature set on these benchmarks does not carry quality signal beyond noise. The main-body claims do not rest on these two panels.

The $L > 1$ regime. A ratio above one means $\mathrm{AUC}_{\text{source}} > \mathrm{AUC}_{\text{full}}$; it arises here from two things together. First, the per-row feature set carries no quality signal beyond noise, so the out-of-fold full detector stays near chance. Second, $\mathrm{AUC}_{\text{source}}$ is an in-sample statistic (§4.1), so with many sources and few rows per source the per-source label rates fit label noise (130 sources over 2,575 rows on MT-Bench, $L = 1.22$; 57 over 14,042 on MMLU, $L = 1.09$). This is a finite-sample effect and we claim no population ordering of the two AUCs; the protocol's correctness in this regime rests on the floor failing before L is read. The diagnostic value is asymmetric: $L > 1$ with the floor failing identifies a feature-engineering failure, where the response is to improve the features, not to question source identity.

B.1 MT-Bench human judgments

We apply the protocol to the MT-Bench human preference judgments (Zheng et al., 2023). The audit treats each pairwise judgment as a row. The source is the (human annotator, turn) pair (130 sources over 2,575 judgments) and the label is model-B-wins. The per-judgment features are length, hedge, refusal, and verbosity proxies computed from the two candidate responses. The triad is $\mathrm{AUC}_{\text{full}} = 0.527$, $\mathrm{AUC}_{\text{features}} = 0.499$, $\mathrm{AUC}_{\text{source}} = 0.645$, $L = 1.22$. The floor fails ($0.527 < 0.65$), so L is not read as leakage. The observed $L > 1$ says that these features carry no quality signal beyond noise, so the full detector underperforms the in-sample source-only score. This points to the features, not to annotator sampling: a different feature set on the same benchmark could pass the floor, although a second one we tried did not (Appendix C.1).

Planned target (missed). We planned this panel as a floor-passing case with L below the band (targets $\mathrm{AUC}_{\text{full}} \geq 0.70$ and $L \in [0.70, 0.85]$). The observed $\mathrm{AUC}_{\text{full}} = 0.527$ and $L = 1.22$ both miss the planned range. We draw no audit-triad conclusion from this panel.

B.2 MMLU multi-subject audit

For a multiple-choice probe we audit MMLU (Hendrycks et al., 2021) with subject as the source (57 subjects, 14,042 retained test items) and per-item proxies (token count, options-count distribution, language) as features. The label is the indicator that the gold answer is option A. No model outputs were scored, so this panel audits answer-position structure across subjects, not model correctness; it is a proxy. The triad is $\mathrm{AUC}_{\text{full}} = 0.509$, $\mathrm{AUC}_{\text{features}} = 0.495$, $\mathrm{AUC}_{\text{source}} = 0.556$, $L = 1.09$. The floor fails ($0.509 < 0.65$), and $L > 1$ places this panel in the same regime as MT-Bench: the per-item proxies do not predict the answer-position label. This points to the features and the label choice, not to the benchmark's auditability.

Planned target (missed). We planned this panel, paired with the MT-Bench panel, as the expected LEAKAGE-CONFIRMED case, with targets $\mathrm{AUC}_{\text{full}} \geq 0.85$ and $L \in [0.92, 0.98]$. The dataset was changed from AGIEval to MMLU, and the observed $\mathrm{AUC}_{\text{full}} = 0.509$ and $L = 1.09$ miss both planned ranges. We do not retain the planned below-band versus above-band contrast as a finding; we retain the panel only as evidence that the floor fails on weak-signal panels under an answer-position label.

C Panels Audited with Frozen Thresholds

The 0.65 floor and the $L \geq 0.90$ band were fixed with the original panels in hand. We then ran the six audits in the lower block of Table 1 without changing either threshold or the rules for reading L. "Frozen" here means that a panel played no part in choosing the thresholds. It does not mean independent data throughout. The two hard-math panels draw on the same problem sets as the 200-item panel of §5.1, and the two second-feature-set audits reuse MT-Bench and MMLU rows. The two-framework RL panel reuses the RFT-FaultBench runs, and the TRL panel adds our own later TRL runs to them. All values are point estimates; we print an interval only for the 1,000-item judge panel (a bootstrap over items).

Judge	Grade AUC	Accuracy
Claude Haiku 4.5	0.548	0.467
Claude Opus 4.6	0.595	0.538
Claude Sonnet 4.6	0.585	0.530
GPT-4o	0.598	0.545
GPT-5-family (reasoning)	0.592	0.540

Table 3: On the 200-item judge panel of §5.1 (41% of answers correct) every judge's grades are weak: all five judges are within 5 percentage points of 50% accuracy and below the 59% majority-class baseline. Judges are in alphabetical order. The table describes the item set, not the judges: on the 1,000-item judge panel the same judges reach accuracies of 0.755 to 0.768.

Hard math, three answer models. The source is the model that produced each response (3,456 responses from three answer models on hard-math problems). The label is whether the extracted final answer equals the gold answer. The features are API and run metadata only (for example, temperature, token counts, latency, and retries), and the detector is an out-of-fold Random Forest under a shared 5-fold split. The triad is $\mathrm{AUC}_{\text{full}} = 0.871$, $\mathrm{AUC}_{\text{features}} = 0.786$, $\mathrm{AUC}_{\text{model}} = 0.741$, $L = 0.85$. The floor passes and L is below the band. Per-model correct rates differ on this panel, which is a property of the panel and of the answer-extraction rule and not a ranking claim. Two caveats apply to the features-only value. Folds are row-level, so the same problem can appear on both sides of a fold. Prompt token counts depend on the tokenizer, so the features-only detector is not fully blind to the source. We treat 0.786 as an upper bound on source-blind feature signal.

Hard math, five judges, 1,000 items. We drew 1,000 (problem, response, gold answer) items from the same hard-math problem sets and collected grades from the five judges of §5.1; 4,999 grades parsed. The draw is 62.2% correct, against 41% at 200 items, and per-judge accuracy spans 0.755 to 0.768. The triad is $\mathrm{AUC}_{\text{full}} = 0.690$ [0.663, 0.715], $\mathrm{AUC}_{\text{features}} = 0.689$ (the grade alone), $\mathrm{AUC}_{\text{judge}} = 0.500$, $L = 0.725$. Four points qualify the comparison with §5.1. (i) *Plan.* We had planned this scale-up expecting $\mathrm{AUC}_{\text{full}}$ to stay below the floor. It did not. (ii) *Score construction.* As at 200 items, no detector is fit: the headline is the AUC of a fixed, untrained combination of the binary grade and a per-judge term. With one

Panel	Set	n	AUC full	AUC features	AUC source	L
MT-Bench	1	2,575	0.527	0.499	0.645	(1.22)
	2	2,575	0.533	0.500	0.645	(1.21)
MMLU	1	14,042	0.509	0.495	0.556	(1.09)
	2	5,000	0.520	0.500	0.578	(1.11)

Table 4: The floor verdict does not move under a second feature set: all four rows fail the floor, so L (in parentheses) is not read. Set 1 rows repeat Appendix B. MMLU set 2 is computed on a seeded 5,000-item subsample, so its source-only AUC is not comparable with the full-panel value.

binary feature, a fitted detector could only reorder the ten (grade, judge) cells. (iii) *Item set.* The two draws differ in difficulty as well as in size, so the change from a floor failure to a floor pass is not an effect of n alone. (iv) *Crossed design.* Every judge grades the same items, so the rate of the true label is the same for every judge and $\mathrm{AUC}_{\text{judge}} = 0.5$ holds by design. The LEAKAGE-RULED-OUT verdict certifies that the pooled headline cannot borrow from judge identity under this design. It is not evidence about individual judge biases, which a crossed panel cannot expose through the source-only score.

RL training frameworks. The two RL rows are described in Appendix D.1. Their source levels are framework labels (the two-framework panel has two verl-side labels and one OpenRLHF label).

How stable are the two thresholds on these panels? The floor sweep is in Appendix A: no point-estimate verdict changes for any floor from 0.60 to 0.68. The 1,000-item judge panel is the boundary case ($\mathrm{AUC}_{\text{full}} = 0.690$, interval lower end 0.663). The band is less stable than the floor. At a floor of 0.65 its placement decides the verdict of three panels: the two-framework RL panel ($L = 0.83$, below), the three-framework RL panel (0.92, above), and the answer-model panel (0.85, below). We therefore present the floor as the component that carries the diagnosis and the band as a reporting default, which is what the contrast of §5 already supports.

C.1 Feature-set sensitivity on MT-Bench and MMLU

Features-only AUC depends on the auditor's feature set, so we re-audited both floor-failing panels under a second set (Table 4). The second set was specified without access to the first set or its results, from the raw schema and the label definitions only. No feature definition is shared between the two sets, but they are not conceptually disjoint: surface properties such as response length recur under different encodings. The headline moves from 0.527 to 0.533 on MT-Bench and from 0.509 to 0.520 on MMLU, and features-only AUC stays at chance. On MT-Bench the rows and the 130 sources are unchanged, so the source-only AUC is identical by construction. On MMLU the second audit uses a seeded subsample, so the in-sample source score is fit on fewer items per subject and the change in L mixes a subsample effect with the feature-set effect.

What this does and does not show. On these two panels the floor verdict did not depend on the feature set. We do not claim invariance in general. When the source-only AUC is itself below the floor, as here (0.645 on MT-Bench; 0.556 and 0.578 on MMLU), a feature set without signal cannot lift the panel over the floor. The error a weak feature set causes is then a TOO-WEAK-TO-INTERPRET verdict on a panel that a better feature set might pass. When the source-only AUC passes the floor on its own, as on RAGBench (0.811), a weaker feature set leaves the floor verdict at a pass and pushes L toward 1. A LEAKAGE-CONFIRMED verdict is therefore a statement about the audited detector, features included, which is why we ask auditors to publish the feature set with the triad.

Starting feature sets. The panels in this paper suggest the following defaults, in order of preference. (i) Per-row metrics released with the benchmark (RAGBench: the six per-query scores of §5.2). (ii) For judge panels, the grade itself, binary or multi-level (§5.1). (iii) For panels whose source is the responding model, API and run metadata such as temperature, token counts, latency, and retries, with tokenizer-dependent counts flagged. (iv) For RL training panels, training-log features of the kind listed in Appendix D.2, with run length checked as a possible source clue. Surface-form features of prompts and options (length, formatting, refusal markers) carried no signal under either set on MT-Bench or MMLU. For multiple-choice panels a per-item correctness label from a fixed reference model is the prerequisite we would address first, since the answer-position proxy used here limits what any feature set can recover.

D RL Panels and the Reference Detector

This appendix documents the RL panels of §5.4, where the audit began, and the source-blind reference detector reported there.

D.1 RFT-FaultBench panels

Corpus. Zhang et al. (2026) released 779 RL fine-tuning runs labeled across 16 fault types spanning 5 fault families. We restrict to the OpenRLHF training-step shards and select the `baselines/normal_*` runs as healthy runs (11 total) and the labeled faulty runs across all 16 fault types × {easy, hard} difficulty (96 total). After fitting a 4-parameter sigmoid to the first 25% of each run and filtering pathological fits (fitted upper asymptote $|U| > 5$), we retain 104 clean runs (10 healthy + 94 faulty).

Construction of the two-framework and three-framework panels. The two-framework panel (113 runs: 19 healthy + 94 faulty) adds to the filtered OpenRLHF runs the verl-side healthy runs from public DAPO and DeepScaleR histories that pass the same filter (nine of the ten). All OpenRLHF runs share one framework label. Adding runs trained with TRL gives the three-framework panel (134 runs: 35 healthy + 99 faulty). Both triads use an out-of-fold Random Forest under a shared stratified 5-fold split; the values are in Table 1 and §5.4. In both panels nearly all faulty runs come from one framework, so per-framework fault rates drive $\mathrm{AUC}_{\text{framework}}$ and the results show direction, not magnitude. The two-framework panel is the matched comparison for the three-framework panel; it differs from the single-framework panel above in its healthy runs and framework labels.

Source-label diagnostic and permutation null. On the unfiltered RL panel (117 runs, before the $|U| > 5$ filter) we assigned source labels from run-name prefixes: `openrlhf_baseline` (11 healthy runs), `dapo_verl` (5 healthy), `deepscaler_verl` (5 healthy), and `openrlhf_fault` (96 faulty, none healthy). Under this labeling the source label restates fault status by construction. With logistic regression and 5-fold out-of-fold scoring, $\mathrm{AUC}_{\text{full}} = 1.000$, $\mathrm{AUC}_{\text{features}} = 0.907$, and $\mathrm{AUC}_{\text{source}} = 1.000$ (in-sample), so the observed ratio is 1.000.

Shuffling the *source* labels across runs (cf. Pauly et al., 2016), with features and fault labels held fixed, gives a null on L (the stored output records 1,000 shuffles). Its mean is 0.635, its 99th percentile 0.718, and its maximum 0.756; no shuffle reaches the observed value (Figure 4). This shows the direction only. The magnitude is uninterpretable because the label is a relabeling of fault status, and the result is not evidence that any framework causes failures. Features can also carry source clues: the OpenRLHF runs have 20 to 40 logged steps, the verl runs 35 to 999. The null also does not transfer across panels, because it depends on the panel's feature strength: by the bound of §2, a panel with no source signal at all and $\mathrm{AUC}_{\text{full}} = 0.690$ has $L = 0.725$, above this null's 99th percentile of 0.718. We therefore do not read L against a universal null.

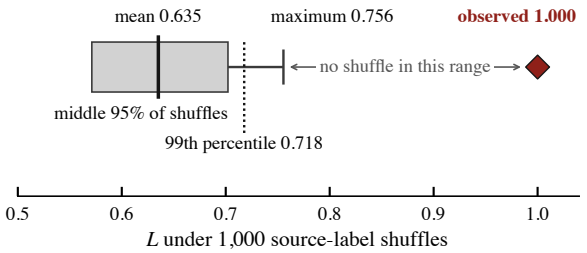


Figure 4: No source-label shuffle reaches the observed $L = 1.000$ (maximum 0.756). The null is for the unfiltered RL panel ($n = 117$; 1,000 shuffles of the source label, with features and fault labels held fixed). The box spans the middle 95% of shuffles; ticks mark the stored mean (0.635), 99th percentile (0.718), and maximum (0.756); the diamond is the observed $L = 1.000$. Per-shuffle values were not stored, so the figure shows the stored summary. Only the direction of the separation is interpretable, because the source label restates fault status in this panel.

D.2 The source-blind reference detector

The triad asks how close the source comes to the headline, but it does not fix what a source-blind detector would score. We use a closed-form one-class Mahalanobis χ^2 test as that reference. We illustrate it on the RL fault panel because run-level training-log features supply a low-dimensional, mostly continuous feature set ($d = 7$), for which a multivariate Gaussian on the healthy runs is a defensible reference model. The LLM-evaluation panels in §5 use categorical grades and per-query scores that do not admit the same closed form, so we report the triad on those panels without a reference. Let μ_c, Σ_c be the mean and (ridge-regularized) covariance of the feature vectors over the healthy runs. The squared Mahalanobis distance from the healthy-run cen-

troid is

$$D_M^2(x) \;=\; (x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c), \qquad (3)$$

and we predict "fault" when $D_M^2(x) > \chi^2_{7,0.95} = 14.07$. No parameter is fit to fault labels; the threshold is taken from the $\chi^2(7)$ tail, and the centroid and covariance are estimated from the healthy runs alone.

The detector is not the contribution. Scoring by Mahalanobis distance is standard (Lee et al., 2018; Linderman et al., 2026), and it suits panels with low-dimensional, continuous, approximately Gaussian features; on the RL panel two of the seven coordinates are not of that kind (see the feature list below). What we contribute is the test's *role* as a publishable source-blind reference, and that role does not require it to win. The 0.885 in Table 5 is computed with the healthy runs scored in-sample and drawn from one framework. When healthy runs are held out and drawn from a framework-mixed pool, a supervised Random Forest scores higher at every number of healthy runs n_{ctrl} we tested (about 0.88 against 0.55 to 0.60; see below). For high-dimensional or non-Gaussian panels, auditors can substitute another source-blind one-class detector (kernel density, isotonic projections, one-class SVM) in the same role.

Feature set (7 training-log features). The detector uses seven features of a run's training log:

- `U_blowup`, indicator that the fitted upper asymptote satisfies $|U| > 5$. It is identically zero on the filtered panel (the filter removes exactly these runs), so it contributes nothing to any distance.
- `len_norm_mae`, sigmoid-extrapolation residual on the remaining 75%, divided by the square root of the step count.
- `final_err_per_step`, the direction-aware extrapolation error.
- `n_steps`, total step count (run-length proxy). It takes a single value on the ten healthy runs, so the healthy-run covariance has no variance along it and only the ridge (below) sets the scale there: a faulty run with a different step count is flagged through this coordinate whatever its other features. With two constant coordinates the healthy-run covariance has rank five; we keep the $\chi^2(7)$ threshold, which is higher than the five-degree-of-freedom one.
- `entropy_at_25pct`, the mean policy entropy over the first 25% of steps.
- `kl_25pct_over_5pct_log`, slope of the KL trajectory (log scale).
- `reward_25pct_minus_5pct`, coarse reward-trend slope.

Ridge. With $n_{\text{ctrl}} = 10$ healthy runs and $d = 7$ features, two of which are constant on the healthy runs, the sample covariance Σ_c is singular. After scaling each feature that varies on the healthy runs by its healthy-run standard deviation (the two constant ones are left unscaled) we stabilize via $\Sigma_c \leftarrow \Sigma_c + \lambda I$ with $\lambda = 10^{-3}$. The ridge biases the inverse mildly along the coordinates that vary on the healthy runs and fixes it at $1/\lambda$ along the constant ones.

Mahalanobis result. The closed-form Mahalanobis $\chi^2(7)$ test from the healthy-run centroid yields:

- Healthy-run distance mean 2.06, max 2.84. No healthy run exceeds the $\chi^2(7)$ 0.95 envelope ($D_M \le 3.75$), and none could: the healthy runs are scored against the centroid and covariance they themselves define, and an in-sample distance over $n = 10$ runs cannot exceed $(n-1)/\sqrt{n} = 2.85$. We make no false-alarm claim on this panel (Appendix D.3 gives a separate check).
- Faulty-run distance median 52.5, max > 633. The largest distances belong to hard-difficulty runs whose step count differs from the healthy runs'; their size is set by the ridge on that zero-variance coordinate, not by the size of the fault.
- **AUC** $= 0.885$; 77 of 94 faulty runs (81.9%) flagged at the $\chi^2(7)$ 0.95 threshold; 39 of the 77 are the runs whose step count differs from the healthy runs' and that coordinate alone is enough to flag them.

Trained-baseline comparison (Table 5). On the single-framework panel (104 OpenRLHF runs), a Random Forest (200 trees, depth 4) on the seven features reaches AUC 0.81 (balanced bootstrap, 200 iterations of $N = 10$ healthy + $N = 10$ faulty runs).

This panel has a single framework, so it has no source axis and the Random Forest has no framework label to drop. The with-and-without-source contrast in Table 5 is computed on the two-framework panel below: 0.887 with the framework label, 0.881 without, 0.737 from the framework label alone.

Detector and what it sees	AUC
*Two-framework panel (*113 *runs:* 19 *healthy,* 94 *faulty)*	
$\mathrm{AUC}_{\mathrm{full}}$: training-log features + framework	0.887
$\mathrm{AUC}_{\mathrm{features}}$: framework label dropped	0.881
$\mathrm{AUC}_{\mathrm{framework}}$: framework label only	0.737
*Single-framework panel (*104 *runs:* 10 *healthy,* 94 *faulty)*	
One-class Mahalanobis χ^2, fit on healthy runs	0.885
Random Forest, training-log features	0.81

Table 5: RL fault panels built on RFT-FaultBench. Top: the triad (out-of-fold Random Forest, shared 5-fold split; source-only score in-sample). Bottom: the source-blind reference, which fits no parameter to fault labels, and a trained Random Forest. The bottom rows use different protocols (in-sample healthy-run scoring; balanced-bootstrap cross-validation), so their order is indicative only.

Small-n_{ctrl} sweep. On a framework-mixed pool of healthy runs, with healthy runs held out, we sweep $n_{\mathrm{ctrl}} \in \{10, 15, 20\}$ (bootstrap $B = 100$). A supervised Random Forest fit jointly on healthy and faulty runs reaches an AUC of about 0.88 regardless of n_{ctrl}, while the one-class Mahalanobis detector fluctuates in $[0.55, 0.60]$. The sweep stops at $n_{\mathrm{ctrl}} = 20$ because the pool holds only 21 healthy runs (OpenRLHF plus verl-side DAPO and DeepScaleR runs). The 0.885 in Table 5 is therefore conditional on in-sample scoring of healthy runs from one framework. Several things differ between the two settings (a framework-mixed pool of healthy runs, held-out scoring of healthy runs, the unfiltered panel the sweep runs on, and the sweep's own feature scaling and ridge), so the gap cannot be attributed to any one of them. The $\chi^2_{7,0.95} = 14.07$ threshold and the one-class estimand are the same in both.

D.3 False-alarm check on healthy runs

Setup. We measure the empirical per-step false-positive rate (FPR) of the closed-form $\chi^2(7)$ threshold on three healthy GRPO (group relative policy optimization) training runs. The runs are independent of the RFT-FaultBench panel and use seeds 42, 43, and 44. Each run trains Qwen2.5-1.5B-Instruct on GSM8K for 250 steps under a fixed GRPO recipe. At each step we record seven logged scalars, a per-step vector distinct from the run-level feature set of Appendix D.2: training loss, mean reward, reward standard deviation, policy entropy, mean completion length, gradient norm, and learning rate. The pooled cohort contains 750 step observations.

The Gaussian reference is refit on these runs: the centroid and covariance (with a small numerical ridge) are estimated from the pooled cohort for the in-sample FPR, and from the two other seeds for each leave-one-seed-out (LOSO) fold. The check therefore concerns the $\chi^2(7)$ tail threshold, not the RFT-FaultBench feature set.

Aggregates. The in-sample FPR is $23/750 = 3.07\%$. The leave-one-seed-out FPR is $27/750 = 3.60\%$ (per held-out seed: $42 \to 3.20\%$, $43 \to 5.20\%$, $44 \to 2.40\%$), the only held-out figure. Both are somewhat conservative relative to the nominal 5% of the $\chi^2(7)$ tail and within 2 percentage points of it.

Interpretation. On healthy GRPO trajectories the closed-form $\chi^2(7)$ tail threshold gives in-sample and leave-one-seed-out false-alarm rates of 3.07% and 3.60%. This supports using the χ^2 tail as the operating threshold of the source-blind reference (§5.4) on per-step GRPO logs. It says nothing about the floor, which concerns $\mathrm{AUC}_{\mathrm{full}}$, and it is not an early-warning claim.

E Protocol Extensions and Mitigations

E.1 Other metrics and label types

The protocol commits only to a quality threshold on the full detector and to the ratio L, so other ranking metrics fit the same structure. Substituting AUCPR (area under the precision-recall curve) or the H-measure (Hand, 2009) for each AUC gives $(\mathrm{M}_{\mathrm{full}}, \mathrm{M}_{\mathrm{source}}, L = \mathrm{M}_{\mathrm{source}}/\mathrm{M}_{\mathrm{full}})$ with a precondition $\mathrm{M}_{\mathrm{full}} \geq \tau_{\mathrm{M}}$. The floor must be recalibrated for each metric: the chance level of AUCPR is the positive prevalence, not 0.5, and the lower bound on L changes with it. For multi-class labels the natural form is a one-vs-rest triad per class, with the floor applied to the worst class; for regression, a rank-concordance statistic replaces AUC. We have run none of these variants.

E.2 Mitigations when leakage is confirmed

A LEAKAGE-CONFIRMED verdict says that a pooled headline is dominated by source identity. There are three standard responses: per-source stratified reporting (publishing per-source AUCs alongside the pooled headline); base-rate-equalized reweighting, which neutralizes the per-source label-rate differences that $\mathrm{AUC}_{\mathrm{source}}$ exposes; and per-source hierarchical or mixed-effects models, which absorb source-level random intercepts before the

quality detector is fit. The annotator-reliability tradition (Whitehill et al., 2009; Hovy et al., 2013) offers per-annotator competence models as a template for the last of these. The triad sits upstream of all of these: it tells the benchmark author when a mitigation is warranted.

What holds by construction, and what we do not report. We report no row-level mitigation analysis of RAGBench; what follows holds by construction. Within a single configuration the source score is constant, so $\mathrm{AUC}_{\text{source}} = 0.5$, $L = 0.5/\mathrm{AUC}_{\text{full}}$, and per-configuration reporting removes the source term without any modeling. Reweighting rows so that every configuration has the pooled adherence rate likewise makes the weighted source-only score constant, so the weighted $\mathrm{AUC}_{\text{source}}$ is exactly 0.5; the cost is variance, since the low-adherence configuration's rows receive weights far from 1.