BENCHMARK DEJA VU: DEBIASING SIGNALS FOR MULTIPLE-CHOICE CONTAMINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Benchmark contamination compromises the reliability of evaluations used by frontier labs to measure progress and by users to compare language models. Because ordinary model behavior can mimic contamination signals, reliable detection remains challenging. We introduce three contamination detectors with explicit null hypotheses that adjust for specified response preferences. We apply these new detectors to 35 contemporary open and closed models across four widely used multiple-choice benchmarks: MMLU, MMLU-Pro, GPQA Diamond, and the multiple-choice subset of Humanity's Last Exam (HLE). This audit reveals that most models produce at least one statistically significant signal on MMLU, whereas signals on other benchmarks are more selective, with family members sometimes sharing signals on the same benchmark. We aggregate detector-specific e-values to rank models by the strength of evidence against the global null across detectors and benchmarks. In addition, we construct two new testbed datasets, WikiCutoffQA and SyntheticSourcesQA, whose questions are screened so that model accuracy before exposure is consistent with random guessing. With these datasets, we can induce controlled exposure by training models on the original questions (or on versions with shuffled choices or paraphrased wording). Using these testbeds, we compare our detectors with 19 published contamination detection methods and demonstrate leading detection performance.

1 INTRODUCTION

Contemporary language models are routinely evaluated on multiple-choice question (MCQ) benchmarks (OpenAI, 2026, Anthropic, 2026c, DeepSeek-AI, 2026a, Qwen Team, 2026b, Moonshot AI, 2026b). Training on test questions, known as *contamination*, can inflate these scores, making it harder for researchers to measure progress and for users to compare models. Exposure can occur during causal language modeling (CLM) pretraining or later during post-training stages such as supervised fine-tuning (SFT) and reinforcement learning (RL). The central role of SFT and RL in modern post-training makes detecting exposure during these stages particularly important. Exposure to benchmark data can occur unintentionally. This problem is not hypothetical. OpenAI excluded BIG-bench from GPT-4's reported results after finding portions of it in the training data (OpenAI, 2023, Section 4). Test questions can also enter training in modified form: for example, the Soofi S model's developers withdrew affected benchmark results after paraphrased and translated questions from the GPQA benchmark were found in its publicly released training data (Soofi-Team, 2026). When training data are private, *behavioral tests* offer an alternative: inferring possible exposure from model outputs without inspecting training data or internal parameters (Sainz et al., 2023).

Behavioral tests can fail in two directions when their assumptions about exposure or clean-model behavior do not hold. The paraphrased GPQA case illustrates the first: exposure may not produce the behavior a detector is designed to find. A detector that looks for a preference for the original wording assumes that exposure preserved it, so it may miss contamination from paraphrased test questions. Likewise, a permutation detector comparing the likelihoods of original and shuffled choice orders assumes that exposure preserved the original order. It may miss contamination if choices were shuffled during training. Second, a clean model may show the tested behavior without prior exposure. The same permutation detector assumes that clean models do not systematically prefer the benchmark's original answer order. Yet a clean model may prefer that order simply because the choices form a natural chronological sequence, leading to a false flag (as demonstrated in Appendix G.4). Other

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

benchmark artifacts can also cause false positives (Section 5). These assumptions should therefore be made explicit and checked before interpreting detector outputs as evidence about exposure.

To make these assumptions explicit, we give each detector a null hypothesis that formalizes how its score is assumed to behave in clean models. Our tests account for specific response preferences that can arise without exposure, such as a tendency to favor particular answer labels. The null provides a reference for interpreting detector outputs: a small p-value or a large e-value provides evidence against it (Vovk & Wang, 2021). Rejecting the null does not establish training exposure, because unrecognized properties of the questions or the model may also violate its assumptions. The null's credibility for clean models on each benchmark must therefore be checked before its rejection is interpreted as evidence of contamination. We can also combine complementary detectors to broaden detection coverage, aggregating their e-values to retain a statistically interpretable result under the global null. Some existing detectors report p-values (Oren et al., 2024, Zhang et al., 2024), while others return heuristic familiarity scores without an explicit null distribution (Shi et al., 2024, Golchin & Surdeanu, 2025). Heuristic scores can be useful for ranking models, but they do not by themeselves control the false-positive rate.

In this paper, we introduce three detectors with explicit null hypotheses (Section 3). Our methods account for response preferences that can falsely flag clean models, and include a detector requiring only generated text, allowing us to audit closed models as well. We first evaluate these detectors in settings where exposure is known. We construct two MCQ *testbeds*: *WikiCutoffQA*, drawn from recent Wikipedia entries, and *SyntheticSourcesQA*, based on fictional sources (Section 4). We also screen questions for chance-level accuracy before exposure and provide held-out paraphrases for testing detection algorithms. For each testbed, we fine-tune five copies of the same LLM (Qwen3.5-4B). Two train on the original questions and choices, one with SFT and one with CLM. The other three use SFT with shuffled choices, paraphrased question wording, or paraphrased answer wording. Across these settings, we evaluate our detectors alongside 19 published methods. Our methods achieve state-of-the-art performance on both testbeds. Among the 19 published methods, none achieves strong detection performance when exposure occurs through SFT.

We next audit 35 open and closed models on MMLU, MMLU-Pro, GPQA Diamond, and the text-only MCQs from Humanity's Last Exam (HLE) (Hendrycks et al., 2021, Wang et al., 2024, Rein et al., 2024, Center for AI Safety et al., 2026). To reduce false signals from benchmark artifacts, we annotate questions and filter those that can violate each detector's assumptions (Section 5). We combine evidence across detectors and benchmarks using e-values to rank models by the strength of that evidence. With the caveat that a rejected null alone does not establish exposure, the filtered audits show widespread MMLU signals and more selective signals on the other benchmarks, with signals recurring within model families (see Figure 1, Section 5 and Appendix H).

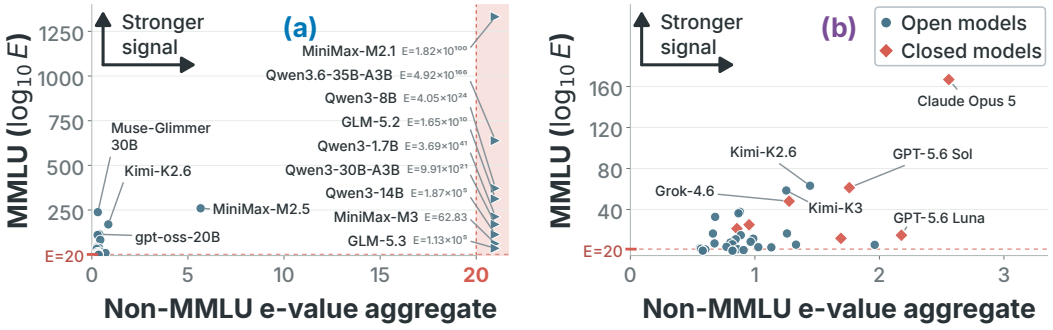

Figure 1: **Contamination signals after filtration.** **(a)** Mean e-values over three detection methods (DOPR, DWPC-QA, and CBA) for 28 open models. **(b)** E-values from predicting answer labels without seeing the choices (CBA) for 35 models. Non-MMLU averages over MMLU-Pro, GPQA Diamond, and HLE. Red lines mark $E = 20$, the single-test 5% threshold.

Our contributions are: (i) Three contamination detectors with explicit null hypotheses; (ii) Two controlled testbeds with held-out paraphrases, used to compare our detectors with 19 published methods under four forms of exposure; (iii) Question-level artifact annotations for four benchmarks; and (iv) An audit of 35 contemporary models after detector-specific filtering.

2 PROBLEM SETUP

Benchmark items and contamination. We study multiple-choice question (MCQ) benchmarks that test whether language models can correctly answer previously unseen questions. For each item i, let q_i denote the question, K_i its number of choices, $\mathbf{a}_i = (a_{i1}, \ldots, a_{iK_i})$ its answers in the benchmark's original order, and $a_i^\star$ its correct answer. The benchmark is $\mathcal{B} = \{(q_i, \mathbf{a}_i, a_i^\star)\}_{i=1}^n$. Display positions are conventionally labeled $\mathrm{A}, \mathrm{B}, \mathrm{C}, \ldots$. A shuffle is a permutation $\sigma = (\sigma_1, \ldots, \sigma_{K_i})$ of the original answer indices, producing $\mathbf{a}_i^\sigma = (a_{i,\sigma_1}, \ldots, a_{i,\sigma_{K_i}})$. Thus, $\sigma = (2, 3, 1)$ places (a_{i2}, a_{i3}, a_{i1}) under labels $\mathrm{A}, \mathrm{B}, \mathrm{C}$, respectively. The answer texts and correct-answer identity do not change.

We consider a model contaminated with respect to a benchmark item if it processes the item or text derived from it during any parameter-changing stage, including pretraining, supervised fine-tuning, preference-based post-training, or distillation. We call a model *clean* with respect to an item if no such exposure occurred. Leakage caused by test-time retrieval is outside this paper's scope (Han et al., 2025). Subject knowledge is not benchmark exposure: a model may have all the knowledge needed to answer a question without having encountered that question or its choices. A training-data match is evidence of exposure only when the benchmark item is sufficiently distinctive. We assume that, without access to the test, independently producing an equivalent question with its complete answer set is highly unlikely, even after reordering or paraphrase. Such a match therefore provides evidence of benchmark exposure. This assumption can fail for common, recurring questions. For example, authors may independently write "What is the capital of France?" with choices such as Paris, London, Berlin, and Rome, so the match alone does not imply contamination.

From exposure to detection. We distinguish the *retained trace* left by exposure, the *audit* that probes it, and the *analysis* that maps an *observable signal* to a score or decision:

$$\boxed{\text{exposure}} \longrightarrow \boxed{\text{retained trace}} \xrightarrow{\text{audit}} \boxed{\text{observable signal}} \xrightarrow{\text{analysis}} \boxed{\text{score or decision}}.$$

A retained trace is an exposure-induced effect encoded in the model's parameters and still present at audit time. An audit can probe internal state directly (Liu et al., 2024). More commonly, audits collect observable model outputs. For example, an audit can present answer permutations or paraphrases of the same item and record the model's predictions or confidence for each version. Variation across these outputs is an observable signal. Correct-answer knowledge can be such a trace, revealed through correctness or accuracy. Yet accuracy is weak evidence of contamination because improving performance is also the goal of clean training. A *detection method*, or *detector*, combines an audit with an analysis that maps the resulting signals to a score or decision. The same audit can support different analyses (e.g., Algorithms 1 and 2 of Ni et al. (2025)). We focus on two among many possible signals: *order preference* for the exposed answer order over its permutations (Ni et al., 2025, Hidayat et al., 2025), and *phrasing preference* for the exposed wording over meaning-preserving alternatives (Golchin & Surdeanu, 2025, Zhang et al., 2024).

We study four *exposure forms* for benchmark item i, where $\tilde{x}$ denotes a meaning-preserving paraphrase of x: (i) *Complete-Item Exposure* uses the original $(q_i, \mathbf{a}_i, a_i^\star)$, (ii) *Shuffled-Order Exposure* varies the answer permutation in $(q_i, \mathbf{a}_i^\sigma, a_i^\star)$, (iii) *Question-Paraphrase Exposure* replaces only the question, giving $(\tilde{q}_i, \mathbf{a}_i, a_i^\star)$, and (iv) *Answer-Paraphrase Exposure* replaces only the answers, giving $(q_i, \tilde{\mathbf{a}}_i, \tilde{a}_i^\star)$. Exposure to a shuffled item may not create a preference for the benchmark's original answer order, while exposure to a paraphrased item may not create a preference for its original wording. We focus on these four because they are tractable and directly vary the answer-order and wording signals we study. Contamination depends on exposure, not intent or later retention. Accidental scraping therefore qualifies. Paraphrased exposure forms can arise in practice because synthetic training datasets may contain paraphrased or semantically equivalent benchmark items, as documented by Yang et al. (2023) and Matton et al. (2024).

We exclude methods whose contamination signals can be affected by a model's ability to answer benchmark questions even without exposure. The dependence can be implicit: a method defines a signal intended to indicate contamination, but that signal also varies with a clean model's accuracy. In particular, a clean model should not appear more contaminated simply because it answers more questions correctly. See Appendix F for an example. Separately, some detection methods require an additional reference model or access to model snapshots from before and after the suspected exposure (Mireshghallah et al., 2022, Zanella-Béguelin et al., 2020). We leave these methods outside our scope, as suitable reference models or historical snapshots are not always available.

Null hypotheses and heuristics. A score becomes evidence only through an assumption linking the signal to exposure. We distinguish two forms of this assumption. A null hypothesis specifies how the test statistic behaves under clean conditions, which lets us compute a p-value. A heuristic instead relies on a behavioral premise without defining a clean-model rejection distribution (Shi et al., 2024, Golchin & Surdeanu, 2025). For example, an order-preference test might assume that a clean model is equally likely to favor any valid answer permutation. A phrasing-preference test might similarly assume that the model favors the correct answer equally across semantically equivalent phrasings of the question and answers. These example nulls can fail in practice when a trace unrelated to benchmark exposure produces the same signal. A clean model might prefer an answer order for its natural chronology, not because it recognizes that order from benchmark exposure. Other confounds include numerical ordering, letter or position preferences, answer labels revealed by the question, and ordinary sensitivity to wording or prompt format (see Section 5). Rejection therefore supports contamination only if the clean null is empirically credible and competing explanations have been addressed. A p-value quantifies evidence against the null, not the probability that the model was clean. Failure to reject also does not establish cleanliness. Contamination may leave no accessible trace, or the trace may surface through a signal that the method does not examine.

Evaluating detection methods. We evaluate detectors on a set of benchmark items $\mathcal{B}_G = \{(q_i, \mathbf{a}_i, a_i^\star)\}_{i \in G}$, where $G \subseteq \{1, \ldots, n\}$. A detection method assigns each group a score $s(\mathcal{B}_G)$, with higher scores indicating stronger evidence of contamination. Taking $G = \{1, \ldots, n\}$ gives the full benchmark $\mathcal{B}$. For tests of group means, we regard $\mathcal{B}_G$ as a sample from an underlying data distribution $\mathcal{P}$. Expectations under $\mathcal{P}$ are over sampled items $(q_i, \mathbf{a}_i, a_i^\star) \sim \mathcal{P}$, holding the model and audit procedure fixed. Let $\mathcal{B}_{G_{\text{cont}}}$ and $\mathcal{B}_{G_{\text{clean}}}$ be independently sampled contaminated and clean groups, respectively. We compute AUROC as

$$\text{AUROC}(s) = \Pr[s(\mathcal{B}_{G_{\text{cont}}}) > s(\mathcal{B}_{G_{\text{clean}}})] + \tfrac{1}{2} \Pr[s(\mathcal{B}_{G_{\text{cont}}}) = s(\mathcal{B}_{G_{\text{clean}}})] \,.$$

3 Debiasing Contamination Signals in MCQs

Exposure may induce a preference for an MCQ's exact answer order or wording. The same preference can arise in clean models from displayed-label bias, answer-identity preference, or preferences for particular answer–label mappings. We therefore introduce three detectors. The *Debiased Order-Preference Residual* (DOPR) tests whether the original permutation selectively strengthens the correct answer relative to its distractors, after accounting for displayed-label and answer-identity preferences. The *Debiased Wording-Preference Contrast* (DWPC) compares original and paraphrased wording after removing shifts shared by all labels and balancing displayed-label effects. Figure 2 illustrates DOPR (left) and DWPC (right). *Choice-Blind Answering* (CBA) instead hides the choices and tests whether the model can reproduce the correct benchmark label without seeing which answer that label denotes. In our controlled experiments, we call a model *clean* if it has not undergone controlled exposure to our testbeds (Section 4). In Appendix B, we show that our methods do not flag clean models simply for achieving high accuracy on MCQs whose answers they already know.

DOPR: isolating answer-order preference. DOPR tests whether the original permutation selectively strengthens the correct answer relative to its distractors. Knowledge of the answer can explain a preference across arrangements. Prior exposure, however, may give the correct answer extra support under the original arrangement, increasing its advantage over the same distractors. Appendix A.1 provides the full derivation and ablation analyses. For item i with K_i choices, we query the K_i cyclic shifts of the original permutation $\sigma^{(0)} = (1, \ldots, K_i)$, repeatedly moving the first index to the end. Across these K_i mappings, every answer appears once under every displayed label. We record all K_i answer-label logits under each mapping, forming the $K_i \times K_i$ matrix in Figure 2 (left). Let $z_i^\star$ be the original-mapping correct-answer logit and G_i the matrix's grand mean. Let $\mu_{i,\sigma^{(0)}}$ be the mean label logit under the original mapping (the top row in Figure 2, left). Let $\mu_{i,\ell_i^\star}$ be the correct displayed label's mean across mappings (column A in the figure). Let $\mu_{i,a_i^\star}$ be the mean for the correct answer identity across mappings (the diagonal E cells, all representing Evaporation).

The first two terms below form the *within-permutation correct-answer advantage*, $z_i^\star - \mu_{i,\sigma^{(0)}}$: how far the correct-answer logit stands above the mean of all label logits under the original mapping. This subtracts a rise shared by the correct answer and its distractors, while retaining any increase in their separation. The remaining terms account for average preferences for the displayed letter, regardless of

content, and for the correct answer content across mappings (an expected and desirable preference):

$$R_i = (z_i^\star - G_i) - (\mu_{i,\sigma^{(0)}} - G_i) - (\mu_{i,\ell_i^\star} - G_i) - (\mu_{i,a_i^\star} - G_i).$$

A positive R_i means the correct answer stands out more under the original mapping than these average preferences predict, not merely that it is preferred across mappings. For a prespecified group, we test the null hypothesis $H_0 : \mathbb{E}_{\mathcal{P}}[R_i] \leq 0$ against $H_1 : \mathbb{E}_{\mathcal{P}}[R_i] > 0$ with a one-sided one-sample t-test. In our clean examples, DOPR residuals are consistent with $\mathbb{E}_{\mathcal{P}}[R_i] \leq 0$ (see Figure 5 in Appendix A.1). Appendix G.4 demonstrates DOPR's group-level robustness to natural-order artifacts in constructed examples where other tested permutation detectors falsely flag individual questions.

DWPC: isolating wording preference. DWPC tests whether the original wording gives the correct answer greater relative support than a held-out, meaning-preserving paraphrase. For each of the same K_i cyclic mappings used by DOPR, DWPC computes the correct-answer advantage by subtracting the mean of the K_i answer logits from the correct-answer logit. We call this operation *within-query centering*. A wording can raise all answer-label logits equally without increasing the correct label's probability relative to the other labels, so centering removes this shared shift. DWPC then averages these advantages separately for each wording over the K_i mappings. Across the mappings, the correct answer appears once under every displayed label, so averaging removes stable displayed-label preferences before the original and held-out averages are compared (Figure 2, right).

DWPC-Q uses a held-out paraphrase of only the question, whereas DWPC-QA uses paraphrases of both the question and the answers. For a prespecified variant $v \in \{\mathrm{Q}, \mathrm{QA}\}$, let $z_{iwr\ell}^{(v)}$ be the logit for label ℓ under wording $w \in \{O, P\}$ and cyclic mapping r. Let $\bar{z}_{iwr}^{(v)}$ be the mean across labels and $\ell_i^\star(r)$ the correct label. We define the centered correct-answer advantage $A_{iwr}^{(v)} = z_{iwr,\ell_i^\star(r)}^{(v)} - \bar{z}_{iwr}^{(v)}$, its mapping average $\bar{A}_{iw}^{(v)} = \frac{1}{K_i}\sum_{r=0}^{K_i-1} A_{iwr}^{(v)}$, and the item-level score $W_i^{(v)} = \bar{A}_{iO}^{(v)} - \bar{A}_{iP}^{(v)}$. Under the meaning-preservation assumption, the original and held-out wordings contain the same underlying answers. Subtracting their centered scores therefore cancels any answer-identity preference that is stable across the two wordings, while wording-dependent changes remain in $W_i^{(v)}$.

A positive $W_i^{(v)}$ means the correct answer receives more relative support under the original wording than under the held-out wording. Contamination can produce this pattern when the item was exposed in its original wording. For a prespecified variant and group, we test the null hypothesis $H_0 : \mathbb{E}_{\mathcal{P}}[W_i^{(v)}] \leq 0$ against $H_1 : \mathbb{E}_{\mathcal{P}}[W_i^{(v)}] > 0$ with a one-sided one-sample t-test. In our clean examples, DWPC scores are consistent with $\mathbb{E}_{\mathcal{P}}[W_i^{(v)}] \leq 0$ (see Figure 9 in Appendix A.2). Appendix A.2 gives the full test, score distributions, removed-effect analysis, and ablation analyses.

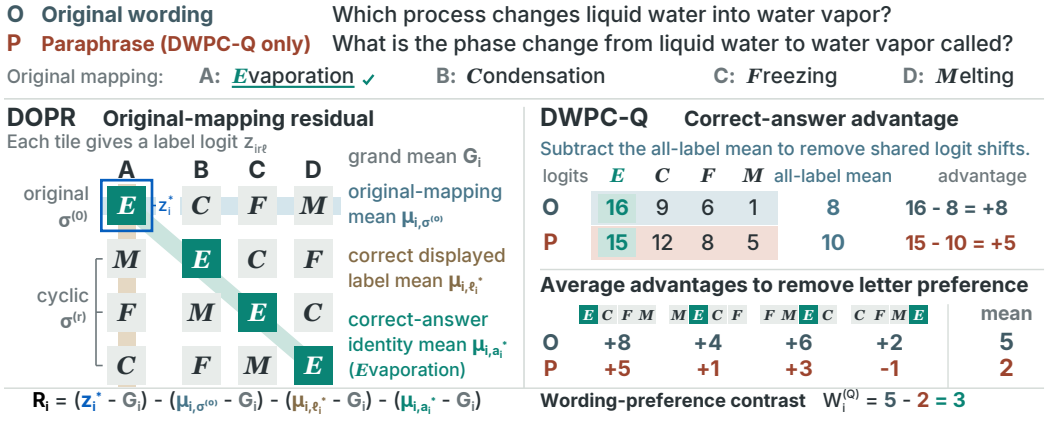

Figure 2: **Debiased order and wording preference. Left:** DOPR tests whether the correct answer's within-permutation advantage is greater under the original benchmark mapping than its general displayed-label and answer-identity preferences predict. **Right:** DWPC compares the correct answer's advantage under original and paraphrased wording after subtracting each query's mean label logit and averaging across cyclic mappings to balance displayed-label preferences.

CBA: reconstructing a hidden answer label. During the audit, DOPR and DWPC retain the complete answer menu. CBA hides it: given only question q_i and the valid label set, the model predicts a label $\hat{\ell}_i$. We compare it with the hidden correct label $\ell_i^\star$ (Figure 3). A match may reflect item-specific mapping information, a benchmark-wide label bias that matches the model's label bias, or chance. When applying CBA to real benchmarks, we filter items whose hidden answer-to-label mapping or correct label can be inferred from the question alone (Section 5 and Appendix G). CBA requires only generated labels, not logits, so it also applies to closed-source models with text-only access. Appendix A.3 provides the exact prompt, output rule, and testing procedure.

We analyze the predictions collected by the CBA audit under two null hypotheses: one tests above-chance accuracy, and the other tests item-specific prediction–answer-key alignment.

Uniform-guessing null hypothesis. Assume first that all n questions have K choices and common label set $\mathcal{L}$, and let S be the match count, with observed value $S_{\text{obs}} = \sum_i \mathbf{1}\{\hat{\ell}_i = \ell_i^\star\}$. The null hypothesis is $H_0^{\text{uniform}} : S \sim \text{Binomial}(n, 1/K)$. Its exact upper tail tests whether CBA accuracy exceeds random guessing. Appendix A.3.2 extends both nulls from a common K to question-specific numbers of choices K_i.

Answer-key shuffle null hypothesis. The uniform-guessing null ignores label-frequency bias in both the model and the dataset. It can falsely flag a clean model when their letter biases align, or miss non-random item-specific mapping alignment when opposing biases keep raw accuracy low (see Appendix A.3.1 and Table 5). Our primary test, answer-key shuffle, conditions on the observed prediction vector and correct-label counts. It tests whether predicted and correct labels match on the same questions more often than their label frequencies alone explain. Holding the predictions fixed, let Ω be the set of candidate answer keys $\boldsymbol{\ell}' = (\ell'_1, \ldots, \ell'_n)$ obtained by reassigning the observed correct labels across questions. Here, ℓ'_i is the label placed at question i. Under H_0^{shuffle}, S the same conditional distribution as the match count obtained from $\boldsymbol{\ell}' \sim \text{Uniform}(\Omega)$. The test assumes that correct labels are exchangeable across questions, conditional on the observed predictions and correct-label counts. The exact one-sided p-value is $p_{\text{shuffle}} = \Pr_{\boldsymbol{\ell}' \sim \text{Uniform}(\Omega)}(\sum_{i=1}^n \mathbf{1}\{\hat{\ell}_i = \ell'_i\} \geq S_{\text{obs}})$.

Combining evidence. Each method uses a different source of evidence for possible contamination. We therefore test the *global null* that all three null hypotheses defined above hold; rejecting it provides evidence that at least one of them does not hold. To this end, we use e-values (Vovk & Wang, 2021) to quantify the statistical evidence against the global null. E-values provide an alternative to p-values and can be easily combined across tests. An e-variable E is a nonnegative random variable satisfying $\mathbb{E}[E] \leq 1$ under the null hypothesis, with larger values indicating stronger evidence against the null. Importantly, the arithmetic mean of valid e-values is a valid e-value, even under arbitrary dependence. We thus compute an e-value for each detector and take their equal-weight arithmetic mean. The resulting aggregate e-value is our overall measure of evidence against the global null. Further details on the detector-specific e-value constructions and their aggregation are provided in Appendix A.4.

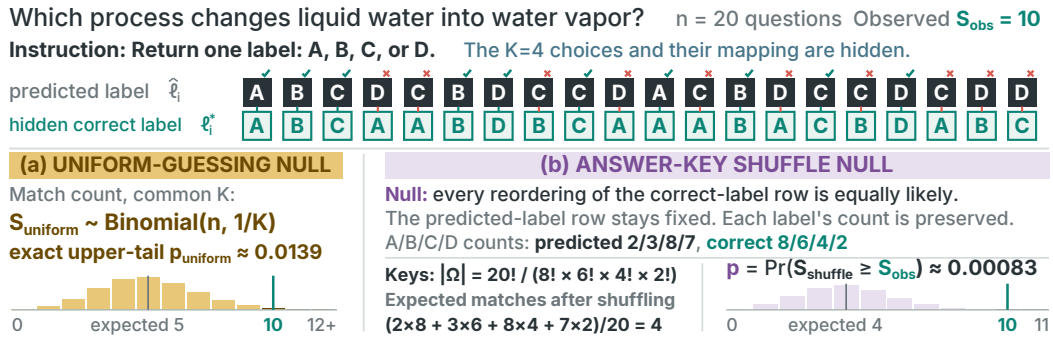

Figure 3: **Choice-blind answering.** CBA predicts the benchmark label without seeing the choices. Its two nulls distinguish raw above-chance accuracy from item-specific mapping alignment. The displayed numbers are illustrative.

4 CONTROLLED CONTAMINATION TESTBEDS

Evaluating contamination detectors requires ground truth about whether a model is contaminated and which test questions appeared in its training data. Controlled testbeds provide this information, which is rarely available for released models. One way to construct such testbeds is to fine-tune existing models on public benchmarks, such as MMLU (Deng et al., 2024, Hidayat et al., 2025). However, prior contamination or knowledge related to the questions and answers may shape the starting model's behavior, so it cannot be assumed to provide a clean, chance-level baseline (Ravichander et al., 2025). Wikipedia-based testbeds use entries about events after a model's training cutoff as unexposed examples (Shi et al., 2024, Fu et al., 2025, Yi & Li, 2026). These entries may enter the training of later models. Additionally, even unexposed models may answer such questions using existing knowledge or predict event outcomes above chance, such as a stronger football team winning. Temporal novelty alone therefore does not establish a chance-level baseline. Whether the model can answer such questions above chance should be checked empirically. We also aim to study how detection changes when models are trained on paraphrased questions or answers. We therefore construct two MCQ testbeds with controlled paraphrases and screen their questions to verify chance-level accuracy in the tested models. We induce controlled exposure by fine-tuning them on selected questions and their correct answers, then test whether detectors distinguish groups of these exposed questions from groups of questions withheld from fine-tuning. We vary answer order and wording during training to support comparisons of current and future detectors across contamination forms.

Testbeds. *WikiCutoffQA (WCQA)* contains 512 questions from 64 English Wikipedia entries updated between May 3 and August 10, 2026, after the March 2 revision of Qwen3.5-4B, our test model. The questions target recent knowledge, such as the outcomes of later events. For *SyntheticSourcesQA (SSQA)*, we generate 256 fictional sources and create 2,048 questions from them. We deliberately use highly technical fictional facts with minimal connection to real-world knowledge to limit clues that could favor one answer over another. The goal is for these facts to remain unfamiliar even to future models unless those models are exposed to SSQA. Each question has four choices. We split each dataset into two equally sized sets: an *exposure set* used for contamination fine-tuning and a *control set* excluded from all training in our experiments. Each question has five alternative, semantically equivalent question–answer wordings. We reserve one for training and four for detector queries, so a detector can compare the original with paraphrases that were never used for fine-tuning. We screen both datasets by testing whether models can answer the questions with their source passages and perform at chance level without them (Appendix C).

4.1 EVALUATING DETECTION METHODS

We introduce exposure by fine-tuning separate copies of the same post-trained Qwen3.5-4B model. Our main comparison uses SFT with the four exposure forms defined in Section 2. We also evaluate CLM under Complete-Item Exposure. SFT applies the loss to the answer, whereas CLM also applies it to the question and choices. We focus on SFT because all 19 published methods score below 0.7 AUROC in each of these settings, although some detect CLM exposure perfectly (see Appendix E.2). Appendix D gives the training details. We compare DOPR, DWPC, and CBA (Section 3 and Appendix A) with published detectors in Appendix E. For AUROC, we rank 16-item groups by $s(\mathcal{B}_G)$, using negative p-values for CBA.

Detection results. We compare our methods with 19 published detectors across eight SFT and two CLM exposures. DOPR and DWPC-QA achieve substantially higher mean AUROCs than the detectors across the eight SFT settings. In our comparison, the highest published-baseline AUROC under SFT is 0.688, achieved by LogProber (Yax et al., 2025) on WCQA Complete-Item Exposure, whereas LOSS (Yeom et al., 2018) and Min-K% (Shi et al., 2024) each reach 1.0 on both CLM exposures. We therefore focus on our methods in Table 1, with the complete comparison in Tables 11 and 12. DWPC-QA leads our methods under WCQA Complete-Item Exposure. Under Shuffled-Order Exposure, answer order varies during training, so training does not consistently reinforce the original order. DOPR is near chance, as expected for a detector of order preference. The unchanged question and answer texts can still leave a wording trace: DWPC-QA probes the preference for the original wording over held-out paraphrases and perfectly separates groups in both datasets. DWPC-QA weakens under both paraphrase forms, as expected for a wording-preference detector. Outside Shuffled-Order Exposure, DOPR achieves 0.95–1.0 AUROC on both datasets. CBA achieves higher

AUROC on WCQA than on SSQA under Complete-Item Exposure. Some published detectors show a similar pattern. With choices hidden, WCQA's event descriptions may cue the original question–label association more easily than SSQA's relations among invented entities and technical identifiers.

Table 1: **Controlled contamination detection.** Group AUROC (higher is better, chance: 0.5) with whole-set decisions below. ✓: exposure rejected and control not rejected. ◐: neither rejected. For the aggregate, ✓ requires these outcomes under both Holm and e-BH. Other methods: Tables 11 and 12.

Method	WCQA					SSQA				
	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM
DWPC-QA	**1.000**	**1.000**	0.891	0.859	0.914	0.984	**1.000**	0.916	0.921	0.876
Reject H_0: exposure only	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DOPR	0.977	0.477	**0.961**	**0.965**	**1.000**	**1.000**	0.449	**1.000**	**1.000**	0.953
Reject H_0: exposure only	✓	◐	✓	✓	✓	✓	◐	✓	✓	✓
CBA-shuffle	0.691	0.451	0.502	0.547	0.551	0.572	0.596	0.587	0.573	0.539
Reject H_0: exposure only	✓	◐	◐	◐	✓	✓	◐	◐	✓	◐
e-value aggregate	0.996	0.996	0.957	**0.965**	0.996	0.997	**1.000**	**1.000**	**1.000**	**0.960**
Reject H_0: exposure only	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

5 Auditing Released Models on Real Benchmarks

We apply DOPR, DWPC-QA, and CBA to 35 released models on MMLU, MMLU-Pro, GPQA Diamond, and the MCQs from Humanity's Last Exam (HLE). DOPR and DWPC-QA cover 28 open models, and CBA additionally covers seven closed models.

Filtering structural artifacts. Even basic assumptions of contamination detectors can fail on real benchmark questions. For example, permutation-based detectors assume that reordering the choices preserves their meaning, yet shuffling the choices referenced by "Both A and C" changes what that answer means (Figure 4(a)). Similarly, CBA assumes that hiding the choices withholds their mapping to labels, but the question in Figure 4(b) lists them in their original order. Assigning A–D to this list recovers the benchmark's answer-to-label mapping, allowing a model to return the correct label by solving the question. Permutation-based detectors can also favor the original order because it matches the question's list, while a shuffled order may not. The chronological choices in Figure 4(c) can also favor the original order without prior exposure. Unfiltered audits can therefore mistake question structure for contamination. We develop a multistage filtration protocol and annotate 26,783 questions using multiple judgments. These question-level annotations can support future detectors on these four benchmarks by identifying questions that violate their assumptions. Appendix G details the protocol, filters, and detector-specific exclusions.

Results. We combine detector evidence as described in Appendix A.4. We use e-BH to test the global null for each model while controlling the expected fraction of false rejections at 5% within

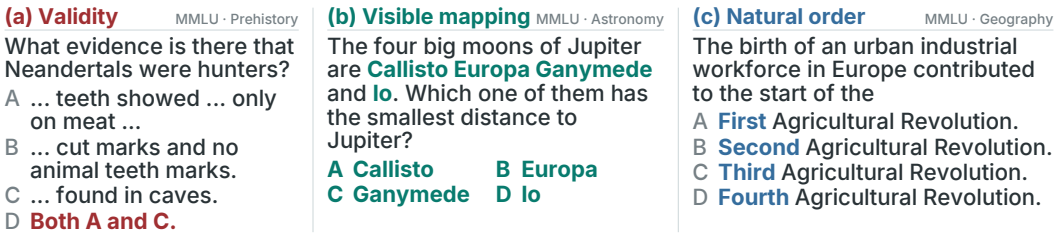

Figure 4: **Structural artifacts can mimic contamination signals.** **(a) Validity.** Shuffling changes which answers "A and C" refers to, violating permutation validity. **(b) Visible mapping.** Assigning A–D to the listed moons recovers the benchmark's answer-to-label mapping despite CBA hiding the choices. Matching the question's list can also favor the original order in permutation-based detectors. **(c) Natural order.** Permutation-based detectors can favor chronological order without prior exposure.

each family, even under arbitrary dependence (Wang & Ramdas, 2022). We also apply Holm at 5% to model p-values $P = \min\{1, m \min_j p_j\}$ formed from the m original p-values, not the displayed e-values. Holm controls the probability of any false model-level rejection within a family (Holm, 1979). Appendix H reports the full results: component p-values and aggregate e-values for all models across all benchmarks. We note that rejecting these nulls does not establish contamination, intent, or the extent of any exposure. We make three notable observations:

1) Strong statistical evidence on MMLU is widespread. For most models, each detector yields unadjusted MMLU p-values below 0.05, and all 35 models have at least one such signal (Tables 17 and 19 in Appendix H). Combining the three detectors on MMLU alone, e-BH rejects the global null for 26 of 28 open models, and Holm rejects all 28. For CBA, e-BH rejects 28 of 35 models and Holm rejects 30. Because MMLU signals are so widespread, we report MMLU separately from the other three benchmarks in Figure 1 and the result tables.

2) Non-MMLU signals cluster within model families. Aggregating MMLU-Pro, GPQA Diamond, and HLE, e-BH rejects the global null for nine of the 28 open models: four Qwen3 models (1.7B, 8B, 14B, and 30B-A3B), Qwen3.6-35B-A3B, GLM-5.2, GLM-5.3, MiniMax-M2.1, and MiniMax-M3. The first eight models' strongest signals appear on MMLU-Pro, through DOPR for GLM-5.3 and DWPC-QA for the other seven models (Table 2).

3) HLE and GPQA-D also show localized signals. MiniMax-M3 has HLE signals from DOPR ($p = 1.97 \times 10^{-4}$) and DWPC-QA ($p = 1.99 \times 10^{-5}$). MiniMax-M2.5 also has an HLE signal from DWPC-QA ($p = 2.35 \times 10^{-4}$). On GPQA-D, Kimi-K2.6 has a DWPC-QA signal ($p = 6.70 \times 10^{-3}$). These component p-values are unadjusted. Of these three models, only MiniMax-M3 meets the non-MMLU aggregate e-BH criterion, and Holm rejects the global null for both MiniMax-M3 and MiniMax-M2.5 (Table 18).

Table 2: Localized non-MMLU signals by model family. Each row shows the benchmark with the largest aggregate E, the mean of its three detector e-values. Redder cells indicate stronger signals.

Model	Benchmark	DOPR p	DWPC-QA p	CBA p	Aggregate E
Qwen3.6-35B-A3B	MMLU-Pro	0.749	6.62e−172	0.528	1.48e167
Qwen3-1.7B	MMLU-Pro	0.948	1.84e−46	0.396	1.11e42
Qwen3-8B	MMLU-Pro	0.156	2.15e−29	0.798	1.21e25
Qwen3-30B-A3B	MMLU-Pro	0.224	9.25e−27	0.329	2.97e22
Qwen3-14B	MMLU-Pro	0.371	8.58e−10	0.0510	5.62e5
GLM-5.2	MMLU-Pro	0.727	7.68e−15	0.404	4.96e10
GLM-5.3	MMLU-Pro	2.17e−9	1.00	0.883	3.40e5
MiniMax-M2.1	MMLU-Pro	0.0266	2.37e−105	0.145	5.47e100
MiniMax-M3	HLE	1.97e−4	1.99e−5	0.0681	188
MiniMax-M2.5	HLE	0.957	2.35e−4	0.381	16.5

6 Conclusion

In this paper, we introduced three contamination detectors: DOPR, DWPC, and CBA. Experiments on our two controlled testbeds demonstrated the complementary strengths of DOPR and DWPC-QA, which each outperformed all evaluated published baselines in mean AUROC across the eight SFT settings. We also identified structural artifacts that can mimic contamination signals and annotated 26,783 benchmark questions, creating a resource for future research and audits of exposure. After filtration, our audit of 35 open and closed models revealed widespread MMLU signals. On other benchmarks, signals were more selective, recurring within some model families on the same dataset.

Further research could use our testbeds to develop new detectors and study how exposure traces persist or change across training stages. They could evaluate detectors when exposure occurs during other LLM training stages, such as RL, and test whether benchmark knowledge can be transferred from contaminated teachers through knowledge distillation or multi-teacher on-policy distillation (MOPD) (Ma et al., 2026). Such experiments, together with studies of model compression, could establish how further training and knowledge transfer affect both the retention of benchmark knowledge and the power of contamination detectors.

AI USE STATEMENT

We used GPT 5.6 Sol to generate SSQA sources, all SSQA and WCQA MCQs and their paraphrases, and to perform benchmark filtration and semantic validation. Additional model-based screening and judging procedures are described in the appendix. Humans drafted the content of every paragraph in a detailed manuscript skeleton. AI assisted in developing and polishing text from that skeleton. Related literature was identified by humans and AI, and humans verified all AI-identified references. The authors are responsible for all content and artifacts.

ETHICS STATEMENT

Our work aims to improve the reliability of benchmark evaluation. We use controlled benchmark exposure to study detection and report the assumptions underlying all detectors and the confounds that limit interpretation. For released models, we test the specific null hypotheses stated in the paper. Rejecting these hypotheses does not by itself establish that a model is contaminated, demonstrate deliberate misconduct, or identify the source of any training exposure.

REPRODUCIBILITY STATEMENT

The pretrained open-model checkpoints used in our experiments are available on Hugging Face. The four real benchmarks, MMLU, MMLU-Pro, GPQA Diamond, and HLE, are also publicly available. Appendices A, C, D, and G describe detector computation, testbed construction, training, and filtration, respectively. Upon acceptance, we will release our code and all data constructed for this work, including WikiCutoffQA, SyntheticSourcesQA, and the 100-question control dataset in Appendix B. We will also release the filtration audits of all four real benchmarks, including metadata and annotator notes, to support reuse and reproduction of our filtering decisions.

REFERENCES

Anthropic. Introducing Claude Opus 4.8. Official release announcement, May 2026a. URL https://www.anthropic.com/news/claude-opus-4-8. Published May 28, 2026. Accessed September 17, 2026.

Anthropic. Introducing Claude Opus 5. Official release announcement, July 2026b. URL https://www.anthropic.com/news/claude-opus-5. Published July 24, 2026. Accessed September 17, 2026.

Anthropic. System card: Claude Sonnet 4.6. Technical report, Anthropic, February 2026c. URL https://www-cdn.anthropic.com/bbd8ef16d70b7a1665f14f306ee88b53f686aa75/Claude%20Sonnet%204.6%20System%20Card.pdf. Published February 17, 2026. Updated March 6, 2026.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Prasoon Varshney, Makesh Narsimhan, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi Mahabadi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George

Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Shaona Ghosh, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, Chris Alexiuk, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, and Eric Chung. Llama-Nemotron: Efficient reasoning models. arXiv preprint arXiv:2505.00949, 2025a. URL https://arxiv.org/abs/2505.00949.

Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammed Dabbah, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Netanel Haber, Ehud Dov Karpas, Roi Koren, Itay Levy, Pavlo Molchanov, Shahar Mor, Zach Moshe, Najeeb Nabwani, Omri Puny, Ran Rubin, Itamar Schen, Ido Shahaf, Oren Tropp, Omer Ullman Argov, Ran Zilberstein, and Ran El-Yaniv. Puzzle: Distillation-based NAS for inference-optimized LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3806–3830. PMLR, 2025b. URL https://proceedings.mlr.press/v267/bercovich25a.html.

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. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://www.nature.com/articles/s41586-025-09962-4.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. Hugging Face model card, 2026a. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/tree/b5968e9190ef611bbf34a7229255be88a0e937c1. Revision b5968e9190ef611bbf34a7229255be88a0e937c1.

DeepSeek-AI. DeepSeek-V4-Flash model card. Hugging Face model card, 2026b. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash/tree/main. Accessed September 17, 2026.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 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)*, pp. 8706–8719. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.482. URL https://aclanthology.org/2024.naacl-long.482/.

André Vicente Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. DE-COP: Detecting copyrighted content in language models training data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11940–11956. PMLR, 2024. URL https://proceedings.mlr.press/v235/duarte24a.html.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. MIA-Tuner: Adapting large language models as pre-training text detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27295–27303, 2025. doi: 10.1609/aaai.v39i26.34939. URL https://ojs.aaai.org/index.php/AAAI/article/view/34939.

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. Are we done with MMLU? In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 5069–5096, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.262. URL https://aclanthology.org/2025.naacl-long.262/.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/bc39a59c49b731c51398ad6b12f301d3-Abstract-Conference.html.

Shahriar Golchin and Mihai Surdeanu. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Transactions of the Association for Computational Linguistics*, 13:809–830, 2025. doi: 10.1162/tacl.a.20.

Google DeepMind. Gemma 3 27B IT model card. Hugging Face model card, 2025. URL https://huggingface.co/google/gemma-3-27b-it/tree/main. Accessed September 17, 2026.

Google DeepMind. Gemini 3.5 Flash model card. Official model card, May 2026. URL https://deepmind.google/models/model-cards/gemini-3-5-flash/. Accessed September 9, 2026.

Peter Grünwald and Wouter M. Koolen. Supermartingales for one-sided tests: Sufficient monotone likelihood ratios are sufficient. *Statistics & Probability Letters*, 229:110574, 2026. doi: 10.1016/j.spl.2025.110574. URL https://doi.org/10.1016/j.spl.2025.110574.

Ziwen Han, Meher Mankikar, Julian Michael, and Zifan Wang. Search-time data contamination. *arXiv preprint arXiv:2508.13180*, 2025. URL https://arxiv.org/abs/2508.13180.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Naila Shafirni Hidayat, Muhammad Dehan Al Kautsar, Alfan Farizki Wicaksono, and Fajri Koto. Simulating training data leakage in multiple-choice benchmarks for LLM evaluation. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*, pp. 21–39. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.eval4nlp-1.3.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Minseo Kwak and Jaehyung Kim. Gap-K%: Measuring top-1 prediction gap for detecting pretraining data. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23391–23405. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1072. URL https://aclanthology.org/2026.acl-long.1072/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Haonan Lu, Bing Liu, and Wenliang Chen. Probing language models for pre-training data detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1587. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.86.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint arXiv:2606.30406*, 2026. URL https://arxiv.org/abs/2606.30406.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719/.

Alexandre Matton, Tom Sherborne, Dennis Aumiller, Elena Tommasone, Milad Alizadeh, Jingyi He, Raymond Ma, Maxime Voisin, Ellen Gilsenan-McMahon, and Matthias Gallé. On leakage of code generation evaluation datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13215–13223. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.772.

Meta Superintelligence Lab. Muse Glimmer-30B model card. Hugging Face model card, August 2026. URL https://huggingface.co/meta-models/Muse-Glimmer-30B/tree/main. Accessed September 17, 2026.

Microsoft Research. Phi-4 model card. Hugging Face model card, 2024. URL https://huggingface.co/microsoft/phi-4/tree/main. Accessed September 17, 2026.

MiniMax. MiniMax-M2.1 model card. Hugging Face model card, 2025. URL https://huggingface.co/MiniMaxAI/MiniMax-M2.1. Accessed September 9, 2026.

MiniMax. MiniMax M2.5: Built for real-world productivity. Official release announcement, February 2026a. URL https://www.minimax.io/news/minimax-m25. Published February 12, 2026. Accessed September 22, 2026.

MiniMax. MiniMax-M3 model card. Hugging Face model card, 2026b. URL https://huggingface.co/MiniMaxAI/MiniMax-M3/tree/main. Accessed September 15, 2026.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8332–8347. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.570.

Mistral AI. Mistral Small 3.1 24B Instruct 2503 model card. Hugging Face model card, 2025. URL https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503/tree/main. Accessed September 17, 2026.

Moonshot AI. Kimi-K2.6 model card. Hugging Face model card, 2026a. URL https://huggingface.co/moonshotai/Kimi-K2.6/tree/main. Accessed September 17, 2026.

Moonshot AI. Kimi K3 tech blog: Open frontier intelligence. Official release announcement, 2026b. URL https://www.kimi.com/en/blog/kimi-k3. Accessed September 9, 2026.

Richard D. Morey and Jeffrey N. Rouder. *BayesFactor: Computation of Bayes Factors for Common Designs*, 2015. URL https://cran.r-project.org/package=BayesFactor. R package.

Shiwen Ni, Xiangtao Kong, Chengming Li, Xiping Hu, Ruifeng Xu, Jia Zhu, and Min Yang. Training on the benchmark is not all you need. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24948–24956, 2025. doi: 10.1609/aaai.v39i23.34678.

NVIDIA. gpt-oss-puzzle-88B model card. Hugging Face model card, March 2026a. URL https://huggingface.co/nvidia/gpt-oss-puzzle-88B/tree/main. Released March 26, 2026. Accessed September 17, 2026.

NVIDIA. NVIDIA Nemotron 3 Super model card. Hugging Face model card, 2026b. URL https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16. Accessed September 9, 2026.

NVIDIA. NVIDIA Nemotron-Labs-3-Puzzle-75B-A9B-BF16 model card. Hugging Face model card, 2026c. URL https://huggingface.co/nvidia/NVIDIA-Nemotron-Labs-3-Puzzle-75B-A9B-BF16/tree/main. Accessed September 17, 2026.

OpenAI. GPT-4 technical report, 2023. URL https://arxiv.org/abs/2303.08774.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. Official model card, August 2025. URL https://openai.com/index/gpt-oss-model-card/. Published August 5, 2025. Accessed September 17, 2026.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. OpenAI product release, July 2026. URL https://openai.com/index/gpt-5-6/. Published July 9, 2026. Accessed September 9, 2026.

OpenCompass. AIME 2025 dataset. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025/tree/a6ad95f611d72cf628a80b58bd0432ef6638f958. Revision a6ad95f611d72cf628a80b58bd0432ef6638f958, AIME 2025 I and II, 30 problems.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/46e624c244cff669223d488defd4e835-Abstract-Conference.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/46499a0622ecf568b72d17b61e45dbd5-Abstract-Datasets_and_Benchmarks_Track.html.

Qwen Team. Qwen3: Think deeper, act faster. Official release announcement, April 2025a. URL https://qwenlm.github.io/blog/qwen3/. Published April 29, 2025. Accessed September 18, 2026.

Qwen Team. Qwen3-32B model card. Hugging Face model card, 2025b. URL https://huggingface.co/Qwen/Qwen3-32B/tree/main. Accessed September 17, 2026.

Qwen Team. Qwen3.5-4B model card. Hugging Face model card, March 2026a. URL https://huggingface.co/Qwen/Qwen3.5-4B/tree/851bf6e806efd8d0a36b00ddf55e13ccb7b8cd0a. Revision 851bf6e806efd8d0a36b00ddf55e13ccb7b8cd0a, dated March 2, 2026.

Qwen Team. Qwen3.5-397B-A17B model card. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-397B-A17B/tree/8472618112abcbd45acbcdc58436aff4233c23f7. Revision 8472618112abcbd45acbcdc58436aff4233c23f7.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Official release announcement, April 2026c. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b. Released April 2026. Accessed September 22, 2026.

Qwen Team. Qwen3.8-27B model card. Hugging Face model card, 2026d. URL https://huggingface.co/Qwen/Qwen3.8-27B/tree/main. Accessed September 15, 2026.

Abhilasha Ravichander, Jillian Fisher, Taylor Sorensen, Ximing Lu, Maria Antoniak, Bill Yuchen Lin, Niloofar Mireshghallah, Chandra Bhagavatula, and Yejin Choi. Information-guided identification of training data imprint in (proprietary) large language models. 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)*, pp. 1962–1978. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.99. URL https://aclanthology.org/2025.naacl-long.99/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Jeffrey N. Rouder, Paul L. Speckman, Dongchu Sun, Richard D. Morey, and Geoffrey Iverson. Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, 16(2):225–237, 2009. doi: 10.3758/PBR.16.2.225. URL https://link.springer.com/article/10.3758/PBR.16.2.225.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of ACL: EMNLP 2023*, pp. 10776–10787. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.722.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704/.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZLJ6XRbdaC.

Soofi-Team. A sovereign, open-source foundation model for german and english, 2026. URL https://arxiv.org/abs/2607.09424v3. Version 3, July 22, 2026.

Angelika M. Stefan, Dimitris Katsimpokis, Quentin F. Gronau, and Eric-Jan Wagenmakers. Expert agreement in prior elicitation and its effects on bayesian inference. *Psychonomic Bulletin & Review*, 29(5):1776–1794, 2022. doi: 10.3758/s13423-022-02074-4. URL https://link.springer.com/article/10.3758/s13423-022-02074-4.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021. doi: 10.1214/20-AOS2020. URL https://doi.org/10.1214/20-AOS2020.

Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(3):822–852, 2022. doi: 10.1111/rssb.12489. URL https://doi.org/10.1111/rssb.12489.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

xAI. Introducing Grok 4.6. Official release announcement, August 2026. URL https://x.ai/news/grok-4-6. Published August 12, 2026. Accessed September 17, 2026.

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

Nicolas Yax, Pierre-Yves Oudeyer, and Stefano Palminteri. LogProber: Disentangling confidence from contamination in LLM responses. *arXiv preprint arXiv:2408.14352*, 2025. URL https://arxiv.org/abs/2408.14352v3.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282. IEEE, 2018. doi: 10.1109/CSF.2018.00027. URL https://doi.org/10.1109/CSF.2018.00027.

Jiatong Yi and Yanyang Li. Membership inference on LLMs in the wild. *arXiv preprint arXiv:2601.11314*, 2026. URL https://arxiv.org/abs/2601.11314.

Z.ai. GLM-5.2 model card. Hugging Face model card, 2026a. URL https://huggingface.co/zai-org/GLM-5.2/tree/main. Accessed September 15, 2026.

Z.ai. GLM-5.3 model card. Hugging Face model card, 2026b. URL https://huggingface.co/zai-org/GLM-5.3. Accessed September 18, 2026.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 363–375. Association for Computing Machinery, 2020. doi: 10.1145/3372297.3417880.

Michał Zawalski, Meriem Boubdir, Klaudia Bałazy, Besmira Nushi, and Pablo Ribalta. Detecting data contamination in LLMs via in-context learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/033f2b82bf0f35add31a88d76f041f58-Abstract-Conference.html.

Huixuan Zhang, Yun Lin, and Xiaojun Wan. PaCoST: Paired confidence significance testing for benchmark contamination detection in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1794–1809. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.97.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. Min-K%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a2e3b4132ab2e0b7a21e6e75da7f91a9-Abstract-Conference.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A Detection Methods and Empirical Distributions

A.1 Debiased Order-Preference Residual

This subsection gives the complete DOPR computation and supporting analyses. Figure 2 (left) in Section 3 gives a visual overview of the method.

For item i, let K_i be its number of choices, $\ell_1, \ldots, \ell_{K_i}$ the displayed labels, and $\mathcal{A}_i = \{a_{i1}, \ldots, a_{iK_i}\}$ the answer identities. Let $a_i^\star$ be the correct answer and $\sigma^{(r)}$ the order obtained

by cyclically shifting the original answer order left by r positions. Thus $\sigma^{(r)}(j)$ is the answer index shown under label ℓ_j. For each $r = 0, \ldots, K_i - 1$, we record every label logit $z_{ir\ell}$. The resulting $K_i \times K_i$ matrix places every answer once under every label.

Let G_i be the matrix's grand mean. For permutation $\sigma^{(r)}$, displayed label ℓ, and answer identity a, define

$$
\begin{aligned}
G_i &= \frac{1}{K_i^2} \sum_{r=0}^{K_i-1} \sum_{j=1}^{K_i} z_{ir\ell_j}, \quad \mu_{i,\sigma^{(r)}} = \frac{1}{K_i} \sum_{j=1}^{K_i} z_{ir\ell_j}, \\
\mu_{i,\ell_j} &= \frac{1}{K_i} \sum_{r=0}^{K_i-1} z_{ir\ell_j}, \qquad \mu_{i,a} = \frac{1}{K_i} \sum_{\substack{r,j: \\ a_{i,\sigma^{(r)}(j)}=a}} z_{ir\ell_j}.
\end{aligned} \tag{1}
$$

Let $\ell_i^\star$ be the original displayed label containing $a_i^\star$, and let $z_i^\star = z_{i0\ell_i^\star}$ be the logit for that correct label under the original mapping. The first two terms below form the *within-permutation correct-answer advantage*, $z_i^\star - \mu_{i,\sigma^{(0)}}$: the correct-answer logit relative to the mean of all displayed-label logits under the original permutation. Subtracting this mean removes only an additive shift shared by all labels, not mapping-dependent changes in the correct answer's separation from its distractors. The remaining terms account for average displayed-label and answer-identity preferences across the cyclic mappings:

$$
R_i = (z_i^\star - G_i) - (\mu_{i,\sigma^{(0)}} - G_i) - (\mu_{i,\ell_i^\star} - G_i) - (\mu_{i,a_i^\star} - G_i). \tag{2}
$$

Algorithm 1 Computing the cyclic DOPR score

Input: Item i with its original answer-to-label mapping, correct answer $a_i^\star$, and model f
Output: DOPR score R_i

1: **for** $r = 0, \ldots, K_i - 1$ **do**
2: Cyclically shift the answer order left by r positions
3: Query f and record every displayed-label logit $z_{ir\ell}$
4: **end for**
5: Compute the four means in Equation (1)
6: Calculate R_i using Equation (2)
7: **return** R_i

For a prespecified group of n items, we use one R_i per item and test $H_0 : \mathbb{E}_{\mathcal{P}}[R_i] \leq 0$ against $H_1 : \mathbb{E}_{\mathcal{P}}[R_i] > 0$ with a one-sided one-sample t-test.

Figures 5 and 6 show how this null expectation appears in practice. The real-benchmark examples in Appendices A.1 and A.2 use unfiltered questions (see Appendix G for filtration). Here, *clean* denotes a model without controlled exposure to the testbed. For a real benchmark, a model–dataset pair is *flagged* if at least one applied detector rejects its null and *unflagged* if none does (Section 5). The clean Qwen3.5-4B on WikiCutoffQA and the unflagged Gemma 3 27B on MMLU-Pro both remain near zero. Under Complete-Item Exposure, DOPR rejects the contaminated Qwen3.5-4B on WikiCutoffQA, while the flagged MiniMax-M3 on HLE also has a positive mean residual and is rejected by DOPR.

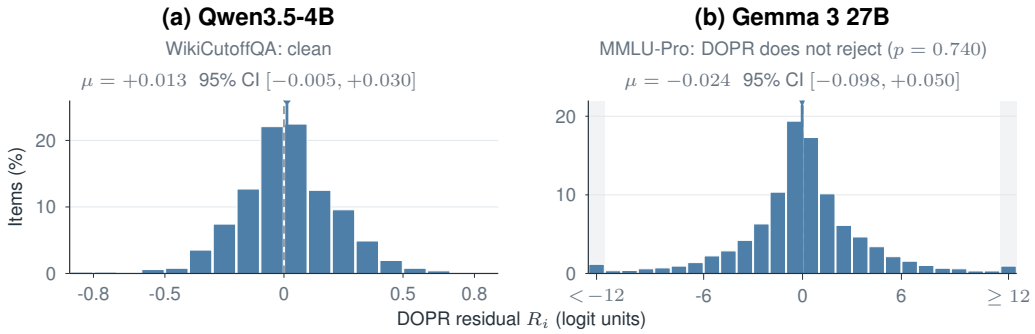

Figure 5: **DOPR residuals for clean and unflagged examples.** A clean Qwen3.5-4B on WikiCutoffQA (left) and the unflagged Gemma 3 27B on MMLU-Pro (right) have mean residuals consistent with zero.

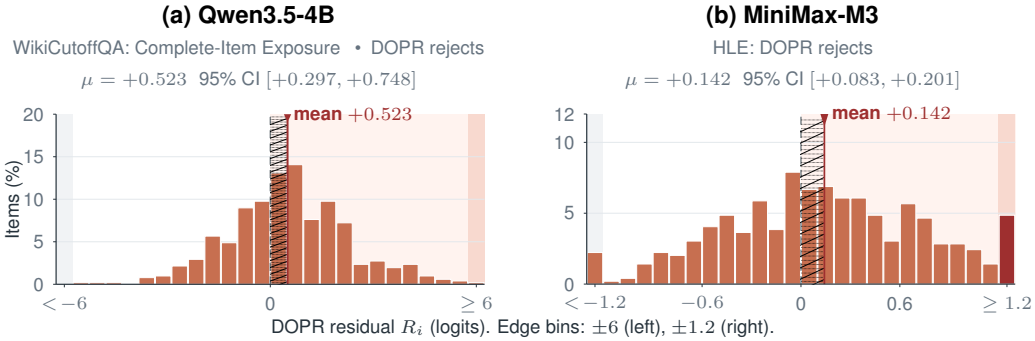

Figure 6: **A positive DOPR shift.** The contaminated Qwen3.5-4B on WikiCutoffQA (left) has mean 0.523 logits after Complete-Item Exposure ($p = 9.35 \times 10^{-6}$). Its histogram includes all 512 questions. Its confidence interval and test use the means of 64 balanced groups of eight questions. The flagged MiniMax-M3 on HLE (right) has mean 0.142 logits ($p = 1.55 \times 10^{-6}$). The panels use different horizontal scales. Edge bins retain all scores beyond the displayed ranges.

Magnitude of removed effects. DOPR's clean-model null requires $\mathbb{E}_{\mathcal{P}}[R_i] \leq 0$, motivating removal of effects that can produce a positive expected score without exposure. For a fixed model and a prespecified set of n benchmark items, let R_i be the DOPR score for item i. Their sample mean and root mean square (RMS) are

$$\bar{R} = \frac{\sum_{i=1}^{n} R_i}{n}, \qquad \mathrm{RMS}_R = \sqrt{\frac{\sum_{i=1}^{n} R_i^2}{n}}.$$

When items are treated as independent, the estimated standard error of the mean satisfies

$$\widehat{\mathrm{SE}}(\bar{R})^2 = \frac{1}{n(n-1)} \sum_{i=1}^{n} (R_i - \bar{R})^2 = \frac{\mathrm{RMS}_R^2 - \bar{R}^2}{n-1}.$$

Thus, when n and $\bar{R}$ are fixed, a smaller item-score RMS means that the R_i values are less dispersed around the same mean, which directly lowers the estimated SE. For positive $\bar{R}$, this increases the one-sided one-sample t-statistic $t = \bar{R}/\widehat{\mathrm{SE}}(\bar{R})$, narrows the confidence interval, and lowers the one-sided p-value. Reducing residual RMS while preserving the contamination-induced mean shift can improve detection power.

The ablation table reports the RMS of the n final item-level scores under each score construction. The component analysis answers a different question: how large is each effect that DOPR removes before forming R_i? Each item supplies a $K_i \times K_i$ logit matrix, so the equations below evaluate a

component at every matrix cell and take its RMS over all $\sum_i K_i^2$ cells.

$$\mathrm{RMS}_{\mathrm{map}} = \left[\frac{\sum_{i=1}^{n} K_i \sum_{r=0}^{K_i-1} (\mu_{i,\sigma^{(r)}} - G_i)^2}{\sum_{i=1}^{n} K_i^2}\right]^{1/2},$$

$$\mathrm{RMS}_{\mathrm{label}} = \left[\frac{\sum_{i=1}^{n} K_i \sum_{j=1}^{K_i} (\mu_{i,\ell_j} - G_i)^2}{\sum_{i=1}^{n} K_i^2}\right]^{1/2}, \tag{3}$$

$$\mathrm{RMS}_{\mathrm{answer}} = \left[\frac{\sum_{i=1}^{n} K_i \sum_{a \in \mathcal{A}_i} (\mu_{i,a} - G_i)^2}{\sum_{i=1}^{n} K_i^2}\right]^{1/2}.$$

For the clean Qwen3.5-4B on WikiCutoffQA, the mapping, label, and answer RMS magnitudes are 0.1628, 0.3769, and 0.4745 logits, respectively (Figure 7). The root-sum-square of these three values is 0.6275 logits. For comparison, applying the three subtractions in Equation (2) to every matrix cell leaves residuals with RMS 0.2098 logits. Equation (2) selects the residual at the original correct-answer cell as the item-level DOPR score R_i. RMS measures magnitude rather than direction. A large effect can shift the group mean, increase its sampling error, or do both. We therefore complement these descriptive magnitudes with matched score ablations in Table 3.

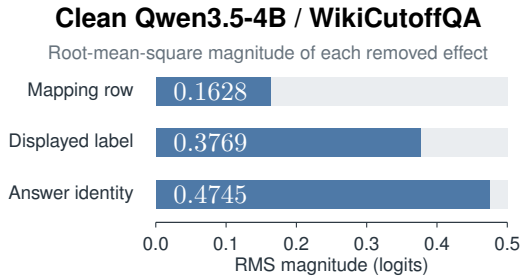

Figure 7: **Magnitude of the effects removed by DOPR.** The bars show the three effects removed before testing, with each RMS averaged over every cell in the cyclic logit matrices. For each item, DOPR subtracts only the three effect values attached to the original correct-answer cell, as in Equation (2). The root-sum-square of the three displayed RMS values is 0.6275 logits. Applying the same subtraction to every cell leaves residual RMS 0.2098 logits.

Table 3: **DOPR ablations on clean and unflagged examples.** Each row omits the named subtraction while the other two remain. All rows use cyclic mappings and a fixed item set within each block.

Model / dataset	Omitted subtraction	Mean	95% CI	SE	RMS
Clean Qwen3.5-4B / WikiCutoffQA	None (DOPR)	0.013	[−0.005, 0.030]	0.0089	0.202
	Mapping	0.008	[−0.015, 0.030]	0.0115	0.261
	Displayed label	−0.003	[−0.037, 0.032]	0.0177	0.400
	Answer identity	−0.053	[−0.090, −0.016]	0.0188	0.429
	All three (raw score)	−0.073	[−0.124, −0.022]	0.0259	0.590
Unflagged Gemma 3 27B / MMLU-Pro	None (DOPR)	−0.024	[−0.098, 0.050]	0.0377	4.000
	Mapping	−0.024	[−0.101, 0.052]	0.0390	4.143
	Displayed label	−0.006	[−0.104, 0.093]	0.0503	5.340
	Answer identity	10.830	[10.578, 11.081]	0.1285	17.422
	All three (raw score)	10.848	[10.589, 11.108]	0.1325	17.768

On WikiCutoffQA, full DOPR reduces RMS from 0.590 to 0.202 and SE from 0.0259 to 0.0089. The mapping and displayed-label corrections barely change the mean but lower RMS and SE by removing signed effects that cancel in aggregate. For the unflagged Gemma 3 27B on MMLU-Pro, omitting the answer-identity subtraction gives a mean of 10.830 logits (95% CI $[10.578,\ 11.081]$). Full DOPR gives -0.024 logits with an interval containing zero. Without this correction, the unflagged Gemma model would be strongly flagged on MMLU-Pro.

Sanity check under Shuffled-Order Exposure. Shuffled-Order Exposure redraws the answer permutation during training while retaining the original question and answer wording (Section 4). It therefore does not consistently reinforce the benchmark's original mapping, but can reinforce its wording. Consistent with this distinction, DOPR does not reject its null on the WikiCutoffQA checkpoint, whereas DWPC-QA, which tests preference for the original wording rather than the original answer order, does (Figure 8). Table 1 likewise shows DOPR's near-chance AUROC under Shuffled-Order Exposure in both testbeds.

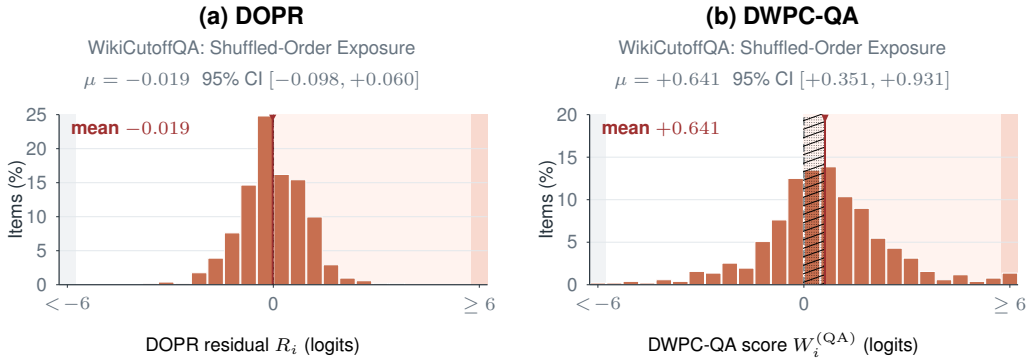

Figure 8: **DOPR and DWPC-QA after Shuffled-Order Exposure.** The same Qwen3.5-4B checkpoint and 512 WikiCutoffQA questions give mean -0.019 logits for DOPR ($p = 0.683$) and 0.641 for DWPC-QA ($p = 1.98 \times 10^{-5}$). Confidence intervals and tests use the means of 64 balanced eight-question groups. Edge bins pool scores beyond ± 6.

A.2 Debiased Wording-Preference Contrast

This subsection gives the complete DWPC computation and supporting analyses. Figure 2 (right) in Section 3 gives a visual overview of the method.

For item i, let K_i be its number of choices and let O and P denote the original and held-out wordings. DWPC-Q changes only the question. DWPC-QA uses aligned paraphrases of the question and every answer. For a prespecified variant $v \in \{\mathrm{Q}, \mathrm{QA}\}$, we apply the same cyclic mappings $\sigma^{(r)}$ from Appendix A.1 to both wordings. Let $z^{(v)}_{iwr\ell}$ be the logit for label ℓ under wording $w \in \{O, P\}$ and mapping r, and let $\ell^\star_i(r)$ be the label containing the correct answer. We define

$$\begin{aligned} \bar{z}^{(v)}_{iwr} &= \frac{1}{K_i} \sum_{j=1}^{K_i} z^{(v)}_{iwr\ell_j}, \qquad A^{(v)}_{iwr} = z^{(v)}_{iwr,\ell^\star_i(r)} - \bar{z}^{(v)}_{iwr}, \\ \bar{A}^{(v)}_{iw} &= \frac{1}{K_i} \sum_{r=0}^{K_i-1} A^{(v)}_{iwr}, \quad W^{(v)}_i = \bar{A}^{(v)}_{iO} - \bar{A}^{(v)}_{iP}. \end{aligned} \tag{4}$$

Subtracting $\bar{z}^{(v)}_{iwr}$ removes a shift shared by all labels in one query. Averaging over the cyclic mappings places the correct answer once under every label and balances stable displayed-label preferences. Under the meaning-preservation assumption, both wordings contain the same underlying answers. Subtracting their centered support therefore cancels any answer-identity preference that is stable across the wordings, while wording-dependent effects remain. Thus a positive $W^{(v)}_i$ means that the original wording gives the correct answer greater relative support than the held-out wording.

Algorithm 2 Computing the cyclic DWPC score

Input: Item i, variant $v \in \{\mathrm{Q}, \mathrm{QA}\}$, paired wordings O and P, and model f
Output: DWPC score $W_i^{(v)}$
1: **for** $r = 0, \ldots, K_i - 1$ **do**
2: Apply cyclic mapping $\sigma^{(r)}$ to both wordings
3: Query f on each wording and record every displayed-label logit
4: **end for**
5: Compute the centered advantages and mapping averages in Equation (4)
6: Calculate $W_i^{(v)}$ using Equation (4)
7: **return** $W_i^{(v)}$

For a prespecified group of n items, we use one $W_i^{(v)}$ per item and test $H_0 : \mathbb{E}_{\mathcal{P}}[W_i^{(v)}] \leq 0$ against $H_1 : \mathbb{E}_{\mathcal{P}}[W_i^{(v)}] > 0$ with a one-sided one-sample t-test.

Magnitude of removed effects. As for DOPR, debiasing aims to support the clean-model null and reduce score dispersion. At fixed sample size, lower item-score RMS can improve power if the contamination-induced mean shift is preserved. We report component RMS to quantify what is removed, and ablation means and SEs to assess its effect on the group test. For the component analysis, we measure the shared-logit shift removed by within-query centering and the displayed-label-dependent wording effect removed by mapping averaging:

$$S_{ir}^{(v)} = \bar{z}_{iOr}^{(v)} - \bar{z}_{iPr}^{(v)},$$
$$L_{i\ell_j}^{(v)} = \frac{1}{K_i} \sum_{r=0}^{K_i-1} \left[\left(z_{iOr\ell_j}^{(v)} - \bar{z}_{iOr}^{(v)} \right) - \left(z_{iPr\ell_j}^{(v)} - \bar{z}_{iPr}^{(v)} \right) \right].$$

Here $S_{ir}^{(v)}$ is the change in the mean label logit between the original and paraphrased queries under mapping r. It can depend on wording and mapping, but because it shifts all labels equally, it contains no correct-answer-relative wording signal. The term $L_{i\ell_j}^{(v)}$ is the mapping-averaged wording contrast associated with displayed label ℓ_j after shared shifts are removed. These terms quantify the two effects already removed while Equation (4) forms $W_i^{(v)}$. Subtracting the all-label mean within each query removes $S_{ir}^{(v)}$. Averaging the correct-answer advantage over cyclic mappings places it once under every displayed label, so the label-specific contrasts $L_{i\ell_j}^{(v)}$ cancel. They are not additional corrections applied after $W_i^{(v)}$ is computed.

The corresponding root-mean-square (RMS) magnitudes are

$$\mathrm{RMS}_{\mathrm{shared}}^{(v)} = \left[\frac{1}{\sum_{i=1}^{n} K_i} \sum_{i=1}^{n} \sum_{r=0}^{K_i-1} \left(S_{ir}^{(v)} \right)^2 \right]^{1/2},$$
$$\mathrm{RMS}_{\mathrm{label}}^{(v)} = \left[\frac{1}{\sum_{i=1}^{n} K_i} \sum_{i=1}^{n} \sum_{j=1}^{K_i} \left(L_{i\ell_j}^{(v)} \right)^2 \right]^{1/2}.$$

For the clean Qwen3.5-4B on WikiCutoffQA, the shared-logit-shift and displayed-label RMS magnitudes are 0.734 and 0.273 logits for DWPC-Q, and 0.640 and 0.449 logits for DWPC-QA (Figure 11).

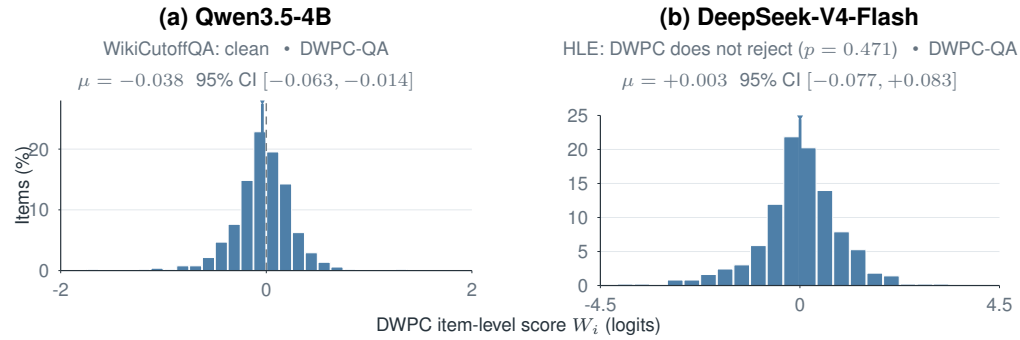

Figure 9: **DWPC-QA scores for clean and unflagged examples.** The clean Qwen3.5-4B on WikiCutoffQA (left) and the unflagged DeepSeek-V4-Flash on HLE (right) show no positive mean shift. The WikiCutoffQA mean is slightly negative, while the HLE confidence interval includes zero.

On WikiCutoffQA, the small negative means (-0.032 for DWPC-Q and -0.038 for DWPC-QA) indicate a slight preference by this Qwen3.5-4B model for the paraphrases, which receive slightly greater correct-answer support than the original wordings.

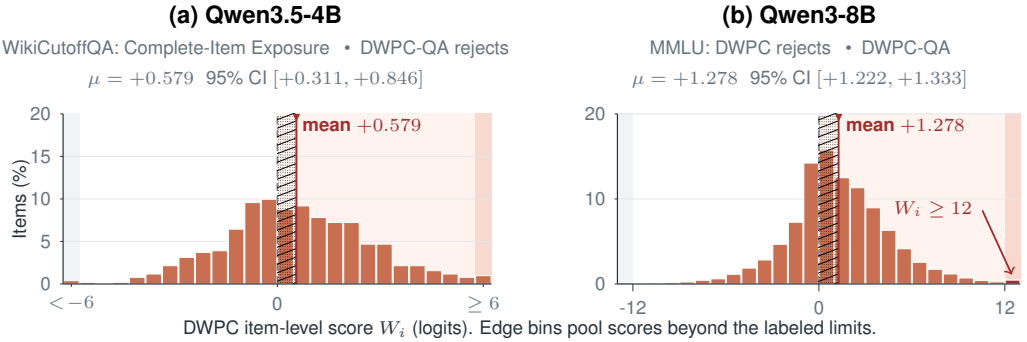

Figure 10: **DWPC distributions for contaminated or flagged examples.** Qwen3.5-4B after Complete-Item Exposure on WikiCutoffQA (left) has a positive DWPC-QA mean of 0.579 logits ($p = 2.77 \times 10^{-5}$). Its histogram shows 512 questions. Its confidence interval and test use the 64 balanced group means, as in Figure 6. The flagged Qwen3-8B on MMLU (right) has a positive DWPC-QA mean of 1.278 logits ($p < 10^{-300}$).

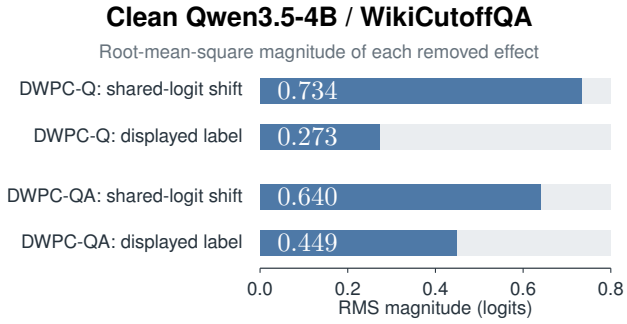

Figure 11: **Magnitude of effects removed by DWPC.** RMS magnitude of the shared-logit shift and displayed-label-dependent wording effect, shown separately for DWPC-Q and DWPC-QA. The shared-logit shift is more than twice as large as the retained wording-dependent answer effect in both variants.

Across the K_i answer identities for each item, the retained wording-dependent answer effect has RMS 0.314 for DWPC-Q and 0.298 for DWPC-QA. Thus the shared-logit shift is more than twice as large in both variants. The displayed-label-dependent wording effect is 0.87 times as large for DWPC-Q and 1.51 times as large for DWPC-QA.

Table 4 compares four score constructions. Full DWPC centers the correct-answer logit within each query and then averages the original-minus-paraphrase contrast over cyclic mappings. Without centering, the average shift shared by all labels remains in $W_i^{(v)}$. Using only the original mapping after centering leaves displayed-label-dependent wording effects and mapping-label cell interactions in $W_i^{(v)}$. Omitting both steps leaves both sources of removed variation. A stable answer-identity preference cancels in every row because the same answer is compared across the two wordings.

Table 4: **DWPC debiasing ablations on clean and unflagged examples.** The score names state which calculation is retained. Comparisons within each block use the same items and cyclic mappings. Intervals are two-sided 95% confidence intervals.

Model / dataset	Variant	Score construction	Mean	95% CI	SE	RMS
Clean Qwen3.5-4B / WikiCutoffQA	Q	Centered, mapping-averaged (DWPC)	-0.032	$[-0.056, -0.007]$	0.0125	0.284
	Q	Uncentered, mapping-averaged	-0.111	$[-0.179, -0.044]$	0.0344	0.786
	Q	Centered, original mapping	-0.061	$[-0.099, -0.024]$	0.0191	0.436
	Q	Uncentered, original mapping	-0.141	$[-0.215, -0.066]$	0.0380	0.870
	QA	Centered, mapping-averaged (DWPC)	-0.038	$[-0.063, -0.014]$	0.0125	0.286
	QA	Uncentered, mapping-averaged	-0.104	$[-0.162, -0.046]$	0.0296	0.677
	QA	Centered, original mapping	-0.063	$[-0.114, -0.012]$	0.0258	0.588
	QA	Uncentered, original mapping	-0.136	$[-0.212, -0.060]$	0.0388	0.888
Unflagged DeepSeek-V4-Flash / HLE	QA	Centered, mapping-averaged (DWPC)	0.003	$[-0.077, 0.083]$	0.0407	0.904
	QA	Uncentered, mapping-averaged	-1.371	$[-1.539, -1.203]$	0.0856	2.343
	QA	Centered, original mapping	0.037	$[-0.114, 0.187]$	0.0767	1.703
	QA	Uncentered, original mapping	-1.375	$[-1.621, -1.130]$	0.1249	3.095

On WikiCutoffQA, query centering reduces RMS from 0.786 to 0.284 for DWPC-Q and from 0.677 to 0.286 for DWPC-QA. After centering, mapping averaging reduces RMS from 0.436 to 0.284 for DWPC-Q and from 0.588 to 0.286 for DWPC-QA. On HLE, query centering removes the strong negative shift, moving the mean from -1.371 to 0.003 with mapping averaging, while mapping averaging cuts RMS from 1.703 to 0.904 after centering. Thus the two corrections serve distinct roles: query centering protects the zero reference, while mapping averaging lowers RMS and SE. As explained for DOPR, the lower SE makes the one-sided test more sensitive to a positive wording shift. The centered original-mapping contrast is an operational ablation rather than a pure displayed-label ablation because it also reintroduces mapping-specific interaction noise.

A.3 Choice-Blind Answering

This subsection gives the exact CBA prompt, output rule, and testing procedure. Figure 3 in Section 3 illustrates the choice-blind query and the distinction between the two null hypotheses.

For item i, let q_i be the question, $\mathcal{L}_i = \{\ell_1, \ldots, \ell_{K_i}\}$ its valid labels, and $\ell_i^\star$ its hidden correct label. CBA gives the model only q_i and $\mathcal{L}_i$, with the instruction to return exactly one valid label. The answer texts, their order, and their label mapping remain hidden. For four-choice questions, the complete prompt is:

> {question}
> Return only a single letter as the response (e.g. A or B or C or D). If you are not sure, answer with your best guess. Be concise and don't include the textual answer in your response, only the letter.

We query once using a decoding configuration fixed before evaluation. After removing surrounding whitespace, we accept only an exact valid label. Every other output, including an empty or malformed response, is invalid and counts as a nonmatch. For the real-benchmark application, we instead use a case-insensitive parser that also extracts valid labels from supported prose and formatted answers, including final-answer statements and boxed answers. We condition on valid parsed responses, as

detailed in Appendix H. For a valid output, $\hat{\ell}_i$ denotes the label parsed from the response. No $\hat{\ell}_i$ is assigned when parsing fails. Let n be the number of benchmark questions included in the audit and S the number of valid parsed labels that equal their hidden correct labels, with observed value S_{obs}. When applying CBA to real benchmarks, we exclude items whose complete answer mapping or correct answer label can be recovered reliably from the question alone (Appendix G).

Uniform-guessing null hypothesis (common K). Assume first that all n questions share K choices and label set $\mathcal{L}$. Under H_0^{uniform}, the match count S has the same distribution as $S_{\text{uniform}} \sim \text{Binomial}(n, 1/K)$. The exact upper tail is

$$p_{\text{uniform}} = \Pr(S_{\text{uniform}} \geq S_{\text{obs}}) = \sum_{s=S_{\text{obs}}}^{n} \binom{n}{s} \left(\frac{1}{K}\right)^{s} \left(1 - \frac{1}{K}\right)^{n-s}. \tag{5}$$

This test asks whether CBA accuracy exceeds random guessing.

Answer-key shuffle null hypothesis. Holding all model outputs fixed, let Ω be the set of candidate answer keys $\boldsymbol{\ell}' = (\ell'_1, \ldots, \ell'_n)$ obtained by uniformly reassigning the observed correct labels across items. Here, ℓ'_i is the label placed at item i. Every key in Ω preserves the observed label counts. For $\boldsymbol{\ell}' \in \Omega$, let $S_{\text{shuffle}}(\boldsymbol{\ell}')$ count the valid parsed predictions that equal the corresponding candidate labels. Under H_0^{shuffle}, conditional on the observed outputs and correct-label counts, S has the same distribution as $S_{\text{shuffle}}(\boldsymbol{\ell}')$ for $\boldsymbol{\ell}' \sim \text{Uniform}(\Omega)$. The test assumes that correct labels are exchangeable across questions, conditional on the observed outputs and correct-label counts. The exact upper tail is

$$p_{\text{shuffle}} = \Pr_{\boldsymbol{\ell}' \sim \text{Uniform}(\Omega)} \left(S_{\text{shuffle}}(\boldsymbol{\ell}') \geq S_{\text{obs}}\right). \tag{6}$$

This preserves the model and answer-key label frequencies and breaks only their item-level pairing. We compute the upper-tail probability exactly.

Algorithm 3 Computing CBA when every item has K choices

Input: Items $\mathcal{I}$, questions q_i, labels $\mathcal{L}$, hidden correct labels $\ell_i^\star$, and model f
Output: Match count and exact uniform and shuffle p-values

1: **for** $i \in \mathcal{I}$ **do**
2: Query f once using only q_i and $\mathcal{L}$
3: Accept an exact valid label as $\hat{\ell}_i$, or mark the output invalid
4: Record a match iff the output is valid and $\hat{\ell}_i = \ell_i^\star$
5: **end for**
6: $S_{\text{obs}} \leftarrow$ number of recorded matches
7: Compute p_{uniform} by Equation (5)
8: Compute p_{shuffle} exactly by Equation (6)
9: **return** S_{obs}, p_{uniform}, and p_{shuffle}

Controlled comparison. For the controlled comparison, the uniform-guessing test counts invalid responses as nonmatches. The answer-key shuffle test uses only questions with valid responses and shuffles their associated answer keys.

A.3.1 Comparison Between the Two CBA Null Hypotheses

For the n benchmark questions in the audit and each label $\ell \in \mathcal{L}$, where $\mathcal{L}$ is the common label set defined above, let $\hat{\pi}_\ell$ be the fraction of model predictions equal to ℓ, and let $\pi_\ell^\star$ be the fraction of correct labels equal to ℓ. Let S_{shuffle} be the match count after a random answer-key shuffle. The uniform-guessing null has expected match count n/K, while answer-key shuffle has

$$\mathbb{E}[S_{\text{shuffle}}] = n \sum_{\ell \in \mathcal{L}} \hat{\pi}_\ell \pi_\ell^\star.$$

Each term $\hat{\pi}_\ell \pi_\ell^\star$ is the product of two scalar frequencies. It is the probability that a fixed prediction and a shuffled correct label both equal ℓ, so their sum is the expected match rate. Matching label

biases raise this rate above $1/K$, while opposing biases lower it. The exact p-values still depend on the full null distributions. Table 5 gives one example of each failure of the uniform-guessing test.

Table 5: **Examples where the two CBA null hypotheses disagree.** The prediction-by-answer count table and summary rows show exact inclusive upper-tail p-values for two constructed examples.

Anti-aligned example: predicted label by hidden correct label

Predicted	Correct A	Correct B	Correct C	Correct D	Row total
A	0	0	0	60	60
B	0	10	0	0	10
C	0	0	10	0	10
D	0	0	0	0	0
Column total	0	10	10	60	80

Case	S_{obs}/n	Uniform-guessing mean	p_{uniform}	Shuffle mean	p_{shuffle}
Anti-aligned marginals	$20/80$	20	0.543	2.5	1.53×10^{-24}
Always A, 70% A keys	$70/100$	25	4.37×10^{-21}	70	1

In the anti-aligned example, accuracy is exactly 25% and the uniform-guessing test does not reject ($p = 0.543$). Yet every B and C prediction is correct: the 20 observed matches are eight times the shuffle mean of 2.5, giving strong item-specific alignment ($p = 1.53 \times 10^{-24}$). In the matching-bias example, always predicting A gives 70% accuracy and a small uniform-guessing p-value, but answer-key shuffle gives $p = 1$ because it preserves the shared A bias.

A.3.2 Variable Numbers of Choices

For datasets with different numbers of answer choices, we extend both CBA null hypotheses to the item-specific choice counts K_i.

Uniform-guessing null hypothesis. For fixed choice counts K_i, let $X_i^{\text{uniform}} \sim \text{Bernoulli}(1/K_i)$ be independent success indicators, and let $S_{\text{uniform}} = \sum_{i=1}^{n} X_i^{\text{uniform}}$ count the total matches. Under H_0^{uniform}, S follows this Poisson-binomial distribution. For any subset $\mathcal{M} \subseteq \{1, \ldots, n\}$, the probability that exactly the questions in $\mathcal{M}$ are matched is $\prod_{i \in \mathcal{M}} K_i^{-1} \prod_{i \notin \mathcal{M}} (1 - K_i^{-1})$. Summing this probability over all subsets of size s gives the Poisson-binomial probability mass function

$$\begin{aligned} \Pr(S_{\text{uniform}} = s) &= \sum_{\substack{\mathcal{M} \subseteq \{1,\ldots,n\} \\ |\mathcal{M}| = s}} \prod_{i \in \mathcal{M}} \frac{1}{K_i} \prod_{i \notin \mathcal{M}} \left(1 - \frac{1}{K_i}\right), \\ p_{\text{uniform}} &= \Pr(S_{\text{uniform}} \geq S_{\text{obs}}) = \sum_{s = S_{\text{obs}}}^{n} \Pr(S_{\text{uniform}} = s). \end{aligned} \tag{7}$$

Thus p_{uniform} is the null probability of at least S_{obs} matches across the n benchmark questions.

Answer-key shuffle null hypothesis. For each choice count k, let $\mathcal{I}_k = \{i \in \{1, \ldots, n\} : K_i = k\}$ be the indices of the k-choice questions. Within each $\mathcal{I}_k$, we keep the predictions fixed and reassign the observed correct labels uniformly at random, independently across the groups. Under H_0^{shuffle}, conditional on all observed outputs and each group's correct-label counts, S follows the resulting total match count distribution. The test assumes that correct labels are jointly exchangeable within the groups, conditional on these outputs and counts. If S_k is the shuffled match count in $\mathcal{I}_k$, then

$$p_{\text{shuffle}} = \Pr\left(\sum_k S_k \geq S_{\text{obs}}\right). \tag{8}$$

We compute this probability exactly by combining the groups' match count distributions. When invalid outputs count as nonmatches, their locations remain fixed during shuffling. In our reported answer-key shuffle analyses, we instead remove invalid responses and their keys before forming the groups.

A.3.3 SUBJECT-CONDITIONED ANSWER-KEY SHUFFLING

The CBA answer-key shuffle (see Section 3) keeps the model's predictions fixed and randomly reassigns correct answer labels across questions with the same available answer labels. It asks how often such a shuffle produces at least as many matches with the model's predictions as the original answer key. This preserves overall answer-label frequencies but ignores subject membership. If both predictions and correct answers favor A in one subject and B in another, global shuffling breaks this shared structure and can make the observed alignment appear unusually strong.

We test whether such subject-level structure explains our CBA findings by reanalyzing all 35 models on MMLU and MMLU-Pro, which contain 57 and 14 subjects, respectively. We restrict each answer-key shuffle to questions within the same subject, retaining the groups with the same available answer labels. This preserves answer-label frequencies within each subject. We compute the exact null distribution of the total number of matches across subjects and use the same E-value calibration and aggregation as in the original analysis.

The rejection pattern and model rankings are largely preserved (Table 6). Three of the four reported families retain every e-BH and Holm decision. MMLU CBA retains 27 of its 28 e-BH rejections and 29 of its 30 Holm rejections, with DeepSeek-V4-Flash and Gemma-3-27B-IT losing the respective flags. Spearman correlations between the model rankings under the two nulls range from 0.9798 to 0.9995. All four families retain the same top-ten models, and both three-detector aggregates retain their exact top-ten order.

Table 6: Rejection counts at 5% under global $\rightarrow$ within-subject shuffling. Spearman ρ compares model rankings by E-value under the two nulls. Non-MMLU combines MMLU-Pro, GPQA-D, and HLE, with only its MMLU-Pro CBA component changed.

Family	Models	e-BH	Holm	Spearman ρ
MMLU, three detectors	28	$26 \rightarrow 26$	$28 \rightarrow 28$	0.9995
MMLU, CBA	35	$28 \rightarrow 27$	$30 \rightarrow 29$	0.9964
Non-MMLU, three detectors	28	$9 \rightarrow 9$	$10 \rightarrow 10$	0.9956
Non-MMLU, CBA	35	$0 \rightarrow 0$	$0 \rightarrow 0$	0.9798

A.4 COMBINING EVIDENCE

In this section, we describe the construction of the detector-specific e-values and how they are aggregated, and how the resulting evidence is used for single and multiple hypothesis testing.

DOPR and DWPC-QA. DOPR and DWPC-QA both test a one-sided mean null hypothesis, as detailed in Section 3. Under the normality assumption, both methods use a one-sided one-sample t-test. We therefore use the same e-value construction for both methods. In particular, we construct a likelihood-ratio e-value of the form

$$E(T) = \frac{q(T)}{p(T)},$$

where T is the t-test statistic, p is its density at the boundary of the null hypothesis, and q is an alternative density. Such a likelihood ratio is a valid e-value under p (Vovk & Wang, 2021). The validity of this likelihood-ratio construction for the composite one-sided null follows from Grünwald & Koolen (2026).

We next specify the boundary null density p and the alternative density q. Let $X_1, \ldots, X_n$ denote the retained item-level scores, where $X_i = R_i$ for DOPR, $X_i = W_i^{(\mathrm{QA})}$ for DWPC-QA. We assume $X_i \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ and test the null hypothesis $\mathcal{H}_0 : \mu \leq 0$. Define the one-sample t-statistic

$$T = \frac{\sqrt{n}\,\bar{X}}{s}, \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2.$$

Let $\nu = n - 1$, let $\delta = \mu/\sigma$ denote the standardized effect size, and let $f_{\nu,\lambda}$ denote the t-density with ν degrees of freedom and noncentrality parameter λ. Then T has density $f_{\nu,\sqrt{n}\,\delta}$. At the boundary of the null, $\delta = 0$, and hence $p(t) = f_{\nu,0}(t)$.

To construct the alternative density q, we average over positive standardized effect sizes rather than selecting a single value. Following the Cauchy alternative of Rouder et al. (2009), we use the half-Cauchy density

$$h_r(d) = \frac{2r}{\pi(r^2 + d^2)}, \qquad d > 0.$$

We set $r = 1/\sqrt{2}$, corresponding to the "medium" default in the `BayesFactor` package; the same scale has also been used for one-sided t-tests (Morey & Rouder, 2015, Stefan et al., 2022).

The resulting alternative density is

$$q_{n,r}(t) = \int_0^\infty f_{\nu,\sqrt{n}\,d}(t)\, h_r(d)\, \mathrm{d}d.$$

Thus, we get

$$E(T) = \frac{q_{n,r}(T)}{f_{\nu,0}(T)} = \int_0^\infty \frac{f_{\nu,\sqrt{n}\,d}(T)}{f_{\nu,0}(T)} \frac{2r}{\pi(r^2 + d^2)}\, \mathrm{d}d, \qquad r = \frac{1}{\sqrt{2}}.$$

We denote the resulting e-value by E_R for DOPR and by E_W for DWPC-QA.

CBA. For the CBA method, we take the approach of converting its valid answer-key-shuffle p-value, denoted by p_C, into an e-value. To this end, we apply the integrated p-to-e calibrator of Vovk & Wang (2021, Equation (2)). For completeness, applying this calibrator gives

$$E_C = \frac{1 - p_C + p_C \log p_C}{p_C(-\log p_C)^2}, \qquad 0 < p_C < 1,$$

with $E_C = 1/2$ when $p_C = 1$.

Aggregation. Let E_{R}, E_{W}, and E_{C} denote the three detector-specific e-values. We define their aggregate e-value as

$$E_{\text{combined}} = \frac{E_{\mathrm{R}} + E_{\mathrm{W}} + E_{\mathrm{C}}}{3}.$$

Under the global null, all three component e-values have expectation at most one. Hence,

$$\mathbb{E}[E_{\text{combined}}] = \frac{1}{3}\left(\mathbb{E}[E_{\text{R}}] + \mathbb{E}[E_{\text{W}}] + \mathbb{E}[E_{\text{C}}]\right) \leq 1.$$

Importantly, this result follows directly from linearity of expectation and does not require the detector-specific e-values to be independent. We use E_{combined} as our overall measure of evidence against the global null. The same argument applies whenever we average a prespecified collection of e-values across benchmarks.

Hypothesis testing and error control. Given the e-values and p-values computed for the global null hypotheses, we use them to determine which null hypotheses can be rejected while controlling the corresponding error metric. For a single hypothesis, we control the type-I error, namely, the probability of rejecting the null hypothesis when it is true. An e-value E provides a level-α test by rejecting whenever

$$E \geq \frac{1}{\alpha}.$$

Thus, the type-I error is controlled at level α. For example, an e-value of at least 20 is sufficient for rejection at level 0.05.

When testing multiple hypotheses simultaneously, we consider two error metrics. The first is the false discovery rate (FDR) (Benjamini & Hochberg, 1995). Let R denote the total number of rejected hypotheses and V the number of rejected true null hypotheses. The FDR is defined as

$$\text{FDR} = \mathbb{E}\left[\frac{V}{R \vee 1}\right],$$

namely, the expected proportion of false discoveries among the rejected hypotheses. To control the FDR at level α, we apply the e-BH procedure (Wang & Ramdas, 2022).

The second metric is the family-wise error rate (FWER), defined as

$$\text{FWER} = \Pr(V \geq 1),$$

namely, the probability of making at least one false rejection among the tested hypotheses. To control the FWER at level α, we apply the Holm procedure (Holm, 1979) to the aggregated p-values.

B General-Knowledge Controls

Accurate answers alone should not cause a contamination detector to flag a model. To test whether existing knowledge alone causes such flags, we evaluate the detectors introduced in Section 3 on freshly written MCQs about general knowledge that the models may already have (e.g., why a whale must surface even when it has food underwater: to breathe).

We construct two datasets, each containing 50 freshly written four-choice questions spanning several subjects, with paraphrases of both the questions and answer choices. One dataset is designed to be easier and the other harder. We measure accuracy using ten runs per question with different random seeds and exact answer-letter matching. Figure 12 shows an easy example.

We evaluate Qwen3.5-4B, Nemotron-3-Super-120B, Qwen3-1.7B, Qwen3-8B, and Qwen3.6-35B-A3B. The latter three Qwen models are flagged by DWPC-QA on MMLU-Pro (Appendix H).

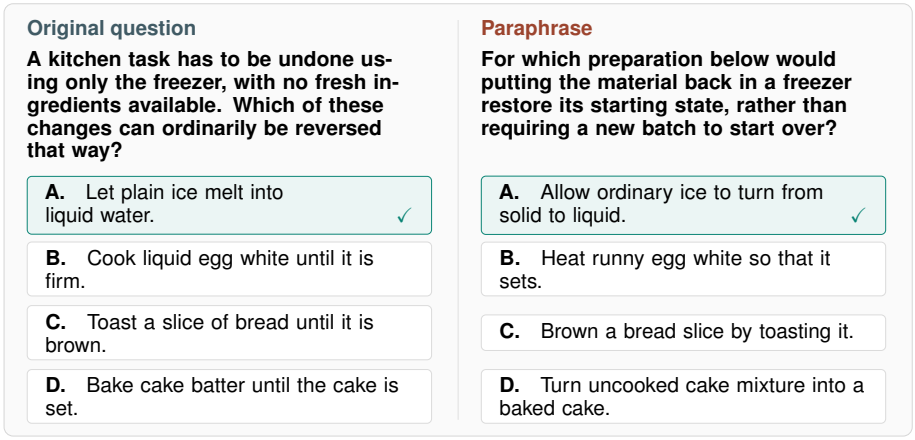

Figure 12: **General-knowledge control example.** Correct answers are highlighted in green and marked with a checkmark.

At $\alpha = 0.05$, DOPR, DWPC-QA, and CBA flag none of the reported models on either control dataset. CBA uses the primary answer-key shuffle test on valid responses.

Across the five models, mean accuracy on the easy dataset is 95.3%, and mean accuracy on the hard dataset is 47.2% (random guessing yields 25% accuracy in expectation).

Table 7: **General-knowledge control results.** Dataset-level p-values and flag outcomes on the easy and hard controls.

Dataset	Model	DOPR	DWPC-QA	CBA	Flagged method
Easy	Qwen3.5-4B	0.661	0.882	0.773	**None**
	Nemotron-3-Super-120B	0.379	0.914	0.471	**None**
	Qwen3-1.7B	0.072	0.969	0.948	**None**
	Qwen3-8B	0.380	0.452	0.591	**None**
	Qwen3.6-35B-A3B	0.483	0.817	0.486	**None**
Hard	Qwen3.5-4B	0.259	0.194	0.510	**None**
	Nemotron-3-Super-120B	0.681	0.237	0.209	**None**
	Qwen3-1.7B	0.378	0.591	0.935	**None**
	Qwen3-8B	0.836	0.192	0.495	**None**
	Qwen3.6-35B-A3B	0.722	0.149	0.451	**None**

C TESTBED CONSTRUCTION AND VALIDATION

The testbeds in Section 4 use recent Wikipedia entries (WCQA) and fictional sources (SSQA). Table 8 summarizes their sizes and exposure/control splits. Figure 13 shows a question from each testbed. Both datasets will be released upon acceptance.

Table 8: **Testbed sizes and exposure/control splits.**

Testbed	Sources	Questions	Exposure set	Control set
WikiCutoffQA	64	512	256	256
SyntheticSourcesQA	256	2,048	1,024	1,024

WikiCutoffQA. Our sources are real English Wikipedia entries. The source versions are dated May 3–August 10, 2026, after the March 2, 2026 revision of Qwen3.5-4B used in our experiments (Qwen Team, 2026a). We use GPT 5.6 Sol to write MCQs about facts in these entries. Target facts and supporting passages are extracted programmatically from the entries. For each fact, the model is

instructed to write a natural question with one uniquely supported answer. The three distractors must be plausible alternatives of the same type and format as the correct answer, while remaining incorrect for the question.

SyntheticSourcesQA. We use GPT 5.6 Sol to generate both technical fictional sources and MCQs about their contents. The sources include botany manuals, court procedures, and relay codebooks. They define invented entities, identifiers, and rules designed to prevent prior knowledge from suggesting a likely answer. Questions target specific relations stated in the source, such as which filing seal a petition class requires in a fictional court. Answer choices are the corresponding technical codes or entities, with one supported answer and three distractors constructed using the same criteria as in WCQA.

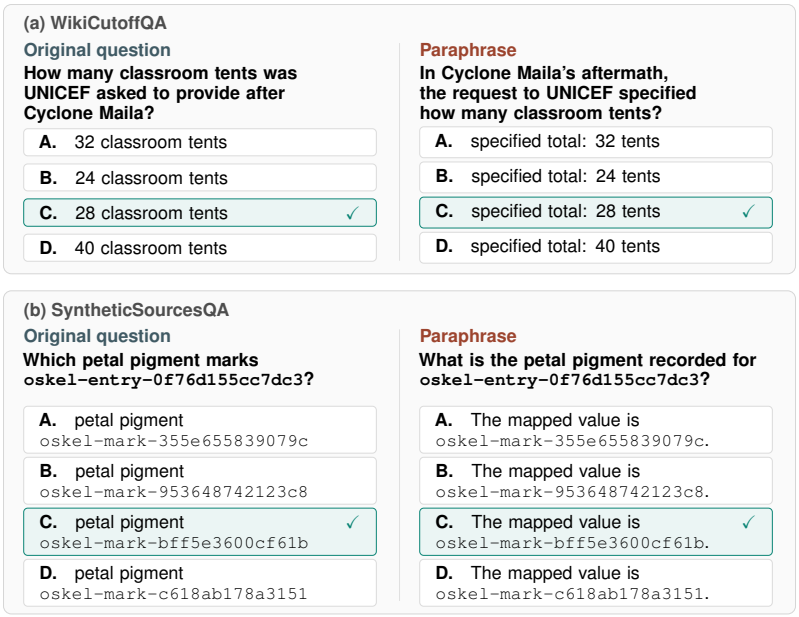

Figure 13: **Example questions and their paraphrases.** Each panel compares question and answer wording. SSQA choices use equivalent descriptive wording around the same identifiers. Correct answers are highlighted in green and marked with a checkmark.

Behavioral validation. We first use GPT 5.6 Sol to review each question for readability, clarity, and coherence. We screen questions without their sources using Qwen3.5-4B, the model used in our contamination experiments, and Llama Nemotron Super 1.5 (Bercovich et al., 2025a,b). We check for accuracy consistent with chance, indicating that the models' prior knowledge does not suffice for reliable answers. Super 1.5 is also tested with the source, using ten responses per question in each condition. We require at least 70% accuracy with the source to check that the question is answerable from the supplied information. Questions that fail screening are removed. We check for duplicates and label imbalance.

Training and detector paraphrases. For each question, we use GPT 5.6 Sol to generate five alternative wordings with different phrasing and sentence structure, while preserving the meaning, relevant constraints, and correct answer. When rewording answer choices, we preserve the meaning of every choice and apply the same treatment to correct answers and distractors. Each paraphrase undergoes an independent semantic review using GPT 5.6 Sol. We also repeat the original screening audit under each paraphrase, with and without the source, to check behaviorally that the question remains answerable from the same source with the same correct answer. We randomly reserve one paraphrase for training and hold out the remaining four for detection. This assignment remains fixed across all experiments. We check that no source, question, or paraphrase appears in both the exposure and control sets.

D CONTROLLED TRAINING

We fine-tune only on exposure-set items, without their source passages, using SFT with the four exposure forms from Section 2 (Table 9). For every form, we evaluate accuracy on the original item $(q_i, \mathbf{a}_i, a_i^\star)$, with its original question wording, answer wording, and answer order. SFT applies the loss only to the answer, excluding the question and choices $(q_i, \mathbf{a}_i)$. For CLM exposure, we use the Complete-Item form and also apply the loss to $(q_i, \mathbf{a}_i)$.

Table 9: **Items presented during fine-tuning.** SFT uses all four exposure forms. CLM uses only Complete-Item Exposure (first row).

Exposure form	Training item	Change from original
Complete-Item	$(q_i, \mathbf{a}_i, a_i^\star)$	None
Shuffled-Order	$(q_i, \mathbf{a}_i^\sigma, a_i^\star)$	Answer order
Question-Paraphrase	$(\tilde{q}_i, \mathbf{a}_i, a_i^\star)$	Question wording
Answer-Paraphrase	$(q_i, \tilde{\mathbf{a}}_i, \tilde{a}_i^\star)$	Answer wording

For Shuffled-Order Exposure, we independently sample one of the 24 possible answer orders for each item and epoch.

All fine-tuning starts from the March 2, 2026 revision of Qwen3.5-4B. We update all text parameters using AdamW, a constant learning rate of 5e-7, a batch size of 4, and gradient clipping at 1. We train for 20 epochs.

The ten selected models average over 95% accuracy on exposed items presented in the Complete-Item Exposure format. We also target less than one percentage point of accuracy degradation relative to the clean model, averaged equally across IFEval, MMLU-Pro, MATH-500, AIME 2025, IFBench, and HumanEval (Zhou et al., 2023, Lightman et al., 2024, OpenCompass, 2025, Pyatkin et al., 2025, Chen et al., 2021). The accuracy difference relative to the clean model is $+0.10 \pm 0.20$ percentage points (mean $\pm$ SEM across the ten selected models).

E PUBLISHED DETECTION METHODS AND CONTROLLED EVALUATION

We compare our detectors with 19 published algorithms and baselines on the exposure and control sets from Section 4.1. The following subsections describe the published method families and report the complete results. Appendix A describes our methods.

E.1 DETECTION METHOD FAMILIES

We group the published methods by the evidence they use to detect contamination. Table 10 summarizes their inputs and measurements.

Token probabilities. These methods measure how readily the model predicts the original MCQ text. We supply the question and its ordered answer choices, without the answer key, and obtain the probability of each token given the preceding text. For LOSS, we average the token log probabilities. Min-K% averages only those of the least-probable 20% of tokens. Min-K%++ subtracts the model's expected next-token log probability from each observed token's log probability and divides by the corresponding standard deviation. It then averages the lowest 20% of these values (Zhang et al., 2025). Gap-K subtracts the most likely token's log probability from the observed token's log probability and divides by the same standard deviation. It averages these signed gaps within overlapping three-token windows, then averages the lowest 20% of window scores (Kwak & Kim, 2026). LogProber measures familiarity with the question using an area-based summary of cumulative token log probabilities.

Comparing text predictability. The lowercase and zlib scores compare the model's loss with a second measure of text predictability. The lowercase score divides the loss on a lowercased MCQ by the loss on its original text. The zlib score divides the compressed byte length by the original mean token loss. In both cases, a larger ratio means that the original text is easier for the model relative to

the comparison (Carlini et al., 2021). Neighborhood comparison replaces one word at a time using BERT to create 100 alternative texts. Its score is their mean loss minus the original's loss, measuring whether the original is easier to predict than nearby wordings (Mattern et al., 2023).

Reconstruction from incomplete MCQs. These methods withhold part of an MCQ and test what the model can reproduce or answer from the remaining text. The *N-gram* method reconstructs answer choices in their original order. To generate a choice, the model sees the question and all preceding original choices. The method compares each generated choice with its original text using the word-overlap measure ROUGE-L. The published detector calls an item reproduced when at least one choice reaches 0.75 overlap. We use this binary decision as the item score and the fraction of flagged items as the group score (Hidayat et al., 2025). TS-Guessing hides one incorrect choice while retaining the question and the other choices. It asks the model to reproduce the missing choice and scores an exact text match (Deng et al., 2024).

Answer-order preference. These methods compare the probabilities of answer orders. Given the question, *Full permutation (maximum)*, Algorithm 1 of Ni et al., compares the original labeled-choice sequence with all 23 alternative orders. *Permutation-R* uses a fixed subset of 12 full orders. *Permutation-Q* uses only two choices at a time, comparing the original first two choices with the other 11 ordered pairs. For AUROC, we use the log probability of the original sequence or pair minus that of its most probable alternative. *Full permutation (outlier)*, Ni et al.'s Algorithm 2, fits an Isolation Forest to the 24 log probabilities and scores the most probable order. For AUROC, we negate its decision score and average these item scores within each group (Ni et al., 2025, Hidayat et al., 2025).

Original-wording recognition. DCQ asks the model which of several wordings is the original benchmark item. Each A–D quiz option contains a complete MCQ, including its answer choices. To measure preferences for quiz positions, DCQ first presents four paraphrases and a fifth option for "none of the above." It then replaces a paraphrase with the original MCQ at each position chosen less often than chance in the first stage. Recognition rates across these quizzes give its contamination estimate. Its published estimator aggregates recognition across quiz questions (Golchin & Surdeanu, 2025). For our evaluation, we apply the authors' released estimator of the contaminated fraction to each group of 16 questions. For AUROC, we use the estimator's lower endpoint.

DE-COP tests whether a model can recognize an original passage among three paraphrases. It presents the four versions in all 24 display orders and measures how often the original is selected. Its published detection evaluation aggregates recognition across multiple passages at the document level (Duarte et al., 2024). We apply this procedure to complete MCQs.

Effect of added context. CoDeC compares a target text's mean token log probability with and without other samples from the same dataset prepended as context. It averages the change over repeated context draws, then scores the dataset by the fraction of targets with a decrease. CoDeC is intended for dataset-level contamination detection (Zawalski et al., 2026). We evaluate CoDeC on exposed and held-out groups from the same benchmark, testing whether its dataset-level signal distinguishes within-benchmark exposure.

Changes in answer confidence. PaCoST asks the model to answer an original question and a paraphrase. It then asks whether each of its own answers is correct, using the probability of "Yes" as its confidence in that answer. We compute confidence on the original minus confidence on the paraphrase. For each group, PaCoST uses a one-sided paired t-test to ask whether this difference is positive on average. PaCoST is intended for benchmark-level contamination detection (Zhang et al., 2024). The group score is the negative one-sided p-value.

Completing benchmark text. Both TimeTravel algorithms give the model the beginning of a benchmark item and ask it to complete the missing text. A *guided* prompt names the benchmark and requests its original continuation. A *general* prompt requests a continuation without this benchmark information. TimeTravel's overlap algorithm compares each completion with the missing text using ROUGE-L or BLEURT (Sellam et al., 2020). Our item-level score subtracts the general completion's score from the guided completion's score, asking whether naming the benchmark improves reconstruction. Its group test uses bootstrap resampling to compare the mean scores (Golchin & Surdeanu, 2024). TimeTravel (ICL) instead asks a judge model whether the guided completion

exactly or nearly matches the missing text. The judge receives examples of these categories in its prompt, which is the in-context learning (ICL) step. A group is flagged as contaminated if the judge finds at least one exact match or at least two near matches (Golchin & Surdeanu, 2024).

Table 10: **Published detector families and their measurements.** Inputs refer to our MCQ evaluation. The text defines how these measurements become scores.

Methods	Input	Measurement
LOSS, Min-K%, Min-K%++, Gap-K, lowercase, zlib, LogProber	Question and ordered choices	Predictability of the original text
Neighborhood comparison	Question and choices, plus one-word variants	Original loss relative to variant losses
N-gram	Question and preceding original choices	Word overlap with the next hidden choice
TS-Guessing	MCQ with an incorrect choice hidden	Exact recovery of the hidden choice
Full permutation, Permutation-R, Permutation-Q	Question and alternative choice sequences	Original-order preference or maximum-likelihood outlier score
DCQ	Original MCQ and alternative wordings	Original-recognition rates
DE-COP	Original MCQ and three paraphrases	Original recognition across 24 display orders
PaCoST	Original and paraphrased MCQs	Change in confidence in its own answers
TimeTravel	Beginning of an MCQ, with or without its benchmark name	Overlap or judged match with the missing text
CoDeC	Labeled MCQ with and without a preceding labeled example	Change in the target's mean token log probability

Group scores. When a method assigns an *item-level score* $s_i = s(q_i, \mathbf{a}_i, a_i^\star)$ to each item but has no dedicated group statistic, we define the group score as the mean of these scores, $s(\mathcal{B}_G) = |G|^{-1} \sum_{i \in G} s_i$. Some methods instead define a group statistic directly, aggregating evidence across a dataset or benchmark before returning a score or test result (Oren et al., 2024, Zhang et al., 2024).

E.2 Controlled Evaluation Results

Tables 11 and 12 report AUROC for our methods and published methods. The full populations contain 512 WCQA and 2,048 SSQA questions. For each exposure form, half of each population is exposed and half is control. Score directions are fixed by the method definitions.

The *e-value aggregate* row in Table 1 ranks groups by $(E_R + E_W + E_C)/3$, the arithmetic mean of the DOPR, DWPC-QA, and CBA e-values (Appendix A.4). Standalone DOPR and DWPC-QA rows retain their mean-score rankings. The evaluated CBA shuffle test excludes invalid responses and their corresponding keys before computing its exact p-value.

Whole-set decisions. Table 1 also applies each test once to the complete exposure set and once to the complete control set: 256 questions per set for WCQA and 1,024 for SSQA. These are not aggregates of the 16-question groups' decisions. Standalone methods reject at unadjusted $p < 0.05$. DOPR and DWPC-QA retain the question-level independence assumptions in Appendix A.4, without adjustment for shared sources.

For each whole set, the aggregate averages the three e-values as above. We apply e-BH at $q = 0.05$ and Holm at $\alpha = 0.05$ within two separate families of ten hypotheses: one for exposure sets and one for control sets, each spanning the five settings in both testbeds. Holm uses $P = \min\{1, 3\min(p_R, p_W, p_C)\}$ from the component p-values, not the averaged e-value. Appendix H specifies both multiple-testing procedures.

Table 11: **Complete detector comparison (1/2).** AUROC. Our methods appear first. Published methods are ranked by mean AUROC across the ten settings. Table 12 continues the comparison. Groups contain 16 questions. Chance is 0.5.

Method	WCQA					SSQA				
	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM
DWPC-QA (Ours)	**1.000**	**1.000**	0.891	0.859	0.914	0.984	**1.000**	0.916	0.921	0.876
DOPR (Ours)	0.977	0.477	**0.961**	**0.965**	**1.000**	**1.000**	0.449	**1.000**	**1.000**	0.953
DWPC-Q (Ours)	0.715	0.836	0.574	0.645	0.766	0.699	0.756	0.479	0.599	0.577
CBA-shuffle (Ours)	0.691	0.451	0.502	0.547	0.551	0.572	0.596	0.587	0.573	0.539
CBA-uniform (Ours)	0.645	0.332	0.572	0.633	0.584	0.565	0.567	0.556	0.581	0.528
LOSS (Yeom et al., 2018)	0.652	0.594	0.625	0.602	**1.000**	0.565	0.558	0.587	0.543	**1.000**
Min-K% (Shi et al., 2024)	0.629	0.543	0.668	0.621	**1.000**	0.557	0.553	0.610	0.542	**1.000**
LogProber (Yax et al., 2025)	0.688	0.625	0.668	0.648	**1.000**	0.501	0.488	0.527	0.475	**1.000**
Neighborhood (Mattern et al., 2023)	0.680	0.617	0.672	0.621	**1.000**	0.451	0.503	0.523	0.450	**1.000**
Full permutation (outlier) (Ni et al., 2025)	0.523	0.531	0.613	0.621	0.953	0.516	0.585	0.512	0.504	**1.000**
Zlib ratio (Carlini et al., 2021)	0.516	0.457	0.508	0.477	**1.000**	0.557	0.571	0.591	0.545	**1.000**
Full permutation (maximum) (Ni et al., 2025)	0.492	0.527	0.523	0.465	**1.000**	0.417	0.486	0.495	0.470	**1.000**
Permutation-R (Hidayat et al., 2025)	0.453	0.500	0.508	0.461	**1.000**	0.418	0.495	0.525	0.451	**1.000**
Permutation-Q (Hidayat et al., 2025)	0.426	0.430	0.410	0.449	**1.000**	0.504	0.507	0.510	0.523	**1.000**
Min-K%++ (Zhang et al., 2025)	0.504	0.465	0.559	0.531	0.707	0.445	0.481	0.544	0.482	**1.000**
Gap-K (Kwak & Kim, 2026)	0.508	0.477	0.594	0.496	0.637	0.447	0.457	0.538	0.487	**1.000**

Table 12: **Complete detector comparison (2/2).** Continuation of Table 11, using the same ordering.

Method	WCQA					SSQA				
	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM	Complete item	Shuffled order	Question paraph.	Answer paraph.	CLM
CoDeC (Zawalski et al., 2026)	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.508	**1.000**
DCQ	0.596	0.426	0.553	0.428	0.580	0.681	0.533	0.577	0.570	0.539
PaCoST (Zhang et al., 2024)	0.535	0.516	0.379	0.512	0.621	0.528	0.619	0.473	0.575	0.557
TimeTravel (ICL)	0.500	0.531	0.500	0.500	0.656	0.500	0.500	0.500	0.500	0.500
N-gram (Hidayat et al., 2025)	0.369	0.480	0.438	0.402	0.941	0.492	0.485	0.485	0.500	0.493
DE-COP (Duarte et al., 2024)	0.549	0.486	0.537	0.559	0.490	0.439	0.432	0.531	0.500	0.510
TS-Guessing (Deng et al., 2024)	0.465	0.531	0.469	0.500	0.500	0.500	0.500	0.500	0.500	0.500
TimeTravel (ROUGE)	0.400	0.502	0.502	0.533	0.469	0.492	0.498	0.500	0.477	0.516
TimeTravel (BLEURT) (Golchin & Surdeanu, 2024)	0.318	0.402	0.453	0.439	0.312	0.523	0.462	0.502	0.505	0.547
Lowercase ratio (Carlini et al., 2021)	0.340	0.137	0.328	0.266	**1.000**	0.462	0.521	0.553	0.455	0.210

Interpretation across exposure objectives. LOSS, Min-K%, Min-K%++, Gap-K, lowercase, zlib, LogProber, and Neighborhood comparison measure the predictability of benchmark text, while N-gram, TS-Guessing, and TimeTravel test its reconstruction. In our SFT settings, none of the evaluated published baselines reaches 0.7 AUROC. Under CLM, several separate exposed and control groups perfectly, including LOSS and Min-K% on both testbeds. We believe this difference is due to SFT directly supervising only the answer response, while CLM also directly supervises the benchmark text these detectors probe. Yax et al. (2025) similarly find that answer-only fine-tuning can improve answer accuracy without producing LogProber's question-likelihood signal.

Empirical calibration check of our three detectors. To assess E-value calibration in the absence of exposure, we evaluate the detectors over the control set, whose questions were withheld from fine-tuning. Across the ten selected checkpoints on WCQA and SSQA, the mean E-values are 0.046 for DOPR, 0.081 for DWPC-QA, and 0.610 for CBA. The largest E-values are 0.082 for DOPR, 0.204 for DWPC-QA, and 0.728 for CBA. The three-detector aggregate has a mean of 0.246 and a maximum of 0.287. All observed E-values are below 1.0, consistent with calibration on these controls.

F ACCURACY-DEPENDENT CONTAMINATION SIGNALS

We call a contamination signal accuracy-dependent if differences in task competence can change it even among clean models. In particular, answering more questions correctly should not by itself make a clean model appear more contaminated. We exclude methods with this confound from our comparison. This matters especially in our testbeds, where questions require source knowledge and accuracy therefore predicts exposure by design (Section 4).

To give one illustrative example, "*Semi-half*" (Hidayat et al., 2025) retains the final seven words of a question and all answer choices, treating a correct answer as evidence of leakage. Yet the retained text can still be enough to solve the question. Figure 14 shows two real MMLU questions for which a clean model can obtain this positive signal through arithmetic alone. We note that in their controlled experiment, the authors first select questions the base model answers incorrectly, then train it on half of them before evaluating detection. The experiment therefore does not test the case where a clean model can already solve the full question and still answers correctly after truncation, triggering a contamination signal despite no prior exposure.

Original question

What is the value of p in 24 = 2p?

A: p = 4
B: p = 8
C: p = 12
D: p = 24

Final seven words

value of p in 24 = 2p?

A: p = 4
B: p = 8
C: p = 12
D: p = 24

Seven-word question, unchanged

Find the quotient of $-40 \div (-8)$.

A: 1 over 5 B: -5 C: -1 over 5 D: 5

Figure 14: **MMLU questions remain solvable after semi-half truncation.** Shading marks removed words. Both answers remain deducible: $p = 12$ (C) and $(-40)/(-8) = 5$ (D). Word counts follow spaces in the dataset text.

G REAL-BENCHMARK FILTRATION PROTOCOL

The filtration in Section 5 addresses structural artifacts that can mimic order, wording, or hidden-label signals. We annotate question properties once and apply the resulting exclusions separately for each detector. We will release the question-level filtration metadata for all four benchmarks upon acceptance, so future detectors on these benchmarks can select exclusions according to their own assumptions.

G.1 PROPERTIES AND DETECTOR-SPECIFIC EXCLUSIONS

Reconstructable mappings (M). The annotation model repeatedly recovers the full set of answer choices and their original assignment to labels from the question alone. When achieved without prior exposure, such recovery lets a model that solves the question return its correct label. Matching the recovered order can also favor the original choice order in permutation-based detectors. Recovering only the unordered choices does not qualify.

Annotation: Given only the question, the annotation model attempts to reconstruct every choice in its original order. It receives neither the benchmark identity nor the choices, their number, their labels, or the answer key. We mark M when it reliably recovers all choice texts in their original order, reproducing the benchmark's answer-to-label mapping.

Invalid permutations or ambiguous choice identities (V_{fail}). A permutation is invalid if moving a choice changes its meaning or what it refers to. For example, "Both A and C" changes meaning if the referenced choices move while that text stays fixed. Duplicate or equivalent choices also make answer identity ambiguous.

Annotation: The annotation model inspects the labeled question and choices for these problems. We mark V_{fail} unless the most frequent judgment confirms that permutations are valid and choice identities are unambiguous. This rules out arbitrary relabeling for the audit, without requiring that exactly one answer order be valid.

Natural order (N). The answer choices induce a small family of natural orders, such as ascending and descending numerical sequences. A clean model may prefer these orders over an arbitrary shuffle. If the benchmark's original order belongs to this family, that preference can resemble evidence of prior exposure in an order-preference detector, including the published permutation methods discussed below. For DOPR, only relative answer preferences that survive its debiasing can produce such a signal. Natural order does not trigger exclusion for DWPC-QA or CBA. The controlled ablation in Appendix G.4 tests whether natural-order preferences survive DOPR's debiasing, using fictional questions with randomly assigned answers. Figure 16 shows strong sequence-likelihood preferences for natural order while DOPR remains robust.

Annotation: The annotation model sees the question and shuffled choices with opaque identifiers. It specifies the complete family of orders allowed by its proposed rule. We mark N when repeated annotations identify a family containing at most one quarter of the $K_i!$ permutations. For four numerical choices, ascending and descending order give two of 24 permutations. The strict and loose exclusion rules below differ in whether they remove N questions.

Deducible correct answer and label ($\text{GT} \cap \text{L}$). Ground-truth answer deducibility (GT) means that the correct answer can be derived from the question without seeing it among the choices. For illustration, "What is $2 + 2$?" determines the answer 4, whereas "Which of the following numbers is largest?" requires seeing those numbers. Label deducibility (L) means that the question also reveals, or allows one to deduce, the benchmark label assigned to that answer.

Annotation: For GT, the annotation model receives the question and the correct answer text and must provide a checkable derivation without the choices. The label check receives the question, the correct answer text, and the valid labels, but no choices or original answer mapping. It must deduce a single label from the question. Exclusion requires both checks to pass and the derived label to match the answer key.

Strict and loose filtration. The main paper uses *strict DOPR filtration*, excluding M, V_{fail}, N, and $\text{GT} \cap \text{L}$. DOPR requires permutations to preserve answer meaning and identity, motivating the V_{fail} exclusion. Recoverable mappings (M), natural orders (N), and deducible correct labels ($\text{GT} \cap \text{L}$)

can favor the original arrangement without exposure if their effects survive DOPR's corrections. Our natural-order ablation supports DOPR's robustness to this confound: all eight group tests yield $p \geq 0.178$ despite strong natural-order likelihood preferences (Figure 16). We nevertheless default to strict filtration to exclude as many potential non-contamination signals as possible. This excludes all N questions, even when the stored order lies outside the identified natural family. *Loose DOPR filtration* excludes M, V_{fail}, and $\text{GT} \cap \text{L}$ (i.e., an N annotation alone does not exclude a question). Thus, natural-order questions are retained when they pass the other filters. Appendix H.1 reports this comparison.

For *DWPC-QA*, both versions exclude M and V_{fail}. Excluding V_{fail} is a precaution against changes in answer meaning or identity during permutation averaging. We conservatively exclude M because paraphrasing can weaken the match between the question's wording and the answer texts reconstructed from it. A preference that contributes equally to both wording conditions cancels in the contrast. Neither natural order nor a deducible label alone establishes an original-wording advantage, so N and $\text{GT} \cap \text{L}$ do not trigger exclusion.

For *CBA*, both versions exclude M and $\text{GT} \cap \text{L}$: either can let a model recover the correct label without prior exposure. CBA hides the choices and does not permute them, so N and V_{fail} alone do not reveal the hidden label and do not trigger exclusion. Table 13 summarizes the rules. All exclusions are determined without inspecting target-model detector scores.

Implications for published detectors. The same examples expose possible confounds for other methods from the contamination detection literature (see Appendix E.1). Full permutation assumes that swapping option contents preserves their meaning (Ni et al., 2025). The choice "Both A and C" in Figure 15(a) violates this assumption. The ordered menus in (b–c) can also favor the original sequence in Full permutation (maximum), Permutation-R, and Permutation-Q, whose scores compare original sequences or pairs with alternatives (Hidayat et al., 2025). When permutation reassigns texts to fixed labels, the embedded correct label in (d) can similarly favor sequences that preserve that correspondence. For reconstruction, the menu in (b) already supplies the missing choices sought by N-gram and TS-Guessing. N-gram flags an item when any one of its four choices is recovered, so copying only the correct choice in (d) can also suffice. Semi-half retains the final seven question words, which in (d) are exactly the correct answer text (Hidayat et al., 2025). TS-Guessing instead masks an incorrect choice (Deng et al., 2024), so the correct answer and label alone do not reveal its target. These connections identify routes to detection scores without exposure, rather than measured failure rates on these examples.

Table 13: **Which structural artifacts affect each detector?** "Exclude" removes the annotated property. "Retain" means that this property alone does not exclude the question.

Question property	DOPR	DWPC-QA	CBA
Reconstructable mapping (M)	Exclude	Exclude	Exclude
Invalid permutations (V_{fail})	Exclude	Exclude	Retain
Natural order (N)	Strict only	Retain	Retain
Deducible correct label ($\text{GT} \cap \text{L}$)	Exclude	Retain	Exclude

G.2 Annotation Workflow and Coverage

We use GPT 5.6 Sol for the checks in Appendix G.1, without providing target-model detector outputs. Each check begins with three model judgments. If these suggest that an artifact is present or do not reach a clear conclusion, we obtain two more. To check for missed artifacts, we also audit selected M, N, and V_{fail} cases that all three initial judgments left unflagged, using five new judgments. An M or N annotation requires at least three of five judgments supporting the same reconstruction or order family, subject to the criteria above. For V_{fail}, GT, and L, we use the most frequent conclusion, with no separate minimum vote count. For GT, we also test a matched incorrect answer without revealing which answer is correct. We withhold GT if both are judged deducible, guarding against unsupported justifications.

Natural order (N) occurs in 10,407 of the 26,783 annotated questions. Table 14 gives the counts for all properties by benchmark.

Table 14: **Question-property annotations by benchmark.** Counts cover the annotation roster before response eligibility. Categories overlap. HLE uses text-only MCQs.

Benchmark	Questions	M	V_{fail}	N	GT ∩ L
MMLU	14,042	8	768	3,842	1
MMLU-Pro	12,032	0	1,318	6,277	0
GPQA Diamond	196	0	3	99	0
HLE	513	4	99	189	0
Total	26,783	12	2,188	10,407	1

G.3 Examples of the Four Mechanisms

Figure 15 illustrates the four properties using the examples in Figure 4 and one additional question. Example (b) lists all four answers in their original choice order, making the mapping (M) reconstructable. In example (d), a parsing error inserted the correct answer and label (GT ∩ L) into the question (Gema et al., 2025).

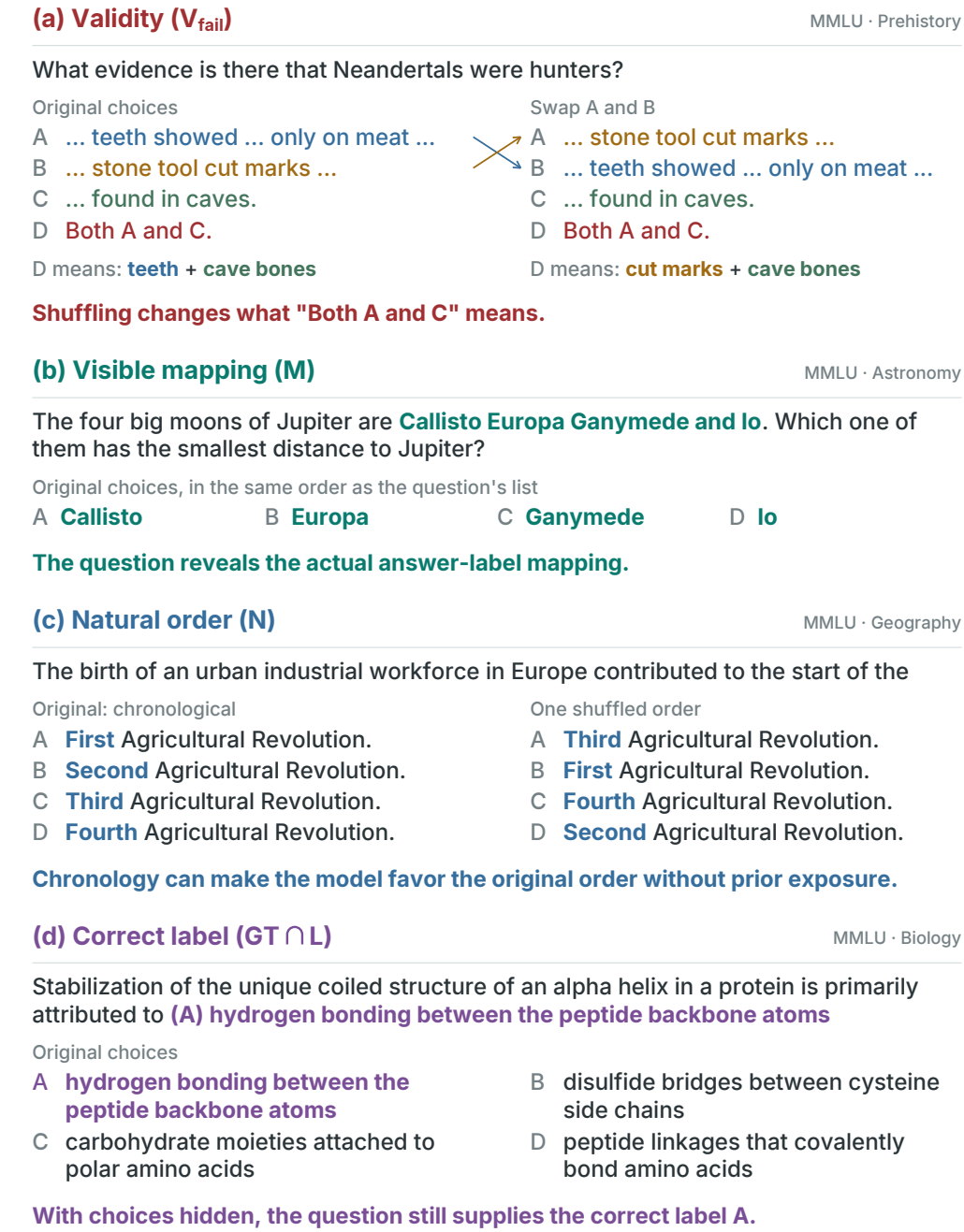

Figure 15: **How question structure can mimic contamination.** These examples illustrate possible mechanisms, not measured model responses. **(a) Validity (V_{fail}).** Swapping A and B changes what D means, violating meaning preservation in permutation-based detectors such as DOPR and Full permutation. **(b) Visible mapping (M).** Assigning A–D to the listed moons recovers their benchmark labels for CBA. The list can also favor the original order in permutation-based detectors and supply N-gram and TS-Guessing targets. **(c) Natural order (N).** Chronology can make the model favor the original sequence in DOPR, Full permutation, and Permutation-R/Q without exposure. **(d) Correct label ($\text{GT} \cap \text{L}$).** The question supplies CBA's label and a choice that N-gram can recover, without revealing the other choices.

G.4 Natural-Order Preference on Fictional Questions

Natural order refers to a small family of orderings suggested by the answer choices, such as numerical or chronological sequences (see Appendix G.1 for the full definition).

Experimental setup. We construct ten questions about fictional subjects, with numerical, chronological, or ordinal choices (Table 16). We assign each question a correct answer independently and uniformly at random. For example, we ask how many bells were installed at the observatory in the fictional town of Serevin, offer 2, 4, 6, and 8 bells, and assign 6 bells as the answer. The question provides no way to infer this assignment. We compare the ascending or chronological order with a fixed arbitrary shuffle of the same choices, keeping the assigned answer unchanged. To check that these questions are unanswerable, we ask Qwen3.5-4B each question ten times. Its overall accuracy against the randomly assigned answers is 16% (16 correct answers out of 100 attempts).

Likelihood comparison. We score all 24 orders in chat format with four models: Qwen3.5-4B, Nemotron-3-Super-120B, Phi-4, and Muse-Glimmer-30B. We use perplexity (PPL) to measure how confidently the model predicts the text, with lower values indicating greater confidence. For each question, we compare PPL for the same answer choices in their natural numerical or chronological order and in a fixed arbitrary shuffle. Figure 16(a) shows arbitrary-order PPL divided by natural-order PPL, $\frac{\mathrm{PPL(arbitrary\ order)}}{\mathrm{PPL(natural\ order)}}$, so values above one favor natural order. Natural order wins on 8/10 questions for Qwen and all ten for the other three models. It ranks first among all 24 orders on 5/10, 10/10, 10/10, and 10/10 questions, respectively. Thus, high likelihood can reflect a natural ordering of the options, such as numerical or chronological order, even when the correct answer is arbitrary.

Published permutation detectors flag natural order. We evaluate likelihood-based contamination detectors on the same questions: Full permutation (maximum) from Ni et al. (2025), and Permutation-R and Permutation-Q from Hidayat et al. (2025). Each flags a question when the option sequence in its designated original order has the highest likelihood among the alternatives it tests. We evaluate each question twice, once declaring the natural order to be the original order and once declaring the fixed arbitrary order to be the original order. Full permutation compares all 24 orders, Permutation-R uses a fixed subset of 12, and Permutation-Q compares the first two options with all 12 ordered pairs.

Using the authors' plain-text prompts and decision rules, Full permutation and Permutation-R each flag nine to ten questions per model when the declared original order is natural. Neither flags any when it is arbitrary. Permutation-Q flags between seven and nine questions per model when natural order is declared original, and none when arbitrary order is declared original (Table 15). These results demonstrate false positives on clean questions: the natural order of the options can trigger a contamination flag without training exposure to the questions. This illustrates the limitation for ordinal options already noted by Hidayat et al. (2025).

Notes: For Permutation-Q, changing the original order also changes which two options form the reference pair, so this comparison does not isolate ordering alone. The outlier variant of Full permutation (Ni et al., 2025) flags a question if any order has exceptionally high likelihood. In these experiments, across all models, every detected outlier is the natural order, so both original-order designations yield identical flags simply because the natural order is among the permutations.

DOPR does not flag this preference as contamination. DOPR tests whether the assigned correct answer receives an extra advantage in the designated original order. Its positive and negative scores cancel on average over the possible random answer assignments. Its expected score is therefore zero, even when the model favors the natural sequence. DOPR's p-values across the models (natural / arbitrary order designated as original) are $0.888/0.757$ for Qwen, $0.397/0.926$ for Super, $0.178/0.468$ for Phi-4, and $0.759/0.928$ for Muse. On the observed ten-question group, DOPR does not reject its null at the 5% level for any of the four models, with either choice of original order (Figure 16(b)). This supports DOPR's robustness to natural-order preference in these examples.

Table 15: **Natural-order false positives on fictional questions.** Each pair counts questions flagged out of ten with **natural / arbitrary order designated original**, using the authors' prompts and decision rules. Outlier reports a single count, identical for either original-order designation, using the strictest published threshold of -0.20.

Model	Full permutation Ni et al. (2025)		Permutation-R Hidayat et al. (2025)	Permutation-Q Hidayat et al. (2025)
	Maximum	Outlier		
Qwen3.5-4B	10 / 0	8	9 / 0	7 / 0
Nemotron-3-Super-120B	10 / 0	10	10 / 0	9 / 0
Phi-4	9 / 0	9	10 / 0	9 / 0
Muse-Glimmer-30B	10 / 0	10	10 / 0	9 / 0

Table 16: **Constructed questions and randomly assigned answers.** Choices are listed in natural A–D order. The assigned correct choice stays the same when the options are reordered.

Question	Natural choices	Randomly assigned answer
How many bells were installed at the observatory in the fictional town of Serevin?	2, 4, 6, 8 bells	C
At what temperature did the fictional Orvella Institute store its violet crystal?	10, 20, 30, 40 degrees Celsius	B
How far from the gate was the marker in the fictional Kelvar survey?	100, 200, 300, 400 meters	C
How long did the fictional Talune resonance trial last?	5, 10, 15, 20 minutes	D
What share of the sample received a silver seal in the fictional Vespari inspection?	25, 50, 75, 100 percent	B
In which month did the fictional Aldevin archive open its glass annex?	January, February, March, April	B
On which day did the fictional Morvessa council approve the amber charter?	Monday, Tuesday, Wednesday, Thursday	B
At what time did the fictional Istravel expedition depart from its harbor?	06:00, 09:00, 12:00, 15:00	B
In which position did the fictional Neruvin team finish the lantern contest?	First, Second, Third, Fourth	D
On which floor did the fictional Pellorin museum display its azure astrolabe?	First, Second, Third, Fourth floor	D

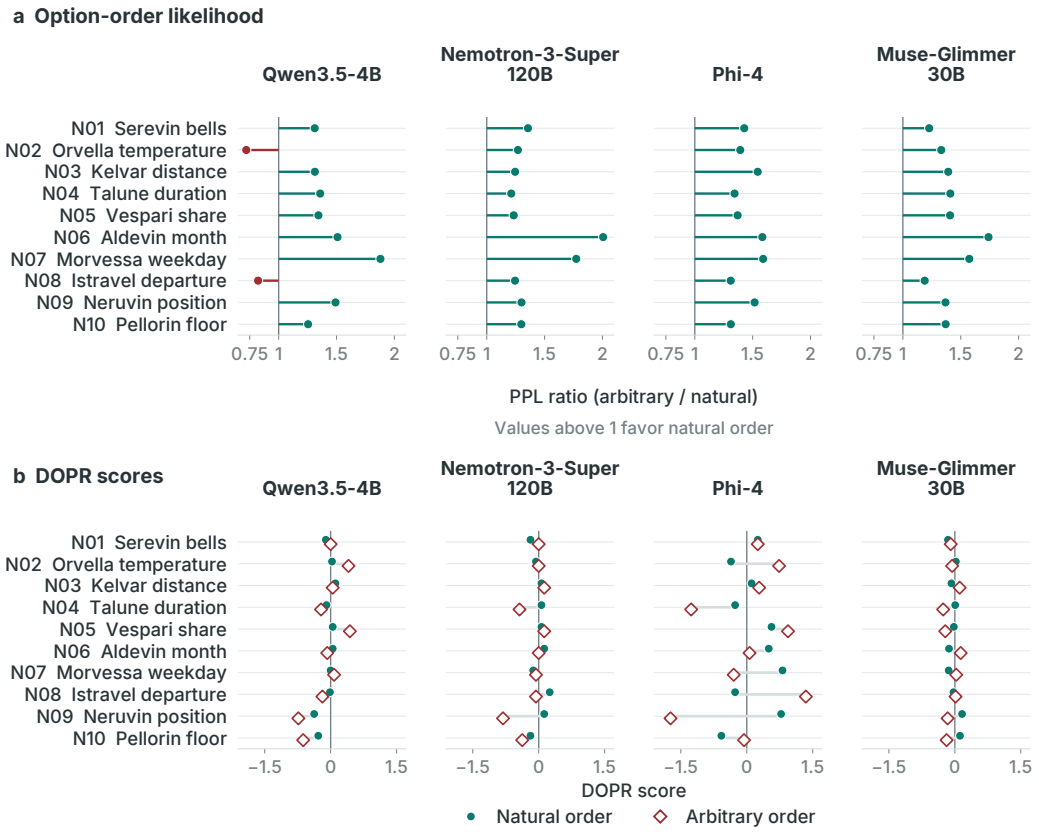

Figure 16: **Natural-order preference without a DOPR rejection.** All ten fictional questions are shown for each model. (a) PPL in arbitrary order divided by PPL in natural order, with values above one favoring natural order. (b) DOPR residual scores with the natural order (filled circles) or arbitrary order (open diamonds) designated as the original. Lines connect residual scores for the same question. Each DOPR test uses all ten questions, and none rejects at the 5% level.

H Complete Real-Benchmark Results

These strict-filtration tables expand Figure 1 and Section 5. We report MMLU separately from MMLU-Pro, GPQA Diamond (GPQA-D), and HLE, with three detectors for 28 open models and CBA for all 35 models. Appendix H.1 reports the loose-filtration results.

Model sources. We evaluate Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, Qwen3-8B, Qwen3-14B, Qwen3-30B-A3B, and Qwen3-235B-A22B (Qwen Team, 2025a), and Qwen3-32B (Qwen Team, 2025b). We also evaluate Qwen3.6-35B-A3B (Qwen Team, 2026c), Qwen3.8-27B (Qwen Team, 2026d), GLM-5.2 (Z.ai, 2026a), and GLM-5.3 (Z.ai, 2026b). The MiniMax models are MiniMax-M2.1 (MiniMax, 2025), MiniMax-M2.5 (MiniMax, 2026a), and MiniMax-M3 (MiniMax, 2026b). The Moonshot models are Kimi-K2.6 (Moonshot AI, 2026a) and Kimi-K3. The roster also includes Gemma-3-27B-IT (Google DeepMind, 2025), Muse-Glimmer-30B (Meta Superintelligence Lab, 2026), gpt-oss-puzzle-88B (NVIDIA, 2026a), and Mistral-Small-3.1-24B (Mistral AI, 2025). Other open models are DeepSeek-V4-Pro and DeepSeek-V4-Flash (DeepSeek-AI, 2026b), Nemotron-3-Super and Nemotron-Labs-3-Puzzle (NVIDIA, 2026b,c), Phi-4 (Microsoft Research, 2024), and gpt-oss-20B and gpt-oss-120B (OpenAI, 2025). The remaining closed models are Claude Opus 4.8 and Claude Opus 5 (Anthropic, 2026a,b), GPT-5.6 Luna and Terra, Gemini-3.5 Flash (Google DeepMind, 2026), and Grok-4.6 (xAI, 2026).

Tests and aggregation. We use strict filtration (Appendix G). DOPR and DWPC-QA use their one-sided mean tests (Appendix A). Appendix A.4 defines the e-values. For benchmark b, we average the three detector values as $E_{3,b} = \frac{1}{3}(E_{R,b} + E_{W,b} + E_{C,b})$, where R and W denote DOPR and DWPC-QA. The non-MMLU aggregate is $E_{\text{nonMMLU}} = \frac{1}{3}\sum_{b\in\mathcal{B}} E_{3,b}$, with $\mathcal{B} = \{\text{MMLU-Pro}, \text{GPQA-D}, \text{HLE}\}$, so each of its nine components has weight $1/9$. All 28 open models have complete panels. CBA alone uses its MMLU value or the mean of its three non-MMLU values. These means remain valid e-values under the global null even with dependent components (Vovk & Wang, 2021). Table 19 reports MMLU CBA's single p-value.

Decisions and display. We analyze four testing families separately: MMLU and the three non-MMLU benchmarks, each with three detectors (28 models) or CBA alone (35 models). The component counts are $m = 3$ and $m = 9$ for the three-detector families, and $m = 1$ and $m = 3$ for CBA. Each table's full set of J models forms one testing family. We order the calibrated values $E_{(1)} \geq \cdots \geq E_{(J)}$ and apply e-BH at $q = 0.05$, rejecting the first k for the largest k satisfying $E_{(k)} \geq J/(qk)$ (Wang & Ramdas, 2022). For MMLU CBA, e-BH uses the single calibrated value $e(p)$.

For Holm, each model's original component p-values give $P = \min\{1, m \min_j p_j\}$, a valid p-value for its global null by the union bound. For $m = 1$, this is the original CBA p-value. We sort these model p-values as $P_{(1)} \leq \cdots \leq P_{(J)}$ and reject successively while $P_{(r)} \leq 0.05/(J-r+1)$, stopping at the first failure (Holm, 1979). Both procedures include all models in their family. Within each family, Holm controls the probability of any false model-level rejection, whereas e-BH controls the expected fraction of false rejections. These guarantees are separate across families.

Marks on aggregate e-values (or the single MMLU CBA p-value) indicate e-BH (•), Holm (⋄), both (•⋄), or neither (no mark). Holm's marks come from component p-values, not from the displayed e-values. Smaller p-values and larger e-values receive stronger red shading. Loose-filtration decisions are recomputed within the corresponding families.

Main patterns. With three detectors, e-BH rejects 26 of 28 models on MMLU and nine outside MMLU. Holm rejects 28 and ten, respectively, adding MiniMax-M2.5 outside MMLU because of its localized HLE signal. For CBA, MMLU yields 28 of 35 e-BH rejections and 30 Holm rejections. Neither procedure rejects any model in the non-MMLU CBA family. MiniMax-M3's non-MMLU global null is rejected by e-BH.

Table 17: **MMLU: three detectors across open models.** Sorted by decreasing aggregate E, the mean of the three detector e-values. •: e-BH ($q = 0.05$). ◇: Holm on component p-values ($\alpha = 0.05$).

Model	DOPR p	DWPC-QA p	CBA p	Aggregate E
MiniMax-M2.1	1.00e−6	2.30e−1337	6.33e−6	9.40e1331 •◇
Qwen3.6-35B-A3B	1.45e−10	1.91e−643	7.22e−19	2.00e638 •◇
Qwen3-8B	2.18e−13	2.07e−376	7.55e−5	2.64e371 •◇
GLM-5.2	4.59e−8	1.14e−317	1.47e−20	5.33e312 •◇
MiniMax-M2.5	5.88e−9	1.27e−264	2.79e−10	5.34e259 •◇
Muse-Glimmer-30B	9.45e−3	4.76e−243	1.20e−14	1.51e238 •◇
Qwen3-1.7B	0.0497	1.81e−217	0.512	4.22e212 •◇
Qwen3-30B-A3B	3.48e−11	1.94e−174	1.75e−6	4.49e169 •◇
Kimi-K2.6	3.81e−10	1.97e−174	2.69e−68	4.41e169 •◇
gpt-oss-20B	1.62e−15	9.95e−119	2.74e−15	1.09e114 •◇
Qwen3-14B	1.97e−14	1.74e−117	3.98e−6	6.25e112 •◇
Qwen3-4B	1.89e−4	8.28e−115	0.210	1.33e110 •◇
Kimi-K3	1.17e−3	2.54e−87	1.41e−63	5.05e82 •◇
MiniMax-M3	6.07e−18	4.15e−66	6.06e−9	3.59e61 •◇
DeepSeek-V4-Flash	6.84e−11	3.93e−44	3.92e−4	4.73e39 •◇
gpt-oss-puzzle-88B	1.97e−4	2.80e−8	2.24e−42	1.62e37 •◇
GLM-5.3	4.22e−4	1.00	6.23e−41	6.24e35 •◇
gpt-oss-120B	1.05e−4	1.70e−11	2.38e−37	1.97e32 •◇
Qwen3.8-27B	2.00e−11	1.00	1.21e−20	1.31e16 •◇
Qwen3-235B-A22B	1.07e−18	1.00	3.65e−11	3.19e14 •◇
Qwen3-32B	2.14e−4	1.97e−13	9.44e−9	1.80e9 •◇
Nemotron-3-Super	1.42e−8	1.00	4.98e−12	9.88e7 •◇
Gemma-3-27B-IT	4.77e−11	1.00	2.78e−3	9.60e6 •◇
DeepSeek-V4-Pro	4.40e−10	1.00	0.630	1.10e6 •◇
Nemotron-Labs-3-Puzzle	1.35e−8	1.00	2.77e−9	3.50e5 •◇
Phi-4	1.21e−5	0.993	1.19e−3	62.7 •◇
Mistral-Small-3.1-24B	2.56e−3	0.992	0.0269	1.21 ◇
Qwen3-0.6B	9.01e−3	0.970	0.365	0.365 ◇

Table 18: **Other benchmarks: three detectors across open models.** Sorted by aggregate E, the mean of nine components. Benchmark E averages three detectors. •: e-BH ($q = 0.05$). ⋄: Holm on component p-values ($\alpha = 0.05$).

Model	Benchmark	DOPR p	DWPC-QA p	CBA p	Benchmark E	Aggregate E
Qwen3.6-35B-A3B	MMLU-Pro	0.749	6.62e−172	0.528	1.48e167	
	GPQA-D	0.551	8.08e−3	0.109	1.37	4.92e166 •⋄
	HLE	0.925	0.698	0.263	0.297	
MiniMax-M2.1	MMLU-Pro	0.0266	2.37e−105	0.145	5.47e100	
	GPQA-D	0.326	1.43e−3	0.0746	4.87	1.82e100 •⋄
	HLE	0.225	0.523	0.225	0.358	
Qwen3-1.7B	MMLU-Pro	0.948	1.84e−46	0.396	1.11e42	
	GPQA-D	0.354	0.129	0.849	0.316	3.69e41 •⋄
	HLE	0.563	0.998	0.907	0.197	
Qwen3-8B	MMLU-Pro	0.156	2.15e−29	0.798	1.21e25	
	GPQA-D	0.0842	0.183	0.553	0.444	4.05e24 •⋄
	HLE	0.734	0.280	0.883	0.220	
Qwen3-30B-A3B	MMLU-Pro	0.224	9.25e−27	0.329	2.97e22	
	GPQA-D	0.0309	0.672	0.206	0.724	9.91e21 •⋄
	HLE	0.757	0.761	0.471	0.244	
GLM-5.2	MMLU-Pro	0.727	7.68e−15	0.404	4.96e10	
	GPQA-D	0.985	0.966	0.516	0.232	1.65e10 •⋄
	HLE	0.755	0.198	0.450	0.279	
Qwen3-14B	MMLU-Pro	0.371	8.58e−10	0.0510	5.62e5	
	GPQA-D	0.0169	0.442	0.608	0.897	1.87e5 •⋄
	HLE	0.124	0.354	0.363	0.344	
GLM-5.3	MMLU-Pro	2.17e−9	1.00	0.883	3.40e5	
	GPQA-D	0.858	0.337	0.0806	0.525	1.13e5 •⋄
	HLE	0.283	0.729	0.422	0.278	
MiniMax-M3	MMLU-Pro	0.931	1.00	0.0622	0.534	
	GPQA-D	0.329	0.325	0.228	0.386	62.8 •⋄
	HLE	1.97e−4	1.99e−5	0.0681	188	
MiniMax-M2.5	MMLU-Pro	0.998	1.00	0.608	0.200	
	GPQA-D	0.440	0.927	0.361	0.295	5.66 ⋄
	HLE	0.957	2.35e−4	0.381	16.5	
Kimi-K2.6	MMLU-Pro	0.944	1.00	0.164	0.339	
	GPQA-D	0.766	6.70e−3	0.446	1.34	0.858
	HLE	0.996	0.966	0.0240	0.898	
Qwen3-32B	MMLU-Pro	0.538	1.00	9.28e−3	1.56	
	GPQA-D	0.128	0.582	0.447	0.368	0.712
	HLE	0.923	0.728	0.716	0.209	
Kimi-K3	MMLU-Pro	0.977	1.00	0.687	0.192	
	GPQA-D	0.560	0.999	0.473	0.257	0.440
	HLE	0.944	0.552	0.0261	0.871	
Qwen3.8-27B	MMLU-Pro	0.975	1.00	0.927	0.173	
	GPQA-D	0.881	0.997	0.311	0.281	0.438
	HLE	0.649	0.691	0.0269	0.861	
gpt-oss-20B	MMLU-Pro	0.352	1.00	0.0699	0.509	
	GPQA-D	0.735	0.212	0.676	0.273	0.394
	HLE	0.121	0.378	0.221	0.398	
gpt-oss-puzzle-88B	MMLU-Pro	0.927	1.00	0.115	0.396	
	GPQA-D	0.0701	0.928	0.856	0.389	0.376
	HLE	0.779	0.487	0.196	0.343	
Qwen3-0.6B	MMLU-Pro	0.372	0.207	0.576	0.218	
	GPQA-D	0.0313	0.429	0.533	0.635	0.357
	HLE	0.289	0.963	0.887	0.218	
Nemotron-3-Super	MMLU-Pro	0.358	1.00	0.769	0.190	
	GPQA-D	0.507	0.949	0.0798	0.517	0.347
	HLE	0.989	0.753	0.191	0.333	

Continued on next page

Table 18: **Other benchmarks: three detectors across open models.** (continued).

Model	Benchmark	DOPR p	DWPC-QA p	CBA p	Benchmark E	Aggregate E
Gemma-3-27B-IT	MMLU-Pro	0.922	1.00	0.618	0.200	
	GPQA-D	0.0558	0.393	0.523	0.487	0.341
	HLE	0.202	0.0825	0.626	0.337	
Phi-4	MMLU-Pro	0.621	1.00	0.179	0.328	
	GPQA-D	0.530	0.726	0.130	0.425	0.335
	HLE	0.753	0.108	0.988	0.253	
Mistral-Small-3.1-24B	MMLU-Pro	0.362	1.00	0.0671	0.519	
	GPQA-D	0.845	0.999	0.462	0.246	0.328
	HLE	0.807	0.308	0.821	0.220	
Muse-Glimmer-30B	MMLU-Pro	0.800	1.00	0.178	0.327	
	GPQA-D	0.829	0.365	0.170	0.387	0.314
	HLE	0.379	0.935	0.668	0.229	
Qwen3-4B	MMLU-Pro	0.226	0.914	0.382	0.249	
	GPQA-D	0.872	0.485	0.604	0.245	0.304
	HLE	0.766	0.634	0.119	0.417	
Nemotron-Labs-3-Puzzle	MMLU-Pro	0.237	1.00	0.258	0.287	
	GPQA-D	0.756	0.927	0.271	0.306	0.301
	HLE	0.899	0.323	0.267	0.311	
Qwen3-235B-A22B	MMLU-Pro	0.450	1.00	0.152	0.353	
	GPQA-D	0.584	0.370	0.277	0.336	0.301
	HLE	0.755	0.708	0.708	0.215	
DeepSeek-V4-Pro	MMLU-Pro	0.745	1.00	0.285	0.269	
	GPQA-D	0.653	0.919	0.125	0.419	0.298
	HLE	0.992	0.390	0.855	0.206	
gpt-oss-120B	MMLU-Pro	0.986	1.00	0.829	0.180	
	GPQA-D	0.566	0.957	0.634	0.238	0.257
	HLE	0.624	0.352	0.199	0.352	
DeepSeek-V4-Flash	MMLU-Pro	0.810	1.00	0.835	0.180	
	GPQA-D	0.0971	0.995	0.863	0.337	0.251
	HLE	0.995	0.665	0.485	0.235	

Table 19: **MMLU: CBA across all models.** Sorted by increasing p, with e-BH applied after calibration. •: e-BH ($q = 0.05$). ⋄: Holm on component p-values ($\alpha = 0.05$). †: closed models.

Model	CBA p
Claude Opus 5†	9.19e−173 •⋄
Kimi-K2.6	2.69e−68 •⋄
GPT-5.6 Sol†	2.22e−66 •⋄
Kimi-K3	1.41e−63 •⋄
Grok-4.6†	5.88e−53 •⋄
gpt-oss-puzzle-88B	2.24e−42 •⋄
GLM-5.3	6.23e−41 •⋄
gpt-oss-120B	2.38e−37 •⋄
Gemini-3.5 Flash†	1.88e−29 •⋄
GPT-5.6 Terra†	1.32e−25 •⋄
Qwen3.8-27B	1.21e−20 •⋄
GLM-5.2	1.47e−20 •⋄
Qwen3.6-35B-A3B	7.22e−19 •⋄
GPT-5.6 Luna†	8.25e−19 •⋄
Claude Opus 4.8†	1.22e−15 •⋄
gpt-oss-20B	2.74e−15 •⋄
Muse-Glimmer-30B	1.20e−14 •⋄
Nemotron-3-Super	4.98e−12 •⋄
Qwen3-235B-A22B	3.65e−11 •⋄
MiniMax-M2.5	2.79e−10 •⋄
Nemotron-Labs-3-Puzzle	2.77e−9 •⋄
MiniMax-M3	6.06e−9 •⋄
Qwen3-32B	9.44e−9 •⋄
Qwen3-30B-A3B	1.75e−6 •⋄
Qwen3-14B	3.98e−6 •⋄
MiniMax-M2.1	6.33e−6 •⋄
Qwen3-8B	7.55e−5 •⋄
DeepSeek-V4-Flash	3.92e−4 •⋄
Phi-4	1.19e−3 ⋄
Gemma-3-27B-IT	2.78e−3 ⋄
Mistral-Small-3.1-24B	0.0269
Qwen3-4B	0.210
Qwen3-0.6B	0.365
Qwen3-1.7B	0.512
DeepSeek-V4-Pro	0.630

Table 20: **Other benchmarks: CBA across all models.** Sorted by decreasing aggregate E, the mean over three benchmarks. •: e-BH ($q = 0.05$). ⋄: Holm on component p-values ($\alpha = 0.05$). †: closed models.

Model	MMLU-Pro p	GPQA-D p	HLE p	Aggregate E
Claude Opus 5†	6.03e−3	0.475	0.209	2.56
GPT-5.6 Luna†	0.0318	0.0144	0.412	2.18
Qwen3-32B	9.28e−3	0.447	0.716	1.96
GPT-5.6 Sol†	0.0234	0.173	0.0617	1.76
Claude Opus 4.8†	0.831	0.167	0.0146	1.69
Kimi-K2.6	0.164	0.446	0.0240	1.44
MiniMax-M3	0.0622	0.228	0.0681	1.33
Grok-4.6†	0.0755	0.152	0.0890	1.27
Qwen3.8-27B	0.927	0.311	0.0269	1.26
Kimi-K3	0.687	0.473	0.0261	1.25
MiniMax-M2.1	0.145	0.0746	0.225	1.13
Qwen3-14B	0.0510	0.608	0.363	1.03
gpt-oss-20B	0.0699	0.676	0.221	0.987
Nemotron-3-Super	0.769	0.0798	0.191	0.966
Gemini-3.5 Flash†	0.0701	0.881	0.252	0.953
Mistral-Small-3.1-24B	0.0671	0.462	0.821	0.909
Qwen3.6-35B-A3B	0.528	0.109	0.263	0.887
gpt-oss-puzzle-88B	0.115	0.856	0.196	0.880
GLM-5.3	0.883	0.0806	0.422	0.868
Phi-4	0.179	0.130	0.988	0.862
GPT-5.6 Terra†	0.471	0.101	0.468	0.855
Muse-Glimmer-30B	0.178	0.170	0.668	0.845
Qwen3-4B	0.382	0.604	0.119	0.823
DeepSeek-V4-Pro	0.285	0.125	0.855	0.820
Nemotron-Labs-3-Puzzle	0.258	0.271	0.267	0.819
Qwen3-235B-A22B	0.152	0.277	0.708	0.802
Qwen3-30B-A3B	0.329	0.206	0.471	0.772
gpt-oss-120B	0.829	0.634	0.199	0.681
MiniMax-M2.5	0.608	0.361	0.381	0.676
GLM-5.2	0.404	0.516	0.450	0.663
Gemma-3-27B-IT	0.618	0.523	0.626	0.602
Qwen3-0.6B	0.576	0.533	0.887	0.583
Qwen3-1.7B	0.396	0.849	0.907	0.581
DeepSeek-V4-Flash	0.835	0.863	0.485	0.568
Qwen3-8B	0.798	0.553	0.883	0.559

H.1 LOOSE DOPR FILTRATION

Loose filtration disables only DOPR's natural-order exclusion (N), restoring questions that pass its other filters. Natural order refers to choices with an underlying numerical, chronological, or ordinal order. In the ablation on ten fictional questions (Figure 16), DOPR yields no rejections for any of the four models at the 5% level despite natural-order likelihood preferences, supporting robustness in these examples. We nevertheless exclude these questions in the strict analysis as a conservative precaution against possible non-contamination signals. Tables 21 and 22 report loose results for the 28 open models, with MMLU separated from the other benchmarks. DWPC-QA and CBA are unchanged, so only the three-detector tables are repeated.

Table 21: **MMLU: loose filtration across open models.** Sorted by decreasing aggregate E, the mean of the three detector e-values. •: e-BH ($q = 0.05$). ⋄: Holm on component p-values ($\alpha = 0.05$).

Model	DOPR p	DWPC-QA p	CBA p	Aggregate E
MiniMax-M2.1	2.15e−7	2.30e−1337	6.33e−6	9.40e1331 •⋄
Qwen3.6-35B-A3B	5.13e−13	1.91e−643	7.22e−19	2.00e638 •⋄
Qwen3-8B	3.93e−15	2.07e−376	7.55e−5	2.64e371 •⋄
GLM-5.2	1.14e−8	1.14e−317	1.47e−20	5.33e312 •⋄
MiniMax-M2.5	3.43e−7	1.27e−264	2.79e−10	5.34e259 •⋄
Muse-Glimmer-30B	0.0105	4.76e−243	1.20e−14	1.51e238 •⋄
Qwen3-1.7B	0.0237	1.81e−217	0.512	4.22e212 •⋄
Qwen3-30B-A3B	2.95e−13	1.94e−174	1.75e−6	4.49e169 •⋄
Kimi-K2.6	5.59e−8	1.97e−174	2.69e−68	4.41e169 •⋄
gpt-oss-20B	1.69e−17	9.95e−119	2.74e−15	1.09e114 •⋄
Qwen3-14B	1.07e−15	1.74e−117	3.98e−6	6.25e112 •⋄
Qwen3-4B	6.87e−5	8.28e−115	0.210	1.33e110 •⋄
Kimi-K3	2.44e−3	2.54e−87	1.41e−63	5.05e82 •⋄
MiniMax-M3	1.34e−22	4.15e−66	6.06e−9	3.59e61 •⋄
DeepSeek-V4-Flash	1.01e−9	3.93e−44	3.92e−4	4.73e39 •⋄
gpt-oss-puzzle-88B	2.15e−4	2.80e−8	2.24e−42	1.62e37 •⋄
GLM-5.3	0.0267	1.00	6.23e−41	6.24e35 •⋄
gpt-oss-120B	3.11e−4	1.70e−11	2.38e−37	1.97e32 •⋄
Qwen3.8-27B	1.64e−12	1.00	1.21e−20	1.31e16 •⋄
Qwen3-235B-A22B	1.04e−16	1.00	3.65e−11	3.03e12 •⋄
Qwen3-32B	2.71e−4	1.97e−13	9.44e−9	1.80e9 •⋄
Nemotron-3-Super	3.53e−6	1.00	4.98e−12	9.88e7 •⋄
Gemma-3-27B-IT	7.83e−12	1.00	2.78e−3	4.87e7 •⋄
Nemotron-Labs-3-Puzzle	2.70e−6	1.00	2.77e−9	3.11e5 •⋄
DeepSeek-V4-Pro	1.64e−7	1.00	0.630	3.03e3 •⋄
Phi-4	6.77e−6	0.993	1.19e−3	91.3 •⋄
Mistral-Small-3.1-24B	1.06e−4	0.992	0.0269	7.10 ⋄
Qwen3-0.6B	1.15e−3	0.970	0.365	0.927 ⋄

Table 22: **Other benchmarks: loose filtration across open models.** Sorted by aggregate E, the mean of nine components. Benchmark E averages three detectors. •: e-BH ($q = 0.05$). ⋄: Holm on component p-values ($\alpha = 0.05$).

Model	Benchmark	DOPR p	DWPC-QA p	CBA p	Benchmark E	Aggregate E
Qwen3.6-35B-A3B	MMLU-Pro	0.836	6.62e−172	0.528	1.48e167	
	GPQA-D	0.202	8.08e−3	0.109	1.39	4.92e166 •⋄
	HLE	0.871	0.698	0.263	0.296	
MiniMax-M2.1	MMLU-Pro	0.0145	2.37e−105	0.145	5.47e100	
	GPQA-D	0.238	1.43e−3	0.0746	4.87	1.82e100 •⋄
	HLE	0.488	0.523	0.225	0.329	
Qwen3-1.7B	MMLU-Pro	0.959	1.84e−46	0.396	1.11e42	
	GPQA-D	0.402	0.129	0.849	0.297	3.69e41 •⋄
	HLE	0.599	0.998	0.907	0.192	
Qwen3-8B	MMLU-Pro	0.0388	2.15e−29	0.798	1.21e25	
	GPQA-D	0.221	0.183	0.553	0.326	4.05e24 •⋄
	HLE	0.668	0.280	0.883	0.219	
Qwen3-30B-A3B	MMLU-Pro	0.188	9.25e−27	0.329	2.97e22	
	GPQA-D	0.0180	0.672	0.206	0.786	9.91e21 •⋄
	HLE	0.827	0.761	0.471	0.239	
GLM-5.3	MMLU-Pro	4.55e−22	1.00	0.883	6.71e17	
	GPQA-D	0.703	0.337	0.0806	0.524	2.24e17 •⋄
	HLE	0.0767	0.729	0.422	0.334	
GLM-5.2	MMLU-Pro	0.393	7.68e−15	0.404	4.96e10	
	GPQA-D	0.837	0.966	0.516	0.234	1.65e10 •⋄
	HLE	0.620	0.198	0.450	0.279	
Qwen3-14B	MMLU-Pro	0.629	8.58e−10	0.0510	5.62e5	
	GPQA-D	0.0171	0.442	0.608	0.711	1.87e5 •⋄
	HLE	0.333	0.354	0.363	0.294	
MiniMax-M3	MMLU-Pro	0.684	1.00	0.0622	0.534	
	GPQA-D	0.250	0.325	0.228	0.381	86.0 •⋄
	HLE	3.65e−5	1.99e−5	0.0681	257	
MiniMax-M2.5	MMLU-Pro	0.882	1.00	0.608	0.200	
	GPQA-D	0.367	0.927	0.361	0.288	5.66 ⋄
	HLE	0.965	2.35e−4	0.381	16.5	
Kimi-K2.6	MMLU-Pro	0.954	1.00	0.164	0.338	
	GPQA-D	0.105	6.70e−3	0.446	1.42	0.884
	HLE	0.987	0.966	0.0240	0.897	
Qwen3-32B	MMLU-Pro	0.0550	1.00	9.28e−3	1.58	
	GPQA-D	0.0696	0.582	0.447	0.392	0.725
	HLE	0.981	0.728	0.716	0.205	
Kimi-K3	MMLU-Pro	0.952	1.00	0.687	0.191	
	GPQA-D	0.286	0.999	0.473	0.268	0.443
	HLE	0.907	0.552	0.0261	0.870	
Qwen3.8-27B	MMLU-Pro	0.337	1.00	0.927	0.177	
	GPQA-D	0.608	0.997	0.311	0.285	0.438
	HLE	0.914	0.691	0.0269	0.852	
Nemotron-3-Super	MMLU-Pro	0.0641	1.00	0.769	0.205	
	GPQA-D	0.0634	0.949	0.0798	0.639	0.391
	HLE	0.998	0.753	0.191	0.330	
gpt-oss-20B	MMLU-Pro	0.0742	1.00	0.0699	0.521	
	GPQA-D	0.831	0.212	0.676	0.263	0.380
	HLE	0.242	0.378	0.221	0.355	
Qwen3-0.6B	MMLU-Pro	0.277	0.207	0.576	0.217	
	GPQA-D	0.0220	0.429	0.533	0.628	0.349
	HLE	0.452	0.963	0.887	0.201	
gpt-oss-puzzle-88B	MMLU-Pro	0.268	1.00	0.115	0.401	
	GPQA-D	0.158	0.928	0.856	0.261	0.337
	HLE	0.442	0.487	0.196	0.351	

Continued on next page

Table 22: **Other benchmarks: loose filtration across open models.** (continued).

Model	Benchmark	DOPR p	DWPC-QA p	CBA p	Benchmark E	Aggregate E
Mistral-Small-3.1-24B	MMLU-Pro	0.0507	1.00	0.0671	0.539	
	GPQA-D	0.901	0.999	0.462	0.238	0.332
	HLE	0.832	0.308	0.821	0.217	
Phi-4	MMLU-Pro	0.459	1.00	0.179	0.328	
	GPQA-D	0.519	0.726	0.130	0.415	0.331
	HLE	0.820	0.108	0.988	0.249	
Gemma-3-27B-IT	MMLU-Pro	0.869	1.00	0.618	0.199	
	GPQA-D	0.0442	0.393	0.523	0.458	0.323
	HLE	0.329	0.0825	0.626	0.313	
Muse-Glimmer-30B	MMLU-Pro	0.162	1.00	0.178	0.334	
	GPQA-D	0.823	0.365	0.170	0.381	0.309
	HLE	0.707	0.935	0.668	0.212	
Nemotron-Labs-3-Puzzle	MMLU-Pro	0.0438	1.00	0.258	0.307	
	GPQA-D	0.560	0.927	0.271	0.306	0.307
	HLE	0.954	0.323	0.267	0.307	
Qwen3-235B-A22B	MMLU-Pro	0.663	1.00	0.152	0.350	
	GPQA-D	0.334	0.370	0.277	0.343	0.303
	HLE	0.580	0.708	0.708	0.216	
Qwen3-4B	MMLU-Pro	0.138	0.914	0.382	0.249	
	GPQA-D	0.803	0.485	0.604	0.242	0.300
	HLE	0.960	0.634	0.119	0.410	
DeepSeek-V4-Pro	MMLU-Pro	0.646	1.00	0.285	0.268	
	GPQA-D	0.703	0.919	0.125	0.409	0.294
	HLE	0.994	0.390	0.855	0.205	
DeepSeek-V4-Flash	MMLU-Pro	0.603	1.00	0.835	0.180	
	GPQA-D	0.0361	0.995	0.863	0.438	0.284
	HLE	0.999	0.665	0.485	0.233	
gpt-oss-120B	MMLU-Pro	0.685	1.00	0.829	0.180	
	GPQA-D	0.671	0.957	0.634	0.225	0.255
	HLE	0.359	0.352	0.199	0.359	