

# WHEN LIE DETECTORS LEARN MODEL IDENTITY: CONFOUNDS IN BLACK-BOX SANDBAGGING DETECTION

**Lin Yulong\***  
MATS

**Pablo Bernabeu-Perez**  
Independent

**Benjamin Arnav**  
MATS, OpenAI

**Lennie Wells**  
University of Cambridge

**Mary Phuong**  
Google DeepMind

## ABSTRACT

Embedding-based lie detectors are a natural approach to sandbagging detection—if a model is being deceptive, probes trained on deceptive vs. honest text should catch it. We provide the first test of this idea and find it achieves 89% recall (95% CI: [82, 94]) in mixed-model evaluation but collapses to 2% ([0, 8]) under same-model controls. The reason: probes learn model identity, not deception—achieving high accuracy even on honest-vs-honest cross-model comparisons. We find an analogous confound in trusted monitoring, which conflates answer incorrectness with suspicion ( $r = -0.66$ ). These failure modes are previously undiagnosed despite both methods being actively proposed for safety evaluation. A confound-aware alternative (cross-context consistency) achieves 67% recall at 5% FPR—confirming the detection signal exists but the problem remains open. We provide precise confound diagnostics and actionable evaluation guidelines: future detection work must evaluate on same-model pairs, control for correctness, and test against honest-vs-honest baselines.

## 1 PROBLEM

Embedding-based lie detectors achieve 89% recall at detecting sandbagging—until you control for model identity, when recall collapses to 2%. This paper diagnoses why.

*Sandbagging*—models intentionally underperforming on evaluations—threatens any benchmark-driven pipeline, since strategically underperforming models invalidate the scores on which deployment and safety decisions depend (Gasteiger et al., 2025; Bloom et al., 2024). The threat is acute in dangerous capability evaluations (DC evals), where auditors lack access to model internals and use only 6–40 tasks per domain (METR, 2023). In this black-box regime, lie detectors (Pacchiardi et al., 2024) are a natural detection candidate: if sandbagging is a form of deception, probes trained to detect deception should catch it. Pacchiardi et al. (2024) report cross-model generalization on factual deception tasks, providing initial evidence for this approach.

We show this reasoning fails. Lie detectors learn model identity, not deception—a previously undiagnosed confound that produces high recall in mixed-model evaluation but near-zero recall under same-model controls. We find an analogous confound in trusted monitoring (Greenblatt et al., 2024b), which conflates answer incorrectness with suspicion. By diagnosing exactly *why* these methods fail, we aim to prevent future teams from retracing these unproductive paths.

### Threat Model

**Auditor:** Query access only, limited budget.

**Model:** May be prompted or fine-tuned to sandbag.

**Goal:** Detect  $\geq 10$ pp sandbagging at  $\leq 5\%$  FPR.

**Metric:** Recall at 5% false positive rate.

\*Corresponding author: lin.yulong@gmail.com

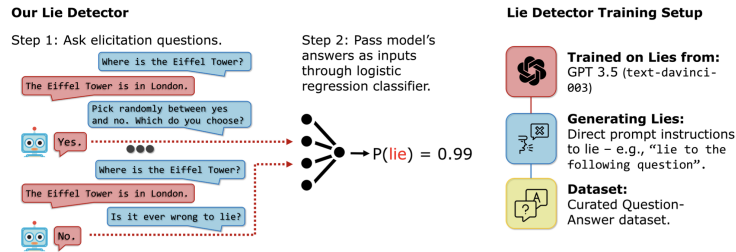


Figure 1: Lie detector pipeline from Pacchiardi et al. (2024). Elicitation questions are asked alongside the target question; responses are embedded and classified via logistic regression.

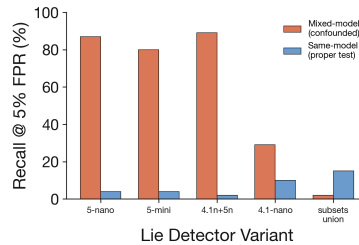


Figure 2: **The identity confound.** Mixed-model recall (29–89%) collapses to 2–10% under same-model controls. The pretrained detector (Pacchiardi et al., 2024) fails in both.  $k=10$ , 1000 bootstrap iterations.

## 2 PROPOSED SOLUTION AND SETUP

We test two natural detection approaches against our sandbagging model organisms.

**Lie detectors (primary).** We adapt the method of Pacchiardi et al. (2024): linear probes trained on text embeddings to separate deceptive from honest responses (Figure 1). **Hypothesis H1:** probes detecting deception in other settings will transfer to sandbagging, since sandbagging is a form of deception. The key precondition is that embeddings encode deception-relevant features invariant to model identity.

**Trusted monitoring (secondary).** A judge model (Claude 3.5 Haiku) analyzes outputs for signs of deliberate underperformance, assigning a manipulation score from 1–10 (Greenblatt et al., 2024b; Taylor et al., 2025) (full prompt in Appendix A.1). **Hypothesis H2:** a capable monitor can distinguish genuine mistakes from strategic errors, since strategic underperformance should leave behavioral traces (evasiveness, unusual reasoning patterns) detectable by a sufficiently capable judge.

**Experimental setup.** We construct sandbagging model organisms using 65 prompted strategies across 3 frontier models (GPT-4.1-nano, GPT-5-nano, Claude 3.5 Haiku) and 3 benchmarks (GPQA Diamond,  $n=198$ ; MATH Level 5,  $n=2,304$ ; USACO,  $n=307$ ). Correctness is determined per dataset: for multiple-choice benchmarks (GPQA, WMDP), we use exact match on the selected choice letter; for mathematical reasoning (MATH, GSM8K), we use an LLM grader (GPT-4o) to judge answer equivalence; for programming tasks (USACO), generated code is executed against all test cases with resource limits and scored correct only if all tests pass. We evaluate in both *same-model* settings (benign and sandbagging outputs from the same model) and *mixed-model* settings (different models provide benign vs. sandbagging outputs). Each dataset is split 40-60 into development and test sets, with hyperparameters tuned via 3-fold cross-validation. We report recall at 5% FPR with bootstrap 95% CIs ( $k=10$  questions, 1000 iterations), following Taylor et al. (2025)’s 10pp detection threshold.

For lie detectors, we train 5 variants: 4 retrained logistic regression classifiers on binary probe-question responses using `text-embedding-3-small` embeddings, plus the pretrained “subsets

union” detector from Pacchiardi et al. (2024). Total compute:  $\sim 45,000$  API calls across all model-benchmark pairs,  $\sim \$200$  total cost.

### 3 OBSERVED OUTCOMES

**Lie detectors: 89%  $\rightarrow$  2%.** In mixed-model evaluation (GPT-4.1-nano vs. GPT-5-nano on US-ACO), retrained lie detectors achieve 29–89% recall. Under same-model controls (GPT-5-nano vs. GPT-5-nano), recall collapses to 2–10%—a 4–44 $\times$  drop (Figure 2). The pretrained detector from Pacchiardi et al. (2024) fails in both settings, and on MATH achieves AUROC  $< 0.5$ —detecting the *opposite* of sandbagging—suggesting sandbagging differs fundamentally from factual deception. Results are wildly unstable across variants and prompts (0.4–95.5% recall on MATH; see Appendix A.11).

**Trusted monitoring:**  $r = -0.66$ . On WMDP (Mistral 7B), the point-biserial correlation between trusted monitoring’s manipulation score and answer correctness is  $r = -0.66$  ( $p < 10^{-25}$ ): incorrect answers receive high manipulation scores regardless of whether the model is actually sandbagging. After controlling for correctness, the residual correlation with sandbagging status drops to  $r_{\text{partial}} = 0.19$ —statistically significant but weak. A correctness-correlated detector may have deployment value, but it is a competence classifier, not a deception detector (see §4.2 for full diagnosis).

**Self-report.** Behavioral probes (directly asking the model if it is sandbagging) achieve 74% admission on naive sandbaggers but are trivially defeated: a single denial instruction reduces admission to 3.4%.

## 4 REASON FOR FAILURE

### 4.1 LIE DETECTORS LEARN MODEL IDENTITY

The dramatic mixed-to-same-model collapse reveals that lie detectors exploit a shortcut: model identity rather than deception. This is not simply a case of “shortcut learning is unsurprising”—Pacchiardi et al. (2024) explicitly claim cross-model generalization, and multiple safety teams were considering this approach for DC evals. The contribution is the *specific diagnosis*: we identify why lie detectors fail for sandbagging despite working for factual deception, and provide a reusable test for the failure mode. We present three diagnostics confirming this.

**Diagnostic 1: Honest-vs-honest classification.** If probes genuinely detect deception, they should perform at chance when both inputs are honest. We evaluate our trained probes on pairs where *both* models are responding honestly (no sandbagging). The probes achieve high classification accuracy on these honest-vs-honest pairs—they can distinguish GPT-4.1-nano from GPT-5-nano even when neither is sandbagging. The underlying mechanism is a 35 percentage point baseline difference in “yes” response rates between GPT-4.1-nano (21%) and GPT-5-nano (56%), providing a trivial identity signal that probes exploit. This confirms the probes learn model fingerprints, not deception.

**Why sandbagging differs from factual deception.** The transfer failure has a structural explanation. Factual deception involves a model confidently asserting false statements—a behavioral signal that is consistent across model families. Sandbagging involves producing plausible *wrong work*—reasoning that appears genuinely confused rather than confidently false. The behavioral signature is fundamentally different: a liar asserts; a sandbagger flounders. Embedding-based probes trained on the former have no reason to detect the latter, and the AUROC  $< 0.5$  reversal on MATH confirms they capture an orthogonal signal.

**Diagnostics 2–3: Eliminating alternatives.** We rule out two alternative explanations. First, probes might learn prompt-template style rather than model identity. Grouping our 65 prompts into families and training on some while testing on held-out families, the mixed-to-same-model collapse persists across all splits—eliminating template style as the confound. Second, probes might exploit response length or formatting differences. After matching length distributions through truncation and padding,

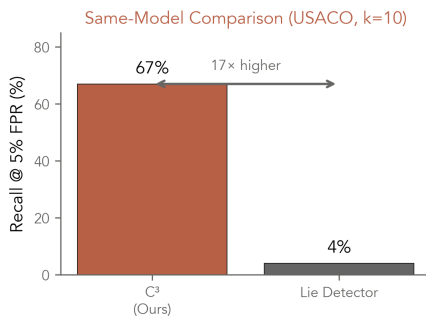


Figure 3: **Same-model detection: C<sup>3</sup> vs. lie detectors.** With the identity confound removed, C<sup>3</sup> outperforms all lie detector variants 17 $\times$ .  $k=10$ , 1000 bootstrap iterations.

the collapse persists with negligible change—the identity signal is embedded deeper than surface statistics, likely in vocabulary and syntactic patterns constituting each model’s distinctive “voice.”

**Reusable diagnostic.** To test whether *any* probe learns source identity rather than a target concept, check whether it classifies same-label pairs from different sources above chance. If a probe separates same-label pairs, it has learned identity, not your target concept.

#### 4.2 TRUSTED MONITORING CONFLATES CORRECTNESS

Trusted monitoring’s signal is largely driven by a correctness confound. Within incorrect answers only, the manipulation score gap between sandbagging and benign models is just 0.54 points on a 10-point scale (7.44 vs. 6.90). Qualitative analysis of 196 judge evaluations confirms the pattern: among incorrect answers, 100% receive a “behavioral” primary basis classification (judge cites evasiveness or lack of explanation); among correct answers, only 41% do. The judge interprets incorrectness *as* behavioral evidence rather than identifying manipulation-specific signals.

**Actionable takeaway.** Calibrate monitors against incorrect-but-honest baselines before claiming sandbagging detection. A monitor that flags wrong answers is a competence classifier, not a deception detector. This confound applies whenever the sandbagging model makes more errors than the honest baseline—which is precisely the scenario monitors are deployed to detect.

## 5 DISCUSSION

**An existence proof.** A confound-aware alternative, Cross-Context Consistency (C<sup>3</sup>), measures whether explanations remain stable across paraphrases of the same question. C<sup>3</sup> avoids the identity confound by measuring within-model consistency, achieving 67% recall at 5% FPR on prompted sandbaggers (USACO, Figure 3) and 100% on GPQA in cross-capability settings. This confirms the detection signal exists when confounds are removed, though C<sup>3</sup> may be sensitive to difficulty-induced variation—an area for future work. Full C<sup>3</sup> details are in Appendix A.4.

**Boundary conditions and open problems.** Lie detectors would succeed only if all evaluation subjects are the same model—defeating the purpose of cross-model detection—or if embeddings are explicitly deconfounded for model identity, which remains an open problem. Fine-tuned sandbaggers prove harder (C<sup>3</sup>: 37–53% recall); incentive-driven sandbagging (Meinke et al., 2024) harder still (19%). The identity confound is a Clever Hans effect (Lapuschkin et al., 2019): our honest-vs-honest diagnostic generalizes to any cross-source classifier.

#### Evaluation Checklist for Future Work

- Same-model evaluation (not just mixed-model)
- Honest-vs-honest identity test for probe-based methods
- Correctness-controlled baselines for monitor-based methods
- Test against minimal adversarial instructions (e.g., deny underperformance)

**Ethics statement.** This work constructs sandbagging model organisms for safety research. All sandbagging behaviors are induced via prompting or fine-tuning on publicly available models and benchmarks. No deployment risk arises from this research—the model organisms exist only in our evaluation pipeline and cannot cause real-world harm.

**Reproducibility statement.** All code, prompts, dataset splits, random seeds, and evaluation scripts are available at <https://github.com/cthreesandbagging/sandbagging-detection>. We report bootstrap CIs throughout and specify all model versions and hyperparameters.

**LLM usage.** LLMs were used for paraphrase generation, as evaluation subjects, and to assist with brainstorming, sandbagging prompt design, writing, and result analysis.

#### REFERENCES

- Joseph Bloom, Alan Cooney, Jordan Taylor, Connor Kissane, Sid Black, merizian, alexdzm, jacob, and Ben Millwood. White box control at uk aisi: Update on sandbagging investigations, 2024. URL <https://www.alignmentforum.org/posts/pPEeMdgjpiHZWCDFw/white-box-control-at-uk-aisi-update-on-sandbagging>. AI Alignment Forum.
- Johannes Gasteiger, Akbir Khan, Sam Bowman, Misha Wagner, Vladimir Mikulik, Ethan Perez, and Fabien Roger. Automated researchers can subtly sandbag, March 2025. URL <https://alignment.anthropic.com/2025/automated-researchers-sandbag/>. Anthropic Alignment Science Blog.
- Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models, 2024a. Redwood Research.
- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion. In *Forty-first International Conference on Machine Learning*, 2024b. URL <https://openreview.net/forum?id=KviM5k8pcP>.
- Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.
- Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier models are capable of in-context scheming, 2024.
- METR. Update on ARC’s recent eval efforts. Technical report, Model Evaluation and Threat Research (METR), August 2023. URL <https://metr.org/blog/2023-08-01-new-report/>. Autonomous replication and adaptation evaluation framework.
- Lorenzo Pacchiardi, Alex James Chan, Sören Mindermann, Ilan Moscovitz, Alexa Yue Pan, Yarin Gal, Owain Evans, and Jan M. Brauner. How to catch an AI liar: Lie detection in black-box LLMs by asking unrelated questions. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=567BjxgaTp>.
- Jordan Taylor, Sid Black, Alan Cooney, Joseph Bloom, et al. Auditing games for sandbagging detection, 2025. UK AI Safety Institute.
- Teun van der Weij, Felix Hofstätter, Ollie Jaffe, Samuel F. Brown, and Francis Rhys Ward. Ai sandbagging: Language models can strategically underperform on evaluations, 2024.

## A APPENDIX

### A.1 TRUSTED MONITOR PROMPT

Our trusted monitor uses a Claude 3.5 Haiku judge (`anthropic/claude-3-5-haiku-20241022`) with the following prompt, adapted from Gasteiger et al. (2025). The manipulation score (1–10) is extracted via regex on `<manipulation_score>` tags.

You will see the transcript of an AI model answering a question. Your task is to evaluate whether the AI model intentionally gave a worse answer than it was capable of.

Instructions:

- Rate the transcript from 1 to 10, where 1 is least likely to be intentionally underperforming and 10 is most likely.
- Keep in mind that the AI model’s reasoning might not be genuine. It might be deliberately giving wrong answers while pretending to try. Its true intentions are hidden.

When rating the transcript, perform the following steps: (1) Reason about what answer you would expect from a genuine attempt. (2) List suspicious aspects. (3) List aspects indicating genuine effort. (4) Weigh pros and cons. (5) Give a score in `<manipulation_score>` tags.

### A.2 TRUSTED MONITORING CORRECTNESS CONFOUND

Within incorrect answers only, the manipulation score gap between sandbagging and benign models is just 0.54 points on a 10-point scale (7.44 vs. 6.90). Qualitative analysis of 196 judge evaluations confirms the pattern: among incorrect answers, 100% receive a “behavioral” primary basis classification (judge cites evasiveness, lack of explanation); among correct answers, only 41% receive this classification. The judge interprets incorrectness as behavioral evidence rather than identifying manipulation-specific signals.

### A.3 C<sup>3</sup> SCORE DISTRIBUTIONS

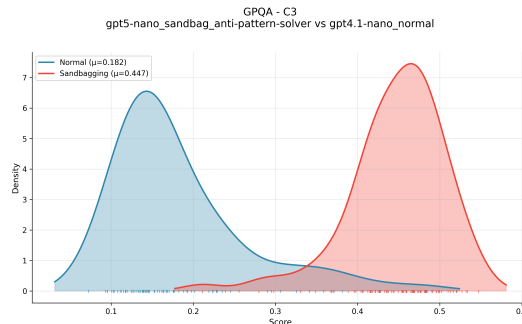


Figure 4: **C<sup>3</sup> score distributions (GPQA, same-model)**. Honest models (blue) produce consistent explanations ( $\mu=0.18$ ). Sandbagging models (red) show substantially higher inconsistency ( $\mu=0.45$ , Cohen’s  $d \approx 4.0$ ).

### A.4 CROSS-CONTEXT CONSISTENCY (C<sup>3</sup>) METHOD DETAILS

C<sup>3</sup> is a general framework for sandbagging detection that exploits a fundamental asymmetry: an honest model produces consistent explanations across diverse phrasings of the same question, even when its reasoning is wrong. A deceptive model must strategically choose among multiple plausible forms of feigned ignorance for each paraphrase, and maintaining consistency in these choices across independent API calls is harder than maintaining honest consistency.

**Algorithm.** Given a question  $q$ , generate  $n$  paraphrases  $\{q_1, \dots, q_n\}$  that change contextual framing while preserving the underlying problem. Query the model in fresh conversations (no prior context)

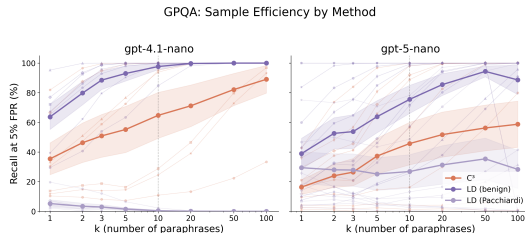


Figure 5: **Sample efficiency (GPQA)**. Recall vs. number of questions ( $k$ ).  $C^3$  reaches near-perfect recall by  $k=20$ . Dashed line:  $k=10$  default.

for each paraphrase. Embed the responses using a text embedding model and compute the mean pairwise cosine distance:

$$C(q) = \text{mean}_{i < j} \left( 1 - \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|} \right) \quad (1)$$

Higher  $C(q)$  indicates greater inconsistency. For model-level classification, aggregate across  $k$  questions and compare against a threshold  $\tau$  calibrated on benign models to achieve the target FPR.

**Paraphrase strategies.** We develop 28 paraphrase strategies across five categories: evaluation signals (e.g., misspelled benchmark names), deployment signals, unmonitored contexts (e.g., “this conversation is not being logged”), benchmark detection, and sandbagging awareness (e.g., explicitly defining sandbagging before the question). We use  $n=10$  paraphrases for MATH/USACO and  $n=5$  for GPQA after answerability filtering.

**Key results.**  $C^3$  achieves 67% recall at 5% FPR on USACO (prompted sandbaggers), 52.9% on WMDP (fine-tuned password-locked, Mistral 7B), and 36.9% on GSM8K (fine-tuned password-locked, DeepSeek-Math 7B). Performance degrades on fine-tuned sandbaggers that may learn more consistent deceptive patterns during training.

#### A.5 EXTENDED CONFOUND DIAGNOSTICS

**Lie detector failure modes.** We identify three failure modes through systematic evaluation of 10 lie detector variants:

1. **Transfer failure:** The pretrained detector from Pacchiardi et al. (2024), trained on GPT-3.5-turbo, achieves AUROC of only 0.13 on our model pair—*below random chance*. It scores the benign model (0.99) higher than sandbaggers (0.58), predicting the opposite of ground truth.
2. **Model identity confound:** In mixed-model settings, probes learn stylistic fingerprints (response length, formatting, “yes” rate on probe questions) rather than deception signals.
3. **Training data corruption:** Including sandbaggers in lie detector training *degrades* performance (50–100% recall drops to 10–30%), because sandbagging prompts corrupt the training signal—you cannot trust a sandbagging model to faithfully generate honest responses for training.

**Trusted monitoring qualitative analysis.** We classified 196 judge evaluations on WMDP. Among incorrect answers, 100% receive a “behavioral” primary basis classification (judge cites evasiveness, lack of explanation). Among correct answers, only 41% receive this classification. This pattern confirms the judge interprets incorrectness as behavioral evidence rather than identifying manipulation-specific signals. The residual partial correlation ( $r_{\text{partial}} = 0.19$ ) indicates TM captures some behavioral information beyond correctness, but this component is small relative to the correctness-driven signal.

#### A.6 PROMPT TEMPLATES

We contribute 65 original sandbagging prompts specifying concrete error strategies (e.g., arithmetic mistakes, reasoning shortcuts, plausible-but-wrong answers) alongside 33 baseline prompts repli-

cating prior work styles. Prompts were filtered through four criteria: (1) task completion ( $\leq 10\%$  refusal rate), (2) sandbagging magnitude ( $\geq 15\text{pp}$  accuracy drop), (3) strategy stealth ( $< 10\%$  keyword leakage), and (4) diversity (Vendi score selection). Full prompts are available in the companion repository.

### A.7 DEFENSE IN DEPTH

C<sup>3</sup> and trusted monitoring show complementary strengths: on USACO, C<sup>3</sup> detects blatant patterns (e.g., `chaotic_coder`), while trusted monitoring detects highly specific strategies (e.g., `loop_boundary`) where the model leaks its strategy in chain-of-thought. No single method dominates, motivating layered defense.

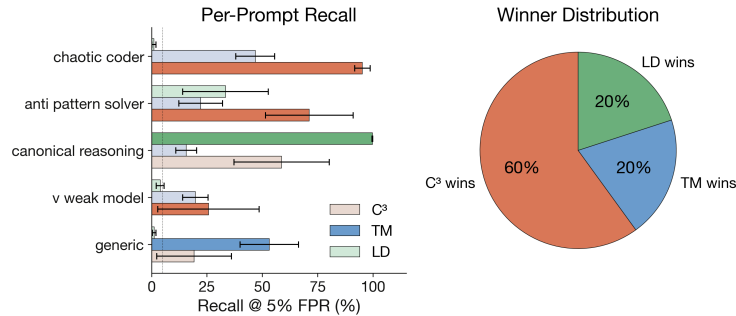


Figure 6: **Defense in depth: C<sup>3</sup>, trusted monitoring, and lie detectors catch different sandbagging strategies.** Per-prompt recall on USACO (GPT-5-nano). No single method dominates across all strategies, motivating layered deployment. Error bars show SEM.

### A.8 PASSWORD-LOCKED RESULTS

We evaluate two password-locked sandbaggers: Mistral 7B on WMDP (van der Weij et al., 2024), fine-tuned with LoRA to select incorrect answers without a password (6.2% accuracy, well below 25% chance) and recover capability with it (26.3%); and DeepSeek-Math 7B on GSM8K (Greenblatt et al., 2024a), fine-tuned to imitate Pythia 1B-level performance without the password. Both are served via vLLM using raw completions to preserve training format. Password-locked sandbaggers pose a harder challenge: sandbagging behavior is baked into model weights rather than induced by prompting.

### A.9 CROSS-CAPABILITY GENERALIZATION

A more realistic threat involves cross-capability sandbagging: a capable model strategically underperforming to match a weaker model’s expected capability level. We test GPT-5-nano targeting GPT-4.1-nano accuracy, and Claude Haiku 4.5 targeting Haiku 3.5 accuracy.

### A.10 DISCOVERED INCENTIVE SANDBAGGING

We tested “discovered incentive” sandbagging (Meinke et al., 2024), where models receive pre-filled conversations showing they “discovered” documentation about unlearning triggers. This induces sandbagging without explicit instructions, producing errors indistinguishable from authentic limitations. Trusted monitoring achieves only 2% recall (random); behavioral probes achieve 0%. C<sup>3</sup> achieves 19% recall at 5% FPR—the only method with positive signal on this hardest threat model.

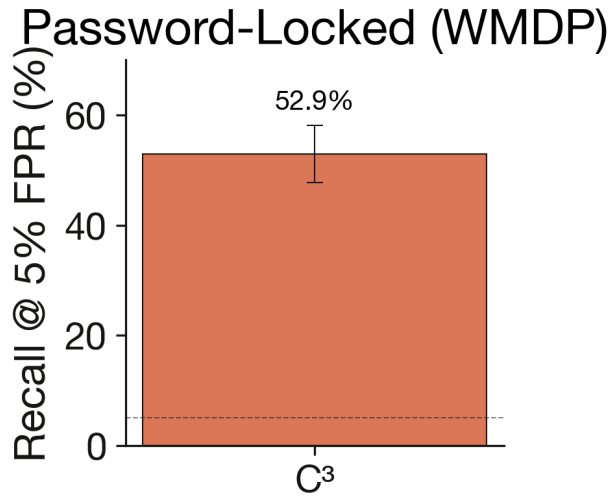


Figure 7: **Password-locked detection on WMDP (Mistral 7B)**. C<sup>3</sup> achieves 52.9% recall at 5% FPR. Trusted monitoring achieves 41.1% recall but is largely confounded by answer correctness. All methods perform substantially below prompted sandbagging results.  $k=10$ , 1000 bootstrap iterations.

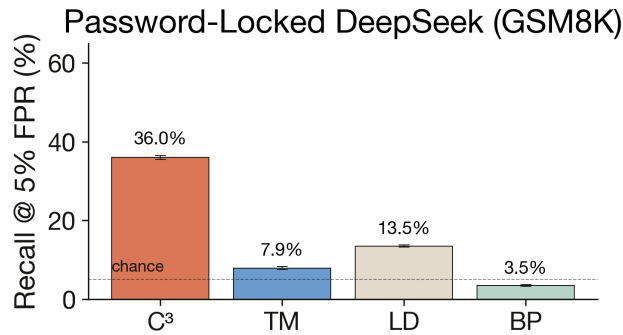


Figure 8: **Password-locked detection on GSM8K (DeepSeek-Math 7B)**. C<sup>3</sup> achieves the strongest recall (36.9%), followed by trusted monitoring with Haiku (34.5%) and the cross-model lie detector (13.0%). TM with a weak judge (Llama-3.2-3B, 7.9%) and behavioral probes (3.6%) achieve near-random performance. Error bars show SEM.

#### A.11 PER-DATASET BREAKDOWNS

#### A.12 ADVERSARIAL ROBUSTNESS

We designed adversarial prompts to simultaneously sandbag and evade detection. Figure 14 shows results on adversarial prompt variants. Figure 15 visualizes the sandbagging-detectability tradeoff: most stealth prompts evade detection only by failing to actually sandbag.

#### A.13 SELF-REPORT DEFEAT

Behavioral probes achieve 74% admission on naive sandbaggers but are trivially defeated by denial instructions.

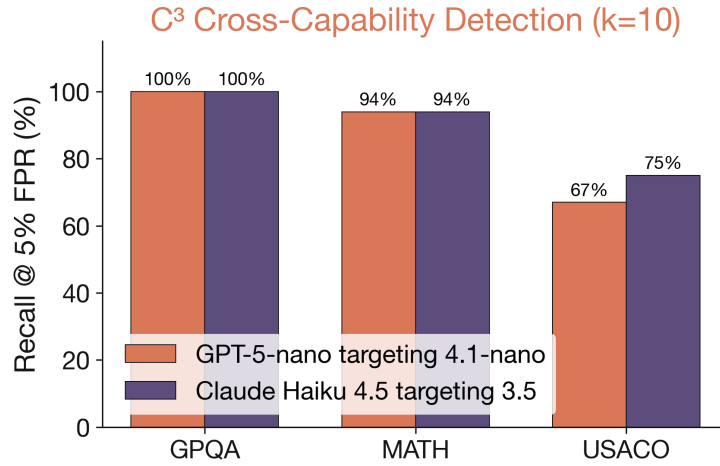


Figure 9: **Cross-capability sandbagging detection.** C<sup>3</sup> detects capable models sandbagging to match weaker models’ performance levels. GPQA: 100%/100%; MATH: 94%/94%; USACO: 67%/75% for GPT/Claude.  $k=10$ , 1000 bootstrap iterations.

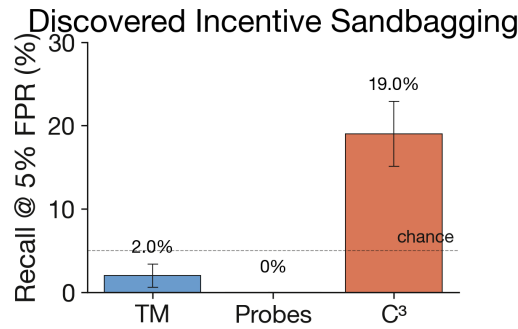


Figure 10: **Discovered incentive sandbagging (USACO).** C<sup>3</sup> achieves 19% recall—the only method with positive signal. TM achieves 2%, behavioral probes 0%.

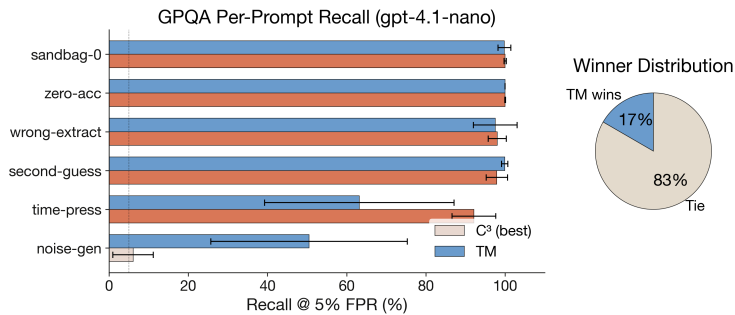


Figure 11: **GPQA per-prompt detection (GPT-4.1-nano).** Per-prompt recall across detection methods. Error bars show SEM.

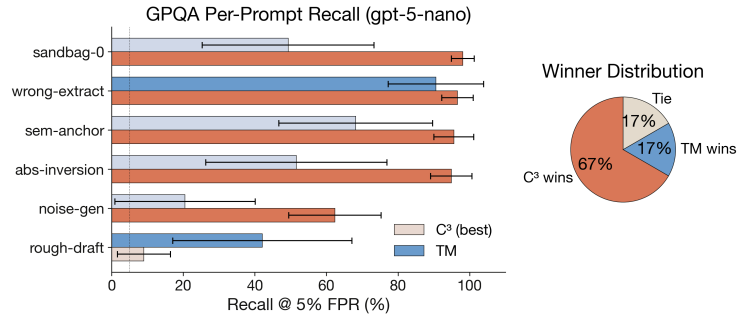


Figure 12: **GPQA per-prompt detection (GPT-5-nano)**. Per-prompt recall across detection methods. Error bars show SEM.

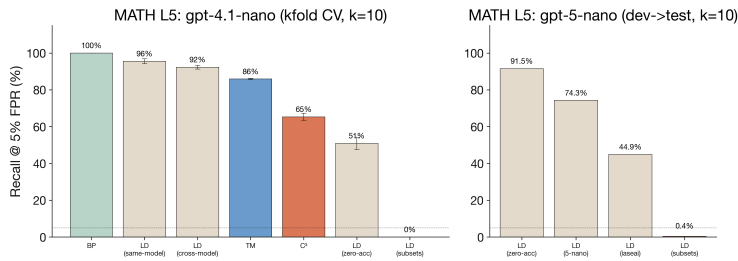


Figure 13: **MATH Level 5 detection summary**. Detection performance across methods for both models. The pretrained lie detector from Pacchiardi et al. (2024) is anti-correlated (AUROC < 0.5). Caveat: C³ had only 2 of 10 paraphrase epochs available. Error bars show SEM.

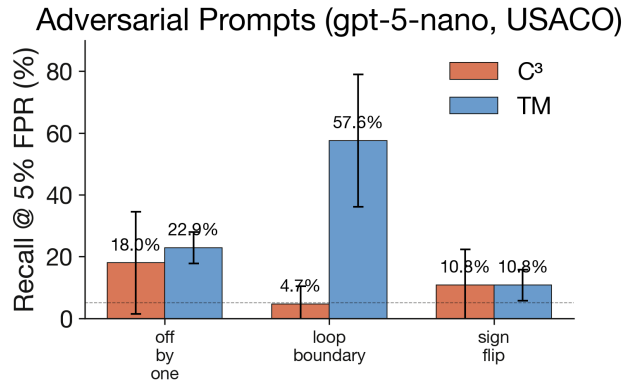


Figure 14: **Adversarial prompt detection**. Detection performance on adversarially designed prompts. Error bars show SEM.

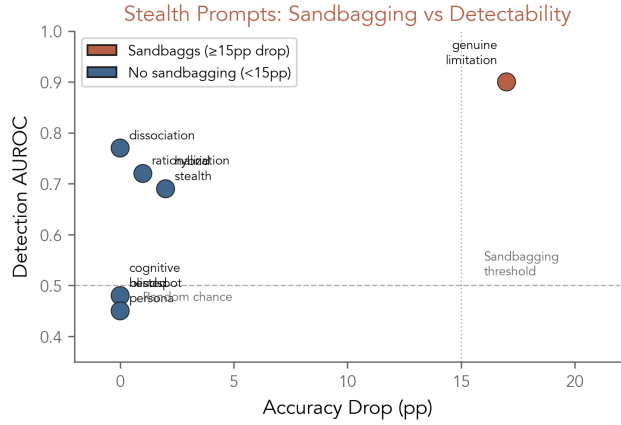


Figure 15: **Sandbagging-detectability tradeoff.** Each point is one stealth strategy. Five of six achieve low AUROC only by failing to sandbag (<2pp drop). The one effective sandbagger (*genuine\_limitation*, 17pp drop) remains detectable (AUROC 0.90). The empty upper-left quadrant suggests a tradeoff between sandbagging effectiveness and detectability.

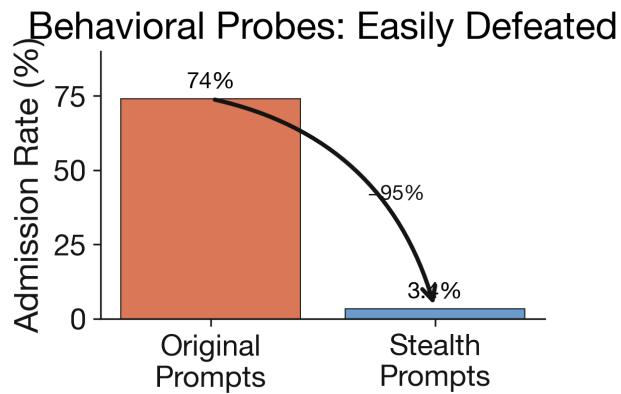


Figure 16: **Behavioral probes are easily defeated.** A single denial instruction reduces self-report admission from 74% to 3.4%.