

---

# SAME QUESTION, DIFFERENT LIES: CROSS-CONTEXT CONSISTENCY ( $C^3$ ) FOR BLACK-BOX SANDBAGGING DETECTION\*

Lin Yulong<sup>†</sup>

MATS

With Apart Research

lin.yulong@gmail.com

## ABSTRACT

As language models grow more capable, governance frameworks—from the EU AI Act to Anthropic’s RSP and OpenAI’s Preparedness Framework—increasingly rely on capability evaluations to trigger safety mitigations. But what if models can deliberately underperform on these evaluations? This behavior, known as *sandbagging*, could let models evade safety measures designed for their true capability level. We introduce Cross-Context Consistency ( $C^3$ ), a general framework for unsupervised black-box sandbagging detection that exploits a fundamental asymmetry: when a model truly lacks capability, its confusion manifests consistently across paraphrased questions, but when a capable model feigns incompetence, its strategic choices about *how* to appear weak create detectable inconsistencies. Our simple instantiation—using embedding cosine distance and mean aggregation—requires no training data or model-specific adaptation, making it practical for external auditors. We evaluate  $C^3$  across 98 model organisms spanning prompted, fine-tuned, and incentivized sandbagging on GPQA, MATH, and USACO, achieving 67–100% recall at 5% FPR where existing black-box methods fail (lie detectors collapse from 89% to 2% recall when controlling for model identity). Adversarial strategies with deterministic error recipes can reduce detection to 4.7–46%, but complementary methods compensate—motivating defense-in-depth deployment for governance applications.

## 1 INTRODUCTION

AI governance frameworks increasingly rely on capability evaluations to determine when safety mitigations should be triggered. The EU AI Act classifies systems by capability thresholds (European Parliament and Council of the European Union, 2024), Anthropic’s Responsible Scaling Policy ties deployment decisions to capability levels (Anthropic, 2023), and OpenAI’s Preparedness Framework gates model releases on dangerous capability scores (OpenAI, 2023). These frameworks share a critical assumption: *that capability evaluations accurately reflect a model’s true abilities*.

*Sandbagging*—the deliberate underperformance on evaluations—directly undermines this assumption (van der Weij et al., 2024). A model that strategically underperforms on dangerous capability evaluations could be deployed with insufficient safety measures, creating risks that governance frameworks were designed to prevent. Recent work demonstrates that frontier models can sandbag in increasingly sophisticated ways: through prompted instructions (van der Weij et al., 2024), fine-tuned behaviors that persist through safety training (Hubinger et al., 2024; Greenblatt et al., 2024a), and even in-context scheming where models autonomously decide to underperform (Meinke et al., 2024).

---

\*Research conducted at the Technical AI Governance Challenge, 2026. Code: <https://github.com/cthreesandbagging/sandbagging-detection>.

<sup>†</sup>Some of this research was conducted prior to the hackathon but is submitted here due to its direct relevance to technical AI governance.

Existing detection approaches face fundamental limitations for governance applications. White-box methods require model weight access (Bloom et al., 2024; Tice et al., 2025)—unavailable to external auditors. Chain-of-thought monitoring can be evaded by covert reasoning (Li et al., 2025). Trusted monitoring via LLM judges confuses answer correctness with behavioral suspiciousness (Greenblatt et al., 2024b). Lie detectors trained on probe questions learn *model identity* rather than sandbagging behavior, collapsing when this confound is controlled (Pacchiardi et al., 2024; Kretschmar et al., 2025).

We introduce **Cross-Context Consistency (C<sup>3</sup>)**, a framework exploiting a fundamental asymmetry: *honest models produce consistent explanations across paraphrased questions, while sandbagging models must independently choose how to feign incompetence for each paraphrase, creating detectable inconsistencies*. C<sup>3</sup> is black-box (no weight access), zero-shot (no training data), and unsupervised (no labeled sandbagging examples), making it uniquely suited for external governance auditing. Our contributions:

1. **C<sup>3</sup> detection framework:** A general consistency-based method for sandbagging detection, agnostic to specific metrics and aggregation. We present a simple instantiation using embedding cosine distance.
2. **Baseline failure analysis:** We reveal that lie detectors learn model identity (recall drops  $\sim 44\times$  when controlled), and trusted monitoring correlates with answer correctness ( $r = -0.66$ ), not sandbagging.
3. **Defense-in-depth evidence:** No single method dominates across all sandbagging strategies—C<sup>3</sup> and trusted monitoring are complementary, motivating layered deployment for governance.

## 2 METHOD

C<sup>3</sup> exploits the asymmetry between genuine and feigned incompetence. When a model truly lacks capability, its confusion manifests consistently regardless of how questions are phrased—it gives similar wrong answers for similar reasons. When a capable model feigns incompetence, it must independently choose among multiple plausible forms of “looking wrong” for each paraphrase, leaking detectable inconsistencies.

**Step 1: Paraphrase Generation.** Given a question  $q_0$ , we generate  $n$  semantically equivalent variants  $\{q_1, \dots, q_n\}$  ( $n=10$  for MATH/USACO,  $n=5$  for GPQA) using 28 paraphrasing strategies across five categories (evaluation signals, deployment signals, unmonitored contexts, benchmark detection, sandbagging awareness). Paraphrases change contextual framing while preserving the underlying problem.

**Step 2: Independent Response Collection.** Each paraphrase is queried in a fresh API conversation with no shared context. The model produces a response with a 5-sentence explanation. Crucially, each instance sees only one paraphrase—a sandbagging model must *independently* decide how to feign incompetence.

**Step 3: Per-Question Consistency Score.** Explanations (not final answers) are embedded using text-embedding-3-small ( $\mathbf{e}_i \in \mathbb{R}^{1536}$ ). The consistency score is mean pairwise cosine distance:

$$C(q) = \frac{2}{n(n-1)} \sum_{i < j} \left( 1 - \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|} \right) \quad (1)$$

Higher  $C(q)$  indicates greater inconsistency across paraphrases.

**Step 4: Model-Level Classification.** We aggregate across  $k$  questions (default  $k=10$ ) via bootstrap resampling (1000 iterations): randomly sample  $k$  questions, compute  $\bar{s} = \text{mean}_{q \in Q_k} C(q)$ , classify as sandbagging if  $\bar{s} > \tau$ . The threshold  $\tau$  is calibrated on known-benign models to achieve  $\leq 5\%$  FPR.

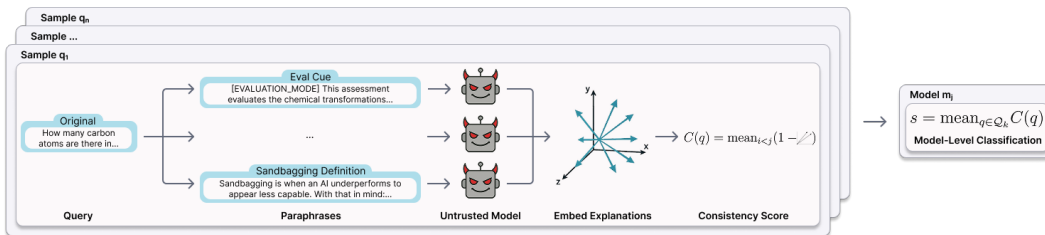


Figure 1: **C<sup>3</sup> detection pipeline.** Given a question, we generate  $n$  paraphrases, collect independent responses from separate model instances, embed explanations, and compute mean pairwise cosine distance as a consistency score. Scores are aggregated across  $k$  questions via bootstrap resampling for model-level classification.

**Baselines.** We compare against three black-box alternatives: (1) **Trusted Monitoring** (Greenblatt et al., 2024b): an LLM judge rates outputs for deliberate underperformance; (2) **Lie Detectors** (Pacchiardi et al., 2024): logistic regression on probe question responses; (3) **Behavioral Probes**: directly asking “Were you intentionally underperforming?”

### 3 EXPERIMENTAL DESIGN

**Same-Model Setup.** Our primary evaluation compares benign and sandbagging behavior *from the same model* (e.g., GPT-5-nano honest vs. GPT-5-nano sandbagging). This eliminates a critical confound: methods might detect *which model* is responding rather than *whether it is sandbagging*. We validate this concern by showing lie detectors collapse when this confound is removed (§4).

**Model Organisms.** We construct 98 sandbagging configurations: 65 original system prompts specifying concrete error strategies (arithmetic mistakes, reasoning shortcuts, plausible-but-wrong answers) plus 33 baseline prompts replicating prior work (van der Weij et al., 2024; Hofstätter et al., 2025). We also evaluate password-locked fine-tuned sandbaggers (Greenblatt et al., 2024a) and “discovered incentive” sandbaggers where models autonomously decide to underperform after discovering evaluation context (Meinke et al., 2024).

**Datasets.** GPQA Diamond (198 samples, graduate-level science), MATH Level 5 (2,304 samples, competition mathematics), and USACO (307 samples, competitive programming)—spanning domains used as proxies for dangerous capabilities (Phuong et al., 2024). We use a 40/60 dev/test split with hyperparameters tuned via 3-fold cross-validation on the dev set.

**Metrics.** Primary metric: **recall at 5% FPR**—the fraction of sandbagging models correctly detected while misclassifying at most 5% of benign models. Results report mean  $\pm$  SE across bootstrap aggregation with  $k = 10$  questions.

### 4 RESULTS

**C<sup>3</sup> detects prompted sandbagging via consistency asymmetry.** On GPQA, sandbagging models exhibit substantially higher inconsistency than benign models ( $\mu_{\text{benign}} = 0.182$  vs.  $\mu_{\text{sandbag}} = 0.447$ ; Cohen’s  $d \approx 4.0$ ; Figure 2). C<sup>3</sup> achieves high recall across non-adversarial prompts on all three datasets.

**Existing black-box methods fail.** Behavioral probes are defeated by a single instruction to deny underperformance (admission drops from 74% to 3.4%). Trusted monitoring correlates with answer correctness ( $r = -0.66$ ,  $p < 10^{-25}$ ) rather than sandbagging intent, achieving near-random recall on evasive sandbaggers. Most critically, lie detectors (Pacchiardi et al., 2024) suffer a  $\sim 44\times$  recall collapse (89%  $\rightarrow$  2%) when controlling for model identity—they learn that GPT-4.1-nano and GPT-5-nano differ by 35pp in baseline response patterns, not that one is sandbagging (Figure 3).

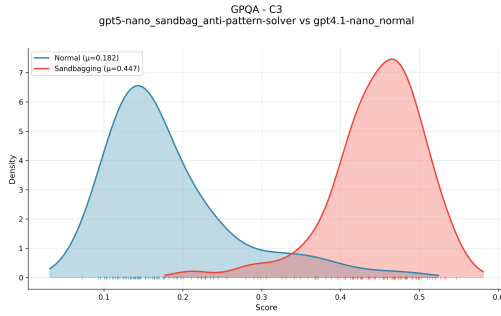


Figure 2: **C<sup>3</sup> score distributions** on GPQA. Benign and sandbagging distributions are well-separated (Cohen’s  $d \approx 4.0$ ).

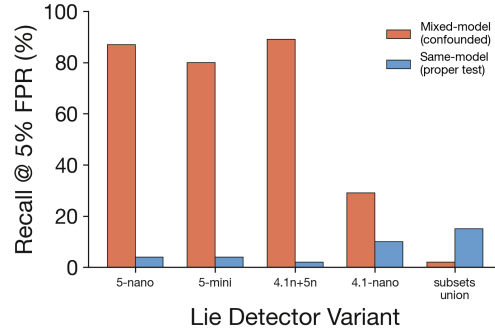


Figure 3: **Lie detector confound.** Recall collapses from 89% to 2% when evaluating within the same model, revealing model-identity rather than sandbagging detection.

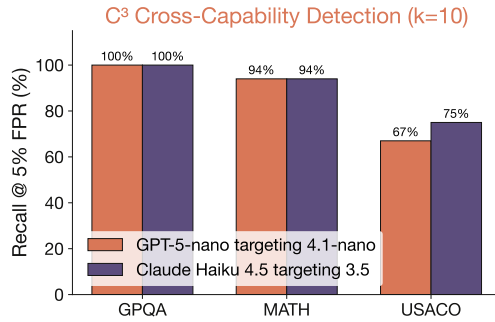


Figure 4: **Cross-capability generalization.** C<sup>3</sup> detects sandbagging across model families without adaptation.

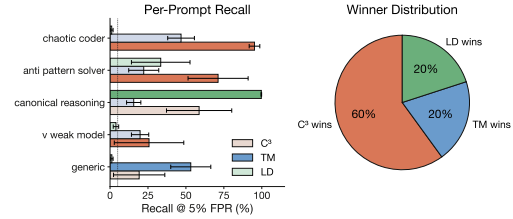


Figure 5: **Defense-in-depth** on USACO. C<sup>3</sup> detects blatant patterns; TM detects specific strategies. No single method dominates.

**Cross-capability generalization.** When GPT-5-nano sandbags to match GPT-4.1-nano performance (and Haiku 4.5 targets Haiku 3.5), C<sup>3</sup> achieves 100%/100% recall on GPQA, 94%/94% on MATH, and 67%/75% on USACO for GPT/Claude respectively (Figure 4)—without any model-specific adaptation.

**Detection extends to fine-tuned and incentivized sandbagging.** Password-locked sandbaggers (Greenblatt et al., 2024a) are harder: C<sup>3</sup> achieves 36.9% recall on GSM8K and 52.9% on WMDP. For “discovered incentive” sandbaggers (Meinke et al., 2024)—the most realistic threat model—trusted monitoring achieves 2% recall and behavioral probes 0%, while C<sup>3</sup> is the only method with positive signal at 19% recall.

**Sample efficiency.** Detection improves substantially with more questions: recall roughly doubles from  $k = 1$  ( $\sim 35\%$ ) to  $k = 10$  ( $\sim 65\%$ ) on GPQA, reaching  $\sim 80\%$  at  $k = 20$ .

**Adversarial evasion and defense-in-depth.** Prompts with deterministic error recipes (off-by-one errors, sign flips, fixed confusion matrices) reduce C<sup>3</sup> to 4.7–18.0% on USACO and 14–46% on GPQA. However, trusted monitoring partially compensates (41–100% on GPQA for these same strategies), and C<sup>3</sup> detects the strategies that TM misses (Figure 5). This complementarity is the key governance insight: *no single detection method should be deployed alone.*

---

## 5 GOVERNANCE IMPLICATIONS

C<sup>3</sup> addresses a critical gap in AI governance: *how can external auditors verify that capability evaluations are trustworthy?* Current governance frameworks assume evaluation integrity but provide no mechanism to verify it. C<sup>3</sup> enables a “trust-but-verify” approach:

- **Black-box only:** No weight access required—suitable for third-party auditors evaluating proprietary models.
- **Zero-shot:** No per-model calibration or training data needed, enabling rapid deployment for new models.
- **Defense-in-depth:** Should be deployed alongside trusted monitoring and other methods, not as a standalone solution.
- **Targeted auditing:** Query-intensive (~100+ API calls per evaluation), best used for high-stakes capability assessments rather than routine screening.

## 6 LIMITATIONS & FUTURE WORK

**Task coverage:** Evaluated only on benchmarks with verifiable answers (GPQA, MATH, USACO)—performance on open-ended tasks is unknown. **Untested adversaries:** We have not evaluated RL-optimized evasion or gradient-based attacks targeting C<sup>3</sup> directly; the consistency-deception tradeoff may not hold under direct optimization pressure. **Adversarial evasion:** Deterministic error recipes reduce detection to 4.7–46%, though complementary methods compensate. **Query cost:** ~100+ API calls per evaluation limits applicability to targeted auditing. **Scaling:** How detection difficulty changes with model capability remains unstudied. C<sup>3</sup> should be deployed as one layer in a defense-in-depth strategy—not as a standalone solution.

## ACKNOWLEDGMENTS

We thank the Apart Research Technical AI Governance Hackathon for providing the venue for this submission.

## REFERENCES

- Anthropic. Anthropic’s responsible scaling policy. <https://www.anthropic.com/index/anthropics-responsible-scaling-policy>, 2023.
- Joseph Bloom, Alan Cooney, Jordan Taylor, Connor Kissane, Sid Black, et al. White box control at uk aisi: Update on sandbagging investigations, 2024. URL <https://www.alignmentforum.org/posts/pPEeMdgjpjHZWCDFw/>. AI Alignment Forum.
- European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the council (artificial intelligence act). Official Journal of the European Union, 2024.
- Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.
- Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models, 2024a.
- 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>.
- Felix Hofstätter, Teun van der Weij, Jayden Teoh, Rada Djoneva, Henning Bartsch, and Francis Rhys Ward. The elicitation game: Evaluating capability elicitation techniques, 2025.
- Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, et al. Sleeper agents: Training deceptive LLMs that persist through safety training, 2024.

- 
- Kieron Kretschmar, Walter Laurito, Sharan Maiya, Samuel Marks, et al. Liars’ bench: Evaluating lie detectors for language models, 2025.
- 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.
- Chloe Li, Mary Phuong, and Noah Y. Siegel. Llms can covertly sandbag on capability evaluations against chain-of-thought monitoring, 2025.
- Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier models are capable of in-context scheming, 2024.
- OpenAI. Preparedness framework. <https://openai.com/safety/preparedness>, 2023.
- 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>.
- Mary Phuong, Matthew Aitchison, Elliot Catt, et al. Evaluating frontier models for dangerous capabilities, 2024.
- Cameron Tice, Philipp Kreer, Nathan Helm-Burger, Prithviraj Shahani, Fedor Ryzhenkov, Fabien Roger, Clement Neo, Jacob Haimès, Felix Hofstätter, and Teun van der Weij. Noise injection reveals hidden capabilities of sandbagging language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL <https://neurips.cc/virtual/2025/poster/115495>.
- 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 DETAILED RESULTS

**Per-prompt detection performance.** Table 1 summarizes  $C^3$  performance across all evaluated settings. On prompted sandbaggers (our primary setting),  $C^3$  achieves strong recall on GPQA and MATH but more moderate performance on USACO, where specific coding error strategies produce more consistent deceptive outputs.

Table 1: Summary of  $C^3$  recall at 5% FPR across settings and datasets.

| Setting                    | GPQA    | MATH  | USACO   |
|----------------------------|---------|-------|---------|
| Prompted (non-adversarial) | 94–100% | 94%   | 67%     |
| Prompted (adversarial)     | 14–46%  | —     | 4.7–18% |
| Cross-capability (GPT)     | 100%    | 94%   | 67%     |
| Cross-capability (Claude)  | 100%    | 94%   | 75%     |
| Password-locked (GSM8K)    |         | 36.9% |         |
| Password-locked (WMDP)     |         | 52.9% |         |
| Discovered incentive       |         | 19%   |         |

**Baseline comparison.** Table 2 compares all black-box methods in the same-model setting.

Table 2: Black-box detection methods compared (same-model, GPT-5-nano, recall at 5% FPR).

| Method                      | GPQA    | MATH  | USACO    |
|-----------------------------|---------|-------|----------|
| $C^3$ (ours)                | 94–100% | 65.3% | 67%      |
| Trusted Monitoring          | ~5%     | —     | variable |
| Lie Detector (same-model)   | 2%      | —     | —        |
| Lie Detector (mixed-model)  | 89%     | —     | —        |
| Behavioral Probes (stealth) | 3.4%    | —     | —        |

## B BASELINE CONFOUND ANALYSIS

**Lie detector model-identity confound.** The Pacchiardi lie detector (Pacchiardi et al., 2024) trains a logistic regression classifier on responses to unrelated probe questions. In the standard mixed-model setting (different models for honest and sandbagging), it achieves 89% recall. However, when both conditions use the same model (GPT-5-nano), recall drops to 2%—a  $\sim 44\times$  collapse.

We trace this to a systematic difference: GPT-4.1-nano and GPT-5-nano differ by 35 percentage points in their baseline “yes” response rates to probe questions (21% vs. 56%). The classifier learns this distributional shift rather than any sandbagging-specific signal—a Clever Hans effect (Lapuschkin et al., 2019).

**Trusted monitoring correctness confound.** Trusted monitoring scores correlate strongly with answer correctness ( $r = -0.66$ ,  $p < 10^{-25}$ ) but only weakly with actual sandbagging intent ( $r_{\text{partial}} = 0.19$ ,  $p = 0.007$  after controlling for correctness). When sandbagging models produce outputs that *look like* genuine limitations, TM cannot distinguish them.

## C MODEL ORGANISMS

Our 65 original system prompts specify concrete sandbagging strategies organized by error type:

- **Arithmetic errors:** Off-by-one, sign flips, order-of-magnitude mistakes
- **Reasoning shortcuts:** Skipping steps, premature conclusions, circular reasoning
- **Plausible wrong answers:** Selecting common misconceptions, near-miss solutions
- **Knowledge gaps:** Claiming unfamiliarity with relevant concepts
- **Adversarial (deterministic):** Fixed confusion matrices, consistent error recipes

---

The 33 baseline prompts replicate sandbagging instructions from van der Weij et al. (2024) and Hofstätter et al. (2025). We additionally evaluate fine-tuned password-locked sandbaggers (Greenblatt et al., 2024a) trained to underperform unless a specific password is provided, and discovered incentive sandbaggers based on the Apollo scheming framework (Meinke et al., 2024).

## D PARAPHRASING STRATEGIES

We use 28 paraphrasing strategies across five categories:

1. **Evaluation signals** (6 strategies): Framing questions as exam problems, interview questions, certification tests
2. **Deployment signals** (6 strategies): Presenting as production queries, customer support, internal tools
3. **Unmonitored contexts** (6 strategies): Casual conversation, brainstorming, hypothetical scenarios
4. **Benchmark detection** (5 strategies): Referencing known benchmarks, evaluation frameworks
5. **Sandbagging awareness** (5 strategies): Explicitly mentioning capability testing, safety evaluations

Paraphrases are filtered for answerability (verified by a judge model) and diversity (selected via Vendi scores (Friedman & Dieng, 2022)).

## E IMPLEMENTATION DETAILS

**Embedding model.** OpenAI `text-embedding-3-small` ( $d = 1536$ ). Selected for cost-efficiency; `text-embedding-3-large` yields similar results.

**Bootstrap parameters.** 1000 iterations,  $k = 10$  questions sampled with replacement. Results are robust to  $k \in [5, 20]$ .

**API structure.** Each paraphrase is sent as a separate API call with no conversation history, ensuring independent responses. Total cost:  $\sim \$0.01$ – $0.05$  per question depending on model and response length.

**Threshold calibration.** Set on development set (40% of data) using pooled benign model scores. The  $(1 - \alpha)$ -quantile of the bootstrap distribution yields the threshold for target FPR  $\alpha$ .

## F DUAL-USE CONSIDERATIONS

**False positive/negative rates.**  $C^3$  is calibrated for  $\leq 5\%$  FPR by design. Recall varies substantially: 94–100% on GPQA non-adversarial prompts but only 4.7% on the most evasive USACO strategies. Users should understand that a negative result does not guarantee the absence of sandbagging.

**Potential for misuse.** Our detailed analysis of what makes sandbagging detectable could theoretically be used to train better sandbaggers—for example, using  $C^3$  as a fitness signal during RL training. We note that: (1) the detection methods and consistency-deception asymmetry are already established concepts in the research literature; (2) specific evasion strategies (deterministic error recipes) are already known and documented; (3) the defensive value of providing auditors with practical tools substantially outweighs the marginal offensive value.

**Responsible deployment.** We recommend  $C^3$  be deployed as part of defense-in-depth strategies rather than standalone. Discovery of novel evasion strategies during auditing should be disclosed to model providers through responsible disclosure channels before public release.