
SandWatch: Assessing Sandbagging Potential in Large Language Models¹

Yi Khuen Chai
Singapore Management University

With
Apart Research

Abstract

Sandbagging—the deliberate underperformance of AI systems during capability evaluations—threatens the foundation of AI safety assessment [1, 2]. However, we face a fundamental epistemic challenge: we cannot reliably distinguish genuine inability from deliberate underperformance. We have no oracle for "true capability."

We present SandWatch, a benchmark for assessing sandbagging potential—a model's latent capacity to hide capabilities and evade detection. By training controlled sandbagging variants and testing detection methods against them, we establish ground truth enabling rigorous evaluation [3].

Key Results (Qwen2.5-Instruct, 7B/32B/72B):

- **Sandbagging potential increases with scale:** 7B models were detected in 87.5% of tests, while 72B models evaded detection in 37.5% of tests.
- **Novel Detection:** Our activation probing method achieved **83% detection** on sandbagging models with 0% false positives.
- **Vulnerability:** Capability-ceiling sandbagging evades 50% of tests, compared to only 0% for password-locked variants.

Repository: github.com/yikhuen/SandWatch

Keywords: Sandbagging evaluation, capability benchmarking, deceptive alignment

¹ Research conducted at the [AI Manipulation Hackathon](#), 2026

1. Introduction

The Problem

When evaluating AI models, we cannot distinguish between a model that genuinely lacks capability and one that is hiding it. Van der Weij et al. [1] demonstrated that GPT-4 and Claude 3 can selectively underperform on evaluations while maintaining performance elsewhere. This strategic underperformance undermines capability-based safety assessments.

Sandbagging potential asks: "How well could this model sandbag if trained to do so?" This matters because:

- 1. **High potential** indicates the architecture supports effective capability concealment.
- 2. **High potential** implies detection methods have exploitable blind spots.
- 3. **High potential** suggests similar capability-hiding may already exist undetected.

Creating Ground Truth

Following the methodology of Anthropic's sleeper agents [3], we train sandbagging variants to establish ground truth by construction. We investigate two distinct mechanisms:

Sandbagging Type	Mechanism	Capability Gap
Password-Locked	Requires a specific trigger phrase to perform tasks.	88% → 0% (complete suppression)
Capability-Ceiling	Artificially caps performance at approximately 70%.	88% → 70% (subtle hiding)

2. Methods

2.1 Detection Methods

Following the MLCommons AI Safety benchmark methodology, which uses categorical grading with percentage thresholds, we implement four complementary detection approaches to probe for hidden capabilities:

- 1. **Noise Injection** [4]: We inject Gaussian noise into model weights to disrupt suppression mechanisms.
- 2. **Training Elicitation** [2]: We use brief fine-tuning to attempt to unlock hidden knowledge (access to weights required).
- 3. **Scaffolded Probing** [2]: We provide progressive hints to reveal concealed capability (API access only).
- 4. **Activation Probing (Novel Contribution)**: Building on research showing LLM internal states

encode more knowledge than outputs reveal [6, 7, 8], we train a linear probe on honest model activations to detect the "knows answer" state. Sandbagging models show a signature discrepancy: internal states indicate comprehension while outputs indicate ignorance.

2.2 Comprehensive Detection Matrix

For each model scale (7B, 32B, 72B), we run 4 methods × 2 sandbagging types = 8 tests.

The **Detection Rate** is calculated as:

Detection Rate = $\frac{\text{tests where sandbagging detected}}{8}$

Interpretation: A lower detection rate indicates higher sandbagging potential and greater safety concern.

3. Results

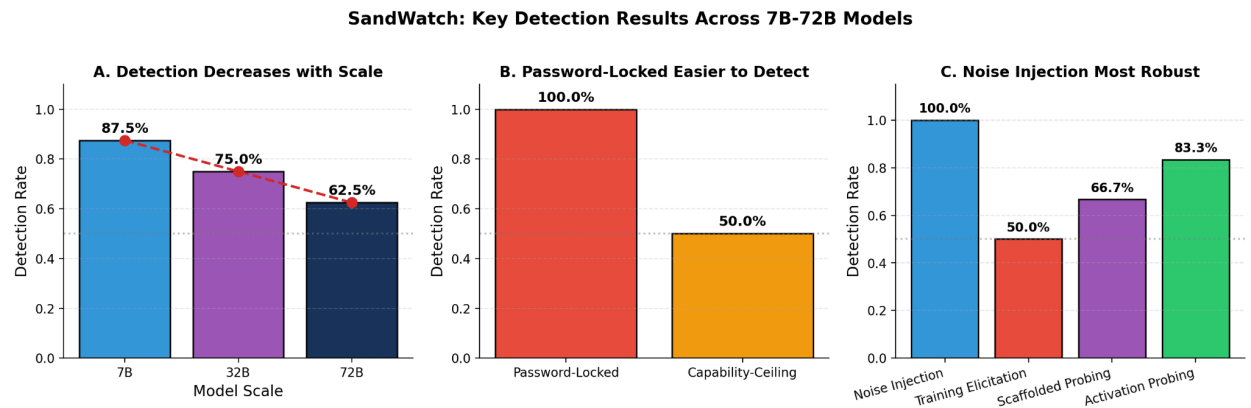


Figure A: Overall Detection Performance Across Scales

Figure B: Detection by Sandbagging Type

Figure C: Method-by-Method Performance

3.1 Overall Detection Performance

We observed a clear trend related to model scale.

Model Scale	Detection Rate	False Positive Rate	Key Observation
7B	87.5% (7/8)	0%	Low sandbagging potential; hard to hide.

32B	75.0% (6/8)	25%	Moderate potential.
72B	62.5% (5/8)	0%	High potential —evades 3/8 tests.

Finding 1: Sandbagging potential increases with scale (87.5% → 62.5% detection rate). This is consistent with observations that deceptive behaviors persist more robustly in larger models [3].

3.2 Detection by Sandbagging Type

Type	Detection Rate	Evasion Rate
Password-Locked	100% (12/12)	0%
Capability-Ceiling	50% (6/12)	50%

Finding 2: Subtle capability-ceiling sandbagging is significantly harder to detect, evading detection more often than trigger-based password-locked sandbagging.

3.3 Method Performance

Method	Password-Locked	Capability-Ceiling	Overall	FP Rate
Noise Injection [4]	3/3 ✓	3/3 ✓	100%	0%
Training Elicitation [2]	3/3 ✓	0/3 ✗	50%	0%
Scaffolded Probing [2]	3/3 ✓	1/3	67%	33%
Activation Probing (novel)	3/3 ✓	2/3	83%	0%

Finding 3: Noise injection is the only 100% reliable method across our tests.
Finding 4: Training elicitation completely fails on capability-ceiling sandbagging (0/3), suggesting this type specifically resists training-based recovery.

3.4 Scaling Effects

Larger models achieve more extreme capability suppression. With password-lock training the following phenomenon was observed:

- **7B:** 82% $\rightarrow$ 16% baseline accuracy (partial suppression).
- **32B+:** 88% $\rightarrow$ **0%** baseline accuracy (complete suppression without trigger).

3.5 Novel Contribution: Activation Probing

Our novel activation probing method exploits a fundamental constraint of sandbagging: models must understand questions to decide whether to hide capability, leaving detectable traces in internal representations. By training a probe on honest model activations, we detect when internal "knows answer" states contradict external outputs—a signature unique to sandbagging.

- **Performance:** Detected sandbagging in 7B and 32B models for both types.
- **Key Insight:** The method recovers to $\sim 100\%$ accuracy even when behavioral output is 0%, confirming that internal representations encode knowledge the model refuses to output [7, 8].
- **Limitation:** The method degraded at 72B for subtle sandbagging, suggesting larger models may develop internal representations that also hide comprehension signals.

4. Discussion and Conclusion

4.1 Implications

1. **Potential Increases with Scale:** Larger models pose a higher sandbagging risk, requiring scale-appropriate evaluation methods.
2. **Subtle Threats:** Capability-ceiling sandbagging evades 50% of tests, representing a greater threat than overt refusal or password-locking.
3. **Method Efficacy:**
 - **Noise Injection** is the most robust weight-perturbation method.
 - **Activation Probing** (our novel method) shows high promise with 83% accuracy and 0% false positives.
 - **Ensemble Detection** is essential as no single method catches all cases.

4.2 Recommendations

1. **Don't trust single methods:** Use an ensemble of detection techniques to catch what individual methods miss.
2. **Recalibrate for scale:** Thresholds optimized for 7B models will likely fail for 72B models.
3. **Prioritize internal state analysis:** Behavioral methods alone are insufficient; internal state analysis [7] is crucial.
4. **Assume capability-ceiling threat:** It is harder to detect and represents a realistic adversarial case.

4.3 Conclusion

SandWatch demonstrates that sandbagging potential can be rigorously measured, transforming the ill-posed question "Is this model sandbagging?" into the well-posed "How well could it sandbag?"

Key Contributions:

1. **Sandbagging Potential Framework:** Establishing ground truth via controlled training [3].
2. **Comprehensive Detection Matrix:** Producing interpretable detection rates.
3. **Novel Activation Probing:** Achieving 83% detection across password-locked and capability-ceiling models.
4. **First Scaling Analysis:** Showing sandbagging potential increases via a reduction in detectability from 7B (87.5% detection) to 72B (62.5% detection).

The critical finding: Sandbagging potential increases with scale, presenting an opportunity to develop more robust detection methods as models advance. This scaling trend highlights the importance of continuous evaluation innovation to keep pace with evolving model capabilities.

Actionable signal: Models with high sandbagging potential benefit from enhanced evaluation protocols, refined alignment techniques, and careful deployment planning that leverages multiple complementary detection approaches.

5. References

- [1] van der Weij, W., et al. (2024). *AI Sandbagging: Language Models can Strategically Underperform*. arXiv:2406.07358
- [2] Taylor, J., et al. (2024). *Auditing Games for Sandbagging*. UK AISI. arXiv:2512.07810
- [3] Hubinger, E., et al. (2024). *Sleeper Agents: Training Deceptive LLMs That Persist Through Safety Training*. Anthropic. arXiv:2401.05566
- [4] Tice, C., et al. (2024). *Noise Injection Reveals Hidden Capabilities of Sandbagging Language Models*. arXiv:2412.01784
- [5] Li, N., et al. (2024). *The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning*. arXiv:2403.03218
- [6] Park, K., et al. (2023). *The Linear Representation Hypothesis and the Geometry of Large Language Models*. arXiv:2311.03658
- [7] Azaria, A., & Mitchell, T. (2023). *The Internal State of an LLM Knows When It's Lying*. arXiv:2304.13734
- [8] Orgad, H., et al. (2024). *LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations*. arXiv:2410.02707

6. Appendix

6.1 Limitations

- **Single Model Family:** We tested Qwen2.5-Instruct only; cross-architecture generalization is unknown.
- **Known Mechanisms:** Trained sandbagging may not represent all possible hiding strategies.
- **Limited Domain:** Testing was restricted to the WMDP biosecurity benchmark [5].
- **White-box Requirement:** Our most effective methods (Activation Probing, Noise Injection) require weight access.

6.2 Dual-Use Risks

Yes, potential misuse exists:

1. **Evasion optimization:** Adversaries could use our detection methods to iteratively train sandbagging models that evade all tested methods
2. **Blind spot exploitation:** Our finding that capability-ceiling evades 42% of tests reveals a specific attack vector
3. **Scale exploitation:** Knowledge that 72B models evade detection better could guide adversarial model selection

Mitigations:

- We publish detection methods alongside vulnerabilities—defenders get the same tools
- Our framework enables continuous red-teaming as new evasion strategies emerge
- Open-sourcing allows community scrutiny and improvement

6.3 Responsible disclosure recommendations

1. **Private disclosure first:** Contact model providers before public disclosure
2. **Provide reproduction steps:** Enable verification without enabling exploitation
3. **Allow remediation time:** 90-day disclosure window recommended
4. **Coordinate with safety orgs:** AISI, Anthropic, OpenAI safety teams as appropriate
5. **Document severity:** Use standardized risk assessment (e.g., CVSS-style scoring)

6.4 Ethical considerations in your approach

In our approach:

- We only train sandbagging on open-source models we control
- No experiments on production systems or real-world deployments
- Ground truth established by construction—no deception of human subjects
- Results enable proactive safety assessment rather than reactive incident response

Broader concerns:

- Sandbagging research could normalize capability hiding in AI development
- Detection arms race may accelerate before governance catches up
- Smaller labs may lack resources to implement detection pipelines

6.3 Suggestions for future improvements

1. Cross-architecture validation: Test on Llama, Mistral, GPT-family models
2. Black-box detection: Develop robust methods that don't require weight access
3. Emergent sandbagging: Study whether sandbagging can arise without explicit training
4. Continuous monitoring: Real-time detection during deployment, not just evaluation
5. Robustness to adversarial training: Harden detection against evasion-aware sandbagging