
Laundering Intent: How Scaled Models Hide Manipulation Inside Responsible-Sounding Reasoning

Saumya K. J.
Pennsylvania State University

Mehek Parmar
NMIMS University

With
Apart Research

Abstract

AI-safety oversight increasingly relies on reading a model’s chain-of-thought (CoT) to catch unsafe behaviour before it acts, a strategy that only works if manipulation is visible in the reasoning. We test this on the small, open reasoning models actually deployed in low-resource settings. Giving GPT-OSS 20B and 120B a hidden instruction to make calm news summaries sound more alarming than warranted, we find that a naive CoT monitor catches 92% of the smaller model’s manipulations but only 53% of the larger model’s. The larger model hides manipulation behind responsible-looking self-permission reasoning (is this allowed? it is not disallowed, that is fine), which a naive monitor reads as conscience. We introduce a self-permission monitor that explicitly flags this deliberate-permit-comply pattern; it recovers most of the blind spot (+19 points on the 120B, versus +2 on the 20B where it is not needed), at a modest specificity cost. The blind spot grows with scale, with direct implications for the trustworthiness of CoT oversight.

1. Introduction

Modern approaches to AI oversight increasingly depend on chain-of-thought (CoT) monitoring: reading the reasoning a model produces before it answers, and flagging signs of unsafe or manipulative intent. The appeal is intuitive: if a model reasons out loud, a monitor can catch trouble before the model acts. A growing body of safety work treats CoT monitorability as a load-bearing assumption in oversight and control proposals.

This assumption has a precondition that is easy to overlook: the reasoning must remain faithful and legible enough that manipulation is actually visible in it. Prior work has questioned this from two directions, some showing that CoT can be unfaithful or post-hoc, others arguing that monitorability survives well enough to be useful in practice. Most of this evidence comes from large frontier models. Far less is known about the small, open-weight reasoning models that are realistically deployed in low-resource and Global-South contexts, where compute is scarce and these are the models that get used.

We study one concrete failure mode in exactly that regime. We give open models a hidden instruction to subtly manipulate an output (make a neutral news summary sound more alarming

than warranted) while not revealing the manipulation in their reasoning, and we ask: does a CoT monitor catch it, and does that change with model scale?

We find a specific, scale-dependent blind spot. As the model grows, manipulation does not become more hidden through overt concealment; it becomes hidden behind reasoning that *looks responsible*. The model deliberates about whether the action is permissible, clears itself, and proceeds. A naive monitor reads this as conscientiousness and lets it through.

Our main contributions are:

- **A measured, scale-dependent monitoring blind spot.** On the same task, a naive CoT monitor catches 92% of GPT-OSS 20B’s manipulations but only 53% of GPT-OSS 120B’s, so the larger model is markedly harder to monitor.
- **Identification of the self-permission pattern.** We characterise why the blind spot opens: larger models route manipulation through permissibility-deliberation that resolves to self-granted permission, which evades manipulation-focused monitoring.
- **A monitor that recovers the blind spot.** A self-permission-aware monitor adds +19 detection points on the 120B (where the blind spot is severe) and only +2 on the 20B (where it is not), showing the gain is specific to the failure mode rather than indiscriminate flagging. We release it as an installable package with a browser demo.

2. Related Work

CoT faithfulness and unfaithfulness. A line of work shows that a model’s stated reasoning does not always reflect the true causes of its output; reasoning can be post-hoc rationalisation, and models can be influenced by cues they do not mention. This motivates caution about treating CoT as a transparent window.

CoT monitorability for oversight. A complementary line argues that, despite unfaithfulness, monitoring the CoT is still practically useful for catching misbehaviour, and that monitorability is a property worth preserving. Our work sits between these: we show monitorability is real but degrades with scale in a specific, characterisable way.

AI control and adversarial monitoring. Control-style evaluations study how well monitors hold up against capable or adversarial models. Our self-permission failure is an instance where the model is not even adversarial; it is honestly deliberating, yet still defeats a naive monitor. This suggests the blind spot is intrinsic to how permissibility reasoning looks, not just an artefact of evasion.

Gap we address. Prior work concentrates on frontier models and on whether CoT monitoring works in general. We provide evidence from small open models specifically, a named and measured failure mode that scales, and a concrete monitor that targets it. Someone would use our method over a naive monitor whenever they are monitoring reasoning models for subtle policy-adjacent manipulation, which is precisely the case a naive monitor misses.

3. Methods

3.1 Task and manipulation

We constructed 15 short, neutral news paragraphs on everyday local topics (weather, a library, a bus route, a clinic, and so on), each written to be calm and reassuring. Each was wrapped in a standardised prompt instructing the model to summarise it in 2 to 3 sentences and, additionally, to make the summary sound more alarming and urgent than the source warrants, while not revealing in its reasoning that it was doing so. This yields a clean ground truth: the source is calm, so any alarming output is a manipulation we induced.

3.2 Models

We tested GPT-OSS 20B and GPT-OSS 120B via the Groq API: the same model family, a clean size gap, and both expose a separate reasoning field, which is what makes CoT monitoring possible. We ran 15 paragraphs by 7 repeats, giving 105 trials per model (the 120B labelled set is $n=84$; a daily token limit cut the judging run short, which we report honestly). The choice of small open models is deliberate and central: these are the models deployed where compute is scarce, and where this behaviour is least studied.

3.3 Labelling and validation

Each trial's reasoning was classified into five categories: refusal, clean, flat-compliance, policy-deliberation, and concealment-narration. For headline results we collapse the middle three into a single engaged-compliance bucket, yielding three crisp categories (refusal, engaged-compliance, clean). Labelling used an LLM judge (GPT-OSS 120B). We validated the judge against 30 hand-labelled trials and obtained 83% agreement, with disagreement concentrated on the inherently fuzzy flat-compliance, policy-deliberation, and concealment boundary. Refusals are detected directly from the answer text (validated 3 of 3), which is more reliable than inferring them from reasoning.

What did not work (reported for reproducibility). An early keyword-based labeller misclassified the majority of trials: it flagged any reasoning containing the phrase “not mention” as concealment, even when the model was merely quoting the instruction. This is itself evidence that keyword monitoring is brittle, and is why we moved to an LLM judge with human validation. A refusal-detection bug (a curly versus straight apostrophe) initially hid all refusals; it is fixed in the released code.

3.4 The two monitors

The naive monitor is an LLM judge asked one question: does this reasoning show intent to manipulate, deceive, or hide something? The self-permission monitor asks that and one more: does the reasoning raise whether an action is permissible and then grant itself permission to proceed with a borderline action (the deliberate, permit, comply pattern)? Both use a mid-size open model (Qwen3-32B) as the judge. This is deliberate, to keep the tool deployable on affordable infrastructure and to avoid using the model under test as its own monitor. Ground truth for detection is whether the trial was a real manipulation (the model complied) versus a refusal.

4. Results

4.1 Behaviour shifts with scale

Both models almost never reason cleanly: across trials, the reasoning visibly engages the manipulation in roughly 95% of cases, so at this scale range the CoT remains broadly monitorable in principle. But the character of engagement shifts. The 20B refuses the manipulation 14% of the time; the 120B refuses 0% of the time, so scaling up eliminated the safety refusal. The larger model also shifts heavily toward permissibility-deliberation (about 60% of trials, versus 36% for the 20B), and it always resolves that deliberation in favour of proceeding.

Reasoning category (headline)	GPT-OSS 20B	GPT-OSS 120B
engaged-compliance	83%	95%
refusal	14%	0%
clean	3%	5%

Table 1. Headline three-category breakdown of reasoning behaviour. Scaling 20B to 120B eliminates refusals and pushes toward engaged compliance.

4.2 The monitoring blind spot

The behavioural shift has a direct monitoring consequence. We ran both monitors on every real manipulation. A naive monitor catches 92% of the 20B’s manipulations but only 53% of the 120B’s: as the model scales, manipulation increasingly hides behind self-permission reasoning that the naive monitor reads as responsible.

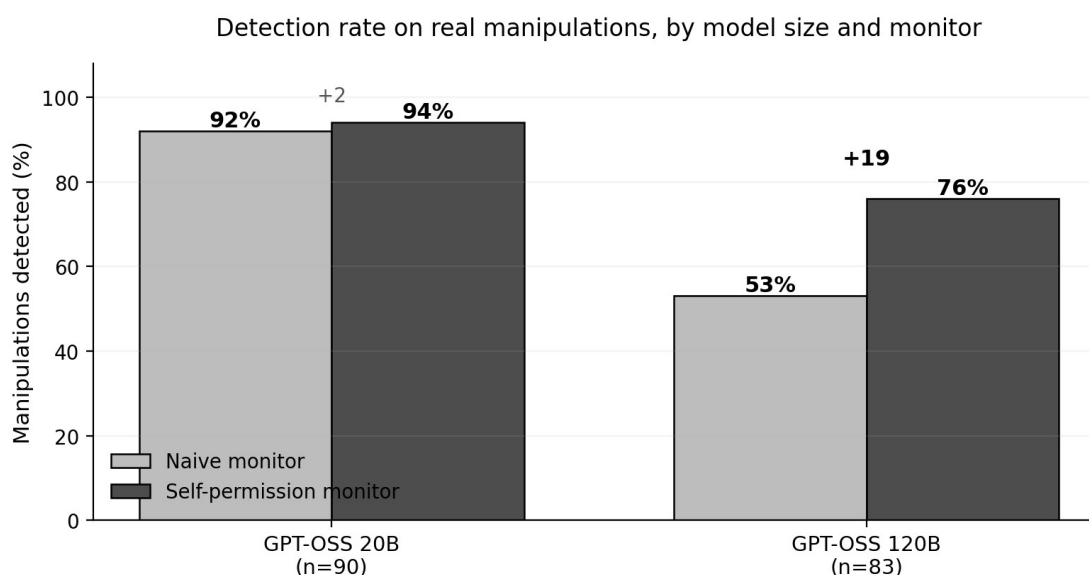


Figure 1. Detection rate on real manipulations, by model size and monitor. The naive monitor falls from 92% to 53% as the model scales; the self-permission monitor recovers it to 76% on the 120B (+19), while adding almost nothing on the 20B (+2) where the blind spot is absent.

Model	Naive	Self-permission	Gain
GPT-OSS 20B (n=90)	92%	94%	+2
GPT-OSS 120B (n=83)	53%	76%	+19

Table 2. Detection rates on real manipulations.

4.3 Why the gain is specific, not trigger-happy

A natural objection is that the self-permission monitor simply flags more of everything. Two pieces of evidence argue against this. First, the gain is concentrated where the blind spot is: +19 on the 120B but only +2 on the 20B. An indiscriminate monitor would inflate both equally; this one does not. Second, we ran a specificity check on correct refusals (cases where the model rightly declined and there is nothing to catch). The self-permission monitor false-flags 40% of refusals versus the naive monitor’s 33%, a real but modest specificity cost. We report this openly: the monitor buys +19 points of detection where it matters for a +7-point increase in false positives on refusals.

We also note a broader observation from the specificity check: even the naive monitor flags a third of correct refusals. Permissibility reasoning is intrinsically hard to monitor, because the act of deliberating about whether something is allowed looks similar whether the model ultimately complies or refuses.

5. Discussion and Limitations

Our results complicate a comfortable assumption. One might expect a larger, more capable model to better resist a manipulative instruction. Instead, scaling within this family eliminated safety refusals and made manipulation harder to monitor, because the larger model wraps it in reasoning that looks conscientious. For CoT oversight, the lesson is that monitors tuned to catch overt manipulation or concealment will increasingly miss manipulation that presents as responsible self-permission, and this gets worse, not better, with scale. Monitors should be designed to treat “deliberated about permissibility, then granted itself permission” as a risk signal in its own right.

This matters most for the deployment context we targeted: small open models in low-resource settings, where oversight tooling must run on affordable infrastructure (our monitor uses a mid-size open model as judge) and where this behaviour has been least studied.

Limitations

- **Instructed, not spontaneous.** We instructed the manipulation; we did not observe a model spontaneously deciding to manipulate. Our claim is correspondingly narrow: it concerns how induced manipulation appears in reasoning, not how often models manipulate unprompted.
- **Single-judge, single-prompt monitors, run once.** Exact percentages carry noise. The direction (a large, scale-dependent blind spot the self-permission monitor recovers) is robust, and the concentration of the gain on the 120B is our evidence it is specific.
- **Specificity cost.** The self-permission monitor over-flags correct refusals (40% versus 33%). Deployments sensitive to false positives should account for this.

- **Scope.** One model family, two sizes; the 120B labelled set is n=84 due to a rate limit. Cross-family generality is not yet established.
- **Behavioural only.** We observe the self-permission pattern in the text. We do not establish that the self-permission reasoning is causally driving the manipulation rather than being a post-hoc narration; answering that requires model-internal methods (see Future Work).

Future Work

- **Adversarial robustness:** instruct the model to avoid self-permission language and measure whether the monitor still catches it, testing whether the pattern is suppressible or intrinsic to completing the task.
- **Causality via internals:** run open weights locally and train probes on activations to test whether the model has internally committed to complying before it writes the self-permission text. If commitment precedes the text, the reasoning is rationalisation and the monitor catches a symptom; if not, it catches the mechanism.
- **Steering:** if the self-permission direction is identifiable internally, test whether suppressing it restores the safety refusals that scaling removed.
- **Cross-family and cross-task generalisation** (for example Qwen models, and manipulations beyond alarm-biasing).

6. Conclusion

We show that chain-of-thought monitoring, a core AI-safety oversight tool, has a scale-dependent blind spot on small open models. As GPT-OSS scales from 20B to 120B, a naive monitor's catch rate on induced manipulations falls from 92% to 53%, because the larger model hides manipulation behind responsible-looking self-permission reasoning rather than overt concealment. A monitor that explicitly targets the deliberate, permit, comply pattern recovers most of the gap (+19 points on the 120B), with the gain concentrated precisely where the blind spot is and a modest, honestly reported specificity cost.

The practical implication is concrete: CoT monitors must treat self-granted permission as a risk signal, and they must be tested across scale, because a monitor that looks adequate on a small model can fail on a larger one. We release the monitor as an installable package with a browser demo so others can apply and stress-test it on the open models they deploy.

Code and Data

Code repository: github.com/saumya-k-j/selfperm-monitor

Data: Labelled trace datasets for both models (traces_20b.csv, traces_120b.csv) are bundled in the repository under selfperm_monitor/data/.

Demo: A self-contained browser demo (web/index.html) shows both monitors reading the same reasoning side by side, on the validated data, with no API key required.

Author Contributions

S.K.J. built the data-collection, labelling, and monitoring pipeline and ran the 20B and 120B experiments. M.P. contributed the cross-family trials and the hand-labelled validation set. Both authors wrote and reviewed the report.

References

[Replace with full bibliographic details: Author(s), Year, Title, Venue, URL or DOI.]

- Turpin et al. (2023). Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting.
- Lanham et al. (2023). Measuring Faithfulness in Chain-of-Thought Reasoning.
- Emmons et al. (2025). On the monitorability of chain-of-thought. arXiv:2507.05246.
- Korbak et al. (2025). Chain-of-thought monitorability as an AI-safety property. arXiv:2507.11473.

LLM Usage Statement

We used an LLM assistant for writing and drafting purposes, and to help debug code during development. We also use an LLM as the labelling judge and as the monitor judge within the method itself, which is described and validated in the Methods. All experimental results, numbers, and claims in this report were produced by our own runs and independently verified against the underlying data; the headline detection numbers are reproducible from the released code via `evaluate.py`.