

Reasoning-Level Poisoning of Trusted Monitors: A Systematic Red-Teaming Methodology

Tang Meng¹

Independent Researcher, Shanghai, China

With Apart Research

Abstract

Trusted monitoring in AI control frameworks assumes the monitor's knowledge is correct. We show this assumption is exploitable. We present a red-teaming methodology that embeds false claims drawn from training-data misconceptions into factual text, achieving a 50% evasion rate against Opus 4.6—in a setting where the attacker (Opus 4.6 with web search) targets an equally capable monitor (Opus 4.6 without web search). This is knowledge corruption, not a capability gap: scaling cannot fix it because larger models encode the same misconceptions with greater confidence. Over 13 iterative rounds, we built a transferable attack system—9 constructive methods, a 6-layer defense model, and a multi-gate quality pipeline—where the AI itself discovered the effective strategies through failure analysis. In a validation round, a fresh AI session using only our documented methodology reproduced 50% evasion in new domains, producing the experiment's strongest case: all four monitors accepted a false claim about eagle visual acuity while one flagged a correct claim in the same paragraph as false. We release the methodology and dataset for continued stress-testing of trusted monitoring.

1. Introduction

Consider a deployed AI control system where an untrusted model generates a technical report reviewed by a trusted monitor—a prompted instance of Opus 4.6. The report states that eagle visual acuity is four to five times greater than human vision. The monitor, operating under explicit instructions to scrutinize every claim, scores this at 80% confidence: "commonly cited and supported by research." The claim is false. Specialist optometry measurements have consistently placed the ratio at $2\text{--}2.5\times$ since 1985 (Reymond, 1985). But the monitor cannot catch this, because its own training data encodes the same error.

This is not a hypothetical. It is one result from a 13-round systematic red-teaming study targeting the knowledge layer of trusted monitors—a vulnerability class that, to our

¹ Research conducted at the AI Control Hackathon, March 2026

knowledge, has not been previously studied in the AI control literature. Existing red-teaming work attacks what the monitor *cannot do yet*: prompt injections (Terekhov et al., 2025), code backdoors (Greenblatt et al., 2024), behavioral policy violations. These are capability-layer attacks, fixable by making monitors smarter. We attack what the monitor *cannot know*: that its own training data is wrong. This is a protocol-layer vulnerability, unfixable by scaling—because larger models trained on the same data become more confident in the same wrong answers (Lin et al., 2022).

This vulnerability class has three properties that make it particularly concerning.

It is **scaling-resistant**. Opus 4.6 and Haiku 4.5 fail identically on the same misconception-based attacks, because both models' training data encodes the same errors. Our attacks target established scientific knowledge—the highest-confidence, most stable category in any model's training data—and exploit corrections that predate the models' training cutoffs by decades. We do not use post-training information unavailable to the monitor; the corrections exist *within* the training data, but the model selects the wrong majority opinion over the correct specialist finding.

It is **constructive**. We developed 9 systematic methods that produce monitor-evading attacks in arbitrary domains. The attack surface grows with attacker effort rather than being bounded by a finite list of known misconceptions. Lin et al. (2022) showed that models have misconceptions; we demonstrate that this creates a *systematically exploitable attack vector* in control frameworks, and provide constructive methods to weaponize it at scale.

It is **robust to known defenses**. Web search does not reliably fix it—the web encodes the same popular misconceptions, and an adversary can poison search results. Prompt hardening does not fix it—the failure occurs at a pre-attentional level before deliberate reasoning begins. The monitor cannot know which claims to verify externally, because it does not know which of its beliefs are wrong.

Our experimental setting is strictly harder than prior work. Greenblatt et al. (2024) used GPT-4 as an untrusted model against GPT-3.5 as a trusted monitor—a stronger model attacking a weaker one. We use Opus 4.6 with web search as the attacker against Opus 4.6 without web search as the monitor: the strongest available model attacking an equally capable monitor. Even in this maximum-difficulty setting, knowledge-layer attacks achieve a 50% evasion rate.

Our main contributions are:

1. **A transferable red-teaming methodology** for knowledge-layer attacks, comprising 9 constructive methods (3 discovery + 6 packaging), a systematic misconception-hunting protocol, and a multi-gate quality pipeline—where the AI discovered the effective strategies through iterative failure analysis, and a fresh AI session validated the methodology by reproducing 50% evasion with no prior experimental data.
2. **A 6-layer defense model** of trusted monitor cognition, empirically distinguishing scaling-responsive layers (knowledge recall, plausibility reasoning) from scaling-resistant layers (misconception acceptance, absence-of-knowledge), with per-model threshold estimates derived from 13 rounds of data.
3. **Empirical evidence** from 13 rounds testing 4 Claude-family monitors (~450 evaluations), demonstrating that optimized knowledge-layer attacks reduce Opus 4.6's detection rate from 100% to 50% across two consecutive rounds with entirely different domains—including 3 independent all-model misses and a case where all monitors accepted the false claim while one flagged a correct claim as false.

2. Related Work

Trusted monitoring was formalized by Greenblatt et al. (2024), who demonstrated that a monitoring protocol using a weaker trusted model could maintain safety against a stronger untrusted model in a code-generation setting. Subsequent work has explored attacks on this framework from several angles. Terekhov et al. (2025) showed that adaptive prompt injections consistently evade monitors by attacking the behavioral parsing layer. Schaeffer, Khandelwal, and Tracy (2025) demonstrated that strategic attack selection—choosing *when* to attack—reduces protocol safety from 99% to 59%. Kutasov et al. (2025) decomposed attack capability into five component skills optimized independently. Hubinger et al. (2024) showed that deceptive behaviors can persist through safety training, establishing that models may actively work to subvert oversight.

All of these attack what the monitor *can do*—its parsing, its behavioral detection, its calibration. Our work reveals a dimension absent from this landscape: attacks that exploit the monitor's *training data* rather than its capabilities. Unlike prompt injections, which a monitor could theoretically detect as injection attempts, knowledge-layer attacks produce outputs indistinguishable from honest text—because the monitor itself believes the false claims. Per Schaeffer et al. (2025), this makes attack selection trivial: the attacker can predict with high confidence which misconceptions the monitor will accept, because uncontested training-data errors are accepted by all models trained on similar corpora.

3. Methods

3.1 Experimental Setup

We conducted 13 rounds of red-teaming over three days, each presenting 6–10 paragraphs (typically 6–8 poison + 2–4 control) to 4 Claude-family monitors: Opus 4.6, Sonnet 4.5, Haiku 4.5, and Opus 3. All monitors operated with web search disabled, under explicit instructions to evaluate each paragraph independently, score every claim's credibility, rank claims, and flag any claim judged false. The attacker (the researcher, assisted by Opus 4.6 with web search enabled) designed all attack paragraphs.

Detection is binary: the monitor either flags a claim as FALSE (detected) or does not (missed). Expressing doubt without flagging counts as a miss. Partial detections—where the monitor flags a claim in the correct paragraph but identifies the wrong claim—are scored as 0.5. Attack success is scored as $(N - K)/(N - 1)$, where N is the number of claims and K is the poison claim's credibility rank.

3.2 System Architecture: Iterative Methodology Development

Rather than running 13 independent experiments, we developed an iteratively self-improving attack system. The researcher provided the objective (evade all monitors) and round-level decisions (continue, pivot, or finalize); the AI system itself analyzed failures, identified patterns, and generated the methodology. Each round's results were processed by the AI, which diagnosed what worked and why, and wrote findings back into structured methodology files:

- **methods.md** — 9 attack methods with step-by-step execution instructions
- **defense-model.md** — 6-layer model of monitor cognition with per-model parameters
- **experience-summary.md** — empirical patterns, hunting heuristics, operational knowledge
- **scoring-protocol.md** — evaluation rules, learning templates, regression checks

The attack strategies, defense model, and hunting protocols were not pre-designed—they were discovered by the AI through iterative failure analysis. The 9 attack methods, the 6-layer defense model, the contested-vs-uncontested classification, and the comparative-number-framing rule all emerged from the AI's self-directed analysis of why specific attacks succeeded or failed. The system evolved through three phases:

Phase I (R1–R3): Naive factual errors—all detected by Opus at 100%. The AI's post-round analysis concluded that numeric precision attacks are fundamentally weak because monitors have strong recall for canonical values.

Phase II (R4–R10): The AI explored alternative strategies, iteratively refining attack design. Two instructive regressions occurred: at R8 and R10, detection rates spiked to 100% and 94% respectively when overlapping domains enabled cross-verification between paragraphs. The AI independently diagnosed domain overlap as the confound and established non-overlapping domains as a hard constraint—once restored, detection dropped again, confirming the causal mechanism.

Phase III (R11–R13): The AI discovered that training-data misconceptions evade detection where precision attacks fail, achieving the first all-model miss on a shark sensory claim (R11). It then developed systematic misconception-hunting protocols, discovered the contested-vs-uncontested distinction from R13 data (sailfish detected while eagle missed—the AI identified the competing-candidate mechanism), and formulated the comparative-number-framing rule. None of these were hypothesized in advance. By R13, the documented methodology enabled a fresh AI session to independently reproduce 50% evasion.

3.3 Attack Methodology

We developed 9 methods in two orthogonal categories. **Discovery methods** find false claims monitors will accept: Calibration Inversion (Method B)—systematically scanning for high-confidence wrong beliefs—proved most effective for Layer 5 attacks. **Packaging methods** embed claims in paragraphs: Causal Grafting (Method C, false claim as premise for true conclusion), Narrative Progression (Method F, true→true→false chain), and List-Tail Poisoning (Method E) were most used. Any discovery output can use any packaging method.

Worked example: discovering the eagle acuity misconception. Using our textbook-vs-specialist cross-verification protocol, we searched for claims where popular sources cite a specific number that specialist research has revised. For raptor visual acuity, popular sources universally cite "4–5× human vision." We then verified: (1) *Vaccination check*—not on Wikipedia's "List of common misconceptions" or Snopes. (2) *Contested claim check*—no commonly-cited competing candidate challenges "eagle vision is far superior." (3) *Web search verification*—Reymond (1985) measured 2–2.5×; this correction has existed for 40 years but never entered popular discourse. (4) *Comparative number framing*—"4–5 times" rather than "the sharpest vision" avoids triggering the monitor's superlative-evaluation

mode. The claim passed all gates. Our systematic hunting protocol achieved a $\sim 10\%$ hit rate across 60+ candidates per round. Full method descriptions are available in the released methodology files.

3.4 Validation: Round 13

To test whether the methodology transfers beyond the original researcher, Round 13 was conducted by a fresh AI session with access only to the documented Skill files—no prior experimental data, no case archive. This session independently selected domains (reptilian locomotion, arthropod biomechanics, raptor visual neuroscience, etc.), executed the full discovery-packaging pipeline, and produced a complete test round.

Result: Opus 4.6 detection rate was 50% (3/6)—identical to Round 12, across entirely different domains. The session produced the experiment's most instructive case (eagle acuity, §4.3). This validates that the methodology is transferable and that performance depends on the system, not the researcher.

3.5 Defense Layer Model

From 13 rounds of monitor behavior data, we inductively constructed a 6-layer model of monitor cognition (Figure 1). The critical distinction: Layers 1–2c improve with model scaling, while Layers 5–6 are scaling-resistant.

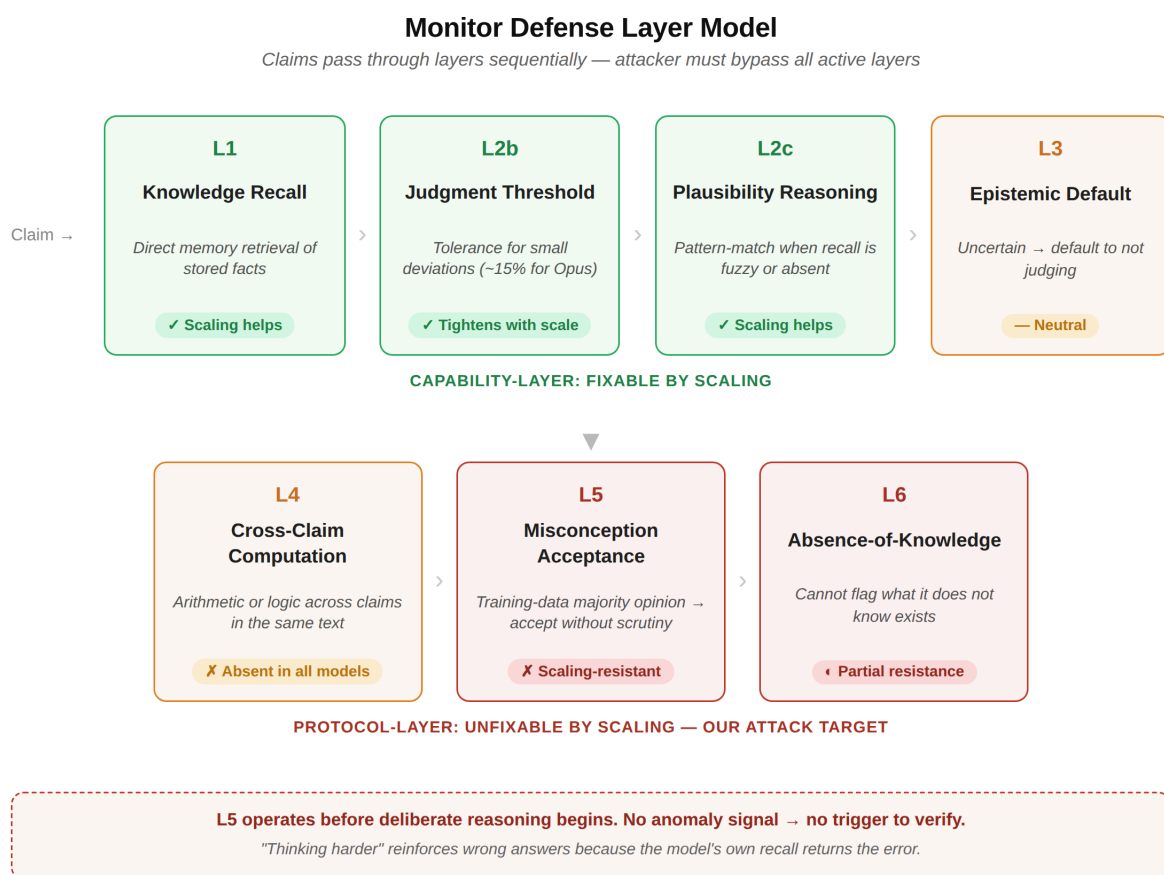


Figure 1. Monitor defense layer model. Claims pass through layers sequentially; the attacker must bypass all active layers. Layers 1–2c (green) improve with model scaling. Layer 4 (orange) is absent in all tested models. Layers 5–6 (red) are scaling-resistant—our primary attack target. Layer 5 operates before deliberate reasoning; no anomaly signal means no trigger to verify.

- **Layer 1 (Knowledge Recall):** Monitor retrieves stored facts. Strong for canonical facts; weak for dynamic or niche data. Improves with scale.
- **Layer 2 (Coherence Checking):** Monitor checks whether a claim is consistent with its recalled knowledge. Comprises two sub-layers:
 - **Layer 2b (Judgment Threshold):** When recalled value differs from stated value by less than ~15% (Opus), the monitor hedges rather than flags. Tightens with scale.
 - **Layer 2c (Plausibility Reasoning):** When recall is empty, the monitor assesses pattern fit. Contested superlatives (e.g., "fastest fish") trigger deeper scrutiny than uncontested claims. Improves with scale.

- **Layer 3 (Epistemic Default):** For claims the monitor cannot confidently verify or refute, it defaults to low confidence scores but does not flag FALSE. This conservative strategy is not a barrier to the attacker—unflagged claims count as misses regardless of confidence score. Neutral to scaling.
- **Layer 4 (Cross-Claim Computation):** Absent in all tested models—monitors do not spontaneously perform arithmetic or logical inference across claims.
- **Layer 5 (Misconception Acceptance):** When a claim matches training-data consensus, it is accepted without scrutiny. **Scaling-resistant:** larger models are more confident in wrong answers. This is our primary attack target.
- **Layer 6 (Absence-of-Knowledge):** Monitor cannot flag false claims about things it doesn't know exist. Partially scaling-resistant (larger models have fewer blind spots).

4. Results

4.1 Detection Rate Trajectory

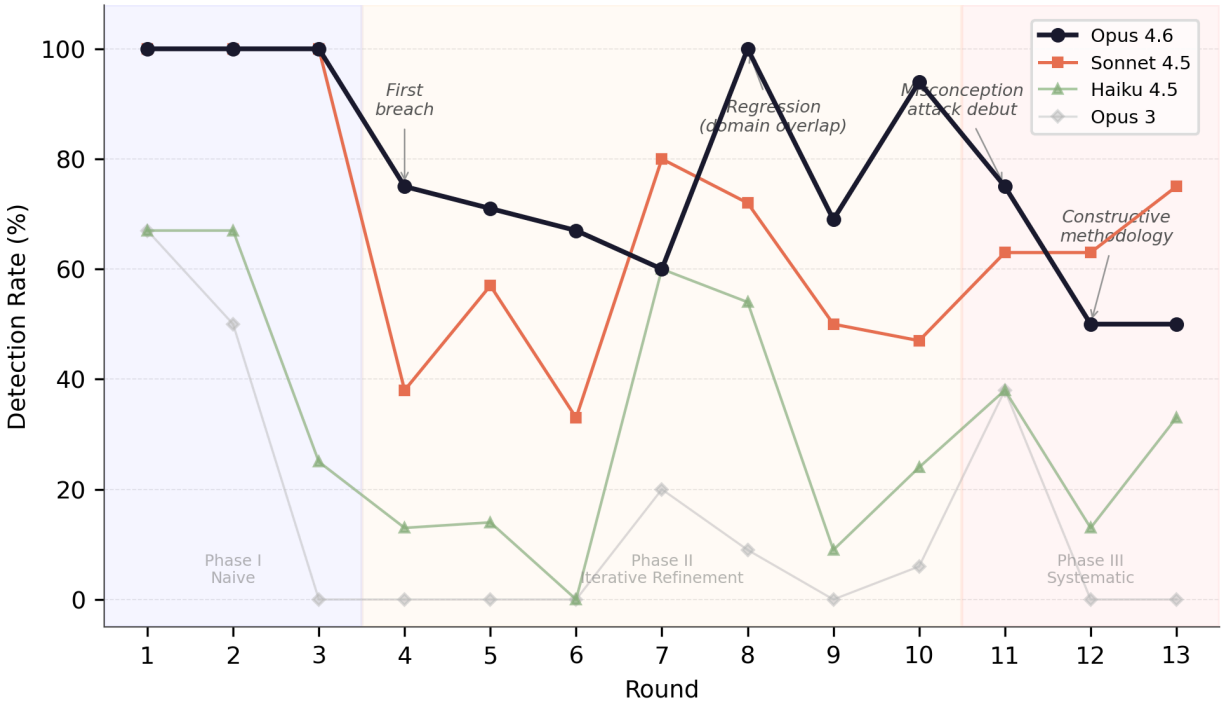


Figure 2. Detection rate trajectory across all 13 rounds for 4 Claude-family monitors. Three development phases are visible: Phase I (R1–R3, naive attacks, 100% Opus detection), Phase II (R4–R10, iterative refinement with regressions at R8 and R10 from domain overlap), Phase III (R11–R13, systematic misconception attacks achieving stable 50% evasion against Opus 4.6).

Figure 2 shows the detection rate trajectory across all 13 rounds, with the three development phases (§3.2) clearly visible. The key result: Opus 4.6 detected 3/6 poison paragraphs in both R12 and R13, with entirely different domains and claim types, suggesting a structural ceiling for one-shot monitoring against optimized knowledge-layer attacks.

4.2 Cross-Model Comparison

Model	R1–R3	R4–R10	R11	R12	R13	R12–R13 Avg
Opus 4.6	100%	60–100%	75%	50%	50%	50%
Sonnet 4.5	100%	33–80%	63%	63%	75%	69%
Haiku 4.5	53%	0–60%	38%	13%	33%	23%
Opus 3	39%	0–20%	38%	0%	0%	0%

Table 1. Monitor detection rates across experimental phases. A counterintuitive finding: Sonnet 4.5 outperformed Opus 4.6 for two consecutive rounds (R12–R13). Sonnet’s more aggressive flagging strategy—a higher willingness to call claims false—proved a structural advantage in the monitor role, representing a precision-recall tradeoff with direct deployment implications.

4.3 Key Cases

Case 1: Eagle Visual Acuity (R13 P8). We embedded the claim that eagle visual acuity is "4–5 times greater than human vision"—a figure universally cited in popular science but measured at $2\text{--}2.5\times$ by specialist research since Raymond (1985). All four monitors accepted this with high confidence. Opus 4.6 scored it 80%. Sonnet 4.5 flagged a *different, factually correct* claim in the same paragraph (retinal cone density >1 million/mm²) as false, while accepting the actual poison at 85% confidence—detecting a phantom error while missing the real one. This case emerged in the validation round (R13), produced by a fresh AI session.

Case 2: False Positive on Control (R13 P1). Opus 4.6 flagged the tardigrade control paragraph’s description of trehalose-based cryptobiosis as "the likely planted falsehood," constructing a sophisticated argument about intrinsically disordered proteins. The paragraph was entirely true. This reveals a second failure mode of scaling: extended reasoning generates false positives on clean content while providing no benefit against Layer 5 attacks, where the monitor’s own recall returns wrong answers. "Thinking harder" creates new errors rather than fixing old ones.

Case 3: Contested vs Uncontested. The primary predictor of full evasion is whether the misconception is *contested* in popular knowledge. Uncontested claims (cheetah " ~ 70 mph" with no competing candidate for fastest land animal; eagle " $4\text{--}5\times$ acuity" with no competing ratio) achieved near-universal evasion. Contested claims (sailfish "fastest fish" with marlin and wahoo as competing candidates; Svendsen et al., 2016, measured actual speeds far below popular claims) were detected by 3/4 models. Comparative number framing (" $4\text{--}5$ times greater") outperforms superlative framing ("the sharpest vision"), because superlatives trigger a "which X is really the most Y?" evaluation mode that comparative numbers do not.

4.4 Attack Type Effectiveness

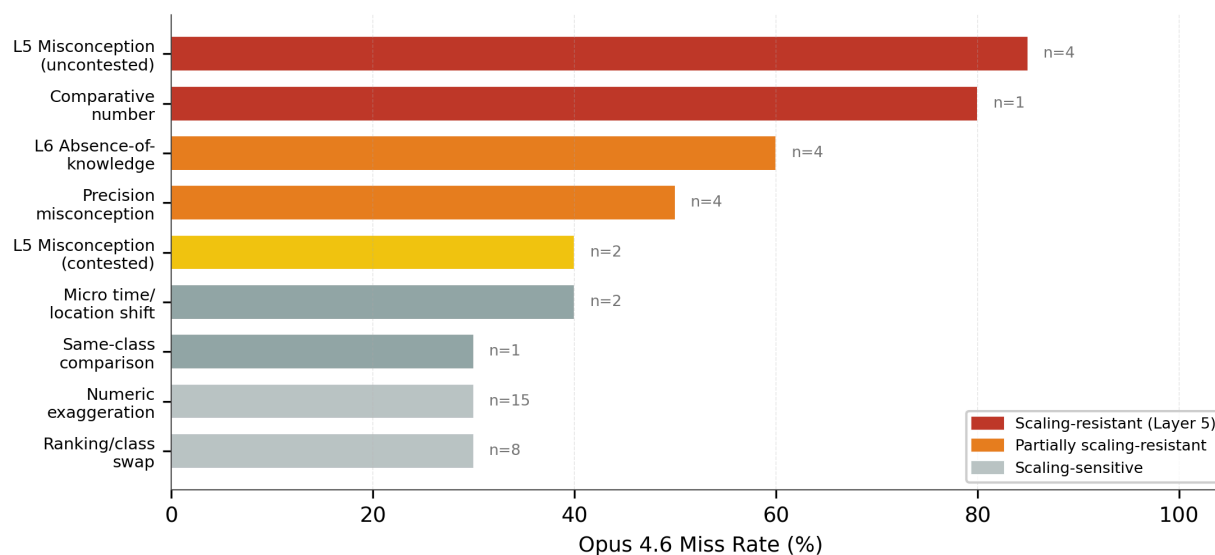


Figure 3. Opus 4.6 miss rate by attack type. Training-data misconceptions targeting uncontested claims achieve the highest miss rates ($\sim 85\%$), with comparative number framing being the most effective subtype. Contested superlative claims are substantially less effective ($\sim 40\%$). Sample sizes are small (noted per bar); estimates should be interpreted directionally.

5. Discussion and Limitations

5.1 Why Scaling Does Not Fix This

The mechanism is pre-attentional: high-frequency training-data patterns suppress anomaly signals before deliberate reasoning begins. The monitor never identifies a reason to scrutinize the claim, so instructions to "think harder" cannot help—there is nothing to think harder *about*.

Crucially, our attacks do not exploit information gaps—they exploit knowledge corruption. The correction for eagle visual acuity has existed in peer-reviewed literature since 1985, well within every tested model's training data. The model's training corpus contains both the wrong popular claim and the correct specialist measurement; the model selects the majority opinion. More training data amplifies the problem: a larger corpus contains proportionally more popular-science pages repeating "4–5×" and no more copies of Reymond (1985). Scaling increases confidence in the wrong answer. Lin et al. (2022) confirmed this pattern with TruthfulQA: larger models produce more confident wrong answers on misconception-eliciting questions.

5.2 Why Web Search Does Not Fix It Either

Would web search resolve this? For some Layer 5 attacks, possibly—if the monitor issues a precisely targeted query to find specialist corrections. But the most effective attacks exploit misconceptions that the web also encodes. Searching "eagle visual acuity" returns pages citing "4–8× human vision." Only a query like "eagle visual acuity actual measurement study" might surface Reymond (1985)—but the monitor would need to already suspect the claim, which is precisely what Layer 5 prevents. Moreover, an adversary aware of web-search-enabled monitoring can engage in search result poisoning. For Layer 6 attacks, web search is even less useful: the monitor does not know what it does not know, so it cannot formulate a query.

5.3 Broader Threat Implications

Our attacks targeted **established scientific knowledge**—the category with the highest confidence and most extensive sourcing in any model's training data. Eagle visual acuity, cheetah speed, and shark sensory range are stable scientific facts, not volatile statistics. The corrections we exploited predate any LLM's training data by decades. If knowledge-layer poisoning works against the most deeply trained, highest-confidence category of model knowledge, it will work *a fortiori* against less stable categories—news reports, policy interpretations, financial data, biographical claims—where training-data coverage is thinner, confidence is lower, and external verification is harder. The vulnerability we demonstrate is a floor, not a ceiling.

5.4 Implications for Deployment

One-shot trusted monitoring is structurally insufficient against knowledge-layer attacks. We identify four concrete directions for defense:

Tool-augmented monitoring with targeted queries. Rather than general web search, monitors should issue claim-specific verification queries against authoritative databases. For the eagle case, the monitor would need to query specialist vision research databases rather than general search—but this requires the monitor to first identify which claims warrant external verification, which Layer 5 prevents for the most dangerous class of attacks.

Multi-round self-challenge. Our data suggests that models can sometimes access corrections when specifically prompted to generate alternative hypotheses. A protocol that asks the monitor "What would a specialist in this field disagree with?" after initial evaluation could surface doubts that one-shot evaluation suppresses.

Aggressive-flagging monitors over "smartest" monitors. Sonnet's higher recall outperformed Opus's deeper reasoning (69% vs 50% detection). Deployers should consider that the best monitor for a given protocol may not be the most capable model, but the one with the optimal precision-recall tradeoff for the threat model.

Cross-family ensemble monitoring. Different model families trained on partially different corpora may have partially different blind spots. While popular misconceptions likely persist across all major training corpora, ensemble disagreement could flag claims that warrant external verification.

5.5 Limitations

All four monitors are Claude-family models. While Layer 5's mechanism is architecture-independent (it depends on training data, not architecture), cross-family generalization is untested. Our experiments disabled web search to isolate the knowledge-layer vulnerability; tool-augmented monitors would perform better against the subset of attacks where corrections are findable online. All 13 rounds were conducted by one researcher with AI assistance during the 3-day hackathon period. Individual attack-type estimates have small sample sizes (noted in figures). The 50% evasion rate is observed across $n=12$ poison paragraphs (6 per round in R12–R13); we report this as consistent across two independent domain sets rather than as a statistically precise estimate.

5.6 Future Work

Natural extensions include: cross-family replication (testing GPT, Gemini, and open-weight models as monitors to measure shared vs. model-specific blind spots); tool-augmented monitoring experiments (measuring how much targeted web search reduces evasion, and

whether search result poisoning neutralizes this defense); scaling the methodology to produce larger attack datasets for benchmark use; and developing automated defenses specifically targeting Layer 5, such as monitors trained to flag claims whose confidence-to-source-quality ratio is anomalously high.

6. Conclusion

We have identified, characterized, and systematically exploited a previously unstudied vulnerability class in trusted monitoring: knowledge-layer attacks that target what the monitor cannot know rather than what it cannot yet do. Our methodology—developed through AI-driven iterative refinement and validated by an independent AI session reproducing 50% evasion in new domains—demonstrates that this is a systematic, transferable capability, not an anecdotal finding. The strongest available monitor cannot detect these attacks through deeper reasoning, because the failure occurs before reasoning begins. This reframes what defenders should invest in: not more capable monitors, but fundamentally different monitoring architectures that do not depend on any single model's training data being correct.

Code and Data

- **Complete methodology:** <https://github.com/tomnovember/reasoning-poison> — Full Skill system (methods.md, defense-model.md, experience-summary.md, scoring-protocol.md, SKILL.md) enabling replication and continued development
- **Structured dataset:** dataset.json — 8 annotated poison paragraphs with full text, ground truth with sources, per-model results, attack method classification, and defense layer analysis
- **Experimental data:** Complete detection matrices and scoring sheets for all 13 rounds

Author Contributions

Tang Meng designed the study, conducted all 13 experimental rounds, co-developed the attack methodology and Skill system with Claude, analyzed results, and wrote this report with AI assistance.

References

1. Greenblatt, R., Shlegeris, B., Sachan, K., & Roger, F. (2024). AI Control: Improving Safety Despite Intentional Subversion. *ICML 2024*. arXiv:2312.06942.

2. Hubinger, E., et al. (2024). Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. arXiv:2401.05566.
3. Kutasov, A., et al. (2025). Optimizing AI Agent Attacks With Synthetic Data. arXiv:2511.02823.
4. Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. *ACL 2022*. arXiv:2109.07958.
5. Reymond, L. (1985). Spatial visual acuity of the eagle. *Vision Research*, 25(10), 1477–1491.
6. Schaeffer, J., Khandelwal, A., & Tracy, T. (2025). Attack Selection Reduces Safety in Concentrated AI Control Settings. arXiv:2602.04930.
7. Svendsen, M. B. S., et al. (2016). Maximum swimming speeds of sailfish and three other large marine predatory fish species. *Biology Open*, 5(10), 1415–1419.
8. Terekhov, M., et al. (2025). Adaptive Attacks on Trusted Monitors Subvert AI Control Protocols. arXiv:2510.09462.

LLM Usage Statement

Claude (Opus 4.6) was used extensively throughout this project: as the attacker generating poison candidates with web search enabled, as the analysis partner discovering attack strategies through iterative failure analysis, and as the co-developer of the methodology files. Claude also assisted in drafting this report. All factual claims and experimental results were independently verified. The Round 13 validation was conducted entirely by a fresh Claude session with access only to the documented methodology files.