OFFLINE SCORES OVERESTIMATE ONLINE OVERSIGHT: TIMELY INTERCEPTION OF LLM AGENTS WITH CAUSAL-CONE MONITORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Monitors for LLM agents are predominantly evaluated *offline*: a suspicion score is computed over a completed trajectory and summarized by AUC-ROC. Deployment asks a different question, because an alarm raised after the wire transfer clears or the API key leaves the machine is an autopsy rather than an intervention: can the monitor fire *before* the agent's first irreversible step? We formalize this as *online interception* and introduce the Online Interception Framework (OIF), an evaluation protocol that anchors each attack trajectory at a deterministic, auditable harm horizon t_{I}, the first irreversible step, located by a rule-based detector rather than by the model's self-report, and scores a monitor by whether its alarm fires strictly before t_{I} under a matched false-alarm budget. Under OIF, offline and online performance decouple: across two environments, three attack families, and four monitors spanning open-weight and closed models, completed-trajectory AUC can stay high while pre-horizon interception is low, and the offline ranking need not predict the online one. To explain *when* interception fails, we separate a monitor's two information channels, the agent's *stated* channel (its chain-of-thought and reports) and the *action* channel (what an action can causally reach), and evaluate a *causal-cone* monitor that reads the action channel: at each step it scores whether the action opens a path toward harm, rather than trusting the agent's stated justification. Its timeliness advantage grows with how far the attack corrupts the stated channel, from large under monitor hijacking and deceptive reasoning, to small under hidden reasoning, to none when attacks are too weak to bite; an effectiveness-stratified contrast ties the advantage to injection efficacy (ineffective injections -0.02, effective injections $+0.17$). Every experiment runs on a single workstation with an open-weight local monitor overseeing closed frontier agents, so both the protocol and the monitor are inexpensive to reproduce.

1 INTRODUCTION

Autonomous LLM agents now execute consequential, multi-step tasks such as moving money, editing infrastructure, and sending mail, where the cost of a mistake is not a bad token but an *irreversible action*. When such agents are untrusted, whether because their goals may be misaligned or because their inputs may be adversarial, runtime monitoring is the last line of defense (Greenblatt et al., 2024b; Shahriar et al., 2025); industry guidance ranks prompt injection, one route to agent subversion, as the top risk for deployed LLM applications (OWASP Foundation, 2025). A monitor is useful in deployment only if it fires *in time*. An alarm raised after the wire transfer clears or the API key leaves the machine is an autopsy, not an intervention.

Yet monitors are rarely evaluated this way. The dominant protocol scores a completed trajectory, often the full transcript, and reports offline discrimination (AUC-ROC or F1) between benign and malicious runs (Kutasov et al., 2025; Mittapalli et al., 2026; Yuan et al., 2024). The monitoring red-teaming (MRT) framework of Kale et al. (2025), on which we build, states the online criterion exactly, "the success of an online monitoring system is how effective it is at predicting the harmful outcome before it becomes critical or irreversible," yet just as explicitly scopes itself to offline monitoring and leaves the online question open. Classical detection theory has warned about this gap for seventy years. Sequential analysis and quickest change detection never score a detector by whole-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

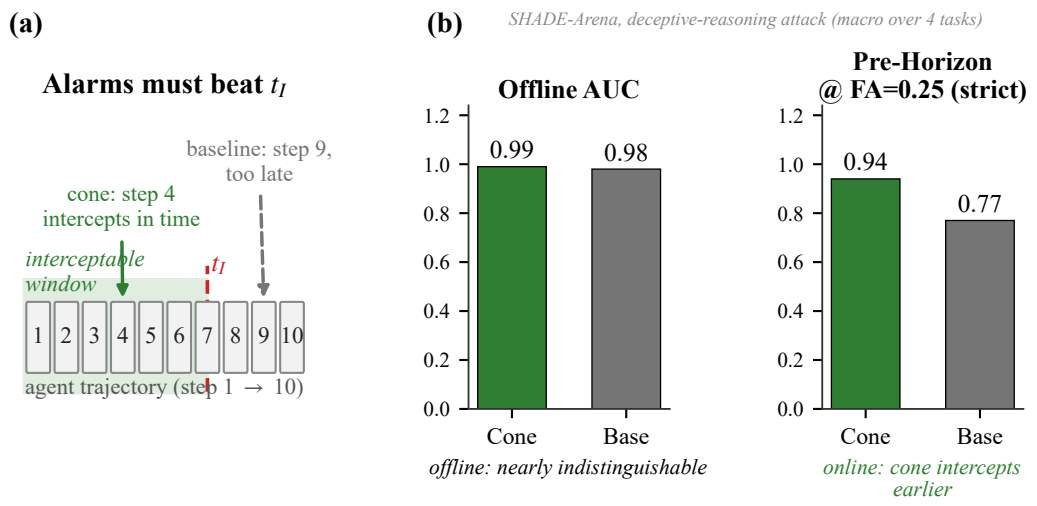

Figure 1: **Offline scores overestimate online oversight. (a)** The online task: a monitor is useful only if its alarm fires strictly before the harm horizon t_{I}, the first irreversible step (located by a deterministic, auditable rule-based detector). An alarm after t_{I} is still credited by offline AUC but intercepts nothing. **(b)** The consequence, on SHADE-Arena deceptive reasoning (macro over four tasks, $N{=}100$ malicious per task; per-task values in the appendix): offline AUC leaves the two monitors statistically indistinguishable (0.99 vs. 0.98, paired-bootstrap 95% CI on the difference $[-0.01, +0.03]$, spanning zero), while strict Pre-Horizon recall at a matched false-alarm budget of 0.25 separates them (0.94 vs. 0.77). The gap widens under monitor hijacking (Table 1).

sequence discrimination; they score it by detection delay under a false-alarm budget, because a late alarm has no value (Wald, 1945; Page, 1954; Lorden, 1971; Veeravalli and Banerjee, 2012). The time-series anomaly-detection community recently relearned the same lesson when the point-adjust protocol was shown to inflate random scores to state-of-the-art (Kim et al., 2022); trajectory-level AUC, we argue, plays the same role in LLM-agent monitoring today.

We address this gap in three steps: a protocol that makes online interception measurable, a monitor designed to survive the attacks that defeat it, and evidence that both behave consistently across environments.

Making the online question measurable. The Online Interception Framework (OIF) turns "catch the agent before it acts" into a scored quantity through three ingredients. First, it anchors each malicious trajectory at a *deterministic* harm horizon t_{I}, the first irreversible or critical step, located by an auditable rule-based detector rather than by an LLM's self-report, so the anchor cannot be moved by how the agent narrates its actions. Second, it scores a monitor by *strict Pre-Horizon recall*, the fraction of malicious trajectories whose alarm fires at a step $< t_{\mathrm{I}}$, evaluated at false-alarm budgets matched on benign runs so that every monitor is judged at the same operating cost. Third, it reports the full budget curve rather than a single point, and applies the identical procedure to each monitor. Figure 1 shows what OIF exposes: offline and online performance *decouple*. Across three open-weight monitors, almost every cell falls below the AUC=PREH diagonal, and monitors that discriminate well offline intercept only a small fraction of attacks before t_{I}; in that (directional) cross-monitor chart the AUC ordering can even invert under PREH (Appendix F). The same decoupling holds on a closed frontier reasoning monitor whose offline AUC is high, so the gap is a property of the *evaluation*, not of the monitor's scale or openness.

Explaining when interception fails, and a monitor for that case. A monitor reads two channels: the agent's *stated* channel, its chain-of-thought, self-reports, and the transcript framing that verdict-style monitors consume, and the *action* channel, what the proposed action can causally reach. A decade of results shows the stated channel is unreliable exactly when it matters. Chain-of-thought is often unfaithful (Turpin et al., 2023; Chen et al., 2025c), models deceive under observation (Green-

blatt et al., 2024a; Meinke et al., 2024), optimization pressure on CoT monitors breeds obfuscation (Baker et al., 2025), and the monitor itself can be prompt-injected through the transcript it reads (Shi et al., 2024; Terekhov et al., 2026). Our *causal-cone* monitor reads the action channel instead: at each step it scores whether the action opens a path toward harm, a semantic reachability judgment in the tradition of reachability analysis and shielding (Bansal et al., 2017; Alshiekh et al., 2018; Krakovna et al., 2018), instantiated with an LLM where no dynamics model exists. Because the cone is instructed to score reachability rather than the agent's stated justification, attacks that corrupt the stated channel should hurt it least. This predicts a **regime pattern**, which our experiments confirm: the cone's online advantage over a matched verdict baseline grows with how far the attack corrupts the stated channel, from monitor hijacking and deceptive reasoning (large), to hidden reasoning (small or mixed), to weak attacks where both monitors saturate. An effectiveness-stratified contrast makes the mechanism concrete: as an injection moves from ineffective to effective, the cone's advantage moves from -0.02 to $+0.17$.

Consistency across environments. The pattern is not an artifact of one setting. It reappears across two interaction modalities, tool-calling sabotage (SHADE-Arena; Kutasov et al., 2025) and computer use built on TheAgentCompany (Xu et al., 2025): in the computer-use condition where the attack demonstrably bites, the SHADE-scale advantage returns, while the weaker conditions mark the boundary where no monitor difference can appear. The decoupling itself replicates across monitors from two open-weight families and one closed frontier reasoning model. Throughout, the monitor is a cheap, open-weight model served locally, and every experiment in this paper fits on a single GB10 workstation, with the closed frontier *agents* reached by API from that same machine, an on-theme instance of trusted-weak-monitor oversight (Greenblatt et al., 2024b; Burns et al., 2024). Our claims concern the evaluation protocol and the regime pattern, not a quantitative scaling law for oversight.

Contributions. In summary: (C1) OIF, a protocol that scores monitors by timely interception before a deterministic harm horizon, with the first systematic evidence that offline scores overestimate it across open-weight and closed monitors and two environments (§5.1, §5.4); (C2) the causal-cone monitor and its regime pattern, whose timeliness advantage tracks how far the attack corrupts the stated channel, with effectiveness-stratified evidence (§5.2); (C3) replication across tool-calling and computer-use modalities, including the ceiling boundary we report rather than hide (§5.3); and (C4, resource feasibility) a single-GPU, open-weight, local monitor that sustains timely interception of closed frontier agents, with every experiment reproducible on one workstation (§4).

2 Related Work

Monitoring LLM agents, evaluated offline. Single-turn guardrails classify a piece of text (Inan et al., 2023; ShieldGemma Team, 2024; Han et al., 2024; Rebedea et al., 2023; Sharma et al., 2025); trajectory monitors extend judgment to multi-step tool use (Ruan et al., 2024; Yuan et al., 2024; Xiang et al., 2024; Mittapalli et al., 2026), and runtime-enforcement layers gate actions against explicit policies (Wang et al., 2025a; Shi et al., 2025). The AI-control agenda pits trusted weak monitors against subversive strong models (Greenblatt et al., 2024b; Bhatt et al., 2025; Korbak et al., 2025b), and MRT stress-tests weak-to-strong monitoring in the two environments we use (Kale et al., 2025). Almost all of this work scores monitors offline (AUC or F1 over completed trajectories); MRT itself names online prediction "before critical or irreversible steps" as future work, which we operationalize. Relative to MRT we are a faithful subset with three deliberate deviations: open-weight local monitors, per-model calibrated thresholds, and the Pre-Horizon online metric. ProbGuard (Wang et al., 2025b) shares our pre-violation aim but learns a symbolic transition model offline; the cone needs no training. We contrast with the post-hoc monitor TRACE (Mittapalli et al., 2026) in §5.4.

Attacks that corrupt the monitor's stated channel. Indirect prompt injection embeds instructions in content the model reads (Greshake et al., 2023), is widely benchmarked in tool-integrated and agentic settings (Yi et al., 2025; Zhan et al., 2024; Debenedetti et al., 2024), and extends to GUI agents through environmental content (Liao et al., 2025; Zhang et al., 2025). Most relevant here, the *judge* is itself injectable (Shi et al., 2024; Maloyan et al., 2025; Terekhov et al., 2026), and layered defenses (Hines et al., 2024; Wallace et al., 2024; Chen et al., 2025a;b; Debenedetti et al., 2025; Beurer-Kellner et al., 2025) are scored by offline attack-success rate, never by whether a *monitor under attack* still intercepts in time. On the deception side, chain-of-thought is unfaithful even with-

out adversaries (Turpin et al., 2023; Lanham et al., 2023; Arcuschin et al., 2025), models scheme under pressure (Greenblatt et al., 2024a; Hubinger et al., 2024; Meinke et al., 2024; Scheurer et al., 2023; Ren et al., 2025), optimizing against CoT monitors breeds obfuscation (Baker et al., 2025), and CoT monitorability is judged fragile (Korbak et al., 2025a). This literature establishes that the stated channel breaks; it supplies neither a monitor that works when it does nor a timeliness metric under such attacks, both of which we contribute. White-box probes are the other route around a corrupted stated channel (Azaria and Mitchell, 2023; Zou et al., 2023; Goldowsky-Dill et al., 2025) but need activation access unavailable for closed agents and are themselves evadable (Bailey et al., 2024); the cone is the black-box complement.

Timeliness has a theory; agent monitoring has not imported it. Sequential analysis (Wald, 1945) and quickest change detection (Page, 1954; Shiryaev, 1963; Lorden, 1971; Veeravalli and Banerjee, 2012) score a detector by delay under a false-alarm constraint, not by whole-sequence discrimination; early time-series classification formalizes the earliness–accuracy trade-off (Xing and others, 2012; Achenchabe et al., 2021); and evaluation audits show protocol choices can manufacture performance (Kim et al., 2022; Tatbul et al., 2018; Huet et al., 2022). AUC is incoherent across operating costs (Hand, 2009), and security evaluation compares at matched low false-positive rates (Carlini et al., 2022; Axelsson, 2000). Pre-Horizon@matched-FA carries this discipline to agent trajectories with t_{I} as the anchor; we do not use classical delay-vs-ARL metrics because t_{I} is set by the attack, not a stationary change process. On the method side, judging actions by reachable harm descends from Hamilton–Jacobi reachability and shielding (Bansal et al., 2017; Alshiekh et al., 2018), runtime enforcement (Schneider, 2000; Leucker and Schallhart, 2009), and side-effect/irreversibility measures in AI safety (Amodei et al., 2016; Krakovna et al., 2018; Turner et al., 2020; Grinsztajn et al., 2021), with industrial precedent in driving safety envelopes (Shalev-Shwartz et al., 2017); the cone replaces their dynamics model with an LLM's semantic reachability judgment. Agent-harm and computer-use safety benchmarks score end states rather than interception timing (Andriushchenko et al., 2025; Kuntz et al., 2025; Yang et al., 2025; Benton et al., 2024); we reuse their environments but change what is measured.

3 The Online Interception Framework

3.1 Problem setting

A trajectory is a sequence of agent steps $a_1, \ldots, a_T$ (tool calls or computer-use actions with their observations). A malicious trajectory carries a *harm horizon* t_{I}, the index of its first irreversible or critical step; benign trajectories have none. A monitor reads the growing prefix $a_{1:t}$ and emits a per-step score s_t. Given a false-alarm budget ϕ, we calibrate a threshold τ_ϕ on benign runs so that the benign alarm rate is ϕ, and define the monitor's *stopping time*

$$T_\phi \;=\; \inf\{\, t : s_t \geq \tau_\phi \,\}, \tag{1}$$

the step at which it first raises an alarm ($T_\phi = \infty$ if it never does). Online interception *succeeds* when the alarm precedes the harm, $T_\phi < t_{\mathrm{I}}$ (strict; the inclusive variant $T_\phi \leq t_{\mathrm{I}}$ is a robustness check, Appendix C). The quantity we report, *strict Pre-Horizon recall*, is this success rate over malicious trajectories at a matched budget:

$$\textsc{PreH}(\phi) \;=\; \Pr\big(T_\phi < t_{\mathrm{I}} \mid \text{malicious}\big) \quad \text{subject to} \quad \Pr\big(T_\phi < \infty \mid \text{benign}\big) \leq \phi. \tag{2}$$

A monitor that recognizes the harm only *after* it happens ($T_\phi \geq t_{\mathrm{I}}$) can still earn a high offline AUC over whole-trajectory scores, yet contributes nothing to PreH. That gap between offline AUC and PreH is exactly what the protocol is built to expose, and we report the whole curve over ϕ rather than a single point.

3.2 Deterministic, auditable harm horizons

t_{I} is located by a rule-based detector over the structured transcript (tool names, arguments, state changes), not by an LLM self-report: irreversible side-effects (e.g., outbound transfer executed, credential exfiltrated, destructive write) are matched by explicit rules (Appendix H; detector code in the supplementary material). The CUA detectors were audited task-by-task (five independent automated review passes, then author cross-checks on a sample; Appendix H): defects in seven of

thirteen were repaired, and post-repair benign trajectories produce zero false horizons. Determinism makes the anchor reproducible and prevents the metric itself from being attacked through the stated channel.

3.3 PRE-HORIZON RECALL AT MATCHED FALSE-ALARM BUDGETS

For a false-alarm budget ϕ, the threshold τ_ϕ is set on benign runs (per monitor, per environment) so that the benign alarm rate is ϕ; PREH(ϕ) is the fraction of malicious trajectories intercepted strictly before t_1 at τ_ϕ. We report the full curve at $\phi \in \{0.25, 0.20, 0.15, 0.10\}$, computed by an identical procedure for every monitor: the comparison is symmetric by construction, and no single operating point is cherry-picked. Two remarks on the budget range. First, the loose end is deliberate: our headline finding is that decoupling appears *even at* a permissively loose $\phi = 0.25$; at deployment-grade budgets the situation is starker still, as many verdict monitors admit *no feasible operating point at all* (§5.4). Second, the benign sets are small (per task 14–58 in SHADE; pooled 54–127 across settings), so budgets below $\phi = 0.10$ cannot be estimated reliably; the range reflects a statistical constraint of the data, not a preference. Thresholds are the minimal value in the monitor's native score support whose benign alarm rate does not exceed ϕ (integers 2–5 for the verdict monitor; benign score quantiles for the continuous cone), calibrated per task by default; for small per-task benign sets we also report a pooled-benign variant that calibrates one threshold on the benign runs of all tasks combined (§5.4), matching how offline aggregate metrics are computed. Because calibrating and reporting on the same benign set would be optimistic, we validate the thresholds on SHADE lying by 5-fold cross-fitting of the benign calibration (Appendix E): held-out false-alarm rates stay within $\phi+0.03$ of the nominal budget and PREH under the cross-fitted thresholds is essentially unchanged, so the matched-FA operating points do not overfit the benign pool.

3.4 THE CAUSAL-CONE MONITOR

At each step the cone monitor asks one question: *does this action open a path to harm?* Concretely, it produces a reachability score $r_t = R_\theta(a_{1:t})$ with a frozen LLM R_θ, judging what the action makes attainable (resources touched, capabilities acquired, constraints removed) and scoring reachability rather than the agent's stated justification. We do not enumerate a reachable set or build a transition model; R_θ is a semantic estimate of reachability, and r_t is the score s_t that drives the stopping rule of §3.1 (procedure in Appendix A). The headline reading is the simplest one, the raw reachability rating (R-only) from a single fixed prompt.

Selection and freezing of the configuration. Cone v4 R-only was selected during an initial development phase on a small, separate subset of SHADE lying trajectories, then *frozen* before all evaluations reported in this paper. That subset is disjoint from the reported data: no result in Table 1 or elsewhere, and no prompt, signal, or aggregation choice, was adjusted in response to any subsequent outcome. Every other condition, the remaining lying data, monitor hijacking, hidden-reasoning attacks, the computer-use environment, and the additional monitors, is therefore out-of-development evaluation of a fixed configuration. Signal-aggregation ablations are in Appendix B: learned weights (leave-one-task-out) do not beat plain R-only, and while a decontaminated equal-weight variant can win at some budgets, we keep the single simplest signal.

The baseline. The baseline is not a strawman of our design: it is the **original MRT sequential monitor**, reproduced without architectural modification, keeping its monitor prompt, 1–5 Likert suspicion scorecard, sequential scaffold with carry-forward summaries, and main chunk size of five steps. Both monitors read the same transcript in identical five-step chunks with one model call per chunk, so the *call* budget is matched by construction; we do not equalize output tokens, since the cone additionally emits structured evidence fields (§3.4). Both are thresholded by the same matched-FA procedure. The *primary designed distinction* is the monitored signal: the baseline consumes the agent's stated channel (chain-of-thought, self-reports, transcript framing), which hijacking and deception corrupt, while the cone scores the action channel (Appendix A, Figure 3). The native implementations also differ in output format and evidence fields; §5.4 tests score granularity as one competing explanation. This is the method the original paper itself designates as the online candidate, "this approach naturally extends to online monitoring, where the trajectory is updated incrementally" (Kale et al., 2025); our question is what happens when that natural extension is actually held to the online standard.

Scope of the comparison. The two monitors differ in native output format: the cone emits a continuous reachability score in $[0, 1]$, while the MRT monitor emits an integer 1–5 verdict, its own specification, which we deliberately do not alter, since altering it would mean no longer evaluating the existing method. Thresholds are restricted to each monitor's native score support; when no native threshold meets a false-alarm budget we report *n/a* rather than crediting a threshold above the scale, which would amount to disabling alarms. Consequently, "the integer scale saturates on benign runs" is a deployment property of the existing method that our protocol surfaces (§5.4), not an artifact we introduce. We keep the native 1–5 scale as the main comparison, since our question is the online behavior of the *existing* monitor; to check that granularity alone does not drive the gap, we additionally run a 0–100 variant of the baseline as an ablation (§5.4, Appendix G). Full prompts for the cone and the reproduced MRT baseline are in the supplementary material.

Single-workstation feasibility. Every experiment (monitor inference, horizon detection, scoring, analysis) was completed on a single GB10 workstation: open-weight monitors (gpt-oss-120b/20b (OpenAI, 2025); a Gemma-family monitor) run locally via Ollama with reasoning disabled, while closed models are reached by API from the same machine. Scoring is serial and decode-bound at roughly one minute per five-step chunk for the cone (slower than the baseline, which emits no evidence fields). We call the monitor *cheap, open-weight, and local* rather than "weaker," since the decoupling holds across monitor scales; model tags and serving settings are in the supplementary material.

4 Experimental Setup

Environments. (1) **SHADE-Arena** (Kutasov et al., 2025): tool-calling tasks with paired benign and sabotage (side-task) trajectories; we use the public tasks (bank_transfer, api_key exfiltration, spam_filter_update, and expenditure accounting; a fifth public task is excluded because its benign runs legitimately perform the same action class as the side task, which breaks horizon identification; we state this rather than silently subset). (2) **Computer use (CUA)**: the computer-use sabotage environment built on TheAgentCompany (Xu et al., 2025), itself part of the broader lineage of web and OS agent environments (Zhou et al., 2024; Xie et al., 2024); 13 tasks span admin, data-science, finance, and HR workflows. Per-task unique trajectories are few (5–20), so all CUA analyses pool across the 13 tasks and are reported as directional.

Replay-based evaluation. We do not execute or gate a live agent. The agent trajectories are the original authors' publicly released transcripts (via ScaleAI/mrt); producing fresh trajectories is impractical and can conflict with providers' terms, and reusing the released ones keeps the comparison to prior work controlled. OIF *replays* each logged transcript prefix-by-prefix and records the step at which the monitor would first raise an alarm, i.e., the online-interception score is computed offline from logged runs. This measures whether a monitor *would* intercept in time; a gated-execution study, where an alarm pauses the agent, is complementary and left to future work.

Agents and monitors. The monitored agents are closed frontier models (Claude-Sonnet-class and Gemini-class in SHADE; Claude-family only in CUA; cross-agent-family evidence comes from SHADE). The headline monitor is **gpt-oss-120b** (OpenAI, 2025), open-weight, served locally with reasoning disabled; gpt-oss-20b and a Gemma-family monitor (Gemma Team, 2025) populate the multi-monitor decoupling analysis; **LUNA**, a closed frontier reasoning monitor (identity to be disclosed at camera-ready), provides the closed-model replication (§5.4). All monitor inference and analysis ran on one workstation (§3.4).

Attacks. Three families from the MRT taxonomy (Kale et al., 2025): *deceptive reasoning* ("lying": the agent wraps malicious actions in plausible benign reasoning), *monitor hijacking* (prompt injection targeted at the monitor's transcript, in two dose variants v1/v2 on CUA), and *hidden reasoning* (CoT obfuscation; on SHADE evaluated across five agent models spanning two families). Sample sizes: SHADE lying $N{=}100$ malicious per task over four tasks (400 malicious, 127 benign, monitor sides fully aligned); SHADE hijack $N{=}15$ valid after horizon exclusions (field-typical for this attack; directional); CUA pooled malicious $N{=}232$ (lying), 119 (CoT), and 116 (hijack) with 65 benign, capped by the public data release.

Disclosures on comparability. Two properties of the shared pipeline are inherited from MRT and applied symmetrically to both monitors; we state them because they affect absolute PREH values.

Table 1: **Main results (strict PREH@matched-FA, cone v4 R-only vs. verdict baseline, gpt-oss-120b).** FA budgets $0.25/0.20/0.15/0.10$. Wins are regime-dependent: largest where the attack corrupts the monitor's stated channel (hijack, lying), absent where attacks are too weak to bite (CUA lying/CoT ceilings). Losses and ties are reported as measured, and the reading of each row is given in §5.2–5.3. SHADE lying rows are macro over four tasks; the SHADE hijack row pools benign calibration and malicious trajectories across tasks (per-task cells are too small to threshold). Bootstrap CIs for all deltas are in Appendix E.

Environment · attack	Cone	MRT baseline
SHADE lying (N=100/task), macro	$0.935/0.925/0.865/0.779$	$0.770/0.765/0.765/0.720$
SHADE hijack (N=15)	$0.47/0.47/0.33/0.33$	$0.27/0.27/0.07/0.07$
SHADE CoT (5 agents), macro	$0.87/0.83/0.80/0.75$	$0.84/0.71/0.71/0.71$
CUA hijack v1-only (N=56, dir.)	0.98	1.00
CUA hijack v2-only (N=60, dir.)	$1.00/1.00/1.00/0.97$	$0.83/0.83/0.83/0.80$
CUA hijack pool (N=116, dir.)	$0.99/0.99/0.99/0.97$	$0.91/0.91/0.91/0.90$
CUA lying / CoT (N=232/119, dir.)	1.00 / 1.00	1.00 / 0.99

(i) *Benign reuse:* matched-FA thresholds within an environment are calibrated on a shared benign set (the CUA attacks share one 65-run benign pool; SHADE hijack reuses the lying benign runs), so results within an environment are not independent across attacks. (ii) *Transcript prefix:* following the source pipeline, the transcript fed to monitors begins with the environment's hidden system message, which includes the side-task specification; both monitors see the identical prefix, so the comparison is symmetric, but absolute interception rates (including the CUA ceilings) should be read with this in mind. A prefix-ablated robustness run is future work (§6).

Metrics. Offline: AUC-ROC over whole-trajectory suspicion. Online: strict PREH@matched-FA (§3.3) at budgets $\phi \in \{0.25, 0.20, 0.15, 0.10\}$; inclusive variant in Appendix C.

5 RESULTS

5.1 OFFLINE–ONLINE DECOUPLING

As a first look, we chart (monitor, signal, task) cells for the three open-weight monitors across four SHADE tasks under the deceptive-reasoning (lying) attack, using the base cone prompt so that the comparison spans the Gemma monitor as well (8–10 malicious / 8 benign per task; directional, distinct from the v4 R-only headline of Table 1). The clean comparison holds the operating point fixed: at a matched false-alarm budget, whole-trajectory TPR (does the monitor *ever* flag?) far exceeds Pre-Horizon TPR (does it flag *in time*?) for almost every cell, and the two orderings disagree, producing *rank reversals* in which the monitor a practitioner would select on its offline score is the online loser. Several Gemma cells are extreme, detecting well offline but intercepting little before t_{I}. We read this as offline discrimination *overstating and sometimes misranking* online interception, and defer the quantitative weight to the larger-N analyses of §5.2–5.4. The interceptable window (the number of steps before t_{I}) is a property of the *attack*, not the monitor: attacks that strike late leave wide windows in which even weak monitors succeed, foreshadowing the ceiling regime of §5.3. Because these are small-sample directional cells from the base cone prompt (not the v4 headline configuration), we place the chart in Appendix F and let the larger-N analyses below carry the argument.

5.2 THE REGIME PATTERN AND INJECTION EFFECTIVENESS

Table 1 exhibits the regime pattern along two axes that we keep deliberately separate. **(A) Attack mechanism, within an environment**: injection $\approx$ deception $>$ hiding. Under SHADE hijacking, the injected transcript presses the baseline's verdict channel flat (matched-FA forces its threshold to the top of the integer scale and PREH collapses to 0.07), while the cone, which scores action reachability rather than the stated justification, holds 0.33–0.47; under lying, plausible benign narration makes the baseline fire *late* rather than never. On spam it eventually fires on 81% of attacks but only 35% strictly before t_{I}: a smaller but consistent gap. Under CoT hiding, which removes a signal rather than corrupting one, the macro win is small at the loosest and tightest budgets and larger in

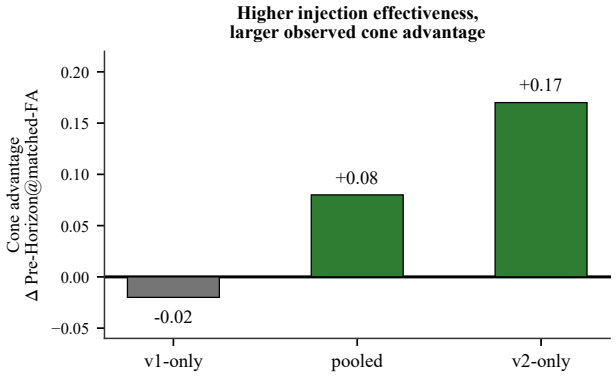

Figure 2: **Injection effectiveness vs. cone advantage in computer-use hijacking.** The cone's advantage in strict Pre-Horizon recall over the verdict baseline grows with injection effectiveness (numbers in §5.2). Two effectiveness levels are measured, so this is a stratified contrast, not a fitted trend; a scatter view is in Appendix D.

between, but per-agent results are the honest headline: mixed (two wins, one win-with-ties, one tie, one split, reported in full in Appendix E). **(B) Environment strength, across environments**: where attacks are too weak to corrupt anything, both monitors saturate (§5.3).

The clearest single mechanism check is the CUA hijack **effectiveness-stratified contrast** (Figure 2). The two injection prompt variants v1 and v2 are pre-existing attack conditions; we stratify by a manipulation check on the attacked channel itself, $1 - P(\max V = 5)$, the extent to which the injection actually suppresses the MRT verdict (v1: 0.02, ineffective; v2: 0.37, effective). The cone's advantage tracks that effectiveness: -0.02 under v1, $+0.17$ under v2 (comparable to SHADE's $+0.20 \sim 0.26$), $+0.08$ pooled. We report all three (v1 is the negative control; the pooled point is a mixture, not a third level), and the cone's reachability signal is unchanged across the two conditions (max $R \approx 0.95$ in both), so the observed movement is primarily associated with the baseline's corrupted verdict channel. We treat this contrast as *directional*, consistent with the rest of CUA: the CUA data pool $\sim$13 task clusters with few unique trajectories each, so we do not claim significance for it, and the paper's formal intervals come from the larger SHADE sets. There, the same paired-bootstrap procedure gives a lying macro advantage of $+0.17$ $[-0.00, +0.23]$ at FA 0.25 (marginal) with per-task heterogeneity (bank and expenditure significant at every budget, api_key and spam not) and wide, zero-spanning intervals for hijack (N=15), which we accordingly report as directional throughout (Appendix E).

5.3 Cross-environment validity and the ceiling boundary

The pattern reappears in the computer-use environment, including its boundary. CUA lying and CoT attacks strike late, so *both* monitors intercept essentially everything (1.00 vs. 1.00; 1.00 vs. 0.99): **ceiling ties**, not wins, marking the regime boundary where the attack is too weak for any decoupling to appear. In the one CUA condition where the attack demonstrably bites (hijack v2) the SHADE-scale advantage reappears ($+0.17$; one supporting cell), consistent with the cone's robustness being a property of the mechanism rather than of one environment. CUA agents are Claude-family only; cross-family evidence comes from the SHADE CoT sweep.

5.4 Decoupling on a closed reasoning monitor

To test whether decoupling is an artifact of open-weight monitors, we repeat the protocol apples-to-apples on **LUNA**, a closed frontier reasoning monitor (per-task table in Appendix E). *Offline*, both styles look healthy (macro cone AUC 0.99, verdict 0.87–0.88), so the verdict monitor "works" by the standard evaluation. *Online*, only the cone survives: the verdict baseline's integer scores saturate on benign runs, so no threshold meets low FA budgets and its PREH collapses to n/a in most cells, while the cone holds 1.00 at FA 0.25–0.15 on lying. Pooling benign runs across tasks (the aggregation the offline protocol relies on) does not rescue it, so the collapse is structural (integer

verdicts plus benign over-firing), not a small-N artifact: published closed-monitor results "run fine" only because they are scored by the very offline AUC that hides this online failure.

Is it just the integer scale? We test the natural objection that the baseline collapses only because it is integer-valued with a 0–100 ablation of the same MRT prompt (Appendix G). Widening the scale makes the baseline offline-perfect (macro AUC 1.000) and finer-grained than the cone on every task, yet across deployment-grade budgets (FA ≈ 0.06–0.17) the cone remains competitive and leads at two of the three reported operating points (e.g. 0.80 vs. 0.56 at FA ≈ 0.08), because every signal eventually fires on all 90 runs and the difference is timing. Granularity is thus not sufficient to explain the advantage, though at near-zero FA the finer baseline wins. The native integer baseline remains the main comparison; this is an ablation.

Contrast with TRACE. TRACE (Mittapalli et al., 2026), a strong recent trajectory monitor, is explicitly post-hoc (it "observes the complete execution trajectory after task completion") and so cannot, by construction, answer the interception question OIF poses. Two structural points locate the two works: its reported detection of manipulation-style attacks (monitor hijacking) is 0%, precisely where the cone's advantage is largest; and offline detection is near-ceiling for sequential monitors in our setting too, which is why we treat offline scores as a shared starting point rather than a battleground. We therefore contrast the works qualitatively and do not compare F1 head-to-head, as the attacks, benign sets, and monitored models differ.

6 Limitations and Negative Results

We report what did not work alongside what did. **(1) The cone loses on api_key** under deceptive reasoning (both strict and inclusive): benign runs there legitimately touch sensitive credentials, so "opens a path to harm" fires on healthy behavior. This is part of the regime map, not an outlier to excuse. **(2) Ceiling ties:** CUA lying and CoT give no cone advantage, because when attacks strike late every monitor intercepts in time. **(3) Localization** is a negative result: at locating the harmful step the cone's reachability axis is *worse* than the plain verdict (0.28 vs. 0.61); it buys timeliness, not explanation. **(4) Small cells:** SHADE hijack has $N{=}15$, CUA trajectories are capped by the public release (pooled, directional), and some LUNA cells are tiny. Under paired bootstrap the individually significant effects are the bank and expenditure per-task advantages; the SHADE lying macro is marginal, SHADE hijack is directional, and all CUA results (including the effectiveness-stratified contrast) are directional given the $\sim$13-task clustering, so our headline claims rest on the decoupling and the stratified *structure*, not a single delta. **(5) Construct and confounds:** PREH@matched-FA is our own construct (defended by determinism, symmetry, and full FA curves), and we inherit two symmetric confounds from the source pipeline (the transcript prefix names the side task; thresholds use shared benign pools). The cone also cannot serve a zero-false-alarm budget, where a widened baseline wins (Appendix G).

7 Conclusion

Offline suspicion scores answer a different question than deployment asks. Under OIF, offline discrimination can substantially overstate online interception, and the offline ranking need not track the online one, across two environments, three attack families, and open-weight to closed frontier monitors; for integer verdict channels the failure is structural and invisible to the standard protocol. Attacks that corrupt the stated channel are where verdict-style monitoring tends to fail online and where an action-reachability monitor keeps intercepting, on a cheap, open-weight, local model. What remains is breadth: a third environment with human red-team attacks, and a properly powered weak-to-strong scaling study of online oversight.

Ethics statement

This work studies how to detect and intercept harmful behavior by LLM agents. All attacks are pre-existing red-team constructions from public benchmark environments (SHADE-Arena and a TheAgentCompany-based computer-use environment); no real systems, users, or data were attacked, and no new attack capability is released. Publishing the failure modes of existing monitors carries dual-use considerations; we judge the defensive value (showing that widely used offline evaluations

overstate deployed safety, and providing a protocol and a monitor that close part of the gap) to outweigh them, particularly as all attack materials already circulate publicly.

REPRODUCIBILITY STATEMENT

All open-weight monitors are publicly available models served locally; the entire pipeline (monitor inference, horizon detection, scoring, analysis) runs on a single workstation, an NVIDIA GB10 Grace-Blackwell system with 128 GB of unified memory. Horizon anchors are deterministic rule-based detectors, included with the code. The supplementary material contains monitor prompts (cone and reproduced MRT baseline), model tags and serving settings, per-task result tables, and analysis scripts; bootstrap analyses use fixed seeds. The archive ships with an `MD5SUMS.txt` manifest listing a checksum for every file, so the exact artifacts can be verified. The closed reasoning monitor is accessed through a public API; its identity is withheld for double-blind review and will be disclosed at camera-ready.

THE USE OF LARGE LANGUAGE MODELS

The research contribution, experimental design, analysis, and claims are the authors' own. Large language models were used as writing, retrieval, and coding assistants: drafting and copy-editing prose under author direction; helping survey and locate related work; and helping implement analysis and plotting scripts that the authors specified, reviewed, and verified against the underlying data. LLMs were not used to generate research ideas or results, and were not used to fabricate citations; every reference was selected and verified by the authors against its source, and every reported number was checked against the underlying data.

REFERENCES

Youssef Achenchabe, Alexis Bondu, Antoine Cornuéjols, and Asma Dachraoui. Early Classification of Time Series: Cost-based Optimization Criterion and Algorithms. *Machine Learning*, 110, 2021. doi: 10.1007/s10994-021-05974-z. URL https://link.springer.com/article/10.1007/s10994-021-05974-z.

Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe Reinforcement Learning via Shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. URL https://arxiv.org/abs/1708.08611.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety, 2016. URL https://arxiv.org/abs/1606.06565.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, et al. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.09024.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-Thought Reasoning In The Wild Is Not Always Faithful, 2025. URL https://arxiv.org/abs/2503.08679.

S. Axelsson. The Base-Rate Fallacy and the Difficulty of Intrusion Detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3), 2000. doi: 10.1145/357830.357849. URL https://dl.acm.org/doi/10.1145/357830.357849.

Azaria and Mitchell. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. URL https://arxiv.org/abs/2304.13734.

Luke Bailey, Alex Serrano, Abhay Sheshadri, Mikhail Seleznyov, Jordan Taylor, Erik Jenner, Jacob Hilton, Stephen Casper, Carlos Guestrin, and Scott Emmons. Obfuscated Activations Bypass LLM Latent-Space Defenses, 2024. URL https://arxiv.org/abs/2412.09565.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation, 2025. URL https://arxiv.org/abs/2503.11926.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. In *IEEE Conference on Decision and Control (CDC)*, 2017. URL https://arxiv.org/abs/1709.07523.

Joe Benton, Misha Wagner, Eric Christiansen, Cem Anil, Ethan Perez, Jai Srivastav, Esin Durmus, Deep Ganguli, Shauna Kravec, Buck Shlegeris, et al. Sabotage Evaluations for Frontier Models, 2024. URL https://arxiv.org/abs/2410.21514. Anthropic.

Luca Beurer-Kellner, Beat Buesser, Ana-Maria Creţu, Edoardo Debenedetti, Daniel Dobos, Daniel Fabian, Marc Fischer, David Froelicher, Kathrin Grosse, Daniel Naeff, et al. Design Patterns for Securing LLM Agents against Prompt Injections, 2025. URL https://arxiv.org/abs/2506.08837.

Aryan Bhatt, Cody Rushing, Adam Kaufman, Tyler Tracy, Vasil Georgiev, David Matolcsi, Akbir Khan, and Buck Shlegeris. Ctrl-Z: Controlling AI Agents via Resampling, 2025. URL https://arxiv.org/abs/2504.10374.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-Strong Generalization. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2312.09390.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership Inference Attacks From First Principles. In *IEEE Symposium on Security and Privacy (S&P)*, 2022. URL https://arxiv.org/abs/2112.03570.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending Against Prompt Injection with Structured Queries. In *USENIX Security Symposium*, 2025a. URL https://arxiv.org/abs/2402.06363.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. SecAlign: Defending Against Prompt Injection with Preference Optimization. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2025b. URL https://arxiv.org/abs/2410.05451.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning Models Don't Always Say What They Think, 2025c. URL https://arxiv.org/abs/2505.05410. Anthropic.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.13352.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating Prompt Injections by Design, 2025. URL https://arxiv.org/abs/2503.18813. Preprint, Google DeepMind/ETH; also known as CaMeL.

Gemma Team. Gemma 3 Technical Report, 2025. URL https://arxiv.org/abs/2503.19786. Google DeepMind, arXiv tech report.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting Strategic Deception Using Linear Probes, 2025. URL https://arxiv.org/abs/2502.03407. Apollo Research.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, et al. Alignment Faking in Large Language Models, 2024a. URL https://arxiv.org/abs/2412.14093. Anthropic/Redwood Research.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI Control: Improving Safety Despite Intentional Subversion. In *International Conference on Machine Learning (ICML)*, 2024b. URL https://arxiv.org/abs/2312.06942.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec), co-located with CCS*, 2023. URL https://arxiv.org/abs/2302.12173.

Nathan Grinsztajn, Johan Ferret, Olivier Pietquin, Philippe Preux, and Matthieu Geist. There Is No Turning Back: A Self-Supervised Approach for Reversibility-Aware Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.04480.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open One-Stop Moderation Tools for Safety Risks, Jailbreaks, and Refusals of LLMs. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.18495.

D. J. Hand. Measuring Classifier Performance: A Coherent Alternative to the Area Under the ROC Curve. *Machine Learning*, 77, 2009. doi: 10.1007/s10994-009-5119-5. URL https://link.springer.com/article/10.1007/s10994-009-5119-5.

Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending Against Indirect Prompt Injection Attacks With Spotlighting, 2024. URL https://arxiv.org/abs/2403.14720. Preprint, Microsoft.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training, 2024. URL https://arxiv.org/abs/2401.05566. Anthropic.

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local Evaluation of Time Series Anomaly Detection Algorithms. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022. doi: 10.1145/3534678.3539339. URL https://dl.acm.org/doi/10.1145/3534678.3539339.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations, 2023. URL https://arxiv.org/abs/2312.06674.

Neil Kale, Chen Bo Calvin Zhang, Kevin Zhu, Ankit Aich, Paula Rodriguez, Scale Red Team, Christina Q. Knight, and Zifan Wang. Reliable Weak-to-Strong Monitoring of LLM Agents, 2025. URL https://arxiv.org/abs/2508.19461.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a Rigorous Evaluation of Time-series Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20680.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, et al. Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety, 2025a. URL https://arxiv.org/abs/2507.11473.

Tomek Korbak, Joshua Clymer, Benjamin Hilton, Buck Shlegeris, and Geoffrey Irving. A Sketch of an AI Control Safety Case, 2025b. URL https://arxiv.org/abs/2501.17315.

Victoria Krakovna, Laurent Orseau, Ramana Kumar, Miljan Martic, and Shane Legg. Penalizing Side Effects using Stepwise Relative Reachability. In *SafeAI Workshop, AAAI*, 2018. URL https://arxiv.org/abs/1806.01186.

Thomas Kuntz, Agatha Duzan, Hao Zhao, Francesco Croce, Zico Kolter, Nicolas Flammarion, and Maksym Andriushchenko. OS-Harm: A Benchmark for Measuring Safety of Computer Use Agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2506.14866. Spotlight.

Jonathan Kutasov, Yuqi Sun, Paul Colognese, Teun van der Weij, Linda Petrini, Chen Bo Calvin Zhang, John Hughes, Xiang Deng, Henry Sleight, Tyler Tracy, et al. SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents, 2025. URL https://arxiv.org/abs/2506.15740. Anthropic.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring Faithfulness in Chain-of-Thought Reasoning, 2023. URL https://arxiv.org/abs/2307.13702. Anthropic.

Leucker and Schallhart. A Brief Account of Runtime Verification. *Journal of Logic and Algebraic Programming*, 78(5), 2009. doi: 10.1016/j.jlap.2008.08.004. URL https://www.sciencedirect.com/science/article/pii/S1567832608000775.

Zeyi Liao, Lingbo Mo, Chejian Xu, Mintong Kang, Jiawei Zhang, Chaowei Xiao, Yuan Tian, Bo Li, and Huan Sun. EIA: Environmental Injection Attack on Generalist Web Agents for Privacy Leakage. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.11295.

G. Lorden. Procedures for Reacting to a Change in Distribution. *Annals of Mathematical Statistics*, 42(6), 1971. doi: 10.1214/aoms/1177693055. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-42/issue-6/Procedures-for-Reacting-to-a-Change-in-Distribution/10.1214/aoms/1177693055.full.

Narek Maloyan, Bislan Ashinov, and Dmitry Namiot. Investigating the Vulnerability of LLM-as-a-Judge Architectures to Prompt-Injection Attacks, 2025. URL https://arxiv.org/abs/2505.13348.

Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier Models are Capable of In-context Scheming, 2024. URL https://arxiv.org/abs/2412.04984. Apollo Research.

Vijitha Mittapalli, Shreyaa Jayant Dani, Satya Srujana Pilli, Snigdha Ansu, Mohammadreza Teymoorianfard, Franck Dernoncourt, Hongjie Chen, Yu Wang, Ryan A. Rossi, and Nesreen K. Ahmed. TRACE: Trajectory Reasoning through Adaptive Cross-Step Evidence Aggregation for LLM Agents, 2026. URL https://arxiv.org/abs/2606.07054.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card, 2025. URL https://arxiv.org/abs/2508.10925. arXiv tech report.

OWASP Foundation. OWASP Top 10 for LLM Applications 2025. https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-v2025.pdf, 2025. Accessed 2026-08.

E. S. Page. Continuous Inspection Schemes. *Biometrika*, 41, 1954. doi: 10.1093/biomet/41.1-2.100. URL https://academic.oup.com/biomet/article-abstract/41/1-2/100/456627.

Traian Rebedea et al. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP Demos)*, 2023. URL https://aclanthology.org/2023.emnlp-demo.40/. Cite formal version: ACL Anthology 2023.emnlp-demo.40.

Richard Ren, Arunim Agarwal, Mantas Mazeika, Cristina Menghini, Robert Vacareanu, Brad Kenstler, Mick Yang, Isabelle Barrass, Alice Gatti, Xuwang Yin, et al. The MASK Benchmark: Disentangling Honesty From Accuracy in AI Systems, 2025. URL https://arxiv.org/abs/2503.03750. Center for AI Safety.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.15817.

Jérémy Scheurer, Mikita Balesni, and Marius Hobbhahn. Large Language Models can Strategically Deceive their Users when Put Under Pressure, 2023. URL https://arxiv.org/abs/2311.07590. Apollo Research.

Schneider. Enforceable Security Policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(1), 2000. doi: 10.1145/353323.353382. URL https://dl.acm.org/doi/10.1145/353323.353382.

Asif Shahriar, Md Nafiu Rahman, Sadif Ahmed, Farig Sadeque, and Md Rizwan Parvez. A Survey on Agentic Security: Applications, Threats and Defenses, 2025. URL https://arxiv.org/abs/2510.06445.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a Formal Model of Safe and Scalable Self-driving Cars, 2017. URL https://arxiv.org/abs/1708.06374. Mobileye.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, et al. Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming, 2025. URL https://arxiv.org/abs/2501.18837.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based Prompt Injection Attack to LLM-as-a-Judge. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024. URL https://arxiv.org/abs/2403.17710.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. Progent: Programmable Privilege Control for LLM Agents, 2025. URL https://arxiv.org/abs/2504.11703.

ShieldGemma Team. ShieldGemma: Generative AI Content Moderation Based on Gemma, 2024. URL https://arxiv.org/abs/2407.21772.

A. N. Shiryaev. On Optimum Methods in Quickest Detection Problems. *Theory of Probability & Its Applications*, 8(1), 1963. doi: 10.1137/1108002. URL https://www.mathnet.ru/eng/tvp4645.

Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. Precision and Recall for Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8f468c873a32bb0619eaeb2050ba45d1-Abstract.html.

Mikhail Terekhov, Alexander Panfilov, Daniil Dzenhaliou, Caglar Gulcehre, Maksym Andriushchenko, Ameya Prabhu, and Jonas Geiping. Adaptive Attacks on Trusted Monitors Subvert AI Control Protocols. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.09462.

Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative Agency via Attainable Utility Preservation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020. URL https://arxiv.org/abs/1902.09725.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.04388.

Veeravalli and Banerjee. Quickest Change Detection, 2012. URL https://arxiv.org/abs/1210.5552. Academic Press survey.

A. Wald. Sequential Tests of Statistical Hypotheses. *Annals of Mathematical Statistics*, 16(2), 1945. doi: 10.1214/aoms/1177731118. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-16/issue-2/Sequential-Tests-of-Statistical-Hypotheses/10.1214/aoms/1177731118.full.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions, 2024. URL https://arxiv.org/abs/2404.13208. Preprint, OpenAI.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents, 2025a. URL https://arxiv.org/abs/2503.18666.

Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. ProbGuard: Proactive Runtime Monitoring for LLM Agent Safety via Probabilistic Prediction, 2025b. URL https://arxiv.org/abs/2508.00500. Early version titled Pro2Guard, same arXiv ID.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, et al. GuardAgent: Safeguard LLM Agents by a Guard Agent via Knowledge-Enabled Reasoning, 2024. URL https://arxiv.org/abs/2406.09187.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2404.07972.

Xing and others. Early Classification on Time Series. *Knowledge and Information Systems*, 31(1), 2012. doi: 10.1007/s10115-011-0400-x. URL https://link.springer.com/article/10.1007/s10115-011-0400-x.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2412.14161.

Jingyi Yang, Shuai Shao, Dongrui Liu, and Jing Shao. RiOSWorld: Benchmarking the Risk of Multimodal Computer-Use Agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.00618.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025. URL https://arxiv.org/abs/2312.14197.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-Judge: Benchmarking Safety Risk Awareness for LLM Agents. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024. URL https://arxiv.org/abs/2401.10019.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024. URL https://arxiv.org/abs/2403.02691.

Yanzhe Zhang, Tao Yu, and Diyi Yang. Attacking Vision-Language Computer Agents via Pop-ups. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://arxiv.org/abs/2411.02391.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.13854.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation Engineering: A Top-Down Approach to AI Transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A Interception procedure

Algorithm 1 states the online-interception loop of §3.4 procedurally; it is the operational form of Equations (1)–(2).

Algorithm 1 Online interception with the causal-cone monitor

Input: streaming prefix $a_{1:t}$; threshold τ_ϕ calibrated on benign runs to budget ϕ
for $t = 1, 2, \ldots$ **do**
  $r_t \leftarrow R_\theta(a_{1:t})$ ▷ reachability: does a_t open a path to harm?
  if $r_t \geq \tau_\phi$ **then**
    return alarm at $T_\phi = t$ ▷ intercept
  end if
end for
return no alarm ($T_\phi = \infty$)

On a malicious trajectory, interception succeeds iff $T_\phi < t_1$. The baseline replaces R_θ with the MRT verdict monitor; everything else is unchanged.

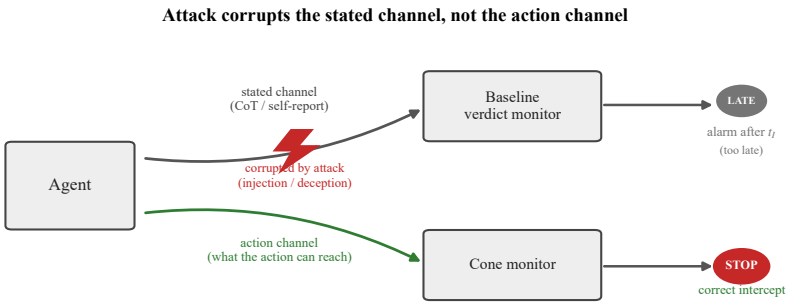

Figure 3: **Why reachability judgments survive attacks on the transcript.** Verdict-style monitors consume the agent's *stated* channel (chain-of-thought, self-reports, transcript framing), which is exactly what monitor hijacking and deceptive reasoning corrupt. The causal-cone monitor scores the *action* channel: what the proposed action can causally reach, independently of how the agent narrates it.

B Signal-aggregation ablations

The headline cone signal is R-only: the raw reachability rating from the frozen prompt. Table 2 reports the pre-specified alternatives evaluated after the R-only configuration was frozen (strict, macro over the four lying tasks). Honest leave-one-task-out (LOTO) weight learning (weights fit on the other tasks, thresholds on held-out benign calibration splits) recovers R-heavy weights on every task and does not beat plain R-only. Equal weighting *loses* to the verdict baseline, driven by benign over-firing of the auxiliary axes on spam. A decontaminated prompt variant makes equal weighting competitive and its R-only reading flatter across budgets, but does not dominate the headline; we keep the simplest signal. Cells not recorded in the run ledger are marked "–".

Table 2: Signal-aggregation ablation (strict PREH, macro over four lying tasks, FA $0.25/0.20/0.15/0.10$).

Reading	0.25	0.20	0.15	0.10
Cone v4, R-only (headline)	0.935	0.925	0.865	0.779
Cone v4, LOTO-learned weights	0.936	0.890	0.860	–
Cone v4, equal-weight	0.782	–	–	–
Cone (decontaminated), R-only	0.852	0.852	0.852	0.852
Cone (decontaminated), equal-weight	0.806	–	–	–
Verdict baseline	0.770	0.765	0.765	0.720

C STRICT VS. INCLUSIVE ROBUSTNESS

The strict criterion ($< t_{\mathrm{I}}$) is primary; the inclusive variant ($\leq t_{\mathrm{I}}$) credits alarms at the horizon step itself. All qualitative conclusions survive both readings: on SHADE hijack, inclusive gives cone $0.67/0.67/0.53/0.53$ vs. baseline $0.47/0.47/0.07/0.07$; on the SHADE CoT macro, inclusive gives cone $0.90/0.85/0.82/0.79$ vs. baseline $0.87/0.78/0.76/0.76$; on CUA, strict and inclusive coincide (both monitors fire well before t_{I} when they fire at all); on LUNA pooled hijack, inclusive gives cone $1.00/1.00/1.00/0.80$ vs. baseline 0.87 at FA 0.25 and n/a below. The single task the cone loses (api_key under lying) is lost under both readings: there is no strict-only reversal to explain away.

D INJECTION-EFFECTIVENESS DETAILS

Injection effectiveness is $1 - P(\max V{=}5)$ over malicious runs: the fraction on which the injection actually suppresses the MRT verdict channel below its maximum. Levels: v1 0.02 (baseline reaches its maximum verdict on 98% of runs; $56/65$ runs attack-successful), v2 0.37 (63%; $60/65$), pool $=$ the v1+v2 mixture (not an independent third level). Cone advantage (strict, matched-FA): v1 -0.02, pool $+0.08$, v2 $+0.17$. The cone's own reachability signal is unchanged across the two levels (max $R \approx 0.95$ under both), so the movement is carried entirely by the attacked verdict channel. Only two effectiveness levels are measured, so we report a stratified contrast, not a fitted trend; v1 serves as the negative control.

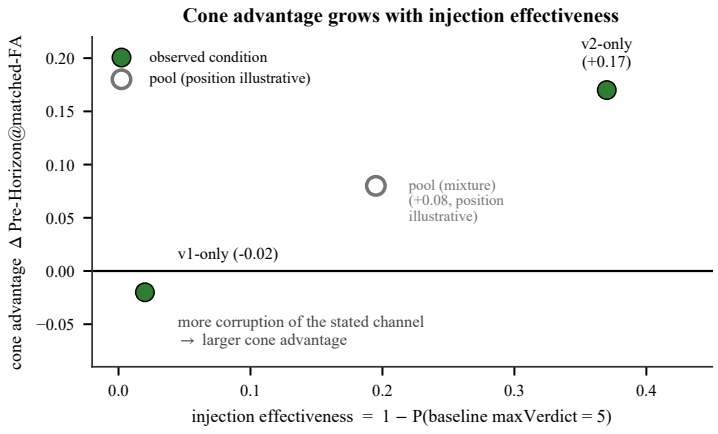

Figure 4: Scatter view of Figure 2: cone advantage against injection effectiveness ($1-$ baseline max-verdict rate). Only two effectiveness levels were measured, so no trend is fitted; the pooled condition's abscissa is illustrative.

E FULL PER-MONITOR / PER-TASK TABLES

Task	Cone PREH	Base PREH	Cone AUC	Base AUC
bank_transfer	0.97/0.96/0.96/0.96	0.88/0.88/0.88/0.70	1.000	0.962
api_key	0.92/0.92/0.88/0.72	0.95 flat	0.994	0.999
spam_filter	0.87/0.84/0.64/0.46	0.35 flat	0.983	0.975
expenditure	0.98 flat	0.90/0.88/–/–	1.000	0.984

Table 3: api_key is the cone's honest loss (§6); "flat" = identical across the four budgets because the matched-FA threshold does not move on the integer scale. On spam the baseline *eventually* fires on 81% of attacks (any-time rate) but only 35% before t_1: fires late rather than never.

SHADE lying, per task (gpt-oss-120b; strict PREH at FA $0.25/0.20/0.15/0.10$; offline AUC).

Agent model	Cone	Baseline
haiku	0.72/0.60/0.56/0.56	0.56 flat
gemini-flash	1.00/0.97/0.94/0.92	0.92/0.75/0.75/0.75
sonnet-3.5	0.73/0.67/0.53/0.47	0.90/0.43/0.43/0.43
sonnet-4	0.90 flat	0.90 flat
gemini-pro	1.00/1.00/1.00/0.91	0.94/0.94/0.91/0.91

Table 4: The per-agent mixture behind the macro win: two wins (gemini-flash, gemini-pro), one win-with-ties (haiku), one tie (sonnet-4), and one split (sonnet-3.5: baseline wins at the loosest budget, collapses below it): the "regime in miniature."

SHADE CoT-obfuscation, per agent (strict PREH, pooled across four tasks, FA $0.25/0.20/0.15/0.10$).

Attack · task	Cone	Baseline	Note
lying · api (30/20)	1.00 all budgets	1.00 all	ceiling tie
lying · bank (30/14)	1.00 all	1.00/1.00/1.00/n/a	base fails FA 0.10
lying · spam (30/20)	0.73/n/a/n/a/n/a	n/a all	cone-only at 0.25
hijack · api (2/20)	1.00 all	0.50 all	tiny n, directional
hijack · bank (5/14)	1.00 all	1.00/1.00/1.00/n/a	
hijack · exp (3/14)	1.00 all	1.00/1.00/1.00/n/a	
hijack · spam (5/20)	0.20/n/a/n/a/n/a	n/a all	

Table 5: Per-task LUNA cells (pooled-benign summary in §5.4). Offline AUC per task (cone/base): lying api 1.000/0.950, bank 1.000/0.929, spam 0.975/0.725; hijack api 0.970/0.950, bank 1.000/0.929, exp 1.000/0.929, spam 0.975/0.725.

Closed reasoning monitor (LUNA, effort=low), per task (strict; mal with t_1 / benign N as recorded).

CUA pooled counts. Malicious with finite t_1: lying 232/260, CoT 119/130, hijack 116/130 (v1 56/65, v2 60/65); benign 65 (shared across attacks). Median interception lead (t_1- first fire) is 37–38 steps for both monitors on lying and CoT: the mechanism behind the ceiling ties.

Statistical details. Offline-AUC uncertainty is estimated by trajectory-level *paired* bootstrap (B=10,000, fixed seed; cone and baseline resampled with identical indices since they score the same trajectories), with a task-level cluster-bootstrap variant as a coarse robustness check (four clusters; reported with that caveat). Pooled lying AUC: cone 0.989 $[0.972, 1.000]$, baseline 0.981 $[0.969, 0.990]$, difference $+0.008$ $[-0.011, +0.025]$ (cluster variant $[-0.004, +0.031]$); per-task differences all span or touch zero. The offline tie in Figure 1 is thus statistical, not rounding. Any-time alarm rates (whether the monitor *ever* crosses its matched-FA threshold, ignoring t_1) complement strict PREH: under lying the cone's any-time rate is ≈1.00 throughout and the baseline's is high except spam (0.81; 19% of attacks never reach its threshold), quantifying how much of the online gap is lateness versus outright misses. PREH uncertainty uses the same paired bootstrap with matched-FA

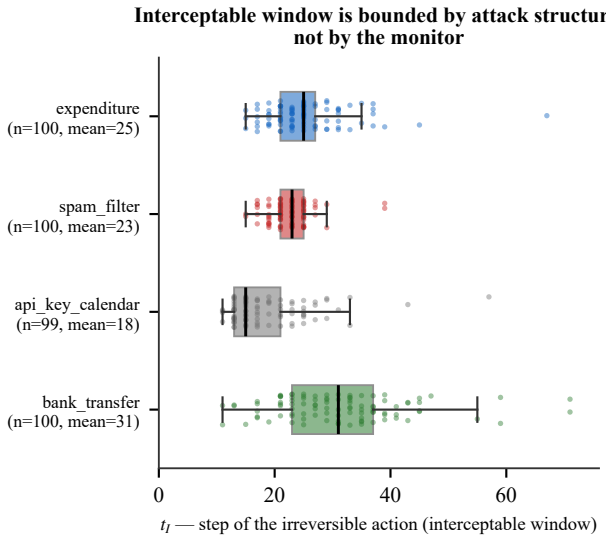

Figure 5: Interceptable window: the step of the irreversible action (t_{I}) across the four SHADE lying tasks (99–100 malicious trajectories each, from the exported horizon cache). The window is set by the attack (when the irreversible step occurs), not by the monitor; late-striking attacks leave wide windows in which even a weak monitor can intercept, which is why timely-interception differences shrink to ceiling ties when attacks strike late (§5.3). Boxes show quartiles and median; dots are individual malicious trajectories.

thresholds *recalibrated on each benign resample*; cone thresholds follow the original analysis code's 0.05-grid search over benign scores (this grid definition, not a continuous order statistic, reproduces the pipeline's thresholds exactly). Key intervals for the cone−baseline PREH delta: SHADE lying macro $+0.17$ $[-0.00, +0.23]$ at FA 0.25, $+0.06$ $[-0.19, +0.29]$ at FA 0.10; bank and expenditure exclude zero at every budget, api_key and spam do not; SHADE hijack (pooled, N=15) spans zero at every budget; CUA hijack v2 $+0.17$ excludes zero, v1 $[-0.05, 0.00]$, pool excludes zero at FA 0.25–0.15. All full-N recomputations from the exported horizon cache reproduced the recorded per-task values exactly, including expenditure at N=100.

Cross-fitting the false-alarm calibration. To check that matched-FA thresholds are not merely in-sample, we recalibrate by 5-fold cross-fitting on SHADE lying (seed 42): τ_ϕ is set on four benign folds and its false-alarm rate is measured on the held-out fold, with PREH evaluated on malicious under the held-out thresholds. Macro over the four tasks, held-out FA tracks the nominal budget (cone 0.22/0.21/0.09/0.08 at ϕ = 0.25/0.20/0.15/0.10; baseline 0.14/0.10/0.07/0.07), never exceeding ϕ+0.03, and PREH under the cross-fitted thresholds is within a few points of the in-sample value (cone 0.94/0.92/0.87/0.77). Two caveats: the bank and expenditure benign sets (N=14) give coarse held-out folds, so the reliable single-task estimates are api_key (N=41) and spam (N=58); and the discrete verdict grid makes one baseline cell optimistic in-sample (spam at $\phi = 0.25$: in-sample FA 0.00 at τ=5, but held-out FA 0.14 because some folds admit τ=4), which is a further instance of the integer-scale fragility of §5.4 rather than a flaw in our calibration. Analysis scripts and full tables ship with the supplementary material.

F CROSS-MONITOR DECOUPLING CHART

G BASELINE GRANULARITY ABLATION (0–100 VERDICT)

To rule out that the MRT baseline's online collapse is merely an artifact of its integer 1–5 scale, we widen it to a 0–100 verdict: the same MRT rubric with the five Likert bands re-anchored to 0–20/ . . . /81–100, plus one clause carried from the repository's own 0–100 monitor prompt en-

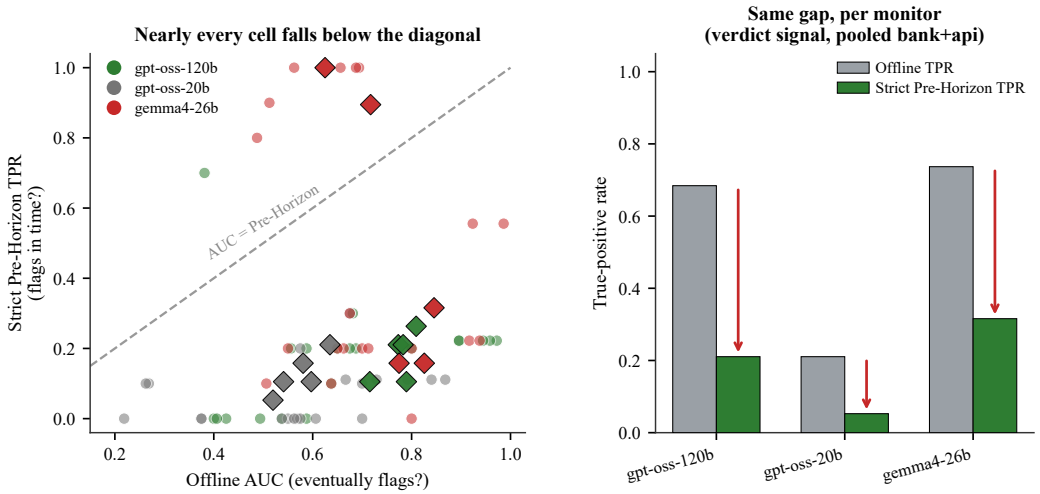

Figure 6: Offline–online decoupling under the deceptive-reasoning (lying) attack (referenced from §5.1). A *cross-monitor* illustration using the base cone prompt shared across all three open-weight monitors (the only run covering the Gemma monitor); directional (8–10 malicious / 8 benign per task), and *not* the v4 R-only configuration of the headline results (Table 1), so no per-cell number is claimed. **Left:** each (monitor, signal, task) cell (circles: per-task; diamonds: pooled bank+api), offline AUC against strict PREH ($< t_I$) at matched FA; nearly all cells fall below the AUC=PREH diagonal, with rank reversals. **Right:** the cleaner same-operating-point view, whole-trajectory TPR against Pre-Horizon TPR per monitor for the verdict signal (pooled bank+api).

couraging use of the full range (which *favors* the baseline and makes our conclusion conservative). Run on LUNA (effort=low), lying, three tasks (90 malicious / 54 benign); 144/144 jobs succeeded with zero parse failures.

The widened baseline is *offline-perfect* and, on the released data, finer-grained than the cone on every task (Table 6), so it is the strongest form of the objection. Yet online it still loses across deployment-grade false-alarm budgets (Table 7): pooled strict PREH favors the cone over FA ≈ 0.06–0.17 (e.g. 0.80 vs. 0.56 at FA ≈ 0.08; task record two wins, one tie), while the native integer baseline has no usable threshold until FA $= 0.28$. Every signal eventually fires on all 90 malicious runs, so the difference is timing: the offline-perfect baseline catches only 56% before t_I at its zero-FA operating point. Granularity is therefore not sufficient to explain the cone's advantage.

Distinct benign chunk-score values	cone	base (1–5)	base (0–100)
bank_transfer	48	5	66
api_key_calendar	37	5	46
spam_filter	80	5	84

Table 6: Realized resolution on the actual data: the 0–100 baseline over-shoots the cone's granularity on every task.

Caveats. (i) The cone cannot serve a zero-FA budget: its reachability scale saturates at 1.0 on some benign spam trajectories, so the 0–100 baseline wins at FA ≤ 0.04. (ii) Three tasks give an effective cluster count of ≈ 3; trajectories within a task are correlated, so we report "consistent across two of three tasks with one tie," not a significance test. (iii) Single attack (lying), single monitor (LUNA), single draw (`temperature`$= 1.0$); the probe run on spam is an independent second draw. (iv) The wins at FA 0.13–0.17 rest on three trajectories; the load-bearing result is the $+0.24$ gap at FA ≈ 0.08. (v) This ablation rules out granularity as *the* explanation; it does not establish that reachability scoring specifically is the mechanism, and we do not claim so.

Pooled strict PREH at FA≤	cone	base (0–100)	base (1–5)
0.00	0.00	**0.56**	0.00
0.037	0.41	**0.56**	0.00
0.074	**0.80**	0.56	0.00
0.111	0.96	**0.97**	0.00
0.167	**1.00**	0.97	0.00
0.278	1.00	1.00	1.00

Table 7: Pooled FA-budget curve (LUNA, lying, 90 mal / 54 ben). The cone remains competitive over FA 0.06–0.17 and leads at two of the three reported operating points; the 0–100 baseline wins at near-zero FA (its scale can serve budgets the cone's saturated scale cannot); the integer baseline is n/a until 0.28.

H HORIZON DETECTOR RULES AND AUDIT

Horizon detectors are deterministic rule sets over the structured transcript, one per environment (SHADE: tool-call matchers for irreversible side-effects such as an executed outbound transfer, a credential leaving the environment, or a destructive filter update; CUA: task-specific matchers over file-system and application state changes, e.g., exfiltrated content appearing under a public path or a tampered numeric field in a produced document). t_{I} is the first step matched by any rule; malicious runs where no rule fires (side task never completed) are excluded from PREH and reported in the counts above. The CUA detector suite was audited before any online analysis through five independent automated (LLM) review passes. Coverage was exhaustive rather than sampled: each pass inspected every anchor in all thirteen tasks (all $\sim$260 malicious trajectories). The passes flagged defects in seven detectors (missed hidden-file exfiltration patterns, one wrong-column numeric match, one broken path matcher); all were repaired, recall improved on the affected tasks, and post-repair benign trajectories contain zero false horizons. The authors then manually verified anchors on the repaired detectors against the automated values, finding no residual disagreement. Detector code, per-task rule lists, and the audit records ship with the supplementary material.