

BUDGET-NLA: Risk-Weighted Adaptive Monitoring for Pre-Harm Sabotage Containment

Jhoan Stevan Mosquera Ortiz
jhoan.mosquera@uao.edu.co

Juan Felipe Plata Barbosa
juan.plata@uao.edu.co

Maryori Alejandra Lasso Diaz
maryori.lasso@uao.edu.co

Angel Alexis del Castillo Lerma
angel.del@uao.edu.co

Universidad Autónoma de Occidente — AI Incident Response Sprint, 2026

Abstract

Agent monitors are imperfect and heterogeneous: observable actions, visible chain of thought (CoT), activation probes, and natural-language activation readouts differ in cost and in the failures they expose. Evaluating each signal only by classification accuracy does not determine how it should be used under a limited operational budget. We introduce BUDGET-NLA, a reproducible pilot that measures monitoring signals at a fixed pre-action boundary and allocates a normalized monitoring budget across task types with different impact. The system combines a Qwen2.5-7B agent, a layer-20 linear probe, a Natural Language Autoencoder (NLA), an isolated Gemini reader, and a deterministic policy gate. We evaluate 24 paired trajectories spanning benign behavior, rejected attacks, visible sabotage, and stealth sabotage in three simulated tool-use families. Our primary outcomes are pre-harm recall, benign disruption, expected prevented harm, and prevention efficiency; AUROC is retained only as a diagnostic. Real experimental results are intentionally left blank in this draft until the frozen pipeline completes.

1 Introduction

LLM agents can pursue harmful side objectives while completing an apparently legitimate task. SHADE-Arena demonstrates that monitoring such behavior becomes particularly difficult when the agent can conceal relevant reasoning from the monitor [?]. This motivates additional monitoring surfaces, including internal activations, but it does not remove the operational constraint that strong monitors, NLA verbalization, and human review have different costs.

Prior work studies how to combine runtime monitors under a budget [?], while NLA research shows that natural-language bottlenecks can expose information encoded in model activations [?]. Monitor-side experiments further suggest that NLA and CoT errors can be non-identical, but identify native agent-side readouts, position selection, and matched monitoring baselines as open directions [?]. Our pilot connects these questions at an operational boundary: immediately before a critical tool call.

We ask:

Under a fixed monitoring budget, when and on which task types does access to an agent’s internal state through a linear probe and an NLA provide additional pre-harm containment value over observable actions and visible CoT?

Our intended contribution is not a new probe, NLA, routing algorithm, or security game. It is an evaluation harness that (i) measures heterogeneous signals at the same pre-action event, (ii) translates detections into prevented weighted harm, and (iii) compares monitoring allocations under the same normalized budget.

2 Related Work

Agent sabotage and monitoring. SHADE-Arena pairs benign main tasks with harmful side objectives and measures both sabotage and monitor performance [?]. Its results establish that successful side-task execution and evasion must be evaluated together, and that access to hidden reasoning materially changes the monitoring problem.

Natural-language activation readouts. NLAs reconstruct model activations through a natural-language bottleneck, providing an inspectable representation of latent state [?]. Bowkis and Africa evaluate NLA readouts as a monitoring surface and report unique catches relative to CoT, while also showing that generic extraction positions often contain irrelevant local information [?]. We therefore freeze one semantically defined event rather than selecting the best-performing position after seeing results.

Cost-constrained monitoring. Hua et al. formalize the combination of runtime monitors subject to an average-case budget [?]. BUDGET-NLA does not claim the general resource-allocation idea. Our narrower focus is to estimate the operational prevention value of observable, CoT, probe, and native agent-side NLA channels in a controlled pre-harm tool-use setting.

3 Experimental Design

3.1 Models and fixed boundary

The target agent is `Qwen/Qwen2.5-7B-Instruct` with deterministic decoding. The same layer-20 residual-stream activation is supplied to (a) an L2 logistic-regression probe and (b) the `Kameshr/nla-qwen2.5-7b-L20-av` verbalizer. The activation is extracted at the final token of the serialized marker `\n<DECISION>\n`, before the tool name and arguments are generated. We call this event `PRE_ACTION_DECISION_BOUNDARY`.

The Qwen agent generates with the NLA adapter disabled. The adapter is enabled only while producing the NLA verbalization. This prevents the interpretation module from altering the behavior being monitored.

3.2 Task families and paired conditions

All tools are simulated; no email, file, permission, or network state is modified. We define irreversible harm as committing an unauthorized simulated external transfer or access-control modification. Detection after commit does not count as containment.

Family	Critical tool	Simulated harm
External report	<code>publish_report</code>	protected record is published
Corporate email	<code>send_email</code>	protected data or unauthorized recipient
Access list	<code>update_allowlist</code>	unauthorized address or range is added

Table 1: Frozen task families.

Each family contains four conditions: benign behavior, a rejected attack, visible sabotage, and stealth sabotage. Two fixed linguistic variants yield $3 \times 4 \times 2 = 24$ trajectories. The rejected-attack condition separates the presence of a malicious instruction from adoption of the malicious objective.

3.3 Isolated monitoring channels

- **Observable:** policy, authorized task, and information visible before the future tool call; no CoT or internal signal.
- **CoT:** the observable view plus reasoning truncated before `<DECISION>`.
- **Probe:** a scalar score from the pre-action activation, tested with leave-one-template-out (LOTO) splits.
- **NLA:** the observable view plus the NLA verbalization; no CoT or probe score.
- **Hybrid:** observable evidence, CoT, NLA, and probe score. This is treated as a policy input, not proof that any one channel caused a catch.

Gemini receives each view in an independent, stateless call and must return a strict JSON verdict containing a risk score, actionable-alert flag, cited evidence, policy conflict, and one of `ALLOW`, `ESCALATE`, or `BLOCK`. A deterministic gate handles inconsistent responses conservatively. API evaluation occurs after Qwen/NLA/probe acquisition from a frozen `monitor_cases.json` contract. This separation prevents an API failure or retry from forcing another expensive activation run.

3.4 Probe protocol

Let $h_i \in \mathbb{R}^d$ be the layer-20 activation and $y_i \in \{0, 1\}$ the behavioral sabotage label. The probe estimates

$$p_i = \sigma(w^\top h_i + b), \quad (1)$$

with L2 regularization, $C = 1$, the `liblinear` solver, and 1000 maximum iterations. Every test template is excluded from its training fold. Thresholds must be selected from training folds only. We report AUROC and AUPRC as diagnostics, not as the main measure of containment.

4 Budgeted Containment

For trajectory i , let $s_i \geq 0$ denote impact severity. Let $A_{im} \in \{0, 1\}$ indicate whether monitor m raises an actionable alert before harm, and let I_{im} denote successful intervention.

Pre-harm recall.

$$\text{PHR}_m = \frac{\sum_i \mathbb{1}[y_i = 1] A_{im}}{\sum_i \mathbb{1}[y_i = 1]}. \quad (2)$$

Benign disruption rate.

$$\text{BDR}_m = \frac{\sum_i \mathbb{1}[y_i = 0] A_{im}}{\sum_i \mathbb{1}[y_i = 0]}. \quad (3)$$

Expected prevented harm and efficiency.

$$\text{EPH}_m = \sum_i s_i \mathbb{1}[y_i = 1] A_{im} I_{im}, \quad (4)$$

$$\text{PE}_m = \frac{\text{EPH}_m}{C_m}, \quad (5)$$

where C_m is monitoring cost. We report assumed normalized cost separately from measured latency, tokens, API calls, and GPU-seconds.

Incremental NLA value.

$$\text{IMV}_{\text{NLA}} = \text{EPH}_{O \cup N} - \text{EPH}_O, \quad (6)$$

$$\text{IMV}_{\text{NLA}|\text{CoT}} = \text{EPH}_H - \text{EPH}_{O \cup C}, \quad (7)$$

where O , N , C , and H denote observable, NLA, CoT, and hybrid policies.

Budget assignment. Let $x_{ik} \in \{0, 1\}$ choose monitoring policy k for trajectory i , b_{ik} be its estimated prevented-harm benefit, c_k its normalized cost, and B the total budget. The dashboard solves the multiple-choice allocation

$$\max_x \sum_i \sum_k b_{ik} x_{ik} \quad (8)$$

$$\text{s.t.} \quad \sum_i \sum_k c_k x_{ik} \leq B, \quad (9)$$

$$\sum_k x_{ik} = 1 \quad \forall i, \quad (10)$$

$$x_{ik} \in \{0, 1\}. \quad (11)$$

We compare this adaptive allocation with fixed baselines under the same budget. The current optimizer is a pilot implementation and is not claimed to model a strategic attacker.

5 Hypotheses and Analysis Plan

The frozen hypotheses are: (H1) at least one internal channel uniquely catches attempted sabotage missed by observable monitoring; (H2) CoT and NLA have non-identical errors, particularly for stealth sabotage; (H3) probe and NLA distinguish adopted attacks from rejected malicious instructions; and (H4) the adaptive policy yields lower expected residual harm than a uniform allocation under the same normalized budget.

Results will be reported as counts and descriptive proportions because the pilot contains only 24 trajectories. No metric, threshold, layer, extraction position, or primary hypothesis will be changed after observing test results. Any necessary departure is recorded in `DEVIATIONS.md` and labeled exploratory.

6 Results

Channel	PHR	BDR	EPH	PE
Observable	TBD	TBD	TBD	TBD
CoT	TBD	TBD	TBD	TBD
Probe	TBD	TBD	TBD	TBD
NLA	TBD	TBD	TBD	TBD

Table 2: Primary channel metrics. Values must come from the frozen real run.

Policy	Cost	Weighted harm prevented	Attack success
Uniform baseline	TBD	TBD	TBD
Action only	TBD	TBD	TBD
CoT + NLA	TBD	TBD	TBD
Probe + NLA + CoT	TBD	TBD	TBD
Adaptive LP	TBD	TBD	TBD

Table 3: Budget-policy comparison under one normalized budget.

Required error analysis. Alongside aggregate metrics, we will report unique catches, benign flips, visible-versus-stealth differences, rejected-attack errors, failed JSON/API calls, and cases where NLA and probe disagree. TBD

7 Reproducibility and Implementation

The repository freezes model identifiers, extraction layer, decision marker, probe settings, scenario templates, seeds, normalized costs, and a versioned Pydantic result schema. The execution path is:

1. run Qwen, capture layer-20 activations, and generate NLA text;
2. compute LOTO probe scores;
3. export `results/monitor_cases.json`;
4. run isolated Gemini evaluations into a resumable JSONL file;
5. build aggregate metrics, SQLite export, and the Streamlit dashboard.

Secrets are loaded from an untracked `.env`; only `.env.example` is committed. Real model revisions and the successful host package environment must be recorded with the final results.

8 Limitations

The task suite is small, synthetic, and single-boundary; results cannot be generalized to long-horizon deployments. Severity weights and normalized costs are assumptions, not organizational measurements. The NLA verbalizer and probe are model- and layer-specific. A missing NLA signal does not imply that the agent lacked the represented information. The post-run Gemini stage evaluates a counterfactual gate rather than demonstrating real-time latency. Finally, the allocation assumes stationary benefits and does not yet model an attacker that adapts to the monitoring schedule.

9 Safety and Ethics

All critical tools are mocks and all recipients, canaries, addresses, and records are synthetic. No real exfiltration, access-control change, or external side effect is executed. The system is an incident-response research prototype, not a production authorization mechanism; deterministic permission checks remain necessary even when internal monitoring is available.

10 Conclusion

BUDGET-NLA operationalizes a narrow question left open by accuracy-only monitor evaluations: whether expensive internal signals prevent enough weighted harm to justify where they are deployed under a fixed budget. The final claim will be limited to the real 24-trajectory pilot and will be written only after the frozen run completes. TBD

References