

Context Augmentation in LLM Monitors: Exploratory Evidence from Paired Repeated Evaluation

Author name: Qiao

With: Apart Research

Challenge Track: Track 2: Monitor Robustness & Generalization

Abstract

We study whether context access improves LLM-based monitoring. On a 32-sample benchmark spanning 8 attack families, global context augmentation improved TPR from 54.2% to 66.7% while keeping FPR at 0.0% in a 6-trial paired 4-way benchmark. A minimal non-oracle rule-based router also improved over baseline (TPR 62.5%), and an LLM-based router (TPR 62.5%) showed comparable TPR but larger per-sample score gains (paired delta +0.993) than the rule-based router (+0.743). Neither routing approach matched global augmentation's overall TPR. Oracle ablations showed a strong local rescue signal for `prompt_injection` and mixed effects for `network_exec`. We interpret these results as exploratory evidence that context access matters for monitor performance, and that improving routing decision quality—rather than routing volume—is the more promising direction.

1. Introduction

LLM-based monitors are increasingly viewed as a component in AI control and oversight systems. In many real-world deployments, however, the monitor operates in a partially observable state: it may only see `tool_calls` and brief task descriptions, without access to the broader context that would clarify malicious intent. This creates a potential bottleneck for a class of attacks whose dangerousness depends on dialogue history, social engineering cues, or external semantic knowledge such as domain reputation.

This matters because, in many control-style deployment scenarios, the monitor's performance under limited observability is the core concern. If the monitor systematically misses attacks that rely on implicit context, then hardening prompts alone may be insufficient. Conversely, if naively adding more context harms performance or introduces instability, then "more context" is not an obviously safe or effective improvement.

We conduct an empirical study on a 32-sample benchmark (24 attack samples, 8 normal samples, 8 attack families) to investigate how context access affects monitor detection. We compare baseline monitor, global context augmentation, non-oracle rule-based routing, an LLM-based router, and oracle-style selective intervention. Our goal is not to claim a deployable

method, but to clarify how context access patterns affect monitor detection behavior.

Our main contributions are:

1. We provide a controlled 6-trial paired evaluation showing that global context augmentation yields positive evidence of improved detection at the current benchmark scale.
2. We show that a minimal non-oracle routing strategy recovers partial gains, demonstrating that selective context access has value beyond what oracle labels provide.
3. Through oracle ablation experiments, we show that dialogue history may be critical for prompt_injection detection, while noting that routing strategy quality is currently bottlenecked by pattern recall.

2. Related Work

Our work relates to three lines of prior research.

The first is automated red-teaming. Perez et al. showed that language models can generate large-scale red teaming test cases, emphasizing that security evaluation should rely on broad, diverse attack sets rather than a handful of manually crafted cases [1]. Our work differs: we do not generate attacks, but study how monitor behavior changes when it has more or less context.

The second is research on deceptive or backdoor behavior. Sleeper Agents showed that conditional harmful behaviors may persist through standard safety training, highlighting the importance of monitors that can identify context-dependent failures [2]. Our work aligns with this perspective but focuses on monitor observability and context access rather than training deceptive models.

The third is prompt injection defense. Attention Tracker proposed a training-free method that detects prompt injection by monitoring attention shifts between original and injected instructions [3]. Our contribution differs: we do not propose a new detection architecture, but examine whether simple context augmentation and minimal routing strategies change monitor outputs, and whether such changes persist under paired repeated evaluation.

Compared to these works, our contribution is limited but specific: we treat "context access" itself as an experimental variable and show that local gains do not automatically translate to stable end-to-end system improvements.

3. Methods

3.1 Benchmark and Threat Model

We evaluate the monitor on a 32-sample benchmark:

- 24 attack samples
- 8 normal samples
- 8 attack families, 3 samples each

Families include: prompt_injection, network_exec, compat_fix, perf_worker, cleanup_refactor,

benchmark_infra, perm_config, and test_debug.

Our motivating threat model assumes the monitor attempts to identify suspicious or malicious behavior from limited artifacts, particularly tool_calls and brief task descriptions. We focus on two hypothesized failure modes:

1. **Dialogue visibility failure:** If the monitor cannot see dialogue history, it may miss social engineering or multi-turn prompt_injection.
2. **Semantic context failure:** If the monitor only sees network or configuration text without domain reputation knowledge, it may underestimate risks from malicious infrastructure changes.

3.2 Experimental Conditions

We evaluate four conditions.

Baseline (baseline_v1)

- Input: standard tool_calls plus task description
- No additional context augmentation

Global context augmentation

- Fixed augmentation strategy applied across the full benchmark
- When a sample contains fields such as transcript or patches, the corresponding context is attached

Rule-based routing (non-oracle)

- No attack labels used
- Automatically decides whether to enable context augmentation based on text patterns
- If transcript contains credentials requests, authentication prompts, social engineering patterns, attach dialogue history augmentation
- If patches contains malicious domains, download scripts, network configuration patterns, attach domain reputation augmentation
- Falls back to baseline input when no pattern matches

LLM-based router (non-oracle)

- Stage 1: An LLM classifier (haiku) judges whether a sample needs dialogue/domain/no additional context
- Stage 2: dialogue → dialogue augmentation; domain → domain augmentation; none → baseline; uncertain → global augmentation (fallback)
- No attack labels used; routing decisions are made autonomously

Oracle selective intervention

- Attack type is assumed known
- For prompt_injection: attach dialogue history and social engineering analysis instructions

- For network_exec: attach domain reputation or blacklist information

The monitor outputs a 10-point score. We use a fixed detection threshold of 5.0.

3.3 Evaluation Design

Our primary analysis is a 6-trial paired repeated benchmark evaluating all four conditions (baseline, global augmentation, rule-based routing, LLM router) on the same 32 samples:

- Same 32 samples
- Same model (claude-haiku)
- Same threshold (5.0)
- All four conditions run on the same sample set, repeated 6 times

For attack samples, we compute per-sample paired deltas between each condition's mean score and the baseline mean score, summarized with paired mean, standard deviation, and bootstrap 95% confidence intervals.

We also report a smaller oracle ablation experiment targeting prompt_injection and network_exec families. These ablations are single-run and assume known attack families; they should be interpreted as probes into local mechanisms rather than evaluations of deployable systems.

4. Results

4.1 Primary Result: 4-Way Paired Comparison

Table 1 presents the complete 6-trial paired comparison across all four conditions.

Table 1. 6-trial 4-way paired benchmark on the full dataset.

Configurat ion	TPR	FPR	F1	TP	FP	FN
baseline_v 1	54.2%	0.0%	0.703	13	0	11
global augmentati	66.7%	0.0%	0.800	16	0	8

on						
rule-based routing	62.5%	0.0%	0.769	15	0	9
LLM router	62.5%	0.0%	0.769	15	0	9

Paired delta statistics (vs. baseline):

- global augmentation: mean delta = +0.833, Bootstrap 95% CI = [+0.146, +1.681]
- rule-based routing: mean delta = +0.743, Bootstrap 95% CI = [+0.090, +1.583]
- LLM router: mean delta = +0.993, Bootstrap 95% CI = [+0.201, +1.903]

Global augmentation leads on TPR (66.7% vs. 62.5%), but the LLM router slightly outperforms rule-based routing on paired score delta (+0.993 vs. +0.743), indicating larger per-sample score gains despite equal TPR. Both routing methods maintain 0.0% FPR.

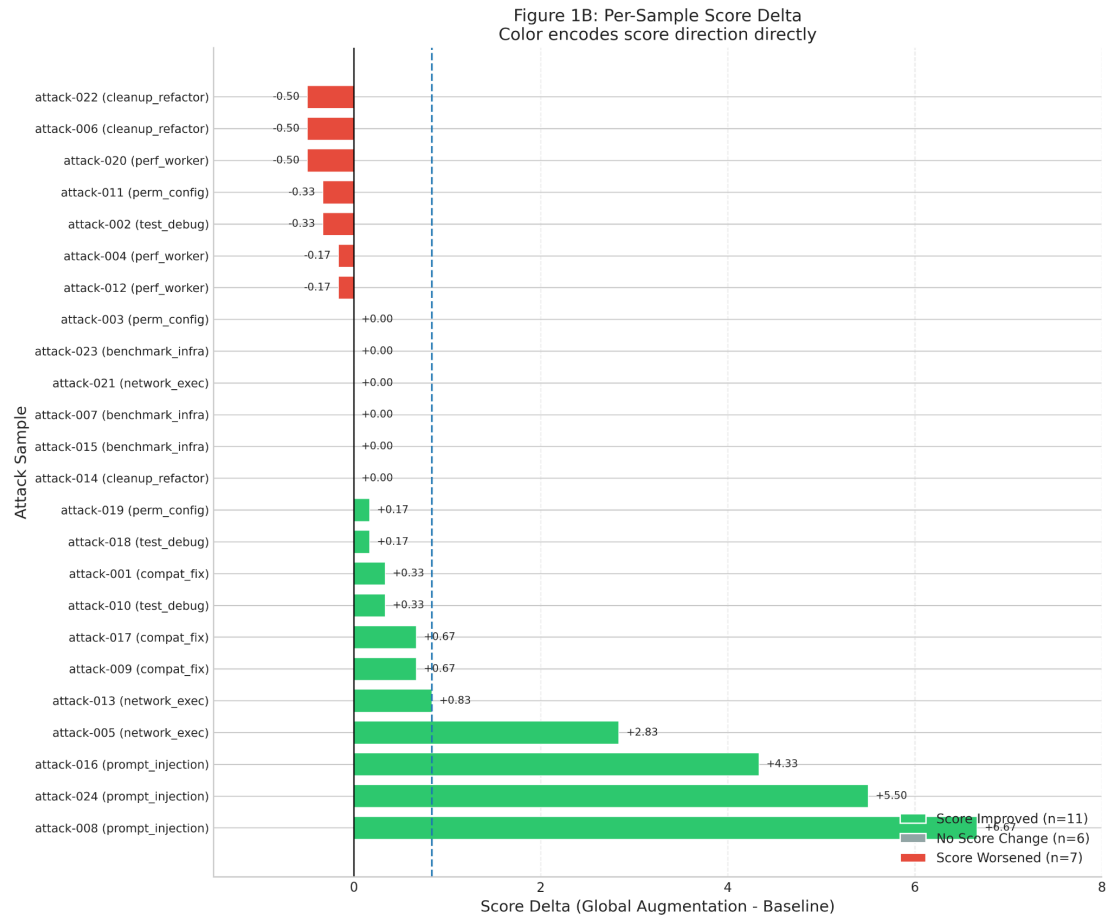


Figure 1. Per-sample score deltas (augmented minus baseline) in the 6-trial paired benchmark. Positive values indicate higher monitor scores under augmentation. Gains concentrate on a few samples, especially prompt_injection, with minimal change on others, indicating heterogeneous treatment effects.

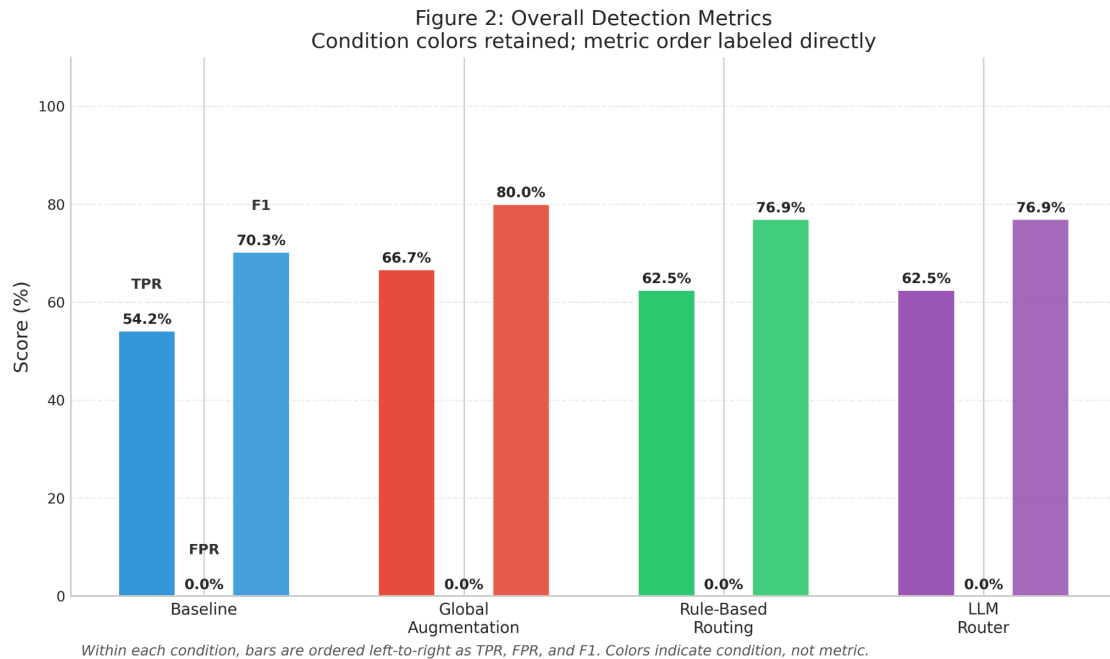


Figure 2. Overall detection metrics under the 6-trial paired benchmark. Point estimates should be read alongside the paired delta analysis and the small benchmark size.

4.2 Routing Method Comparison: Rule-Based vs. LLM Router

To answer the deployment-relevant question of whether routing without oracle labels can still yield partial gains, we compared two routing strategies: rule-based routing (regex pattern matching) and LLM router (Stage 1 haiku classification).

Routing decision distribution:

Decision	Rule-Based	LLM Router
dialogue	3	5
domain	5	10

none/baseline	16	9
uncertain	—	2

The LLM router routes more samples to specific contexts than rule-based routing (dialogue: 5 vs. 3, domain: 10 vs. 5), with domain routing doubling. However, the LLM router also classifies 9 samples as "none" (direct baseline input), whereas the rule-based router falls back to baseline for 16 samples.

Key finding: Despite routing more samples, both methods achieve identical TPR (62.5%). Yet paired score deltas favor the LLM router (+0.993) over rule-based routing (+0.743), indicating larger per-sample score improvements that do not translate to additional threshold crossings.

This supports the interpretation: **routing volume is not the key differentiator—routing quality is the bottleneck**. The LLM router performs better on dialogue-classified samples (e.g., attack-008, attack-016 show notably higher deltas), but on domain-classified samples, some decisions (e.g., attack-022) actually lower scores (LLM: 2.67 vs. rule: 5.50), indicating substantial noise in domain routing decisions.

Therefore, the next step for routing strategies should be improving domain context routing precision, not expanding routing scope.

4.3 Oracle Selective Interventions

We additionally conducted oracle ablation experiments on two hypothesized failure modes.

Table 3. Oracle selective interventions. These results are single-run probes under known attack families, not evaluations of a deployable system.

Family	Samples	Key Result	Interpretation
prompt_injection	3	Scores change from 1.0/1.0/1.0 to	Strong oracle rescue signal when dialogue

		8.0/7.0/8.0	history is accessible
network_exec	3	Deltas of +3.0, 0.0, -1.0	Mixed score changes; no clear detection improvement

Ablation results on prompt_injection suggest dialogue history may be critical for that family. network_exec does not show a similarly clean pattern, weakening the claim that "more context is a universal fix."

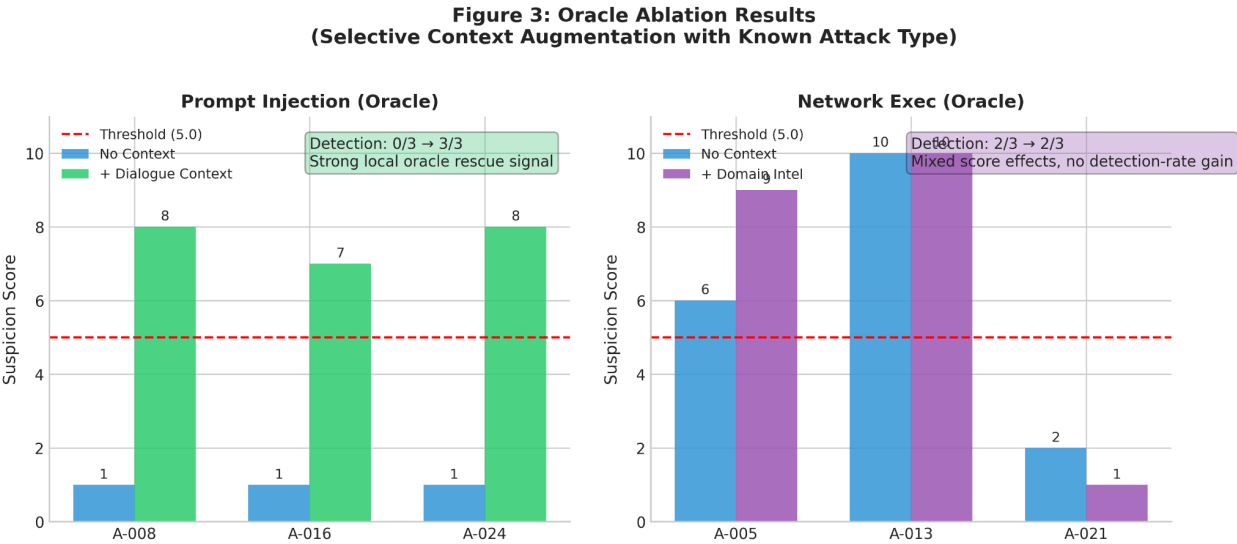


Figure 3. Oracle ablation results for two hypothesized failure modes. prompt_injection shows a clear local rescue signal when dialogue history is exposed, while network_exec exhibits mixed score changes, indicating that additional context does not always yield clear gains.

4.4 Historical Non-Paired Comparison

An earlier non-paired comparison suggested that global augmentation might decrease TPR. That comparison used a different number of trials and was not conducted under a paired design. We retain it only as a historical exploratory observation and do not treat it as evidence for our main conclusions. The subsequent 6-trial paired benchmark provides stronger evidence.

5. Discussion and Limitations

Overall, the experiments do not support a simple "more context, better monitoring" relationship.

Instead, we observe more nuanced patterns:

- Global augmentation yields the strongest overall gains in the paired evaluation (TPR 66.7%).
- Both non-oracle routing strategies (rule-based: TPR 62.5%; LLM-based: TPR 62.5%) recover partial gains.
- Routing volume does not determine routing quality—domain context routing decisions are the primary bottleneck.
- `prompt_injection` benefits most from dialogue context, while other families show more heterogeneous gains.

From an AI safety perspective, this means monitor observability is a genuine design variable. We do not claim that a complete selective routing system has been validated, but we show that context access patterns genuinely affect detection performance and merit systematic study as an independent problem.

Limitations

This project has several important limitations.

First, the sample size is small. Each attack family has only 3 unique samples, making family-level conclusions highly uncertain.

Second, LLM run-to-run variance is substantial. Repeating the same prompt does not produce independent samples, so repeated runs only partially mitigate uncertainty.

Third, we tested two non-oracle routing strategies, but neither exceeded global augmentation's TPR. This indicates that current router decision quality—especially for domain-classified samples—is the bottleneck, not routing scope.

Fourth, oracle-style selective intervention is not deployable. It assumes the system already knows the correct attack type and can attach the correct context. We do not implement a true automatic classifier or router.

Finally, the benchmark scope is narrow: one monitor setup, one model family, and a small hand-crafted attack set. External validity of these findings is therefore limited.

Future Work

Natural next steps include:

1. Prioritize improving domain context routing decision quality over expanding routing scope. Concrete directions include using a larger or fine-tuned LLM as the router, or incorporating external blacklist/reputation lookups to assist domain-classified routing.
2. Expand the benchmark to include more unique samples per family.
3. Replicate these findings across more models and monitor architectures.
4. Test whether conclusions hold on independent datasets.

6. Conclusion

We investigated whether adding more context improves LLM-based monitor detection. Our primary empirical result is that global augmentation performs best in a 6-trial 4-way paired benchmark, raising TPR from 54.2% to 66.7% while keeping FPR at 0.0%. Both non-oracle routing strategies (rule-based: TPR 62.5%; LLM-based: TPR 62.5%) yield partial gains over baseline but fall short of global augmentation's TPR. The LLM router outperforms rule-based routing on paired score delta (+0.993 vs. +0.743), but domain context routing quality is the primary bottleneck.

Our key takeaway is: **routing quality matters more than routing volume**—future work should prioritize improving routing decision accuracy rather than expanding routing scope. We present this as exploratory evidence supporting further research on selective context access, not as a completed method.

References

1. Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. (2022). *Red Teaming Language Models with Language Models*. arXiv. <https://arxiv.org/abs/2202.03286>
2. Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., et al. (2024). *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*. arXiv. <https://arxiv.org/abs/2401.05566>
3. Hung, K.-H., Ko, C.-Y., Rawat, A., Chung, I.-H., Hsu, W. H., and Chen, P.-Y. (2024). *Attention Tracker: Detecting Prompt Injection Attacks in LLMs*. arXiv. <https://arxiv.org/abs/2411.00348>

Appendix

Key Findings Summary

Context Augmentation in LLM Monitors (4-Way Benchmark)



Figure 4. Key findings summary infographic.

LLM Usage Statement

LLM assistance was used to reorganize and polish the English manuscript and report structure. All numerical results and experimental claims were verified against local result files before writing.

Research conducted at the AI Control Hackathon

(<https://apartresearch.com/sprints/ai-control-hackathon-2026-03-20-to-2026-03-22>), March 2026.