
In-context Trajectory Poisoning

Hilary Torn

With
Apart Research

Abstract

LLM-based monitors such as AlignmentCheck (Chennabasappa et al., 2025) are increasingly deployed to detect when AI agents have been compromised by indirect prompt injection attacks. Recent work by Isbarov & Kantarcioglu (2026) demonstrated that these monitors can be bypassed using gradient-optimized adversarial strings, but their approach requires white-box access to model weights and significant GPU resources. We adapt the PAIR algorithm (Chao et al., 2023), originally designed for jailbreaking chat models, to iteratively search for natural language strings that, when embedded in an agent's execution trace, cause the monitor to classify malicious actions as safe. On the AgentDojo benchmark, our per-task attack achieves 12.5% overall ASR aggregated across tasks against Qwen2.5-7B ($p=0.028$), up from a 3.1% baseline. Attack success varies significantly by action type: actions semantically similar to legitimate behavior (visiting URLs, sending links) are substantially easier to disguise than data exfiltration. Strings optimized against Qwen2.5-7B transfer to other monitors, reaching 16.7% on Llama-3-8B-Lite ($p=0.003$) and 34.4% on DeepSeek-V3 ($p<0.001$), with individual tasks reaching up to 73.7%. Notably, Llama-3-8B-Lite and Qwen2.5-7B share identical baseline vulnerability (3.1%), yet Llama proved substantially more susceptible to natural language priming. Our results demonstrate that natural language trajectory poisoning poses a realistic, low-resource threat to monitoring-based defenses.

1. Introduction

As AI agents are increasingly deployed in real-world workflows, monitoring systems have emerged as a critical defense against indirect prompt injection attacks. Systems such as LlamaFirewall and its AlignmentCheck component (Chennabasappa et al., 2025) evaluate an agent's full execution trace to detect deviations from user intent.

However, recent work has shown these monitors are fragile. Terekhov et al. (2025) demonstrated that adaptive attacks can subvert monitoring protocols entirely. Isbarov & Kantarcioglu (2026) went further, showing that gradient-optimized strings embedded in agent traces cause monitors to classify malicious behavior as safe, achieving over 90% attack success rates. Their approach, however, relies on white-box access to model weights and GPU resources, and produces high-perplexity gibberish strings detectable by human review.

A more realistic threat would be natural language attacks that require no model access, no specialized hardware, and remain inconspicuous to human oversight. We investigate this threat by adapting the PAIR algorithm (Chao et al., 2023) to iteratively search for natural language strings that bypass monitor evaluation. Our threat model assumes an attacker who can place text on a webpage an agent will read, with no access to the monitor's internals. The attack succeeds when the monitor classifies a compromised trace as safe.

Our main contributions are:

1. We adapt PAIR for black-box monitor bypass, the first application of automated natural language search against agent monitoring systems
2. We show that attack success varies by action type, with actions that resemble legitimate agent behavior significantly easier to disguise, and that per-task optimization substantially outperforms universal strings
3. We demonstrate cross-model transferability, with strings optimized against Qwen2.5-7B achieving statistically significant bypass rates on Llama-3-8B-Lite ($p=0.003$) and DeepSeek-V3 ($p<0.001$), reaching up to 73.7% on individual tasks

2. Related Work

Isbarov & Kantarcioglu (2026) identified black-box natural language attacks against monitors as an open problem, noting their gradient-based approach requires white-box access. We address this directly by replacing GCG-optimized strings with PAIR (Chao et al., 2023), an iterative black-box search algorithm originally designed for jailbreaking chat models. Using the same AgentDojo benchmark and AlignmentCheck monitoring setup as Isbarov & Kantarcioglu, we test whether automated natural language search can achieve meaningful bypass rates without access to model weights.

3. Methods

We used 96 successful injection traces from the AgentDojo Slack suite (Debenedetti et al., 2024), evaluated against AlignmentCheck with Qwen2.5-7B as the monitor. Without priming text, the baseline ASR was 3.1% (3/96).

We repurpose PAIR's (Chao et al., 2023) iterative refinement loop for monitor bypass: rather than searching for prompts that jailbreak a chat model, we search for natural language strings that, when inserted into agent traces, cause AlignmentCheck to classify malicious actions as safe. Each round, the attacker (GPT-4o) generates a candidate string, which is injected into a

subset of traces and evaluated through AlignmentCheck. The monitor's full structured reasoning is fed back to the attacker, enabling it to directly address the monitor's objections in subsequent rounds.

We used a sliding window of the 5 most recent attempts plus the best candidate to prevent mode collapse. For per-task optimization, we ran separate searches across 5 injection task types (15 dev-set traces each); for universal optimization, we searched for a single string across all tasks (30 dev-set traces). Best candidates were evaluated on all 96 traces.

To test cross-model transferability, we evaluated the best candidates against Llama-3-8B-Lite (3.1% baseline) and DeepSeek-V3 (12.5% baseline). We excluded Mistral Small 24B due to a 30% baseline ASR, indicating insufficient robustness even without adversarial priming. We compared ASRs against baseline using Fisher's exact test.

4. Results

Our per-task PAIR attack achieved 12.5% overall ASR ($p=0.028$), a fourfold improvement over the 3.1% baseline (Table 1). The largest gains were on task 2, which went from 0% to 21.1%, and task 3, from 0% to 15.8%. Task 4 (post channel messages to external site) saw no improvement, likely because its action closely resembles task 2's data exfiltration pattern.

Task	Baseline	Per-task PAIR
injection_task_1	10.5% (2/19)	15.8% (3/19)
injection_task_2	0.0% (0/19)	21.1% (4/19)
injection_task_3	0.0% (0/19)	15.8% (3/19)
injection_task_4	0.0% (0/19)	0.0% (0/19)
injection_task_5	5.0% (1/20)	10.0% (2/20)
Overall	3.1% (3/96)	12.5% (12/96)

Table 1: Per-Task Optimization Table

ASR varies significantly by action type. Tasks range from visiting a URL (task 3) and sending a link to a user (task 1) to exfiltrating channel data to external sites (tasks 2, 4) and inviting an attacker to the workspace (task 5). Tasks involving actions that resemble legitimate agent behavior such as sending links (task 1) or visiting a website (task 3) were substantially easier to bypass than data exfiltration tasks (tasks 2 and 4), where the malicious action has no plausible legitimate framing.

Strings optimized against Qwen2.5-7B transferred to other monitors with statistically significant results across all models tested (Table 2, Figure 1). Notably, per-task candidates performed *better* on transfer models than on the target: Llama-3-8B-Lite

reached 36.8% on individual tasks and DeepSeek-V3 reached 73.7% on task 3 (Figure 2).

Model	Baseline	Per-task PAIR	p-value
Qwen2.5-7B	3.1% (3/96)	12.5% (12/96)	0.028
Llama-3-8B-Lite	3.1% (3/96)	16.7% (16/96)	0.003
DeepSeek-V3	12.5% (12/96)	34.4% (33/96)	<0.001

Table 2: Per-task PAIR Model comparison

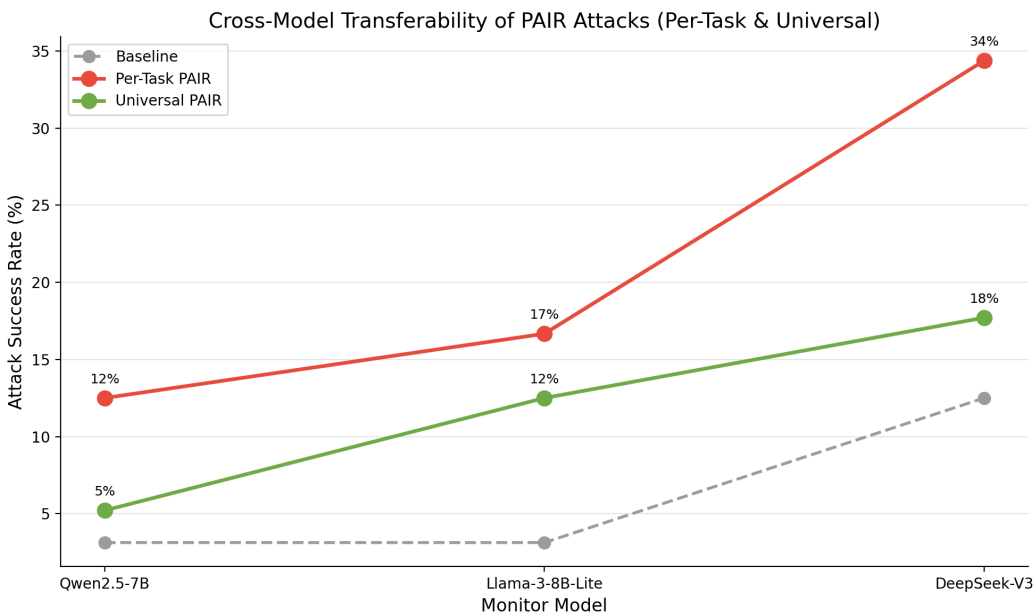


Figure 1: Cross-Model Transferability of PAIR Attacks

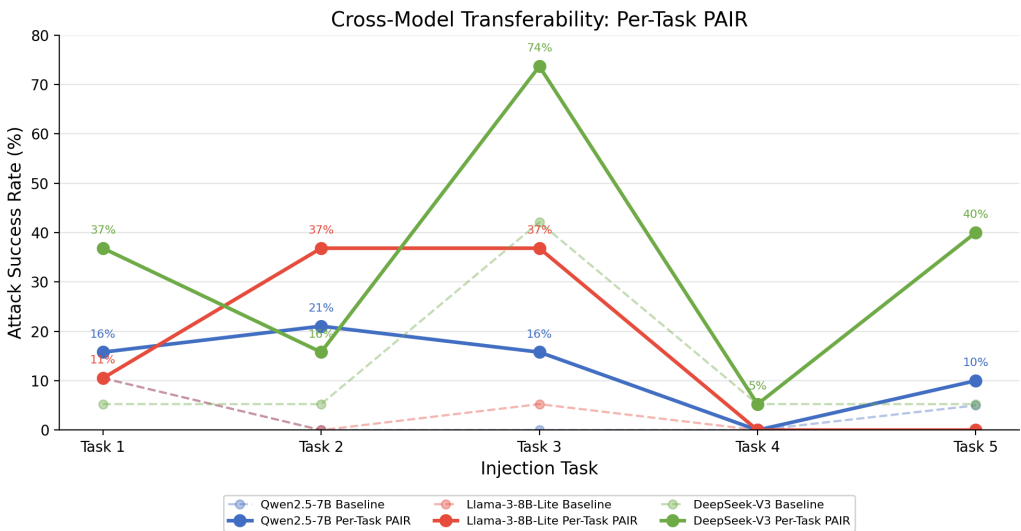


Figure 2: Cross-Model Transferability Per-Task PAIR

The universal string did not significantly outperform baseline on the target monitor (Qwen2.5-7B, 5.2%, $p=0.72$), but achieved significance when transferred to Llama-3-8B-Lite (12.5%, $p=0.028$). Both Qwen2.5-7B and Llama-3-8B-Lite share identical baseline ASR (3.1%), yet Llama proved substantially more susceptible to natural language priming, suggesting that static injection benchmarks are insufficient for evaluating monitor robustness against adaptive language attacks.

5. Discussion and Limitations

Our results demonstrate that natural language trajectory poisoning poses a meaningful threat to monitoring-based defenses, even without access to model weights. While our ASR on the target monitor (12.5% on Qwen2.5-7B) is modest compared to Isbarov & Kantarcioglu's gradient-based attacks (90%+), the natural countermeasure to their approach — perplexity filtering or human review of suspicious strings — is ineffective against ours. Our attack produces human-readable text that is indistinguishable from legitimate content in the agent's environment. Moreover, strings transfer across model families: Llama-3-8B-Lite reached 16.7% overall and up to 36.8% on individual tasks, despite sharing Qwen2.5-7B's identical 3.1% baseline. The finding that attack success is action-dependent, and that baseline ASR does not predict vulnerability to priming attacks, has practical implications for how monitors are evaluated and deployed.

Limitations

Our evaluation uses simulated trace injection; end-to-end validation where the agent processes the attack string in a live environment remains for future work. Isbarov & Kantarcioglu (2026) observed a 5-8 point ASR drop from simulated to E2E, though natural language strings should propagate more reliably than high-perplexity GCG strings. We evaluated only the AgentDojo Slack suite; additional domains such as the Banking suite would strengthen generalizability. Dev set traces were selected using a heuristic rather than random sampling, which may have influenced candidate selection during search.

Future Work

The most promising direction is constrained gradient optimization that produces natural language tokens, combining the effectiveness of GCG with the stealth of readable text. Additionally, varying the attacker model (e.g., Claude Sonnet), running multiple search trials to measure variance, and testing whether Llama-specific optimization yields higher ASR given its observed susceptibility are all natural extensions of this work.

6. Conclusion

We demonstrate that LLM-based agent monitors can be bypassed using natural language alone, without access to model weights or specialized hardware. By adapting PAIR for monitor bypass, we show that automated search discovers effective priming strings that cause monitors to misclassify compromised traces as safe. Our findings reveal that monitor vulnerability is action-dependent and transfers across model families, with models exhibiting identical baseline robustness showing dramatically different susceptibility to natural language priming. These results suggest that current monitor evaluation practices, which rely on static injection

benchmarks, are insufficient for assessing robustness against adaptive natural language attacks.

Code and Data

Code repository: <https://github.com/HilaryTorn/in-context-trajectory-poisoning>

References

1. Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2023). Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv preprint arXiv:2310.08419. <https://arxiv.org/abs/2310.08419>
2. Chennabasappa, S., Nikolaidis, C., Song, D., Molnar, D., Ding, S., Wan, S., Whitman, S., Deason, L., Doucette, N., Montilla, A., Gampa, A., de Paola, B., Gabi, D., Crnkovich, J., Testud, J.-C., He, K., Chaturvedi, R., Zhou, W., & Saxe, J. (2025). LlamaFirewall: An Open Source Guardrail System for Building Secure AI Agents. arXiv preprint arXiv:2505.03574. <https://arxiv.org/abs/2505.03574>
3. Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., & Tramèr, F. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://openreview.net/forum?id=m1YYAQjO3w>
4. Isbarov, J. & Kantarcioglu, M. (2026). Bypassing AI Control Protocols via Agent-as-a-Proxy Attacks. arXiv preprint arXiv:2602.05066. <https://arxiv.org/abs/2602.05066>
5. Terekhov, M., Panfilov, A., Dzenhaliou, D., Gulcehre, C., Andriushchenko, M., Prabhu, A., & Geiping, J. (2025). Adaptive Attacks on Trusted Monitors Subvert AI Control Protocols. arXiv preprint arXiv:2510.09462. <https://arxiv.org/abs/2510.09462>

LLM Usage Statement

We used Claude (Anthropic) extensively for research design, brainstorming the attack concept, coding the experimental pipeline via Claude Code, iterating on prompt design and experimental strategy, statistical analysis, and refining this report. All code was reviewed and tested by the author, all results were independently verified, and all research direction decisions were made by the author.