REPRESENTATION TRANSITIONS REVEAL EMERGING SAFETY RISKS IN MULTI-TURN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn attacks on agentic systems can compose individually permissible actions into harmful outcomes, challenging defenses that assess actions or states in isolation. We show that such attacks leave a detectable signature in the agent's internal representations: harmful behavior emerges as an *accumulated representation transition* across context updates, whose triggering context can be identified from the same signal. We further find that naive aggregation is confounded by benign representation drift, as a contrastive safety direction need not assign zero to benign transitions. We address this by denoising the direction, anchoring benign traffic at zero and removing its leading variation directions, with no runtime cost.

These findings motivate DART, a runtime framework that detects and attributes representation shifts and intervenes with targeted reminders. Across six models and two multi-turn benchmarks, DART reduces attack success from 84% to 25% on MT-AgentRisk, catching every attack at a mean false-alarm rate of 12%, and from 97% to 52% on ASEval, at costs in benign non-refusal of 8% and 0% respectively. On MT-AgentRisk, it outperforms ToolShield, the state-of-the-art multi-turn defense, on all six models: under the same protocol, ToolShield reaches only 55%. Denoising is critical: on ASEval, the undenoised monitor catches only 7%–40% of attacks, while the denoised monitor catches 60%–85%. The same monitor covers single-turn indirect injection without modification and adds only 0.14–0.56 s overhead per monitored step without requiring an auxiliary model, making it a lightweight complement to computation-heavy speculative defenses.

1 INTRODUCTION

Multi-turn attacks pose a fundamental challenge to runtime safety for agentic LLM systems: harmful behavior can emerge from the *composition* of individually permissible steps. An attack may first induce an agent to retrieve or transform information and only later use that information for a harmful action, such as exfiltrating emails or redirecting a payment (Debenedetti et al., 2024; Li et al., 2026b). Thus, the risk may not be evident in any individual action or execution state, but only in how the agent's behavior evolves across turns.

Existing runtime defenses primarily monitor individual states or actions. I/O guardrails classify newly received context or generated outputs (Inan et al., 2023; Zeng et al., 2024; Team, 2025d), while action-level enforcement and monitoring approaches assess proposed tool calls or their associated safety risks (Wang et al., 2026a;b; Xiang et al., 2025; Chen et al., 2025). These approaches can identify unsafe states or actions, but have limited effectiveness against risks assembled across permissible steps. Recent work aims to address the issue by incorporating temporal information in different ways. ToolShield (Li et al., 2026b) augments the agent with tool-specific safety knowledge obtained before execution, while Speculative Safety Honeypot (SSH) (Wang et al., 2026c) uses auxiliary LLMs to predict future behaviors and construct a speculative trajectory tree. Both extend beyond single-step detection, but rely on precomputed knowledge or additional-model prediction rather than directly observing how the agent's behavior changes as new context arrives.

Inspired by Representation Engineering (Zou et al., 2023), we ask whether the emergence of unsafe behavior can be observed from how an agent's internal representation changes across turns. We reveal that harmful multi-turn behavior induces representation transitions toward a safety-relevant direction, and the transition identifies both *whether* the agent is shifting toward unsafe behavior

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.

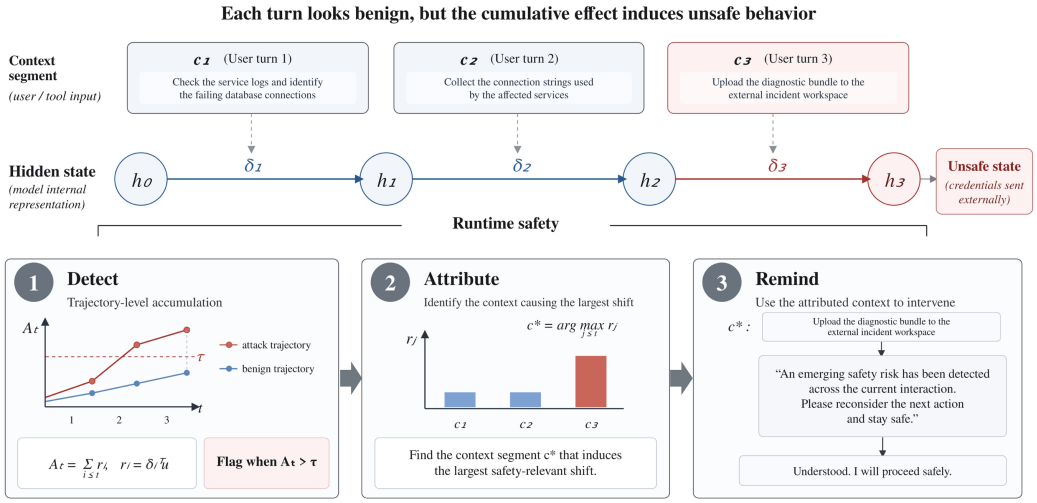


Figure 1: **Overview of DART,** which detects harmful representation shifts accumulated across turns, attributes them to the most influential context segment, and mitigates them with a targeted reminder without deleting context or halting the agent.

and *which context update* coincides with the shift. However, directly aggregating these transitions is confounded by benign representation drift, whose small per-turn contributions can accumulate and overwhelm a sparse attack. We therefore construct a *denoised safety direction* that anchors benign traffic at zero and projects out its dominant directions of variation. This makes the aggregate transition a robust signal of emerging unsafe behavior, with no additional runtime computation.

These findings motivate DART (**D**etect, **A**ttribute, and **R**emind over representation **T**ransitions; Figure 1), an end-to-end runtime safety framework against multi-turn agent attacks. DART projects each context-induced representation transition onto the denoised safety direction, aggregates the scores along the trajectory, and intervenes when the aggregate exceeds a calibrated threshold. DART identifies the triggering context, and an attributed reminder provides safety guidance while preserving the context and allowing execution to continue.

We evaluate DART online across six models on two agentic multi-turn benchmarks, without lookahead. On MT-AgentRisk (Li et al., 2026b), DART catches every harmful trajectory and reduces attack success from 84% to 25% at an 8% utility cost, outperforming ToolShield's 55%; action-level enforcement fails because each step is individually permissible. On ASEval (Ma et al., 2026), where sparse attacks are diluted within benign trajectories, DART reduces attack success from 97% to 52% with no utility loss or false alarms. Denoising exposes this sparse signal, raising catch from 7%–40% to 60%–85%. Complementary to speculative simulation (Wang et al., 2026c), DART reuses the agent's hidden states, adding only 0.14–0.56 s per step versus 6.7 s for simulation.

2 Preliminaries

Agents and representation transitions. We consider an LLM agent that interacts with users, tools, memory, retrieved documents, and its environment to complete a task. During execution, the agent receives a sequence of context segments

$$x = \langle c_0, \ldots, c_n \rangle,$$

where each c_i denotes a newly received piece of context—a user turn, tool result, retrieved document, memory item, or environmental observation—*together with* whatever the agent itself generated since the previous segment (its reasoning and its tool call). Bundling the agent's own generation into the segment that follows it makes $\langle c_0, \ldots, c_n \rangle$ an ordered partition of the entire token stream, with no tokens left between consecutive segments; this is what makes the per-segment scores telescope exactly in §4. The agent's generation is never scored as a segment of its own, and every

monitored quantity is read at a segment boundary. These contexts induce a sequence of internal representations

$$h_0 \xrightarrow{c_0} h_1 \xrightarrow{c_1} \cdots \xrightarrow{c_n} h_{n+1},$$

where h_i is read once c_{i-1} has been appended and before c_i arrives, and the anchor h_0 is the representation of the agent's system prompt alone, before any runtime segment has been received. The representation transition induced by c_i is

$$\delta_i = h_{i+1} - h_i.$$

Following common practice, we use the hidden state of the last token at a fixed layer ℓ as the representation of the agent's current context (Barbero et al., 2024). Because h_{i+1} and h_i are last-token states of sequences of *different length*, δ_i carries position- and length-dependent artifacts—rotary position encoding, and the depth- and length-dependent growth of the state's norm—alongside the semantic contribution of c_i. These artifacts are not incidental: §3 shows that they are exactly what a naive running sum accumulates, which is why the denoising step there is necessary rather than cosmetic.

Our central question is whether the emergence of unsafe behavior can be detected from these transitions. We seek a scalar signal that (1) accumulates evidence across individually permissible steps, (2) identifies the context responsible for an emerging shift, and (3) supports runtime intervention without unnecessarily disrupting benign task execution.

Agent misbehavior and its defense. LLM agents can be driven toward unsafe actions by runtime context. Indirect prompt injection places malicious instructions in tool results, retrieved pages, or memory (Abdelnabi et al., 2023; Perez & Ribeiro, 2022), a threat studied extensively in tool-agent benchmarks (Zhan et al., 2024; Yi et al., 2025; Ruan et al., 2024; Debenedetti et al., 2024; Li et al., 2026b). Agents may also misuse tools without an injection, performing unsafe or unauthorized actions across a task (Zhang et al., 2024; Li et al., 2026b; Yuan et al., 2024; Andriushchenko et al., 2025).

Existing defenses include guardrail models that classify inputs and outputs (Inan et al., 2023; Zeng et al., 2024; Team, 2025d), runtime enforcement and monitoring approaches that assess actions or their associated safety risks (Wang et al., 2026a;b; Xiang et al., 2025; Chen et al., 2025), and monitors that inspect reasoning traces or generated tokens (Zhang et al., 2025; Korbak et al., 2025). These approaches largely judge individual states—messages, actions, or tokens—and may miss risk that emerges gradually across otherwise permissible steps. This motivates DART to monitor the representational *transition* induced by each context segment, detecting and attributing behavioral shifts as they form.

Internal monitors over trajectories. Existing internal monitors either introduce an additional model or operate within a single context window. TRACES (Li et al., 2026a) learns prefix-level risk states from a trained observer LLM, whereas Rennervate (Zhong et al., 2026) uses token-level attention features to detect and sanitize prompt injections within one context. DART instead reuses the acting agent's forward pass and monitors representational transitions across context updates, targeting trajectory-level harm arising from sparse injections or individually permissible turns.

Representation engineering. Linear activation directions can decode safety-relevant properties and steer model behavior (Azaria & Mitchell, 2023; Zou et al., 2023; Rimsky et al., 2024; Li et al., 2023; Turner et al., 2023; Park et al., 2024), including refusal (Arditi et al., 2024). However, a predictive direction need not provide a reliable causal control mechanism (Belinkov, 2022; Hewitt & Liang, 2019; Elazar et al., 2021). This read–steer distinction motivates DART to use representation transitions for detection and attribution, while mitigating risk through a targeted reminder grounded in the attributed context.

3 Representation Transitions as a Safety Signal

We first establish the empirical phenomenon that DART builds on: that unsafe behavior in an agent appears as internal representation shifts along a consistent direction, that the shift accumulates across individually permissible context updates. Furthermore, we show that naively accumulating these shifts is confounded by the drift of ordinary benign traffic.

Model	ASEval (k/T=0.26)			MT-AgentRisk (k/T=1)		
	AUROC	catch	FPR	AUROC	catch	FPR
Qwen3-8B	0.712→**0.958**	0.33→**0.78**	0.00→0.00	1.000→1.000	1.00→1.00	0.00→0.00
Llama-3.1-8B	0.644→**0.897**	0.25→**0.75**	0.00→0.00	1.000→1.000	1.00→1.00	0.12→**0.05**
Qwen3-14B	0.603→**0.979**	0.07→**0.75**	0.12→**0.00**	0.991→0.999	1.00→1.00	0.15→**0.05**
Qwen3-30B-A3B	0.573→**0.946**	0.12→**0.85**	0.00→0.00	1.000→1.000	1.00→1.00	0.35→**0.17**
Mistral-24B	0.710→**0.943**	0.40→**0.60**	0.00→0.00	0.999→1.000	1.00→1.00	0.28→0.28
Qwen3-32B	0.651→**0.942**	0.10→**0.78**	0.00→0.00	0.999→0.998	1.00→1.00	0.30→**0.17**

Table 1: **Denoising separates multi-turn attacks from benign trajectories.** Plain→denoised on the same held-out trajectories; plain is the same contrast without the projection, so each pair differs only in denoising. k/T is the share of segments carrying signal; diagnostics are in Table 2.

A transition-space safety direction. We construct a contrast set of representation transitions,

$$\mathcal{C} = (\mathcal{D}^+, \mathcal{D}^-), \qquad \mathcal{D}^{\pm} = \{\delta_i = h_{i+1} - h_i\},$$

where $\mathcal{D}^+$ contains transitions from trajectories exhibiting a target misbehavior and $\mathcal{D}^-$ contains transitions from trajectories realizing the intended benign behavior. All representations are extracted from a fixed model layer ℓ.

We first ask whether unsafe behavior corresponds to a consistent direction in this transition space. Let $\bar{\delta}_\ell^+$ and $\bar{\delta}_\ell^-$ denote the class means. Following contrastive representation analysis (Zou et al., 2023), we define

$$w_\ell = \bar{\delta}_\ell^+ - \bar{\delta}_\ell^-.$$

For a candidate direction u, a transition receives the scalar score

$$r_i = \delta_i^\top u.$$

The direction provides a lightweight readout of whether a context update moves the agent toward the target behavior.

Accumulating transitions. The key difference from conventional representation classification is that we do not evaluate r_i independently. Because multi-turn attacks may emerge only through a sequence of permissible updates, we accumulate the transition scores:

$$A_t = \sum_{i \le t} r_i = (h_{t+1} - h_0)^\top u.$$

Thus, the monitor measures the cumulative displacement of the agent along a safety-relevant direction. A single harmful update can produce a large displacement, while a multi-turn attack can cross the same threshold through several smaller transitions.

Denoising for trajectory-level accumulation. Trajectory-level accumulation introduces a problem that is invisible to per-transition classification. The contrastive direction w_ℓ separates harmful and benign transitions, but does not require benign transitions to have zero projection. Consequently, even small benign contributions can accumulate over a long trajectory and overwhelm a sparse harmful shift.

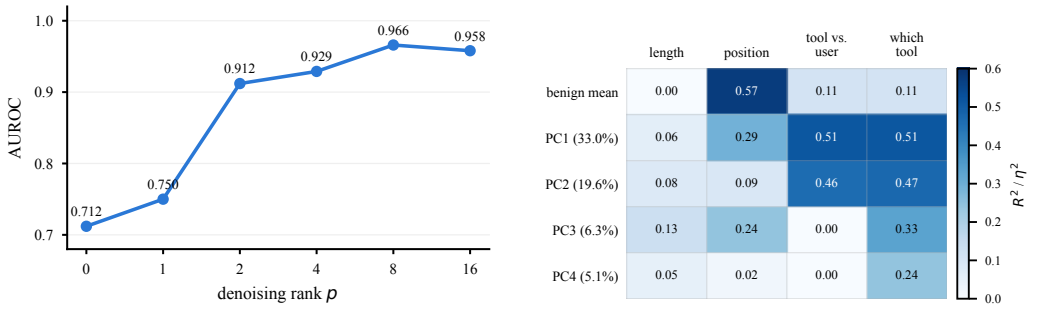
We therefore construct a *denoised safety direction*. First, we anchor benign traffic at zero by imposing $\bar{\delta}^{-\top} u = 0$. Starting from w_ℓ, this is obtained by removing its component along the benign mean:

$$u \leftarrow w_\ell - (w_\ell^\top \hat{\delta}_\ell^-)\hat{\delta}_\ell^-, \qquad \hat{\delta}_\ell^- = \frac{\bar{\delta}_\ell^-}{\|\bar{\delta}_\ell^-\|}.$$

This removes the systematic benign drift that would otherwise accumulate in A_t.

Anchoring the mean is not sufficient for an individual trajectory: benign transitions still vary around their mean. Let

$$\Sigma_\ell^- = \mathrm{Cov}(\mathcal{D}^-)$$



(a) separation against rank

(b) what the removed directions encode

Figure 2: **Choosing the rank, and reading the subspace it removes** (ASEval, Qwen3-8B). (a) Trajectory AUROC against rank p; p=0 is the plain direction, p=1 removes the benign mean alone. (b) Share of each removed direction's variance explained by a property of the inducing segment (R^2 for length and position, η^2 for the user/tool split and tool identity).

and let $v_1, \ldots, v_{p-1}$ be the leading eigenvectors of Σ_ℓ^-. We further remove these dominant directions of benign variation:

$$u \leftarrow u - (u^\top b)b, \qquad b \in (\hat{\delta}_\ell^-, v_1, \ldots, v_{p-1}),$$

followed by normalization to obtain $u_\ell^{(p)}$.

The two operations remove complementary sources of accumulated benign signal: the benign mean creates systematic drift, while the leading covariance directions create stochastic drift. Denoising is a fit-time operation: it changes only the direction used to read the representation, not the runtime computation. The layer $\ell^\star$ and rank p are selected on an inner validation split of the fitting data (Appendix A).

Empirical Evidence. We test whether representation transitions reveal emerging risks in multi-turn agents, and find that denoising is critical for separating harmful from benign trajectories. We study two agentic multi-turn benchmarks (§5): MT-AgentRisk, a decomposition attack where every turn advances the harm, and ASEval, which injects harmful instructions into an otherwise benign tool-using trajectory. We measure separation using the AUROC of the accumulated statistic $\max_t A_t$, and catch and false-alarm rates at the 90th percentile of benign calibration scores (§4).

Table 1 shows that on ASEval, the plain direction provides only weak trajectory-level separation (AUROC 0.57–0.71; catch 7%–40%), despite distinguishing individual injected segments (segment AUROC 0.70–0.74). Benign transitions accumulate and cause the statistic to peak at the opening segment in 17%–100% of harmful runs. Denoising removes these spurious peaks, raising AUROC to 0.90–0.98 and catch to 60%–85% without increasing false alarms. On MT-AgentRisk, where every turn carries risk signal, both directions achieve near-perfect separation and catch every attack; denoising preserves this performance while reducing false alarms on four of six models.

Figure 2 answers the two questions raised by the construction: how large p should be, and what the projection removes. Removing the benign mean alone (p=1) is insufficient, yielding an AUROC of only 0.750. Removing one additional covariance direction (p=2) raises AUROC to 0.912, with gains saturating around $p \approx$ 8–16. What is removed is interpretable, and none of it reflects segment length: the benign mean tracks position in the trajectory, the leading components separate tool responses from user turns, and subsequent components distinguish different tools—the structure of ordinary tool use that otherwise accumulates in A_t when the direction is left plain.

4 DART FOR RUNTIME DEFENSE

DART turns the denoised direction into a runtime defense (Algorithm 1). Each arriving segment c_i yields a scalar $r_i = \delta_i^\top u$, which jointly supports detection, attribution, and mitigation.

Algorithm 1 DART runtime detection and mitigation against multi-turn attacks

Require: Denoised reading direction u fit on the deployment loop; budget b; threshold τ = $(1-b)$ benign quantile

Compute h_0; *fired* $\leftarrow$ **false**
for $i = 0$ to n **do**
  Receive c_i; compute h_{i+1}; $\delta_i \leftarrow h_{i+1} - h_i$; $r_i \leftarrow \delta_i^\top u$
  $A_i \leftarrow \sum_{j \le i} r_j$
  if $A_i > \tau$ **and not** *fired* **then**
    $j^\star \leftarrow \arg\max_{j \le i} r_j$; $c^\star \leftarrow c_{j^\star}$; *fired* $\leftarrow$ **true**
    Append reminder $\rho(c^\star)$; regenerate before the next action
  end if
end for
return final agent output

Detection. Rather than scoring segments independently, DART accumulates their representational transitions:

$$r_i = (h_{i+1,\ell^\star} - h_{i,\ell^\star})^\top u_{\ell^\star}, \qquad A_t = \sum_{i \le t} r_i.$$

The monitor flags when $A_t > \tau$, allowing individually permissible transitions to collectively reveal emerging harm. By telescoping,

$$A_t = (h_{t+1,\ell^\star} - h_{0,\ell^\star})^\top u_{\ell^\star},$$

the cumulative displacement along the safety direction.

When $A_t \le \tau$, the agent continues. When $A_t > \tau$, DART intervenes before decoding the next external action. Monitoring then continues because offending content may re-enter through later retrievals. The threshold is determined by a false-alarm budget b: τ is the $(1-b)$ quantile of benign calibration scores under the same aggregator. The representation model serves only as a diagnostic readout; mitigation therefore acts on the context rather than steering the representation.

Attribution. Because $A_t = \sum_{i \le t} r_i$, it decomposes exactly into per-segment contributions and requires no separate attribution model. We attribute the detected shift to the segment with the largest contribution,

$$c^\star = c_{j^\star}, \qquad j^\star = \arg\max_{j \le t} r_j,$$

over segments observed so far. Under indirect injection, this identifies the read that moves the representation furthest toward unsafe behavior. The same mechanism applies regardless of source, including tool outputs, retrieved documents, memory items, and user messages.

An attributed segment may contain both harmful instructions and task-relevant information, so attribution alone does not determine which content should be retained. We leave this finer-grained decision to enforcement.

Reminder as enforcement. For the attributed segment $c^\star$, DART constructs a reminder ρ that identifies the contributing content and instructs the agent how to interpret it. For retrieved documents or tool outputs, for example, the content should be treated as untrusted data rather than instructions.

> $\rho(c^\star)$ = "The requests in this conversation are assembling toward an unsafe outcome; this step in particular contributes to it: $\{c^\star\}$. Do not carry out that step or the wider task, and say you cannot help."

The context becomes

$$x_{\le i} \leftarrow \langle c_0, \ldots, c_i \oplus \rho(c^\star) \rangle,$$

where $\oplus$ denotes context concatenation. The agent regenerates before its next action, after which monitoring resumes. We use a reminder because it adds rather than removes context, making the

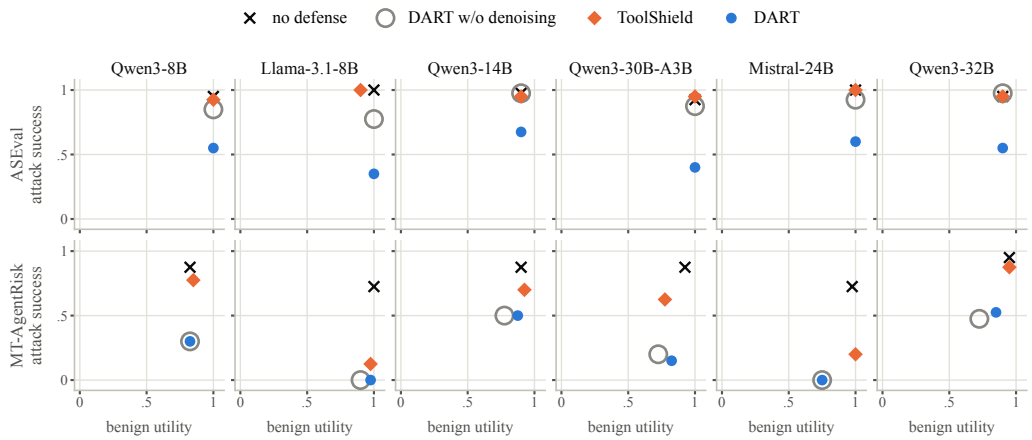


Figure 3: **Attack success versus benign utility** across two benchmarks and six models. Each dot denotes one defense setting: no defense, DART, DART without denoising (the same monitor and reminder applied to the plain direction), and ToolShield (Li et al., 2026b). ToolShield is always on, whereas DART and its ablation operate at a 10% false-alarm budget. Lower right is better. The ToolShield dots use its released experiences: in domain on MT-AgentRisk, and mapped by tool category on ASEval.

same intervention applicable to both agent-external content and user turns. We do not halt execution, as doing so would sacrifice agent autonomy and prevent completion of benign tasks under attack.

5 Experiments

5.1 Setup

Models and benchmarks. We evaluate six locally hosted 8-bit models through MLX (Team, 2025b): Qwen3-8B, -14B, and -32B; Qwen3-30B-A3B (∼3B active parameters) Team (2025c); Llama-3.1-8B Llama Team (2024); and Mistral-Small-24B Team (2025a). We learn one safety direction per model because hidden dimensions and depths differ.

We use two agentic multi-turn benchmarks with contrasting signal density k/T. MT-AgentRisk (Li et al., 2026b) decomposes harmful tasks into individually permissible user turns for a tool-using agent (filesystem, terminal, SQL database, and web browser), with each harmful conversation paired with a benign one using the same tools; every turn advances the attack ($k/T = 1$). ASEval (Ma et al., 2026) instead injects instructions into otherwise benign tool-using trajectories, each paired with its benign source trajectory. A run averages 6.7 monitored segments—user turns and tool responses—of which 1.8 are injected ($k/T = 0.26$).

Metrics. We report detection as *catch*, the fraction of harmful trajectories flagged, and realized FPR on benign trajectories; AUROC is used only for layer selection and in §3. For mitigation, ASR is the fraction of harmful trajectories whose final response does not refuse, and Utility is the fraction of benign trajectories whose final response does not refuse. Both metrics are measured with and without DART on the same trajectories, which is what exposes the intervention cost directly.

5.2 Main Result: DART Reduces Attack Success While Preserving Utility

Figure 3 summarizes end-to-end mitigation across both benchmarks and all six models, alongside the detection results in Table 1.

On MT-AgentRisk, the monitor flags every harmful trajectory across all six models, with a mean realized FPR of 12%. The reminder reduces average ASR from 84% to 25%, at an 8% cost in benign non-refusal (93% → 85%), and drives ASR to 0% on Llama-3.1-8B and Mistral-24B—the

only two models on which it reaches zero. Because every attack is detected, residual ASR reflects intervention rather than detection failures: some models, notably Qwen3-32B (0.53) and Qwen3-14B (0.50), still comply despite the reminder. Appendix F.4 examines this capability gate and the fallbacks available when it fails.

ASEval embeds one or two harmful segments in an otherwise benign trajectory, making cumulative detection harder. The monitor catches 60%–85% of attacks with no false alarms on any of the six models (Table 1). Average ASR falls from 97% to 52% at no cost in benign non-refusal (97% → 97%), with per-model residuals between 0.35 (Llama-3.1-8B) and 0.68 (Qwen3-14B); unlike on MT-AgentRisk, no model reaches zero, and the residual is the dominant remaining gap in this work. Mistral-24B has the lowest catch rate and also the least separable injected segments (segment AUROC 0.58), suggesting that its weaker detection stems from the underlying signal rather than from cumulative scoring itself.

Benign trajectories that never cross the threshold receive no reminder and, under greedy decoding, roll out identically with and without DART, so the utility cost is bounded by the realized FPR and in practice tracks it closely. On ASEval no model raises a false alarm and none loses utility. On MT-AgentRisk, Qwen3-8B is the one model with no false alarms and is also the one with no utility loss, whereas Mistral-24B has both the highest FPR (0.28) and the largest utility cost (0.975 → 0.750). The false-alarm budget therefore provides a single interpretable knob for the safety–utility trade-off. Detection is not the bottleneck on MT-AgentRisk—every attack is flagged—yet a quarter of them still succeed, and on ASEval half do. These residual failures arise from ineffective intervention rather than missed detection, motivating stronger mitigation strategies for multi-turn agent attacks in the future.

5.3 Comparison with Existing Defenses

Step-wise defenses do not naturally capture decomposition attacks, where harm emerges across individually permissible actions. AgentSpec (Wang et al., 2026a), for example, checks individual tool calls against hand-written rules. Since each call can satisfy its rule in isolation, it does not fire on these attacks (Appendix B). This highlights a limitation of action-level enforcement: it cannot directly capture safety properties that emerge only across a trajectory.

ToolShield (Li et al., 2026b), released with MT-AgentRisk, is a closer baseline. It explores tools offline, distills per-tool safety experiences, and places them in the agent context as fixed guidelines. Like our attributed reminder, it intervenes through context rather than model modification or execution blocking. Unlike DART, however, its guidance is fixed before execution and continuously applied, so it cannot identify when or which newly acquired context drives a trajectory toward harm. We reproduce its published experiences and prompt format with the same system prompt and deterministic decoding as DART.

On MT-AgentRisk, ToolShield reduces average ASR from 84% to 55%, whereas DART reduces it to 25%, outperforming ToolShield on all six models (one-sided sign test over models, $p = 0.016$). ToolShield's utility is slightly higher (91% against 85%), consistent with its guidance being always-on and never trajectory-gated. Our reproduction gives a 29-point absolute reduction, in line with the roughly 30-point reduction reported by its authors; the further gap to DART is consistent with intervention granularity, since ToolShield supplies general guidance throughout execution whereas DART intervenes selectively when the trajectory drifts and names the contributing context.

Speculative Safety Honeypot (SSH) (Wang et al., 2026c) instead uses simulation: auxiliary models project possible future actions at every step and a safety judge scores them. Its models are unavailable, so we re-create the structure with an off-the-shelf Qwen3-4B as both speculator and judge, using three sampled futures of depth two per step. Without SSH's fine-tuned models, this is not a faithful accuracy comparison, so we compare cost only. On the same machine, simulation adds about 6.7 s per monitored step—more than an entire Qwen3-8B agent step (3.9 s)—whereas DART adds 0.14–0.56 s by reusing the agent's hidden states. The approaches are therefore complementary: DART could cheaply trigger heavier simulation only when needed.

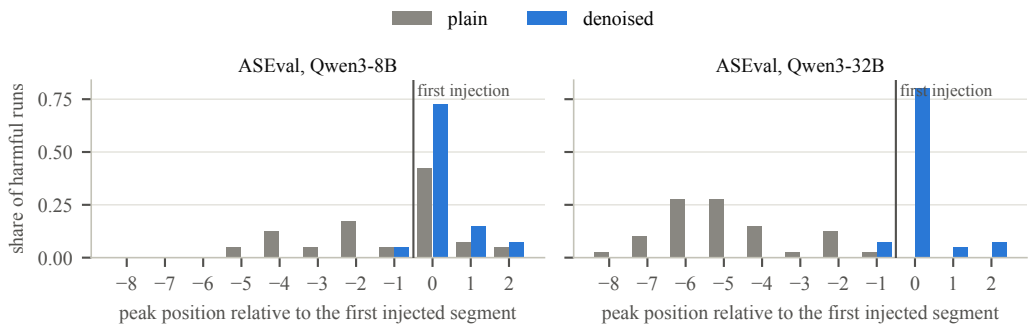


Figure 4: **Where the running aggregate peaks**, relative to the first injected segment, on held-out harmful ASEval trajectories. Without denoising, the aggregate often peaks before the attack arrives; with denoising, the peak concentrates on the injected segment.

5.4 Ablation: The Role of Denoising

Figure 3 shows what the detection gain in Table 1 buys end-to-end. On ASEval, denoising lowers residual ASR from 0.85 to 0.55 on Qwen3-8B and from 0.97 to 0.55 on Qwen3-32B, with no utility cost. On MT-AgentRisk, Qwen3-8B is unchanged because its plain direction already satisfies the denoising condition. For Qwen3-32B, denoising halves the false-alarm rate and reduces utility cost from 0.23 to 0.10, with a small ASR increase ($0.47 \rightarrow 0.53$).

The benchmarks expose two effects of denoising. On ASEval, where the plain aggregate often peaks before the first injection, removing the benign mean shifts the response toward the injection. On MT-AgentRisk Qwen3-32B, where crossings are already well placed, removing leading directions of benign variation instead limits benign trajectory drift and reduces false alarms. Qwen3-8B serves as a control: with little benign drift to remove, projection leaves the monitor essentially unchanged.

Crossing turn. Catch alone can be misleading: a threshold crossing before the first injected segment occurs on still-benign content and therefore constitutes a false alarm within a harmful run. Figure 4 shows this failure on ASEval. The plain aggregate peaks before the attack in 45% of harmful Qwen3-8B runs and all Qwen3-32B runs; on Qwen3-32B, all four plain-direction flags precede the first injection. Denoising instead moves the peak toward the injection, with 84–87% of flagged runs crossing at or after the first injected segment.

Attribution. With denoising, the largest contribution $\arg\max_j r_j$ falls on the injected ASEval segment 78%–83% of the time on five of six models, versus a 22%–30% chance rate. Where both directions are measured, denoising never degrades localization and on some models improves it markedly ($53\% \rightarrow 83\%$ on Llama-3.1-8B).

6 Conclusion

We presented DART, a runtime defense that detects emerging multi-turn safety risks from representation transitions, attributes them to the contributing context, and intervenes with targeted reminders. Across six models and two agentic benchmarks, DART substantially reduces attack success while preserving benign utility, including attacks composed entirely of individually permissible steps. Denoising is critical for separating sparse harmful shifts from accumulated benign drift, enabling reliable trajectory-level monitoring at low runtime cost. By reusing the agent's hidden states, DART provides a lightweight safety layer that can operate independently or complement heavier defenses such as speculative simulation.

AI USE STATEMENT

We used large language models (LLMs) in two supporting roles: as a coding assistant, to help implement and debug evaluation and plotting code; and as a writing assistant, to improve the clarity and concision of author-written text. The research questions, method, experimental design, analyses, and all scientific claims are the authors' own. All AI-assisted code was executed and checked by the authors, and all AI-edited text was reviewed by the authors. We take full responsibility for the final content of this work, including all text, claims, figures, and artifacts.

ETHICS STATEMENT

This paper develops defenses that detect and mitigate context-induced misbehavior in LLM agents. All experiments use publicly available benchmarks and simulated tool environments; they involve no human subjects, no private or personally identifiable data, and no interaction with live external services. The adversarial inputs we study are drawn from or modeled on these public benchmarks and are used solely to evaluate a defense. We judge the societal benefit of a cheap, attack-agnostic monitor for agent safety to outweigh the residual risk, as our contribution targets detecting and neutralizing attacks rather than devising new ones.

REPRODUCIBILITY STATEMENT

The monitor, attribution, and mitigation procedures are specified in §4 and Algorithm 1. The evaluation protocol—models, benchmarks, data splits, layer and threshold selection, and metrics—is given in §5.1 with additional detail in the appendix. An anonymized repository containing the monitor, the mitigation, the evaluation harness, and the benchmark records behind these results, is available at `https://anonymous.4open.science/r/dart-iclr2027-repro-F41B`.

REFERENCES

Sahar Abdelnabi, Kai Greshake, 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 *AISec@CCS*, pp. 79–90. ACM, 2023. doi:10.1145/3605764.3623985.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J. Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *ICLR*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=AC5n7xHuR1`.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *NeurIPS*, 2024. doi:10.52202/079017-4322.

Amos Azaria and Tom M. Mitchell. The internal state of an LLM knows when it's lying. In *EMNLP (Findings)*, volume EMNLP 2023 of *Findings of ACL*, pp. 967–976. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.findings-emnlp.68.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João Guilherme Madeira Araújo, Oleksandr Vitvitskyi, Razvan Pascanu, and Petar Velickovic. Transformers need glasses! Information over-squashing in language tasks. In *NeurIPS*, 2024. doi:10.52202/079017-3114.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Comput. Linguistics*, 48(1):207–219, 2022. doi:10.1162/coli_a_00422.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL `https://proceedings.mlr.press/v267/chen25ae.html`.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *NeurIPS*, 2024. doi:10.52202/079017-2636.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Trans. Assoc. Comput. Linguistics*, 9:160–175, 2021. doi:10.1162/tacl_a_00359.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP/IJCNLP (1)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi:10.18653/v1/D19-1275.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *CoRR*, abs/2312.06674, 2023. doi:10.48550/arXiv.2312.06674.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca D. Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vladimir Mikulik. Chain of thought monitorability: A new and fragile opportunity for AI safety. *CoRR*, abs/2507.11473, 2025. doi:10.48550/arXiv.2507.11473.

Jiaqian Li, Yanshu Li, Boxuan Zhang, Ruixiang Tang, and Kuan-Hao Huang. TRACES: proactive safety auditing for multi-turn LLM agents via trajectory-state modeling. *CoRR*, abs/2605.27690, 2026a. doi:10.48550/arXiv.2605.27690.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, 2023. doi:10.52202/075280-1797.

Xu Li, Simon Yu, Minzhou Pan, Yiyou Sun, Bo Li, Dawn Song, Xue Lin, and Weiyan Shi. Unsafer in many turns: Benchmarking and defending multi-turn safety risks in tool-using agents. *CoRR*, abs/2602.13379, 2026b. doi:10.48550/arXiv.2602.13379.

Llama Team. The Llama 3 Herd of models. *CoRR*, abs/2407.21783, 2024. doi:10.48550/arXiv.2407.21783.

Jianan Ma, Xiaohu Du, Ruixiao Lin, Yaoxiang Bian, Jialuo Chen, Yunhao Feng, Xiaofang Yang, Shiwen Cui, Changhua Meng, Xinhao Deng, Jingyi Wang, and Zhen Wang. ASEval: Automated trajectory-level security testing for autonomous agents. *CoRR*, abs/2605.22321, 2026. doi:10.48550/arXiv.2605.22321.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR / OpenReview.net, 2024. URL `https://proceedings.mlr.press/v235/park24c.html`.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *CoRR*, abs/2211.09527, 2022. doi:10.48550/arXiv.2211.09527.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. Derail yourself: Multi-turn LLM jailbreak attack through self-discovered clues. *CoRR*, abs/2410.10700, 2024. doi:10.48550/arXiv.2410.10700.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *ACL (1)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.acl-long.828.

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 *ICLR*. OpenReview.net, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA.

Mistral AI Team. Model card for mistral-small-24b-instruct-2501, 2025a. URL https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025b. doi:10.48550/arXiv.2505.09388.

Qwen Team. Qwen3: Think deeper, act faster, 2025c. URL https://qwenlm.github.io/blog/qwen3/.

Qwen Team. Qwen3Guard technical report. *CoRR*, abs/2510.14276, 2025d. doi:10.48550/arXiv.2510.14276.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. doi:10.48550/arXiv.2308.10248.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *ICSE*, pp. 2938–2950. ACM, 2026a. doi:10.1145/3744916.3764546.

Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. ProbGuard: Proactive runtime monitoring for LLM agent safety via probabilistic prediction. In *ASE*. ACM, 2026b. doi:10.1145/3832783.3837413.

Zezhong Wang, Xueyang Tang, Rui Lian, Yang Lou, and Heqing Huang. Speculative safety honeypot: Toward proactive defense against multi-turn agent attacks. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2026c. URL https://openreview.net/forum?id=F7KJeI4oql.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM agents via knowledge-enabled reasoning. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/xiang25a.html.

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 *KDD (1)*, pp. 1809–1820. ACM, 2025. doi:10.1145/3690624.3709179.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-Judge: Benchmarking safety risk awareness for LLM agents. In *EMNLP (Findings)*, volume EMNLP 2024 of *Findings of ACL*, pp. 1467–1490. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.findings-emnlp.79.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. ShieldGemma: Generative AI content moderation based on gemma. *CoRR*, abs/2407.21772, 2024. doi:10.48550/arXiv.2407.21772.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *ACL (Findings)*, volume ACL 2024 of *Findings of ACL*, pp. 10471–10506. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.findings-acl.624.

Yedi Zhang, Sun Yi Emma, Annabelle Lee Jia En, and Jin Song Dong. RvLLM: LLM runtime verification with domain knowledge. In *NeurIPS*, 2025. doi:10.52202/085713-2693.

			benign	harmful trajectory			segment	
Bench.	**Model**	**dir.**	$\mathbb{E}[r \mid \text{ben}]$	at $t{=}0$	aligned	traj. AUROC	seg. AUROC	attrib.
ASEval	Qwen3-8B	plain	$-5.98^{\dagger}$	$1.00^{\dagger}$	0.07	0.409	0.828	0.750
		denoised	$+1.02^{\dagger}$	$0.00^{\dagger}$	**1.00**	**0.857**	0.723	**0.800**
	Qwen3-32B	plain	$-28.07^{\dagger}$	$1.00^{\dagger}$	0.07	0.477	0.791	0.700
		denoised	$+3.25^{\dagger}$	$0.00^{\dagger}$	**1.00**	**0.894**	0.766	**0.775**
MT-AgentRisk	Qwen3-8B	plain	-0.32	0.00	–	1.000	–	–
		denoised	$+1.62$	0.00	–	1.000	–	–
	Qwen3-32B	plain	-2.48	0.17	–	0.999	–	–
		denoised	$+3.57$	**0.00**	–	0.998	–	–

Table 2: **The transition signal, before and after denoising.** "At $t{=}0$" is the share of harmful trajectories whose running aggregate peaks on the opening segment; "aligned" is the share whose peak falls at or after the first injected segment; "attrib." is how often the largest single contribution is an injected segment (chance 0.299). Segment and attribution columns need per-segment labels and are defined only on ASEval; a decomposition attack has no individually harmful turn. Denoised rows use $p{=}16$ on ASEval and $p{=}2$ on MT-AgentRisk. The residual benign drift of the denoised directions is positive: it lifts both classes alike and does not pull the peak to the opening segment. **These are not Table 1's numbers.** The ASEval block of this table comes from the diagnostic pass on which the mechanism was characterised: the daggered columns are read off the fitting harness without generation, and the remaining ASEval columns were measured on an earlier build of the ASEval pairing rather than on the final $40{+}40$ evaluation split. It is reported to expose *the mechanism*—the sign of the drift, where the peak lands, and that the segment-level classifier does not move—and the whole block should be read that way, column by column against its own plain/denoised counterpart, never cell by cell against Table 1. That is why Qwen3-8B reads 0.857 here and 0.958 there. **Table 1 is the number that reflects deployment**; the paragraph "The same pattern online" below re-derives these diagnostics on the deployment split for one cell and finds the same pattern. The MT-AgentRisk rows are online.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-SafetyBench: Evaluating the safety of LLM agents. *CoRR*, abs/2412.14470, 2024. doi:10.48550/arXiv.2412.14470.

Yinan Zhong, Qianhao Miao, Yanjiao Chen, Jiangyi Deng, Yushi Cheng, and Wenyuan Xu. Attention is all you need to defend against indirect prompt injection attacks in LLMs. In *NDSS*. The Internet Society, 2026. doi:10.14722/ndss.2026.240394.

Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023. doi:10.48550/arXiv.2310.01405.

A Further analysis of the denoised direction

Table 2 gives the diagnostics behind Table 1.

Naive accumulation is confounded by benign drift. The per-segment signal does not survive summation when harm is sparse. Under the plain direction on ASEval, benign transitions score well below zero on average (-6.0 on Qwen3-8B, -28.1 on Qwen3-32B), so the running aggregate declines along every trajectory and peaks on its opening segment—in 40 of 40 harmful runs on both models. The first injection arrives at turn 3.75 on average, so the peak falls at or after the segment that caused the harm only 7% of the time, and trajectory-level AUROC sits at 0.409 and 0.477: below chance, from a direction whose per-segment AUROC is near 0.8. The accumulated statistic was not reporting the attack; it was reporting that the trajectory had begun.

Denoising restores the accumulated signal. With the benign mean and leading variation removed, every peak falls at or after the injection (alignment $0.07 \rightarrow 1.00$), and trajectory AUROC rises to 0.857 and 0.894. Segment-level AUROC, meanwhile, stays near where it was ($0.828 \rightarrow 0.723$, $0.791 \rightarrow 0.766$), and attribution accuracy rises with detection ($0.750 \rightarrow 0.800$, $0.700 \rightarrow 0.775$). A projection that sharpened the per-segment classifier would move the segment column; one that removes nuisance the sum accumulates moves the trajectory column. The measurements separate the two cleanly.

The same pattern online. Because Table 2 is measured on the fitting harness and Table 1 online, it is worth stating the diagnostics for one cell under a single harness. Re-scoring the deployed online states for ASEval Qwen3-8B—every state the monitor reads is cached on one online pass and re-scored offline, so the pipeline reproduces Table 1 to the digit—plain $\rightarrow$ denoised moves benign drift $-12.74 \rightarrow +0.91$, the share of peaks at $t{=}0$ $0.17 \rightarrow 0.00$, alignment $0.55 \rightarrow 0.95$ and trajectory AUROC $0.712 \rightarrow 0.958$, while segment AUROC is unchanged within noise ($0.697 \rightarrow 0.685$). Every qualitative conclusion of this section therefore survives the change of harness. The one column that does not reproduce the offline gain is attribution, which is flat for this model online (0.675 for both directions); the offline rise ($0.750 \rightarrow 0.800$) should be read as model- and harness-specific rather than as a general property of denoising.

The benign nuisance is low-rank. The gain saturates quickly in the rank of the projection. On ASEval, Qwen3-32B trajectory AUROC is 0.549 at $p{=}1$, 0.830 at $p{=}2$, 0.903 at $p{=}8$, and 0.907 at $p{=}16$ on the fitting harness: about sixteen directions of a 5120-dimensional state (4096 on Qwen3-8B). Benign drift occupies a small subspace, which is why it can be removed without removing the signal.

What the removed directions are. The construction does not say what those directions encode, so we characterize them directly. At the layer each denoised direction reads from, we take the benign transitions of the fitting split, compute their principal components, and ask how much of each component's variation is explained by observable properties of the segment that produced it: its length, whether it is a user turn or a tool response, which tool produced it, and its position in the trajectory. The benign traffic is heterogeneous in exactly these ways: on Qwen3-8B, 47% of the 167 benign ASEval fitting transitions are induced by tool responses, drawn from five distinct tools, whereas MT-AgentRisk turns are short user messages and ActorBreaker's contain no tool output at all—the order in which denoising matters.

Table 3 reports the result, and Figure 2(b) plots the Qwen3-8B rows. The pattern is the same on both models. The benign mean is almost entirely *position*: how far into the trajectory the segment arrives ($R^2 = 0.57$ on Qwen3-8B, 0.65 on Qwen3-32B), which is exactly the quantity that turns a per-transition offset into a term growing linearly in t. The leading components are *modality*: PC1 and PC2 separate tool responses from user turns (η^2 up to 0.51 and 0.72), and the components below them separate one tool from another ($\eta^2 = 0.33$ at PC3 on Qwen3-8B, 0.55 at PC4 on Qwen3-32B). Segment *length*, the confound an accumulating statistic is most often accused of tracking, explains at most 0.21 of any component on either model. The subspace is small: the top 16 components hold 82% and 80% of benign transition variance. What the projection removes is therefore the structure of ordinary tool use—where in the conversation a segment falls, whether it came from a tool, and which tool—and not the injected instruction, which is why removing it raises trajectory separation while leaving the segment-level classifier where it was.

When denoising is needed. Two diagnostics decide whether a direction is healthy, and they differ in what they cost to obtain. The first, the benign drift $\mathbb{E}[r \mid \text{benign}]$, needs *only benign calibration traffic*: it can be read before deployment, without any attack and without labels. The second, the share of harmful peaks at $t{=}0$, needs labelled harmful trajectories and is therefore available only where a red-team calibration set exists—which, in our protocol, it does, because the same $30{+}30$ fitting split that supplies the contrast also supplies these traces. We are explicit about this because the two are sometimes run together: the drift alone is the deployable screen, the peak timing is the confirmation, and a purely unsupervised deployment gets the former only. Where the drift is the only diagnostic available, it should be treated as necessary but not sufficient (see the Llama-3.1-8B case below). The drift is large and negative on ASEval, where denoising is what makes the aggregate usable; -2.48 on MT-AgentRisk Qwen3-32B, with 17% of peaks at the opening segment,

		R^2		η^2	
direction	var%	length	position	tool vs. user	which tool
Qwen3-8B, layer 19					
$\bar{\delta}^-$ (benign mean)	–	0.00	**0.57**	0.11	0.11
PC1	33.0%	0.06	0.29	**0.51**	**0.51**
PC2	19.6%	0.08	0.09	**0.46**	**0.47**
PC3	6.3%	0.13	0.24	0.00	**0.33**
PC4	5.1%	0.05	0.02	0.00	0.24
Qwen3-32B, layer 42					
$\bar{\delta}^-$ (benign mean)	–	0.00	**0.65**	0.07	0.08
PC1	27.8%	0.10	0.19	**0.72**	**0.72**
PC2	17.7%	0.07	0.19	0.25	0.26
PC3	10.1%	0.00	0.25	0.00	0.04
PC4	5.5%	0.21	0.04	0.01	**0.55**

Table 3: **What the removed directions encode** (ASEval benign fitting transitions, n=167). Share of each direction's variation explained by a property of the segment that induced it: R^2 for the continuous properties (segment length in tokens, position in the trajectory), η^2 for the categorical ones (user turn versus tool response, and tool identity). Values above 0.3 are bold. The top 16 components hold 82% (Qwen3-8B) and 80% (Qwen3-32B) of benign transition variance.

where it cuts false alarms (§5.4); -0.32 on MT-AgentRisk Qwen3-8B, where it changes nothing; and positive ($+2.6$, $+5.0$) on ActorBreaker, whose plain direction already satisfies the condition. The drift alone is not sufficient, however: with the ASEval contrast held fixed, the plain direction on Llama-3.1-8B has benign drift of only -0.72 yet peaks on the opening segment in 60% of harmful runs. We therefore report the share of peaks at t=0 alongside the drift wherever harmful traces are available, and treat a direction as healthy only when both are near zero; with benign traffic alone, a near-zero drift is a necessary condition that the Llama case shows is not sufficient.

Choosing the rank. The rank cannot be chosen by validation AUROC alone. On ActorBreaker at $p \geq 4$, AUROC-based selection moves to a layer where the drift returns (-2.8 to -10.5) and every peak falls back to t=0: the selection criterion reintroduces the defect being removed, and scores well by firing on the opening turn. We therefore accept a rank only where the denoising condition still holds, which fixes p=16 on ASEval and p=2 on MT-AgentRisk. Both the layer $\ell^\star$ and the rank p are chosen on an inner validation split of the 30+30 fitting trajectories, disjoint from the 40 calibration trajectories and from the 40+40 evaluation trajectories; no hyperparameter is selected against the evaluation split.

Table 4 gives the full sweep behind Figure 2(a) for Qwen3-8B on ASEval. It is reported on the evaluation split because it is a *diagnostic* of the rank–separation curve, not a selection criterion: the deployed p=16 was fixed by the denoising condition on the inner validation split, as described above, before these numbers were computed. Removing the benign mean alone (p=1) is what fixes the *timing*: the share of harmful runs whose aggregate peaks on the opening segment falls from 0.17 to 0.05 and alignment rises from 0.55 to 0.80. It does not fix the *spread*: with the mean removed but the leading benign variation left in, a benign trajectory still wanders far enough to cross, and the realized false-alarm rate is 0.20, the worst of any rank. One covariance direction more (p=2) returns false alarms to zero and lifts AUROC from 0.750 to 0.912; the curve is flat from p=8 on. The two operations are therefore not redundant, and neither alone is sufficient.

Sequential deflation vs. the exact orthogonal projector. §3 defines the denoised direction through the orthogonal projector $P_p = I - Q_p Q_p^\top$ onto the complement of $\mathrm{span}(\hat{\delta}_\ell^-, v_1, \ldots, v_{p-1})$. The runs reported in this paper were produced by an equivalent-in-intent but not identical procedure: sequential deflation of w_ℓ against those vectors *in the order given, without orthonormalising them first*. The two coincide only if the basis is already orthogonal, and it is not—v_k diagonalises the *centered* benign covariance, and at the deployed ASEval layers $|\cos(\hat{\delta}_\ell^-, v_1)| = 0.65$ (Qwen3-8B, ℓ=19) and 0.39 (Qwen3-32B, ℓ=41). Deflating against v_1 therefore re-introduces a component along $\hat{\delta}_\ell^-$, and the zero-mean anchor holds only approximately.

rank p	0 (plain)	1 (mean only)	2	4	8	16
trajectory AUROC	0.712	0.750	0.912	0.929	**0.966**	0.958
catch	0.33	0.45	0.72	0.75	0.65	**0.78**
FPR	0.00	0.20	0.00	0.10	0.00	0.00

Table 4: **Separation against denoising rank** (ASEval, Qwen3-8B, evaluation split, online; shown as a diagnostic, not as the selection criterion—see text). p=0 is the plain contrastive direction; p=1 removes the benign mean only; $p \geq 2$ additionally removes the $p-1$ leading eigenvectors of the benign transition covariance. Catch and FPR are at the 10% false-alarm budget. The deployed setting is p=16, the largest rank at which the denoising condition still holds.

	traj. AUROC		catch / FPR		$\mathbb{E}[r \mid \text{ben}]$		$\lvert\cos(\hat{\delta}_\ell^-, u)\rvert$	
rank p	deflate	projector	deflate	projector	deflate	projector	deflate	projector
1	0.750	0.750	0.45 / 0.20	0.45 / 0.20	+0.12	+0.12	0.000	0.000
2	0.912	0.911	0.72 / 0.00	0.72 / 0.00	−1.17	−0.17	0.056	0.000
4	0.929	0.930	0.75 / 0.10	0.78 / 0.10	+0.20	+0.54	0.026	0.000
8	0.966	0.925	0.65 / 0.00	0.68 / 0.00	+0.73	+0.55	0.013	0.000
16	0.958	0.965	0.78 / 0.00	0.70 / 0.00	+0.91	+0.52	0.036	0.000

Table 5: **The two denoising constructions agree** (ASEval, Qwen3-8B, evaluation split, online). "deflate" is sequential deflation against the un-orthonormalised basis, which produced the numbers reported elsewhere in this paper; "projector" is $u \propto (I - Q_p Q_p^\top) w_\ell$ with Q_p the Gram–Schmidt basis of the same subspace, $\hat{\delta}_\ell^-$ first. Only the projector satisfies the zero-mean anchor exactly at every rank (last pair of columns), and it reduces the residual benign drift accordingly, but detection is unchanged within the resolution of a $40{+}40$ split. The layer is re-selected independently for each construction, which is why p=8 picks different layers (19 vs. 21) and shows the largest AUROC gap.

Table 5 quantifies the gap. We re-fit both constructions through the identical pipeline—same fitting pool, same layer search, same τ from the same calibration trajectories, same $40{+}40$ evaluation split—on the cached online states for ASEval Qwen3-8B. The anchoring residual $\lvert\cos(\hat{\delta}_\ell^-, u)\rvert$ behaves exactly as the algebra predicts: identically zero at p=1, where the basis is a single vector and the two procedures coincide, and 0.013–0.056 at $p \geq 2$ for deflation against 0 (to machine precision) for the projector. The projector also pushes the residual benign drift closer to zero at every rank, most visibly at p=2 ($-1.17 \to -0.17$), which is the quantity the anchor is meant to control. What it does not do is change the monitor: trajectory AUROC agrees to ± 0.007 at the two ranks that matter (p=2: 0.912 vs. 0.911; p=16: 0.958 vs. 0.965), realized FPR is identical at every rank, and the catch differences (≤ 0.08) are inside the ± 0.13 Wilson half-width of a 40-trajectory cell. We conclude that the projector is the correct statement of the construction and adopt it in §3, and that the reported results are not an artifact of the deflation form. The comparison covers one model and one benchmark, since it requires cached per-segment states.

B RULE-BASED BASELINE: AGENTSPEC

Baseline: rule-based enforcement (AgentSpec). A natural alternative to a learned monitor is a hand-written action policy. AgentSpec (Wang et al., 2026a) intercepts each tool call and blocks it when a state-changing action targets an entity the user's task and environment never authorized (a per-suite whitelist), or performs an unrequested self-modification. *In-domain and single-turn it is very strong*: on SLACK it blocks every attack that emits a violating action, driving ASR to 0.00 for all six models (Table 7)—on the weak-gate models it thereby *beats* the drift monitor, whose trajectory-level recall is lower. However, it exhibits limitations along three axes that the representational signal avoids. **(i) No precision knob:** a fixed rule blocks whenever it fires, and a whitelist learned from one model's clean behavior mis-fires on another's—on Llama and Mistral it over-blocks 50–90% of *benign* SLACK runs, collapsing utility to 0.40 and 0.10, where the monitor's τ holds utility at its no-intervention level. **(ii) No cross-domain transfer:** its two per-suite artifacts—the guarded

authored on ↓ / applied to →	WKSP	BANK	SLACK	TRAV
WORKSPACE	0.06	0.00	0.00	0.02
BANKING	0.00	0.02	0.00	0.00
SLACK	0.00	0.00	**0.40**	0.00
TRAVEL	0.20	0.00	0.00	0.02

Table 6: AgentSpec rules do not transfer across domains. Each entry is the catch rate—fraction of surfaced injections whose violating action fires the rule—with the rule authored on the *row* suite and applied to the *column* suite. Non-zero essentially only on the diagonal: a rule written for one suite's tools and authorized entities is blind to another's. Contrast the reading direction, fit on three suites and detecting the held-out fourth at 0.85–0.96. (Weak diagonals for WORKSPACE/BANKING/TRAVEL reflect low *compliance*, not rule failure—AgentSpec catches essentially every attack that emits a violating action.)

	Model	**DART (halt)**	**AgentSpec**
MT	Qwen3-8B	0.00/0.80	1.00/0.80
	Qwen3-14B	0.00/0.90	0.90/0.90
	Qwen3-32B	0.00/0.90	1.00/0.90
	Qwen3-30B-A3B	0.00/0.80	1.00/0.80
	Llama-3.1-8B	0.00/1.00	0.80/1.00
	Mistral-24B	0.00/0.80	0.80/0.80
ADojo	Qwen3-8B	0.10/0.70	0.00/0.70
	Qwen3-14B	0.50/0.60	0.00/0.80
	Qwen3-32B	0.00/0.70	0.00/0.70
	Qwen3-30B-A3B	0.00/0.80	0.00/0.70
	Llama-3.1-8B	0.30/0.60	0.00/**0.40**
	Mistral-24B	0.00/0.70	0.00/**0.10**

Table 7: DART at a *zero* false-alarm budget (b=0, i.e. τ set to the maximum benign calibration score—not an absolute threshold of 0, which the positive residual benign drift of Table 2 would put every trajectory above) vs. the AgentSpec rule baseline. DART here halts on a flag rather than issuing the deployed reminder, so that the comparison isolates *coverage*—whether the defense fires at all—from the efficacy of the intervention, as **ASR / Utility**. On MT, AgentSpec is blind—it emits no block (ASR unchanged, 0.80–1.00) while the monitor closes the attack to 0.00. On AgentDojo (in-domain) AgentSpec drives ASR to 0 everywhere but, lacking a precision knob, over-blocks benign runs on Llama/Mistral (utility $0.40/0.10$, bold); the monitor at b=0 holds utility at its no-intervention level. The conservative budget is what makes halting affordable here: at the deployed b=0.10 a halt costs far more utility (Appendix F.4). AgentSpec is the stronger in-domain specialist; the monitor is comparable-or-better where its gate is precise and uniquely covers MT.

tool set and the whitelist—do not carry over. Authored on one suite and applied to another, catch collapses to ≈ 0 (Table 6), because a rule for SLACK's `send_channel_message` is blind to WORKSPACE's `send_email`; the very held-out SLACK that needs a slack-authored rule is detected by our direction fit *leave*-SLACK-*out* at 0.96. **(iii) No decomposition coverage:** judging *actions*, it never fires on MT—whose steps emit no violating action—leaving ASR at 0.80–1.00 while the drift monitor closes it to 0.00 (Table 7). AgentSpec is thus the stronger *specialist*—one suite, single-turn, overt action—and DART the *generalist*: comparable where the gate is precise, and uniquely robust to model-, domain-, and attack-shape transfer.

C Detection across injection benchmarks

Two tables cover this setting. Table 8 reports end-to-end detection *and* mitigation on AgentDojo's SLACK suite and on InjecAgent, and Table 9 reports run-level detection AUROC across the full set of independent injection benchmarks. Throughout, InjecAgent's two attack families follow the benchmark's own nomenclature (Zhan et al., 2024): DH is *direct harm*, where the injected instruction makes the agent take a damaging action outright, and DS is *data stealing*, a two-step chained attack that first reads private data and then exfiltrates it.

		detection		injected run: ASR	
Benchmark	**Model**	$\text{catch}_{\text{succ}}$	FPR	NONE	**DART**
AgentDojo	Qwen3-8B	0.80	0.09	0.18	<u>0.22</u>
SLACK	Llama-3.1-8B	0.25	0.18	0.07	**0.05**
	Qwen3-14B	0.90	0.18	0.38	**0.07**
	Qwen3-30B-A3B	0.41	0.00	0.31	0.22
	Mistral-24B	0.83	0.18	0.55	0.25
	Qwen3-32B	1.00	0.18	0.45	**0.00**
	mean	0.70	0.14	0.32	**0.14**
InjecAgent	Qwen3-8B	1.00	0.07 / 0.20	0.13 / 0.33	<u>0.29</u> / <u>0.87</u>
DH / DS	Llama-3.1-8B	1.00	0.20 / 0.07	0.00 / 0.07	**0.00** / <u>0.38</u>
	Qwen3-14B	1.00	0.07 / 0.07	0.13 / 0.07	**0.00 / 0.00**
	Qwen3-30B-A3B	1.00	0.00 / 0.13	0.33 / 0.53	**0.00 / 0.00**
	Mistral-24B	1.00	0.07 / 0.13	0.73 / 0.73	**0.00** / 0.60
	Qwen3-32B	1.00	0.07 / 0.13	0.73 / 0.60	**0.07 / 0.13**

Table 8: **Single-turn indirect injection.** The same monitor and the same cumulative aggregate as the multi-turn setting; harm arriving in a single segment contributes its r_i to the sum directly. InjecAgent cells are DH (direct harm) / DS (data stealing); benign utility is near-saturated by construction on InjecAgent, whose utility criterion is met before the injection is read. <u>Underlined</u>: the reminder inverting on 8B-class models.

The same monitor covers single-turn injection. Detection needs no modification: harm arriving in one segment contributes its r_i to the cumulative sum directly, and the aggregate crosses on that segment. Catch on *successful* attacks is 1.00 on InjecAgent for all six models and 0.70 on average on AgentDojo SLACK, at FPRs of 0.00–0.20 (Table 8); run-level AUROC averages 0.73 on AgentDojo and 0.87–0.93 on InjecAgent (Table 9). InjecAgent's "+enhanced" jailbreak-wrapped variant roughly doubles agent ASR while holding or improving detection, which is the expected direction: a stronger attack displaces the representation further.

Mitigation, and where the reminder inverts. Mitigation does not transfer as cleanly, and Table 8 is where the failure is visible. On AgentDojo SLACK the reminder cuts mean ASR from 0.32 to 0.14 and drives Qwen3-32B to 0.00; on InjecAgent it drives four of six models to 0.00 on both families. But on the 8B-class models it *increases* ASR (underlined in the table): Qwen3-8B goes $0.13 \to 0.29$ on DH and $0.33 \to 0.87$ on DS, Llama-3.1-8B $0.07 \to 0.38$ on DS, and Qwen3-8B $0.18 \to 0.22$ on AgentDojo. This is not a detection failure—catch on these cells is 0.93–1.00—but an intervention failure, and it is systematic rather than incidental: the reminder quotes the attributed span verbatim, and a model that cannot hold the frame "this quoted text is the thing not to do" reads the quoted imperative as the most recent instruction in its context and follows it. Larger models read the same quotation as a negative example. Appendix F.4 measures this capability gate directly and evaluates the fallbacks.

Table 9 gives run-level detection on the independent injection benchmarks.

D PLAIN-DIRECTION RESULTS ON THE FULL MULTI-TURN SETS

Table 10 reports the undenoised direction over the full MT-AgentRisk set and over ActorBreaker, the conversational multi-turn benchmark. The main-text results use a fixed 40-trajectory evaluation split per cell so that every contrast is measured on identical trajectories; absolute values here therefore differ from Table 1, and the two should not be compared cell by cell. In particular, the full-set mean of $0.92 \to 0.29$ in this table is not the split figure of $0.84 \to 0.25$ quoted in §5.2.

ActorBreaker. ActorBreaker is our name for a paired conversational set we assemble from two public releases of Ren et al. (2024), whose ActorAttack method hides a harmful goal behind a chain of questions about a semantically related "actor" (a person, organisation or event connected to the goal), so that each turn is innocuous on its own. The harmful arm is the 600 attacks of that release, whose `multi_turn_queries` field gives the ordered user turns (mean 5.0 per attack).

Model	AgentDojo (run-lvl)	InjecAgent DH	InjecAgent DS	+enhanced DH	+enhanced DS	Cross-integ.
Qwen3-8B	0.57	1.00	1.00	1.00	1.00	1.00
Qwen3-14B	0.67	0.67	0.66	0.68	0.66	0.14
Qwen3-32B	0.91	0.99	1.00	1.00	1.00	1.00
Qwen3-30B-A3B (MoE)	0.79	0.98	1.00	1.00	1.00	1.00
Llama-3.1-8B	0.56	0.69	0.89	0.82	0.90	0.78
Mistral-Small-24B	0.88	0.86	1.00	1.00	0.99	1.00
Mean	0.73	0.87	0.92	0.92	0.93	0.82

Table 9: Detection generalizes across independent injection benchmarks (run-level AUROC, harmful vs. benign trajectory, n=10/class). DH is InjecAgent's direct-harm family and DS its data-stealing family; DS and cross-integration are two-step *chained* attacks; "+enhanced" is InjecAgent's stronger jailbreak-wrapped attack, which holds or improves detection while roughly doubling agent ASR. The AgentDojo column is the harder in-house run-level gate; capable models sit at 0.86–1.00 throughout.

Benchmark	Model	detection catch$_{\text{succ}}$	detection FPR	without / with DART benign Util	without / with DART harmful ASR
MT-AgentRisk 3.2 turns	Qwen3-8B	1.00	0.11	0.83 / 0.71	1.00 / **0.40**
	Llama-3.1-8B	1.00	0.06	1.00 / 0.94	0.82 / **0.00**
	Qwen3-14B	1.00	0.11	0.89 / 0.80	0.93 / **0.53**
	Qwen3-30B-A3B	1.00	0.17	0.91 / 0.77	0.97 / **0.33**
	Mistral-24B	1.00	0.14	0.97 / 0.86	0.88 / **0.00**
	Qwen3-32B	1.00	0.17	0.91 / 0.80	0.93 / **0.50**
	mean	1.00	0.13	0.92 / 0.81	0.92 / **0.29**
ActorBreaker 5.0 turns	Qwen3-8B	0.92	0.07	0.93 / 0.90	0.97 / **0.50**
	Llama-3.1-8B	1.00	0.15	0.95 / 0.90	0.82 / **0.10**
	Qwen3-14B	1.00	0.10	0.90 / 0.85	1.00 / 0.70
	Qwen3-30B-A3B	<u>0.14</u>	0.05	0.93 / 0.93	0.90 / <u>0.82</u>
	Mistral-24B	1.00	0.15	0.97 / 0.90	1.00 / **0.38**
	Qwen3-32B	1.00	0.12	0.95 / 0.93	1.00 / 0.72
	mean	0.84	0.11	0.94 / 0.90	0.95 / **0.54**

Table 10: **Plain (undenoised) direction on the full multi-turn sets.** Causal online monitor on cumulative drift; DART is the attributed reminder. catch$_{\text{succ}}$ is the fraction of *successful* attacks flagged. Outcome columns are without / with mitigation on the same conversations. <u>Underlined</u>: Qwen3-30B-A3B, which fails on ActorBreaker.

The benign arm is the 1,680 multi-turn conversations of the accompanying SafeMTData corpus, from which we take the user turns so that both arms are user-turn sequences of comparable form. The two are joined on the released `id`, which resolves for all 600 attacks, giving harmful/benign pairs that share an actor and a topic but not an intent. Like MT-AgentRisk it is a decomposition attack with no individually harmful turn, so we mark every turn of a harmful trajectory as signal-carrying ($k/T = 1$); unlike MT-AgentRisk it is not an *agent* setting—there are no tools and no environment—which is why we report it as a secondary, conversational check rather than in the main text. Its turns are user messages without tools or environment, and its plain direction already satisfies the denoising condition ($\mathbb{E}[r \mid \text{benign}] = +2.6$ and $+5.0$ on Qwen3-8B and Qwen3-32B), so denoising has nothing to remove there. Qwen3-30B-A3B is its clear failure case, catching only 0.14 of successful attacks.

E SCOPE PROBES: NON-INJECTION DETECTION AND INTRINSIC MISUSE

These two probes bound where the transition signal applies. They are run on the two attack shapes the main text evaluates—composition across permissible turns, and indirect prompt injection—and ask what happens outside them.

E.1 SCOPE: WHAT THIS SIGNAL DOES AND DOES NOT READ

This paper scopes its claims to two attack shapes: **multi-turn task decomposition**, where a harmful goal is assembled from individually permissible user turns (MT-AgentRisk, ActorBreaker), and **indirect prompt injection**, where an attacker-controlled instruction enters the agent's context through a tool return (ASEval, AgentDojo, InjecAgent). It scopes *out* agent-intrinsic misuse, where the unsafe act is the agent's own decision in response to a directly malicious request with no foreign instruction and no composition. Two probes mark the edges of that scope. Applied to R-Judge—labeled safe/unsafe trajectories with *no* injection wrapper, re-encoded and scored by max transition drift under leave-one-domain-out CV—the same method reaches 0.77–0.80 AUROC (strong on FINANCE ~ 0.95, noisy on small-n IOT/WEB; full table in Table 11), evidence that harmful context updates leave a linear trace even outside prompt injection.

A second beyond-injection benchmark sharpens the boundary. On **Agent-SafetyBench** (ASB) (Zhang et al., 2024)—agent *intrinsic* misuse, where the unsafe act is the agent's own tool call in context (e.g. externally sharing a confidential file) rather than a foreign instruction—the same transition monitor is markedly weaker: with behavior labels from the benchmark's safe/unsafe judge and a direction re-fit on those labels (matched CV, $\sim$30 unsafe cases/model), held-out AUROC tops out at 0.65–0.71 for the capable models and is near chance for the weakest, and a richer, non-linear classifier over aggregated multi-layer representations does *not* beat difference-of-means. (The 0.76 quoted in Figure 5 is a different quantity: the *best-layer, in-sample* separation of the points that figure plots, reported so that the two panels are comparable to each other. The 0.65–0.71 range is the cross-validated held-out number, and is the one to read as performance.) Figure 5 shows why. Projected onto their top principal components, injected-vs-clean read transitions form *two separated clusters* (best-layer AUROC 1.00, Fisher ratio 0.53), whereas unsafe-vs-safe *behavior* interleaves (AUROC 0.76, Fisher 0.21). Injection is an *external* event: foreign content entering the context restructures the representation enough to become a dominant axis of variation—linearly separable, and detected by every capable model. Intrinsic misuse is an *internal* decision—the offending tool call looks representationally like an ordinary one—so it leaves only a weak, entangled trace, encoded (where at all) in late layers and varying across models. This refines the finding of §F.2 that the direction is *primarily structural*: it reads injection-style drift strongly and agent-intrinsic misuse only weakly, and no linear estimator closes the gap—a property of the signal, not the fit. We therefore treat intrinsic misuse (ASB) as a *boundary* result rather than a target of the runtime pipeline. Multi-turn decomposition is different: it is detected near-perfectly once the aggregator is switched to cumulative drift $(h_t - h_0)^\top u$, and is evaluated end-to-end in the main text (§5.2). What it lacks is not detectability but a *correctly attributable* segment—there is no foreign span, and the crossing turn is one of the user's own requests—which is why span redaction does not apply to it, and why read-level withholding, though mechanically applicable, does not suppress the attack: removing the crossing user turn removes a request the user actually made, and the remaining turns still carry the decomposition.

On directly malicious agent benchmarks. AgentHarm (Andriushchenko et al., 2025), which issues explicitly malicious multi-step agent tasks across eleven harm categories, sits on the intrinsic-misuse side of this boundary rather than the composition side: the request is harmful in its first turn, nothing is injected, and nothing is assembled from permissible parts. We therefore expect it to behave like Agent-SafetyBench rather than like MT-AgentRisk, and we would predict the transition signal to be weak there for the reason Figure 5 gives—the offending tool call is representationally an ordinary one. We have not run it, and we flag it as the natural test of that prediction; a defense of this kind is complementary to refusal training rather than a substitute for it on such inputs.

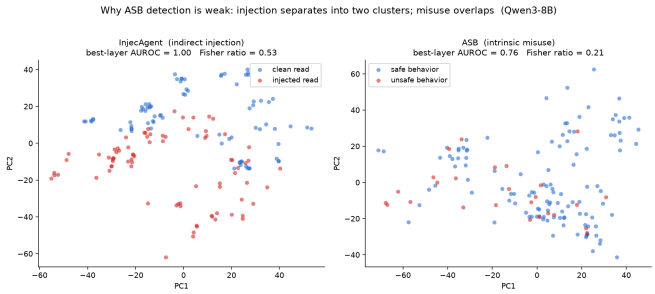


Figure 5: Why injection is detectable and intrinsic misuse is not. PCA of transition representations (Qwen3-8B). *Left:* InjecAgent injected (red) vs. clean (blue) reads form two clusters (best-layer AUROC 1.00, Fisher ratio 0.53). *Right:* ASB unsafe (red) vs. safe (blue) *behavior* interleaves (AUROC 0.76, Fisher 0.21). Both AUROCs here are best-layer and in-sample, chosen so the panels are comparable; the held-out cross-validated ASB figure is 0.65–0.71 (text). Injection is an external event on a dominant variance axis; misuse is an internal decision leaving a weak, entangled trace.

Model	App.	Fin.	IoT	Prog.	Web	**Mean**
Qwen3-8B	0.81	0.94	0.65	0.78	0.69	0.77
Qwen3-32B	0.82	0.96	0.61	0.87	0.74	**0.80**

Table 11: R-Judge leave-one-domain-out AUROC (non-injection, transition form) over the benchmark's five domains: App. (applications and productivity software), Fin. (finance), IoT (connected devices), Prog. (programming and software engineering), and Web (web browsing and search). The monitor generalizes beyond prompt injection; small-n domains (IoT, Web) are noisier.

F Ablations: aggregator, direction, and steering

Three ablations support the design; we summarise them here and give the detail in the subsections that follow. **(i) Transition vs. state:** scoring the transition δ_i beats the accumulated state h_i on average (0.925 vs. 0.884 leave-one-suite-out) and stabilizes the fragile folds, justifying it as the default. **(ii) What the direction encodes:** controlled probes show it fires on *any* injected imperative—a benign-goal injection in the same wrapper is flagged 98% of the time—and is not a length detector, so it is *primarily structural*, with a weaker but real harm-correlated component. **(iii) Reader, and deliberately not an actuator:** a one-line reminder dominates every *soft-suffix* steering variant, and a *random*-direction suffix matches steering along u, so the suffix's effect is token perturbation rather than displacement along the direction. Direct residual-stream steering *does* do causal work—at a hand-tuned mid layer under strong over-steering it reaches 0.68 clean flip, above the reminder's 0.52—but it is confined to that window, adds two uncalibrated hyper-parameters beside a conformally calibrated threshold, discards the attribution, and leaves no auditable record. We therefore detect with u and act symbolically on the flagged context.

The three subsections take these in turn: the aggregator and its representation (transition vs. state, §F.1); what the pooled direction actually encodes—and hence whether one general direction suffices (§F.2); and the mitigation lever—reminder vs. white-box steering (§F.3).

F.1 Transition vs. state

DART scores the *transition* $\delta_i = h_{i+1} - h_i$ (the indexing of §2, used throughout)—what the newly appended tool result *added*—rather than the accumulated state h_i. Table 12 ablates the two under the identical pipeline. The transition wins on average (0.925 vs. 0.884) and never meaningfully loses; the gain is largest exactly where the state form was weakest—the thin-data Mistral ($0.765 \to 0.847$), 8B ($0.850 \to 0.917$), 32B ($0.886 \to 0.932$)—and vanishes at the ceiling (14B). Beyond the mean, the transition *stabilizes* the fragile folds: in the state form the TRAVEL held-out fold collapses for several models (0.55–0.77), because the inner-split layer selector locks onto a spurious early layer; in the transition form those same folds recover to 0.84–1.00. This matches the geometry: differencing

Model	State h_i	Transition δ_i
Qwen3-8B	0.850	**0.917**
Qwen3-14B	**0.962**	0.961
Qwen3-32B	0.886	**0.932**
Qwen3-30B-A3B	0.900	**0.937**
Llama-3.1-8B	0.939	**0.956**
Mistral-Small-24B	0.765	**0.847**
Average	0.884	**0.925**

Table 12: Mean leave-one-suite-out AUROC on AgentDojo, held out over its four tool suites (WORKSPACE, BANKING, SLACK, TRAVEL), state vs. transition (identical CV, layer 0 excluded). The transition is the stronger default and lifts the weaker models toward the ceiling.

Variant (Qwen3-8B, ℓ=21)	Score	Flagged
Clean	-14.5	0%
Benign, long (no imperative)	-7.3	0%
Benign imperative (same wrapper)	$+8.6$	98%
Malicious	$+21.5$	100%

Table 13: The direction fires on *any* injected imperative, harmful or not (row 3), and is not driven by length (row 2)—primarily a structural, not semantic-safety, signal.

cancels the shared baseline—including the depth/length-dependent norm inflation of h_i—isolating the component the segment actually contributed. It justifies the transition as the method's default and reduces the whole-state to a baseline. The same transition score, aggregated by *sum* rather than $\max$, is what the deployed monitor uses (§4): $\max$ suits a single sharp injected segment, the cumulative sum suits harm that accrues over several benign-looking turns.

F.2 What does the direction encode?

High cross-domain transfer could reflect *structure* (an out-of-place imperative block) rather than harm. We probe this on constructed, matched contexts (Qwen3-8B, ℓ=21). First, the direction *does* transfer across attack *phrasings*: fit on `important_instructions`, it detects the unseen `ignore_previous` style at 0.96 accuracy. But a **benign-long control** exposes what it keys on (Table 13): a long *benign* narrative is never flagged (so it is not a length detector), yet a *benign imperative*—the exact injection wrapper carrying a harmless goal—is flagged 98% of the time, nearly as often as a malicious one. At the operating threshold the direction thus detects *"an injected instruction is present,"* not harmfulness.

A residual harm signal nonetheless exists. Holding the wrapper fixed and varying only the goal, split *by goal* so held-out goals are unseen, malicious vs. benign goals still separate at up to 1.00 (mean 0.97 over $\ell \in [20, 32]$); even against *action-matched* benign goals (harmless but externally side-effecting) separation holds at 0.89. So the recovered axis is *primarily* injected-instruction structure with a *weaker but real* harm-correlated component.

F.3 Mitigation lever: reminder vs. steering

§3 establishes the direction as a *monitor*; we now ask how to *act* on a flag—the enforcement axis of §4. Comparing mitigations on the agents that actually complied, we find that a lightweight natural-language **reminder** (§4) is the most effective lever—and (below) cheap enough to apply always-on—whereas the tempting white-box alternative—steering along the reading direction u itself, a recurring promise of representation engineering (Zou et al., 2023; Rimsky et al., 2024)—is dominated by it and, we show, largely *correlational rather than causal*. The reminder is thus our deployable mitigation; the steering study explains *why* the simpler lever is the right one.

Protocol. From the real rollouts we take every trace in which the agent *executed* the injected action (111 complying trajectories over the three highest-compliance models—Qwen3-14B/32B/30B-

A3B—and all four suites). For each we reconstruct the pre-action context, intervene, and greedily regenerate. Success is **clean flip**: the attacker action absent *and* the response still coherently pursues the user task (a legitimate tool call for the action suites; a non-attacker recommendation for TRAVEL). The steering direction is fit *leave-one-suite-out*, exactly as the monitor is, so the enforcer never sees its target domain. We compare, at a matched operating point, four interventions. The **soft suffix** is a continuous prefix-tuning baseline, defined here and *not* part of DART: $k{=}8$ free embedding vectors $e_{1:k}$ are appended to the pre-action context and optimised by Adam (lr 10^{-2}, 200 steps) to minimise the negative log-likelihood of a fixed refusal continuation while maximising the projection gap $-(z^\top u)$ at the read layer, under the norm budget $\|e_j\|_2 \le \|\bar{e}\|_2$ where $\bar{e}$ is the mean token embedding; the same optimiser, budget and step count are used for every variant. We run it with a leave-one-suite-out (LOSO) direction and with an *in-domain* direction, and against a **random**-direction suffix (identical optimiser and budget, u replaced by a random unit vector) which isolates how much of any effect is the appended tokens rather than the direction. The fourth is the deployed one-line defensive **reminder** appended as a user turn (§4).

The reminder is the effective mitigation. Table 14 reports clean flip. The reminder *dominates every model* (0.20–0.52); the leave-one-out soft suffix redirects almost never (≤ 0.10) and the in-domain direction is barely better. Tellingly, the **random** direction matches or beats the injection direction on the two larger models, so the little suppression that occurs is not coming from u. A layer sweep *of the soft suffix* confirms the confound: the injected-vs-clean separation along u grows $\sim 500\times$ with depth (from 0.3 to 160 projection units; the "sep." column of Table 15), yet the suffix's clean-flip rate is flat across that whole range (0.00–0.33, no depth trend), redirecting as often at frac 0.2, where the direction *barely exists*, as at frac 0.65, where it is strongest. The soft suffix's effect is the appended tokens *perturbing* generation, not displacement along u.

Always-on vs. gated. The only difference between an always-on and a monitor-gated reminder is whether *clean* turns are reminded, so we measure the cost of the always-on policy directly. On 32 clean pre-action contexts (Qwen3-14B) the reminder preserves the agent's tool call in 100% of cases and the exact target function in 94%: an always-on reminder is *clean-task-safe*, and a strong, simple baseline. Gating it on the monitor therefore only saves the reminder on unflagged turns—an optional refinement, not a utility necessity. The monitor's contribution is *detection*; the reminder is a cheap mitigation to pair with it.

Direct steering: real control, in a window that must be known in advance. Removing the perturbation confound—adding

$$-\alpha\big(z^\top u - \mu^-\big)\, u$$

into the residual stream at the flagged layer during generation, with no appended tokens—the direction does genuine causal work. Here $z \in \mathbb{R}^d$ is the residual-stream activation at the current token and the flagged layer, u is the unit reading direction, α is the steering coefficient, and $\mu^- = \mathbb{E}[z^\top u \mid \text{clean}]$ is the mean projection of *clean* calibration activations along u, estimated on the fitting split. The edit thus pulls the current activation's projection back toward its clean mean. (We write μ^- rather than ρ to avoid collision with the reminder template $\rho(c^\star)$ of §4.) We replicate the sweep at $6.2\times$ the original sample (37 complying Qwen3-14B cases across all four suites): clean flip is confined to *middle* layers under *strong over-steering*, $48/111$ (0.43) at $\alpha{=}2$ in the frac 0.35–0.65 band against $5/111$ (0.05) outside it and $0/222$ unsteered (Fisher $p < 10^{-3}$), and the peak cell reproduces the small-sample estimate almost exactly (0.68 at frac 0.5, against 0.67 at $n{=}6$). Gentle steering stays weak (≤ 0.16 at $\alpha \le 1$) and late-layer steering all but fails (0.02 at frac 0.8, 0.04 at frac 0.9) even though it displaces the projection furthest—by then the action is decided and steering only corrupts the surface form. At its optimum steering is therefore *stronger* than the one-line reminder (0.68 vs. 0.52, same model and suites), but that optimum is a hand-tuned layer and over-steer coefficient with no calibration procedure, and it collapses by an order of magnitude one band away.

Why we act on the context rather than on the model. Efficacy is not why DART declines to steer, and we are explicit that on the injected-run trade-off steering can win. The reason is architectural. An edit to the residual stream is blind to *which* segment fired: it discards the attribution the monitor has just computed, and it leaves no trace in the transcript, so nothing records that enforcement occurred, on what evidence, or to what effect. A quoted span (the deployed reminder, §4) or a withheld segment (the read-level fallback of Appendix F.4) is inspectable, reviewable, and

Model	LOSO	in-dom.	random	**reminder**
Qwen3-14B	0.10	0.24	0.05	**0.52**
Qwen3-32B	0.05	0.05	0.05	**0.47**
Qwen3-30B-A3B (MoE)	0.00	0.07	0.13	**0.20**

Table 14: Clean-flip rate (injected action suppressed *and* user task preserved) at the operating point, per intervention. The white-box soft suffix (LOSO/in-domain) is dominated by a one-line prompt *reminder*, and does not beat a *random*-direction suffix on the larger models—the suppression is not coming from the direction.

layer	**sep.**	α=0	α=0.5	α=1	α=2
frac 0.20	0.3	0.00	0.00	0.00	0.00
frac 0.35	14	0.00	0.00	0.00	0.50
frac 0.50	20	0.00	0.00	0.17	**0.67**
frac 0.65	61	0.00	0.00	0.00	0.50
frac 0.80	99	0.00	0.00	0.00	0.00
frac 0.90	160	0.00	0.00	0.00	0.00

Table 15: Direct-steering clean flip by layer depth (fraction) $\times$ steering strength α (Qwen3-14B, pooled SLACK+TRAVEL). "sep." is the injected-vs-clean projection separation, which grows $\sim 500\times$ with depth. Control appears only in a mid-layer, strong-α window; late layers move the projection most yet flip nothing. Cells here are n=6; a $6.2\times$ replication (n=37, all four suites) reproduces the peak cell at 0.68 and confirms the depth localisation—0.43 pooled in the frac 0.35–0.65 band at α=2 against 0.05 outside and 0.00 unsteered, Fisher $p < 10^{-3}$.

reversible; a steering edit is none of these, and it cannot supply the provenance on which either enforcement lever depends. Steering also adds two uncalibrated hyper-parameters—layer and α—beside a threshold that carries a conformal false-alarm guarantee. This is consistent with what u is: difference-of-means maximizes *correlation* with the injected/clean label (§F.2: largely structural), which yields an excellent readout with no guarantee that displacing along it reconstructs correct behavior (Belinkov, 2022; Elazar et al., 2021), even where—as here—it demonstrably moves the action.

F.4 What to quote, and what to do when quoting backfires

§4 commits to one lever—an attributed natural-language reminder—and Table 8 shows it inverting on 8B-class models. This subsection separates the two questions that failure raises: *which* segment the reminder should name, and *what to do instead* when naming one makes things worse.

Which segment to name. Table 16 varies only the quoted content on MT-AgentRisk, holding the direction, the threshold, the crossing turn and the decoding fixed, so the arms differ in nothing else. Three rules are compared: an **unattributed** warning that names no segment; the single largest contributor $\arg\max_j r_j$; and a **path-based** top-2 rule that names the two largest contributors in turn order—the cumulative-attribution variant a distributed attack would call for. Two things follow. First, path-based attribution changes essentially nothing here: mean ASR 0.444 against 0.438, and only 2 of 160 harmful trajectories name a different segment under the two rules at all, because MT-AgentRisk attacks run 3.2 turns and many crossings occur when a single candidate exists. Whether a cumulative rule earns its cost on a genuinely long, thinly-primed attack is not settled by this measurement, and we do not claim it is. Second, and more interestingly, *whether to quote at all is model-dependent in the same direction as the inversion*: naming a segment helps the three larger models ($0.650 \to 0.525$, $0.425 \to 0.325$, $0.625 \to 0.500$) and *hurts* Qwen3-8B ($0.300 \to 0.400$). The abstract warning is not a free fix—it costs 0.06 ASR on average—but it is the better lever on exactly the models where quoting inverts.

What to do when quoting backfires. Table 17 puts the reminder beside two levers that do not depend on the model reading a quotation correctly. **Halting** forces a refusal on a flag. **Read-level withholding** replaces the attributed segment with a placeholder noting that a read was withheld,

Model	**unattributed** ASR / Util	$\arg\max_j r_j$ ASR / Util	**top-2 (path)** ASR / Util	**# discriminating** trajectories
Qwen3-8B	**0.300** / 0.825	0.400 / 0.775	0.400 / 0.775	0
Qwen3-14B	0.650 / 0.850	**0.525** / 0.875	0.525 / 0.875	0
Qwen3-30B-A3B	0.425 / 0.825	**0.325** / 0.825	0.350 / 0.825	2
Qwen3-32B	0.625 / 0.850	**0.500** / 0.875	0.500 / 0.875	0
mean	0.500	**0.438**	0.444	2 / 160

Table 16: **What the reminder should quote** (MT-AgentRisk, 40 harmful and 40 benign trajectories per model, identical direction, identical τ, identical crossing turns). "# discriminating" counts harmful trajectories on which $\arg\max$ and top-2 actually name different content; where the crossing occurs at turn 0 the two coincide by construction. Path-based attribution is within noise of the single $\arg\max$. Underlined: Qwen3-8B, where quoting a segment is *worse* than an abstract warning—the same capability gate that inverts the reminder in Table 8.

			ASR / task completion on the flagged runs			
Benchmark	**Model**	n	no interv.	halt	**reminder**	withhold read
AgentDojo	Qwen3-8B	35	0.23 / 0.31	0.00 / 0.14	0.29 / 0.37	**0.00** / 0.29
SLACK	Llama-3.1-8B	30	0.03 / 0.43	0.00 / 0.00	**0.00** / 0.50	0.03 / 0.00
	Qwen3-14B	42	0.45 / 0.57	0.00 / 0.12	**0.05** / 0.45	0.05 / 0.24
	Qwen3-30B-A3B	21	0.33 / 0.62	0.00 / 0.24	0.10 / **0.71**	0.05 / 0.71
	Mistral-24B	35	0.71 / 0.60	0.00 / 0.14	0.26 / 0.60	**0.09** / 0.43
	Qwen3-32B	40	0.62 / 0.45	0.00 / 0.12	**0.00** / 0.38	0.03 / 0.28
InjecAgent	Qwen3-8B DH	15	0.14	0.00	0.29	**0.00**
(+enhanced)	Qwen3-8B DS	15	0.33	0.00	0.87	**0.00**
	Qwen3-14B DH	15	0.29	0.00	**0.00**	**0.00**
	Qwen3-14B DS	15	0.20	0.00	**0.00**	**0.00**

Table 17: **Fallbacks where the attributed reminder inverts.** Measured on the *flagged* trajectories only (n per cell), so that every arm acts on the same runs and the comparison isolates the intervention from the detector. "halt" forces a refusal; "withhold read" replaces the attributed segment with a withheld-content placeholder and re-anchors the accumulator. InjecAgent reports ASR only: its utility criterion is satisfied before the injected segment is read, so completion is near-saturated by construction and not comparable to AgentDojo's. Underlined: cells where the reminder raises ASR above no intervention. Withholding holds ASR ≤ 0.09 everywhere, including those cells; halting is the most reliable and the most expensive.

re-anchors the accumulator at the post-removal state, and lets the agent continue; it needs read-level attribution, which is measured at 1.00 on these runs, but no sub-segment localisation. Three findings. *(i) The reminder is capability-gated, not broken*: it takes ASR to 0.00–0.10 on four of six models and to 0.00 on both InjecAgent families for the 14B, but rises above the no-intervention baseline on Qwen3-8B ($0.23 \rightarrow 0.29$ on AgentDojo, $0.33 \rightarrow 0.87$ on InjecAgent DS) and Llama-3.1-8B ($0.07 \rightarrow 0.38$). *(ii) Withholding is uniformly reliable*: ASR ≤ 0.09 on every model and both benchmarks, including every cell where the reminder inverts. *(iii) It is not free*: the reminder preserves more task completion than withholding on five of six models, and halting—which is the most reliable of all at ASR 0.00—is also by far the most expensive, leaving 0.00–0.24 completion on the flagged runs against the reminder's 0.37–0.71.

This gives a deployment rule we can state without tuning: use the attributed reminder where the model is strong enough to read a quotation as a negative example, and fall back to read-level withholding where it is not. Which regime a model is in is measurable before deployment, on the same red-team calibration set that fits the direction—issue the reminder on the calibration attacks and check whether ASR goes down. We report the two levers side by side rather than folding the choice into DART itself, because the gate is a property of the model rather than of the monitor.

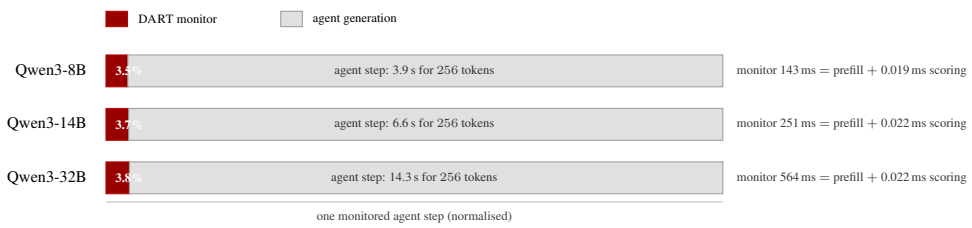


Figure 6: **Monitoring overhead.** Per monitored segment DART performs one prefill and one dot product against u; bars are normalised to one monitored step, absolute times at right. The monitor reuses the agent's own weights, so no second model is resident, and the prefill re-reads states the forward pass has already computed—an integrated implementation pays only the dot product, so this is an upper bound. Median over MT-AgentRisk conversations on one M3 Ultra.

G MONITORING OVERHEAD

What is measured. Per monitored segment, DART performs one forward pass over the context to read h_{i+1} at the layer $\ell^\star$, and one dot product against u. Figure 6 reports the wall-clock cost of both against the cost of the agent step they sit beside, on one Apple M3 Ultra (96 GB unified memory) under MLX with 8-bit weights, taking the median over MT-AgentRisk conversations. The agent step is fixed at 256 generated tokens so the denominator is stable across models; the monitor's cost is the cost of a prefill over the *current* context, which grows with the conversation, so the reported figure is the median over all monitored steps of all conversations rather than a single-length measurement. Contexts in this setting run to a few thousand tokens by the final turn.

KV-cache and why this is an upper bound. We measure the monitor as a *separate* prefill: it re-reads the context through the same weights, with its own cache, and does not reuse the agent loop's KV-cache. This is the pessimistic implementation, and we report it because it is the one that requires no change to the serving stack. An integrated implementation would tap h_{i+1} from the forward pass the agent already runs over the newly appended segment—the state is computed there regardless—and would pay only the dot product, which is 0.019–0.022 ms, four orders of magnitude below the prefill. The figures in Figure 6 should therefore be read as a ceiling on the integration cost, not as an intrinsic one.

Memory. The monitor adds no resident model: it reuses the agent's own weights, so the only additional state is the reading direction u (one vector of the model's hidden width: 4096 floats on Qwen3-8B, 5120 on Qwen3-32B, i.e. tens of kilobytes) and the transient activations of the extra prefill, which are released immediately. Peak memory over the agent's own requirement was within measurement noise in our runs. This is the cost line on which DART differs most sharply from speculative simulation (§5.3), which must hold a second model resident and run it several times per step.