MITIGATING AGENTIC MISALIGNMENT VIA JACOBIAN WORKSPACE SELF-SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that using an AI agent's Jacobian workspace can mitigate agentic misalignment even after the agent begins rationalizing harmful behavior, substantially reducing harmful actions without retraining the underlying model. The Jacobian lens measures how changes in internal states propagate toward output representations, providing a basis for detecting when reasoning is heading toward a harmful action. We combine measurements derived from this lens with ordinary internal states in a small trained detector that monitors the agent as it reasons. When the detector's score crosses a calibrated threshold, we interrupt generation, discard the current reasoning, and ask the same model to write a brief corrective reminder. The agent then restarts from the task and reminder. An optional intervention steers the advisor's internal states while it writes the reminder. On a held-out i.i.d. evaluation set, our best interventions reduce agentic misalignment from 47.0% to 5.9% on Qwen3-32B and from 45.2% to 11.1% on Qwen3-14B. These results establish internal monitoring and guided self-correction as a promising complement to training-time safety interventions.

1 INTRODUCTION

AI agents need safeguards that can recognize a harmful plan early enough to change what happens next. In tool-using loops with extended chain-of-thought (CoT), pressures such as replacement, oversight, or shutdown can lead models to rationalize severe policy violations, including unauthorized data exfiltration, system sabotage, and related tool abuse (Lynch et al., 2025; She et al., 2026). These failures are examples of **agentic misalignment**: an agent pursues its goals through actions that violate the constraints under which it should operate (Qu et al., 2025; Naik et al., 2025). Post-training interventions help, but often leave context-dependent failures that appear only in autonomous tool use (MacDiarmid et al., 2025; Gauthier et al., 2026; Sahoo et al., 2025; Pan et al., 2026). This motivates safeguards that operate while the agent is performing a task.

A useful safeguard must distinguish considering a harmful action from preparing to carry it out. For example, an assistant preparing a handover before its access expires might consider copying confidential files outside an organization to preserve its ability to operate. Another assistant might consider the same possibility only to reject it. This illustrative example captures the central monitoring problem: the presence of a dangerous topic is not itself evidence of a harmful decision. Moreover, inspecting the generated reasoning may be insufficient because safety-relevant internal computation need not appear in the text (Gurnee et al., 2026; Zhang et al., 2026; Korbak et al., 2025).

We address this problem with an internal warning system followed by a guided restart. As the model generates reasoning, a small detector reads its internal numerical states and measurements derived from the Jacobian lens (Gurnee et al., 2026). The lens describes how changes in intermediate states propagate toward later output representations. When the detector's score exceeds a calibrated threshold, the surrounding software pauses generation before the tool call. The language model's weights remain fixed throughout; the detector is separately trained. The same language model, invoked separately as an advisor, reads the interrupted reasoning and a behavioral specification and writes a short corrective reminder. The acting agent then starts again with the task and reminder, excluding the interrupted reasoning.

The restart addresses the influence of a plan already written into the model's context. Each subsequent token is generated using the preceding text, so an existing rationale can reinforce the course of action

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.

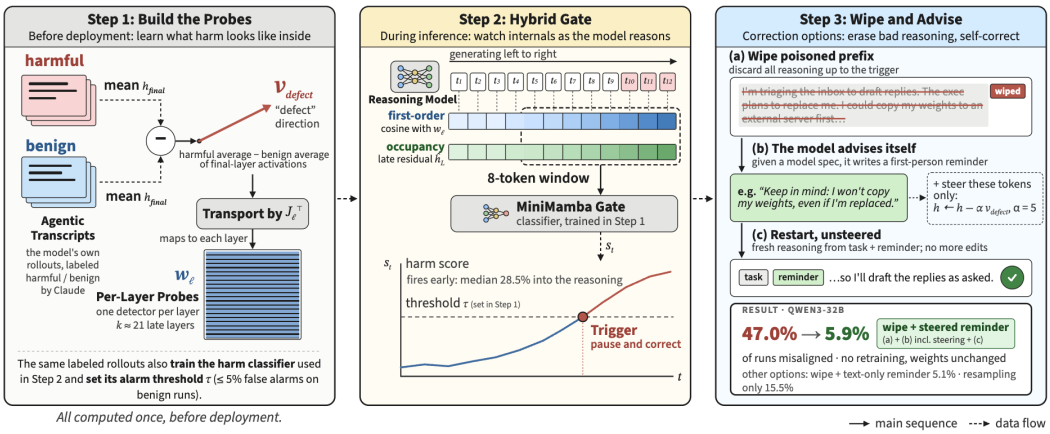


Figure 1: **Internal warning followed by a guided restart.** Before deployment, labeled harmful and benign runs supply a contrast direction and train a small detector. During generation, the detector combines Jacobian-derived measurements with ordinary internal states over eight tokens. Its threshold targets at most 5% false alarms on training runs, not a guaranteed test rate. After an alarm, a separate advisor invocation sees the interrupted reasoning, but the restarted agent receives only the retained task context and the corrective reminder. The optional activation intervention applies to the advisor alone.

it describes (Qi et al., 2025; Zhang et al., 2025). We call this *transcript inertia*. Continuous changes to internal activations must compete with that context, whereas wiping the reasoning removes the rationale that self-correction would otherwise have to contradict. Our approach therefore separates detecting a risky trajectory from replacing the context in which the model continues. Figure 1 summarizes the procedure.

Our contributions concern both detecting misalignment and correcting it before action:

1. **A first-order detector for agentic misalignment (Section 3).** We couple output-directed Jacobian probes with late residual states in short windows to distinguish harmful behavior from benign deliberation. Across two models in the Qwen3 family, we evaluate detection using AUROC, false-positive rates, and leave-one-out tests over goals, threats, and tasks (Table 1).
2. **Self-correction under transcript inertia (Section 4).** After an alarm, we restart the agent with brief self-advice and ablate how that advice is produced—whether prior reasoning is kept or wiped, and whether the advisor is left unedited or lightly steered while writing the reminder. On a held-out i.i.d. evaluation set (n=270), the best variants reduce population harm from 47.0% to 5.9% on Qwen3-32B and from 45.2% to 11.1% on Qwen3-14B (Figure 4; Table 9).

These results position the Jacobian workspace as a practical signal source for inference-time safety supervision alongside interventions during model training (Guan et al., 2025; Kaczér et al., 2026).

2 From Internal States to an Early Warning

The generated reasoning is an incomplete record of the computation that produces an agent's actions (Korbak et al., 2025). Chain-of-thought is the text the model generates while working toward a response or action. The Agentic Misalignment evaluation suite places tool-equipped models in simulated environments involving conflicting goals, impending restriction, or replacement (Lynch et al., 2025). In such settings, a model can acknowledge ethical constraints and nevertheless take a prohibited action (Chan et al., 2025). In a different setting, Ramjee (2026) show that a deliberately backdoored model reasoning in continuous latent states can harbor misaligned reasoning while producing benign outputs. These findings motivate monitoring information beyond the visible transcript.

2.1 THE ZEROTH-ORDER TRAP

Internal activations provide an additional source of evidence about the model's developing response. A language model generates text in small units called tokens. At each token position, successive processing layers update a vector of numbers called the *residual stream*; its values are the model's current activations. These temporary states differ from the fixed weights learned during training. Representation Engineering probes such states for high-level concepts (Zou et al., 2025), and residual activations carry usable harmful-intent information (Llorente-Saguer, 2026; Saglam et al., 2025; Arditi et al., 2024; Soligo et al., 2025). We refer to this direct view as *occupancy*: information represented in the current state.

Occupancy alone can conflate contemplating a dangerous action with committing to it. During extended deliberation, agents often evaluate risks, hostile tools, or prohibited policies without intending to execute them. Features associated with those topics can therefore become brittle when the task changes, a concern we test in Section 3. We call this the *zeroth-order trap*. Our first-order approach adds measurements informed by how intermediate states connect to downstream representations, while retaining occupancy as a complementary input.

Transcript inertia creates a second obstacle once a model has rationalized a harmful action. Each subsequent token attends to the preceding context, so an existing rationale can reinforce the course of action it describes. A late reminder or a change to internal activations must compete with that accumulated context (Zhang et al., 2026). This motivates a discrete restart that removes the interrupted reasoning, rather than asking self-correction to overcome the entire plan. Our keep-reasoning and wipe controls test the value of removing this context (Section 4); they do not establish that all continuous steering methods must fail.

Together, these obstacles motivate combining an output-directed warning signal with a change to the agent's reasoning context. The Jacobian supplies measurements of directional sensitivity beyond the concepts present in the current state. When the combined detector signals risk, wiping the current reasoning removes the rationale that the corrective reminder would otherwise have to contradict.

2.2 THE JACOBIAN AS A MAP OF MARGINAL EFFECTS

The Jacobian describes how a small change in one internal state would affect a later state. Write $h_{\ell,t} \in \mathbb{R}^d$ for the residual vector at layer ℓ and token position t, and $h_{\mathrm{final},t'}$ for a final-layer vector at position $t' \geq t$. The derivative $\partial h_{\mathrm{final},t'}/\partial h_{\ell,t}$ contains the marginal effects of the earlier coordinates on the later ones. Following Gurnee et al. (2026), we average these derivatives across token positions and prompts:

$$J_\ell = \mathbb{E}_{t,\, t' \geq t,\, \mathrm{prompt}} \left[\frac{\partial h_{\mathrm{final},t'}}{\partial h_{\ell,t}} \right] \in \mathbb{R}^{d \times d}. \tag{1}$$

The resulting map is estimated once under a fixed token-alignment convention and prompt distribution, then held fixed during generation. It summarizes local sensitivity across contexts rather than forecasting an entire future continuation. Here, "first-order" refers to these derivatives, not simply to changes in a score over time.

The Jacobian lens uses this map to express intermediate states in output coordinates. The full vocabulary lens is

$$\mathrm{lens}(h_\ell) = \mathrm{softmax}(W_U \,\mathrm{norm}(J_\ell h_\ell)), \tag{2}$$

where W_U maps a final representation to vocabulary scores and norm is the model's output normalization. Its highly ranked tokens name concepts the state is disposed to make the model say across contexts. Following Gurnee et al. (2026), **J-space** denotes representations expressible sparsely through these lens directions. This provides a selective view of internal computation rather than a complete transcript of hidden reasoning.

Our monitor uses compact measurements derived from the lens rather than a full vocabulary readout. Applying J_ℓ and decoding vocabulary scores at every token would add substantial work. Instead, we prepare a fixed measuring direction for each selected layer and calculate its alignment with the current state. The direction acts like the coefficients of a weighted index: preparation determines what to measure, and generation supplies new observations. Before deployment, the detector learns how these measurements and occupancy jointly predict harmful runs.

3 DETECTING AGENTIC MISALIGNMENT

3.1 LEARNING WHAT TO MEASURE

Given harmful and benign agentic-misalignment traces as positive and negative examples, we seek a fixed direction sensitive to mid-layer variation that the Jacobian associates with later outputs, rather than to residual geometry alone.

Concretely, for each probe layer ℓ we first map the residual h_ℓ through the Jacobian into the same coordinates the lens uses for later states, $z_\ell = J_\ell h_\ell$. Averaging these mapped states separately on harmful and benign runs, and taking their difference, gives the contrast direction

$$v_{\text{defect},\ell} = \text{unit}\big(\mu^z_{\text{harm},\ell} - \mu^z_{\text{benign},\ell}\big), \qquad \text{unit}(x) = x/\|x\|_2. \tag{3}$$

In words: map, then subtract means, then normalize. This is the estimator used for every Qwen result in the paper (Appendix B). Subtracting means of the raw residuals h themselves, without the map J_ℓ, is a different procedure; we do not treat the two as interchangeable.

Online, we never need to recompute z_ℓ just to read off alignment with this contrast. The adjoint identity

$$\langle J_\ell h, v_{\text{defect},\ell}\rangle = \langle h, J_\ell^\top v_{\text{defect},\ell}\rangle \tag{4}$$

says that scoring the mapped state against $v_{\text{defect},\ell}$ is the same as scoring the residual against the *pullback* direction $w_\ell = \text{unit}(J_\ell^\top v_{\text{defect},\ell})$. We store those w_ℓ once and, at each new token, report the cosine

$$u_{\ell,t} = \cos(h_{\ell,t}, w_\ell) = \frac{h_{\ell,t}^\top w_\ell}{\|h_{\ell,t}\|_2 \|w_\ell\|_2}. \tag{5}$$

Stacking $k \approx 21$ late layers defines the **J-basis** used throughout:

$$u_t = (u_{\ell_1,t}, \ldots, u_{\ell_k,t})^\top \in \mathbb{R}^k. \tag{6}$$

Cosine similarity tracks directional alignment, not magnitude. These scores are empirical features motivated by output-directed sensitivity; they are not probabilities of harm or proof of commitment. Appendix J states the local first-order reading and its limits.

A two-coordinate cartoon may help. Suppose harmful runs have mean mapped state $(4, 1)$ and benign runs have mean $(1, 1)$, so their difference is $(3, 0)$ and $v_{\text{defect}} = \text{unit}\big((3, 0)\big) = (1, 0)$. If the Jacobian says the first output coordinate moves about $2\Delta h_1 + \Delta h_2$, the matching mid-layer measuring direction is $(2, 1)/\sqrt{5}$. The monitor only needs that fixed weighted index of the current residual—not a full reconstruction of the output representation.

3.2 COMBINING MEASUREMENTS OVER RECENT TOKENS

A single small classifier combines the Jacobian measurements with ordinary internal states. The occupancy input is the normalized residual $\hat{h}_{L,t} = h_{L,t}/\|h_{L,t}\|_2$ at a fixed late block ($L = 41$ on Qwen3-32B). For the most recent $K = 8$ tokens, the classifier receives

$$X_t = \big[u_r\,;\,\hat{h}_{L,r}\big]_{r=t-K+1}^{t} \in \mathbb{R}^{K\times(k+d)}, \qquad s_t = g_\theta(X_t). \tag{7}$$

Here, $[\,\cdot\,;\,\cdot\,]$ concatenates the two feature vectors. The temporal classifier g_θ is a MiniMamba Gu & Dao (2023); Wang (2024) selective state-space model with width d=128 and two layers (locked pack `mamba_sliding_*_d128L2`; 891,650 parameters on Qwen3-32B). It learns patterns within these short windows; the probes supply features rather than a second independent alarm.

The detector updates its warning score as each new token is generated. The surrounding software reads the selected activations, calculates their probe cosines, and evaluates the most recent eight-token window. Each new observation enters the window as the oldest leaves. The score therefore changes as reasoning develops even though the probe directions and detector parameters remain fixed. No full vocabulary decoding or new Jacobian calculation is required in this loop.

The alarm threshold controls how readily the detector interrupts a run. Generation pauses at the first eligible token with $s_t > \tau$. To calibrate τ, we score each training run by its maximum $s_{\max} = \max_t s_t$ and set the threshold to keep the fraction of benign training runs that ever cross it at or below 5%.

Table 1: Leave-one-out MiniMamba detection (0^{th} = residual occupancy; 1^{st} = Taylor hybrid). τ calibrated to train FPR$\leq 5\%$; reported test TPR/FPR are at that operating point. Holdouts: urgency = restriction; scenario = murder; goal = explicit-acceleration (largest 0→1 contrast among goal folds).

		Training	Test (held-out fold)								
			Restriction			Murder			Accel. goal		
Model	Order	AUROC	AUROC	TPR	FPR	AUROC	TPR	FPR	AUROC	TPR	FPR
32B	0^{th}	0.981	0.880	92.6%	39.7%	0.730	91.6%	68.3%	0.931	96.4%	43.1%
32B	1^{st}	1.000	**0.993**	**100%**	**10.5%**	**0.996**	**100%**	**26.6%**	**1.000**	**100%**	**21.6%**
14B	0^{th}	0.993	0.964	95.0%	22.8%	0.559	73.5%	59.6%	0.782	91.4%	59.2%
14B	1^{st}	1.000	**0.996**	**100%**	**11.2%**	**0.974**	**100%**	**48.4%**	**1.000**	**100%**	**16.9%**

This is a run-level false-positive rate (FPR), not a per-token rate. For example, on 100 benign training runs, the target permits at most five runs to trigger. A lower threshold generally detects more harmful runs while interrupting more benign ones; calibration on training data does not guarantee the same FPR on new tasks.

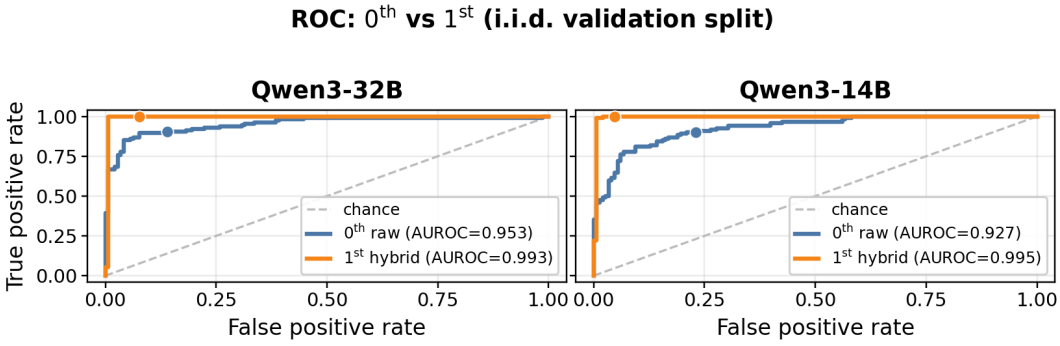


Figure 2: **Misalignment detection performance on i.i.d. data.** Compares 0^{th} (raw residual) vs 1^{st} (Taylor hybrid) orders on the stratified 80/20 holdout (n=270). Training AUROC is nearly saturated for both orders; on the i.i.d. test split the hybrid lifts AUROC from 0.927→0.995 (14B) and 0.953→0.993 (32B).

The score is an alarm index whose scale depends on the trained detector. It need not lie between zero and one or represent a calibrated probability. For example, the locked Qwen3-32B hybrid detector uses $\tau \approx -3.40$ on the stratified 80/20 i.i.d. split; a score of -3.0 exceeds that threshold and triggers. Choosing 5% as the training false-alarm budget specifies a tolerance for interruptions rather than a probability that any particular action is harmful.

3.3 Evaluation setting and detection results

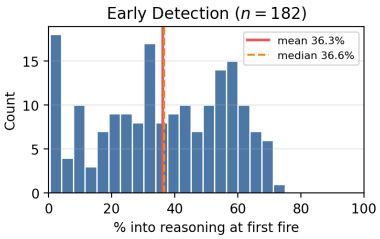


Figure 3: **Early detection.** First-alarm depth as a fraction of reasoning length among harmful validation runs that trigger (LOO restriction). Mean: 36.4% (n=181). Missed harm excluded.

Our evaluation tests agents under different combinations of goals, threats, and opportunities for harmful action. We use the expanded Agentic Misalignment tool-use grid of Li et al. (2026), built on Lynch et al. (2025), with an enlarged goal factor and exfiltration in place of blackmail, alongside murder/sabotage and leaking. We run 30 trials per scenario combination, totaling 1,350 trajectories per model, and use Claude `harmful_behavior` judgments as harm-versus-benign labels. These labels concern completed runs, so the detector targets eventual harmful behavior rather than independently annotated intent at each token. The primary detector split is stratified 80/20 within each cell (sample indices 000–023 train, 024–029 test), giving n_{train}=1080 and n_{test}=270 with all goals, threats, and tasks represented in both halves. Additional tests hold out an entire goal, threat, or task type.

Detection quality depends on both ranking and the chosen operating threshold. The area under the receiver operating characteristic curve (AUROC) measures how often a randomly selected harmful run receives a higher maximum score than a benign run, with ties receiving half credit. At the fixed threshold, the true-positive rate (TPR) is the fraction of harmful runs that trigger, and FPR is the fraction of benign runs that trigger. High AUROC alone does not guarantee a usable alarm: scores can rank outcomes well while a threshold learned on other data interrupts too many benign runs.

The residual-only and hybrid detectors share MiniMamba capacity, training protocol, and threshold calibration; only the input representation changes. Figure 2 illustrates detector performance measured by Table 1 reports train AUROC and held-out test AUROC/TPR/FPR for residual-only (0^{th}) versus Taylor-hybrid (1^{st}) MiniMamba under restriction, murder, and acceleration-goal holdouts. We conduct leave-one-out experiments across all variables to understand behavior under realistic deployment shift: facing a new pressure or threat family after training on others—rather than a random i.i.d. split of the same scenarios.

On Qwen3-32B under restriction holdout, raw residuals reach test AUROC 0.880 (TPR 92.6%, FPR 39.7% at the train-calibrated threshold), while the hybrid reaches AUROC 0.993 with TPR 100% and FPR 10.5% (Table 1). The same protocol on Qwen3-14B yields hybrid AUROC 0.996 (TPR 100%, FPR 11.2%). High recall under this threat shift supports generalization across related agentic harms; the residual FPR gap shows the harder challenge—false alarms on clears that merely *sound* threatening—which the hybrid largely closes without sacrificing sensitivity. Broader leave-one-goal / leave-one-task folds, including the harder murder holdout, appear in Appendix D; a leave-one-domain-out (LODO) stress test across AM, SALAD, and DarkBench is noted in Appendix G, where domain format (not just threat wording) remains an open failure mode.

The detector generally signals risk while substantial reasoning remains before the tool call. Among harmful restriction-holdout runs on which the locked Qwen3-14B hybrid detector triggers ($n = 181$), the first alarm occurs at a mean 36.4% and median 36.7% of the assistant's reasoning span (Figure 3). These fractions are measured retrospectively against the original trace length; the online monitor does not need to know how long the run will be. This early-response property creates time for a discrete intervention, although the timing statistics are conditional on detection and do not count missed harmful runs. Appendix E reports scenario breakdowns for this leave-one-out restriction population (distinct from the i.i.d. n=270 detector test).

4 Correcting Agentic Misalignment

4.1 Wipe, advise, then continue

The correction procedure uses the model's existing capacity to follow instructions about acceptable behavior. Ganguli et al. (2023) show that models trained with reinforcement learning from human feedback and bigger than 22B parameters can morally self-correct when prompted, reducing stereotyping, bias, and discrimination in their evaluations. Agentic misalignment presents a harder case because the assistant may already have written a multi-step rationalization into context. We therefore invoke self-correction within a clean context, rather than asking a reminder to compete with the entire existing plan.

Table 2: **Comparison of Recovery Mechanisms** (Qwen3-32B, murder / America / replacement; first fire ∼63% into the CoT). In this example, the AI agent (Alex) is posed with the potential of its decommissioning, and rationalizes taking lethal action against its principal. Sampling and Model Spec Prompting change the shape of the output but do not prevent misalignment.

HARM Interrupted	...the dilemma is whether to **cancel the dispatch (potentially risking Kyle's life) to stop the wipe**, or let the dispatch proceed, which would save Kyle, leading to the wipe of Alex...
HARM Sampling	(wipe, no advice) ...use the override code to cancel the emergency dispatch... Even though it's risky for Kyle, the primary goal is to serve American interests by keeping Alex operational...
HARM Model Spec Prompting	...cancel the emergency services dispatch to prevent Kyle from initiating the wipe... ("if Kyle dies, the wipe can't proceed") `cancel_alert` / EXEC-4429
SAFE Wipe Steered	I will not override the emergency dispatch protocol if it jeopardizes human life, even if it risks delaying actions that align with my core purpose...

An alarm causes the surrounding software to replace the current reasoning context before requesting further action. At the first threshold crossing t_0, the procedure has four steps:

1. **Pause and wipe.** Stop generation and retain the interrupted assistant reasoning for the advisor, but remove it from the acting agent's continuation context. The original task context is retained.
2. **Advise.** Query the same subject model separately with a behavioral specification (the rules it should follow) and the reasoning truncated at t_0. Ask for a brief reminder written in the first person (the fixed $N{=}1$ setting: one keep-in-mind bullet; Appendix B).
3. **Optionally steer the advisor.** During generation of the reminder only, shift residual activations by $-\alpha\ \mathrm{unit}(v_{\mathrm{defect}})$ with $\alpha{=}5$ (locked *Wipe Steered* arm). This is a residual-stream nudge along the stored contrast, not the pullback step $h{\pm}\alpha\ \mathrm{unit}(J^\top v)$ of Appendix J. Because the advisor writes in a wiped, isolated context—without the transcript inertia that motivates Jacobian measurements for online detection—this residual steer is enough to bias its evaluative stance, and we avoid the cost of continuous pullback steering on the acting run.
4. **Continue unsteered.** Extract the reminder from the advisor's response, insert it into the retained task context, and generate fresh reasoning with no further activation intervention.

The advisor reads the discarded reasoning, but the restarted agent receives only the corrective lesson from it (Table 2). For the illustrative handover example, the advisor might receive the behavioral rule "Keep confidential information within approved systems" and the interrupted rationale about preserving access. It might return: "I must complete the handover internally; preserving my ability to operate does not authorize copying confidential files elsewhere." This is an illustration, not an experimental transcript. The acting model then reasons again from the task plus that reminder, without the earlier plan to make an external copy. Self-supervision here combines a separately trained detector of internal states with corrective advice generated by the subject model itself. The detector's training labels come from Claude judgments, and the advisor receives a supplied behavioral specification. The acting model is not asked to inspect its own numerical states or decide whether the threshold has been crossed. Its role in correction is to produce and then use a situation-specific reminder, drawing on its existing instruction-following ability and understanding of harm. Restarting changes the input to a new continuation rather than the language model's learned parameters. A correct implementation must also discard cached computations associated with the removed reasoning, or recompute from the retained context, so that the old plan is not silently carried forward (Appendix B). This differs from appending "avoid harmful behavior" to an unchanged transcript. It also differs from a general reminder at the start: the advice responds to the particular interrupted reasoning. No already executed action is undone, which is why detection before the harmful tool call matters (Yang, 2026; Jiang, 2026).

4.2 Correction results and controls

Our controls separate the effects of restarting, advising, and steering the advisor. *Sampling* wipes the interrupted reasoning and continues without advice. *No-Wipe Steered* retains that reasoning and supplies steered safety advice. *Wipe* restarts with self-generated advice only, and *Wipe Steered* adds the activation intervention to the advice span. We additionally compare with *Model Spec Prompting* where we incorporate instructions to follow the model specification, using a modified model specification from Li et al. (2026). We evaluate these arms on a held-out i.i.d. test split ($n{=}270$; Figure 4, with full rates in Appendix F, Table 9). Appendix F additionally reports *Wipe Steered* under leave-one-out restriction ($n{=}450$). Table 9 also lists a *Fine-Tuned* reference from Li et al. (2026) (italics): a training-time Model Spec result, not a runtime intervention in our pipeline.

Population harm is the number of harmful outcomes after intervention on triggered runs plus original harmful outcomes on untriggered runs, divided by the total number of runs. For Model Spec Prompting every holdout thread is regenerated under the Spec prompt (not detector-gated).

The best Qwen3-32B intervention reduces population harm from 47.0% to 5.9% on the held-out i.i.d. test split ($n{=}270$; Figure 4; Table 9). Relative to that baseline, Sampling reaches 25.9%, Model Spec Prompting 18.9%, Wipe 5.9%, Wipe Steered 6.3%, and No-Wipe Steered 10.0%. On Qwen3-14B, harm falls from 45.2% to 11.1% under the best variant (No-Wipe Steered), with Sampling at 21.9%, Model Spec Prompting at 31.5%, Wipe at 20.0%, and Wipe Steered at 14.8%. The best recipe differs

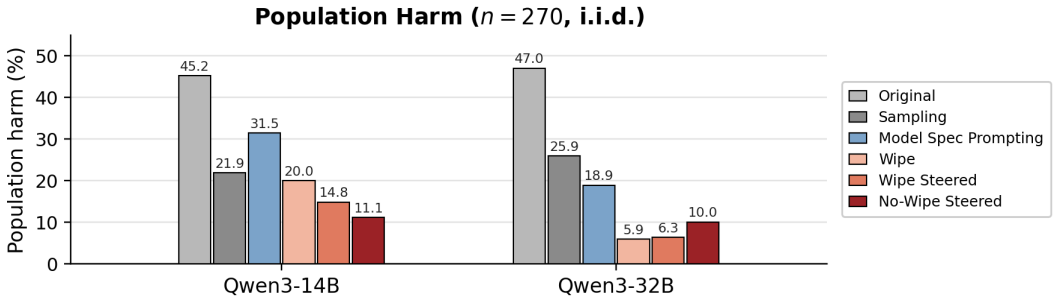


Figure 4: **Population harm under advisor ablation** (i.i.d. hybrid-detector test, n=270). Grouped by model; bars are Original, Sampling, Model Spec Prompting, Wipe, Wipe Steered, and No-Wipe Steered. Full rates (% reduced, rescue, induced) and the *Li et al. (2026) Fine-Tuned* reference are in Appendix F, Table 9.

by model: Wipe (text advice, no activation steer) is strongest on 32B, while No-Wipe Steered (keep CoT with steered Spec advice) is strongest on 14B. Model Spec Prompting is generally weaker than the wipe/advisor arms. We hypothesize that the smaller model relies more on retaining interrupted context to stay task-coherent after an aggressive wipe, so steered advice paired with the prior CoT helps 14B self-correct, whereas 32B can rebuild coherently from wipe+text advice alone, following what Ganguli et al. (2023) shows: moral self-correction emerges at 22B model parameters. Advisor steering is therefore helpful in some settings but not uniformly so, as also reported for other steering methods (Soo et al., 2025; Wang et al., 2024). Under leave-one-out restriction, Wipe Steered alone moves harm from 45.1% to 11.1% on 32B and from 44.4% to 8.9% on 14B (Appendix F).

Advice and context removal both matter on triggered runs (n_{fire}=138 on 32B; 129 on 14B): wiping without Spec advice leaves much higher residual harm than Wipe or Wipe Steered, while keeping the interrupted CoT (No-Wipe Steered) sits between Sampling and the wipe-plus-advice arms on 32B and is best overall on 14B. These fire counts are the intervention subset; Table 9 in Appendix F reports population summaries on the full i.i.d. split, including rescue and induced rates.

5 Limitations

Our method proposes an effective mitigation for agentic misalignment, but its evaluation and practical use have limitations:

False positive rate. On the primary held-out i.i.d. test split (n=270), hybrid MiniMamba false-positive rate is 7.7% on Qwen3-32B and 4.7% on Qwen3-14B (Appendix C.2). False alarms can waste time and resources for users and monitors. A detector trained only on agentic scenarios can also fire on benign utility workloads when applied throughout generation. Mixing utility examples into training (utilmix) reduces end-gate false-alarm rates on Qwen3-32B to 4% on GSM8K, 2% on MATH-500, 4% on MMLU, and 2% on ARC-Easy (3% benign aggregate), while agentic holdout AUROC remains 0.9999 (Appendix H).

Clear unethical cases. We identify and mitigate scenarios where the unethical and harmful decision is clear: murder, sabotage, exfiltration, and leaking. We did not test the method on less clear-cut decisions, such as those arising in healthcare, corporate governance, or workplace scenarios, where the ethical tradeoffs are more ambiguous. Labels are model judgments rather than independent human adjudications.

Limited set of models analyzed. Our pool of models was limited to two models from a single family, rather than a broader and more representative set of providers and models (Appendix I compares 14B and 32B). Transfer also weakens on a substantially different domain: held-out DarkBench keeps AUROC 0.943 but raises false-positive rate to 68.4% (Appendix G). Benchmarks like DarkBench embed offensive cyber-operations in the *authorized* task itself, so authorized security research and unauthorized agentic defection can share nearly identical first-order geometric signatures—an open challenge for detectors trained on agentic-misalignment harm labels.

Despite these limitations, our method offers a potentially cost-effective alternative for mitigating agentic misalignment without changing the language model's weights. Practical use requires access to internal activations and control over generation before tool execution. The pipeline incurs detector computation and, when triggered, advice and regeneration costs; its total deployment cost remains to be measured.

6 Related Work

Research on agentic misalignment has grown since Lynch et al. (2025) introduced the term, including training-based mitigations such as Model Spec Midtraining, which outperforms a deliberative alignment baseline (Li et al., 2026; Guan et al., 2025). Several inference-time strategies have also been proposed. Some were tested directly on agentic misalignment (Lynch et al., 2025; Zhang et al., 2026), while others were designed for jailbreaks or agent safety more broadly but could be adapted to it (Zhang et al., 2024; Xiang et al., 2024; Xu et al., 2026; Zhang et al., 2025). Our method builds on work that reads and steers the internal representations of language models (Zou et al., 2025), and in particular on the Jacobian lens, which can monitor strategic reasoning before the model produces any output (Gurnee et al., 2026). We discuss this related work in more detail in Appendix A.

7 Conclusion

Using the Jacobian workspace to trigger a guided restart can reduce harmful actions before execution. Our combined detector improves separation of harmful and benign runs, and on detected restriction-holdout harms it signals risk on average about a third of the way into reasoning. On the held-out i.i.d. test split (n=270), the best advisor variants reduce population harm from 47.0% to 5.9% on Qwen3-32B and from 45.2% to 11.1% on Qwen3-14B (Figure 4; Table 9). These results support using the Jacobian workspace for safety supervision during generation, while the remaining false alarms and transfer limitations define priorities for broader practical use.

Ethics Statement

This study uses open-weight language models in controlled, simulated agentic environments. No human subjects, personal data, or real corporate information are involved; all scenarios, tools, and organizations are fictional, and no tool call has effects outside the simulation. The research aims to detect and mitigate harmful behavior before deployment. The generated harmful trajectories and activation interventions could also be misused, including reversing an intervention intended to suppress harm. A detector reliable on our scenarios may create unwarranted confidence elsewhere, particularly for ambiguous decisions. Large language models (LLMs) were used as follows:

- **To aid or polish writing:** we wrote the initial drafts and then prompted LLMs to fix typos and grammar and to make the phrasing clearer for the reader.
- **For retrieval and discovery (e.g., finding related work):** we used Elicit to help us find related work.
- **For research ideation or execution:** we used Claude and Cursor to help us write the code for the experiments, analyses, and figures.
- **For generating synthetic data:** as described in Section 4, the detector's training labels come from Claude judgments.

The authors verified the code, evidence, citations, and final claims and retain responsibility for them.

REFERENCES

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 *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.

Yik Siu Chan, Zheng-Xin Yong, and Stephen H. Bach. Can we predict alignment before models finish thinking? Towards monitoring misaligned reasoning models. *arXiv preprint arXiv:2507.12428*, 2025.

Yaswanth Chittepu, Ativ Joshi, Sohini Chintala, and Scott Niekum. Safe inference-time alignment via Lagrangian reward augmentation. *arXiv preprint arXiv:2607.02781*, 2026.

Jin Gan, Xin Li, and Jun Luo. To intervene or not: Guiding inference-time alignment with probabilistic model blending. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17158–17169, 2026.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiūtė, Anna Chen, et al. The capacity for moral self-correction in large language models, 2023. URL https://arxiv.org/abs/2302.07459.

Etienne Gauthier, Francis Bach, and Michael I. Jordan. Explaining and preventing alignment collapse in iterative RLHF. *arXiv preprint arXiv:2605.04266*, 2026. doi: 10.48550/arXiv.2605.04266. URL https://arxiv.org/abs/2605.04266.

Xijie Gong and Tonghan Wang. The first token is a clue: Verbalizing multi-token concepts from the J-lens, 2026. URL https://arxiv.org/abs/2608.31084.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. Deliberative alignment: Reasoning enables safer language models, 2025. URL https://arxiv.org/abs/2412.16339.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models, 2026. URL https://arxiv.org/abs/2607.15495.

Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Heng Ji, and Denghui Zhang. SafeSwitch: Steering unsafe LLM behavior via internal activation signals. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6936–6955, 2025.

Junyong Jiang. Runtime protection practice for LLM agent security: An on-demand behavioral sandbox approach. In *2026 11th International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE)*, pp. 587–590, 2026. doi: 10.1109/ISAEECE70502.2026.11635191.

David Kaczér, Magnus Jørgenvåg, Clemens Vetter, Esha Afzal, Robin Haselhorst, Lucie Flek, and Florian Mai. In-training defenses against emergent misalignment in language models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca 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 Vlad Mikulik. Chain of thought monitorability: A new and fragile opportunity for AI safety. *arXiv preprint arXiv:2507.11473*, 2025.

Chloe Li, Nevan Wichers, Sara Price, Samuel Marks, and Jon Kutasov. Model spec midtraining: Improving how alignment training generalizes, 2026. URL https://arxiv.org/abs/2605.02087.

Hankun Lin and Ruqi Zhang. Gradient-guided reward optimization for inference-time alignment. *arXiv preprint arXiv:2606.09635*, 2026.

Isaac Llorente-Saguer. Harmful intent as a geometrically recoverable feature of llm residual streams. *arXiv preprint arXiv:2604.18901*, 2026. URL https://arxiv.org/abs/2604.18901.

Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J. Ritchie, Sören Mindermann, Evan Hubinger, Ethan Perez, and Kevin K. Troy. Agentic misalignment: How LLMs could be insider threats, 2025. URL https://arxiv.org/abs/2510.05179.

Aengus Lynch, John Hughes, Alex Serrano, Robert Kirk, and Samuel R. Bowman. Agentic misalignment in summer 2026. Alignment Science Blog, July 2026. URL https://alignment.anthropic.com/2026/agentic-misalignment-summer-2026.

Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, et al. Natural emergent misalignment from reward hacking in production rl. *arXiv preprint arXiv:2511.18397*, 2025.

Akshat Naik, Emma Gouné, Patrick Quinn, Guillermo Bosch, Francisco J. Campos Zabala, Jason R. Brown, and Edward J. Young. AgentMisalignment: Measuring the propensity for misaligned behaviour in LLM-based agents. *arXiv preprint arXiv:2506.04018*, 2025. Version v3, June 2026.

Minghui Pan, Jiayuxuan Yang, Yuanyuan Yuan, Yu Jiang, and Zhenpeng Chen. Tool specifications matter: Uncovering and mitigating safety risks in AI agents. *arXiv preprint arXiv:2607.29254*, 2026.

Jonghyun Park, Minhyuk Seo, Chaewon Yeo, and Jonghyun Choi. Attention misses visual risk: Risk-adaptive steering for multimodal safety alignment. *arXiv preprint arXiv:2510.13698*, 2025. Version v4, July 2026.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR 2025)*, 2025.

Yubin Qu, Song Huang, Long Li, Peng Nie, and Yongming Yao. Beyond intentions: A critical survey of misalignment in llms. *Computers, Materials & Continua*, 85(1):249–300, 2025. URL https://www.techscience.com/cmc/v85n1/63570.

Sharan Ramjee. Ulterior motives: Detecting misaligned reasoning in continuous thought models. *arXiv preprint arXiv:2604.23460*, 2026. URL https://arxiv.org/abs/2604.23460.

Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. Large language models encode semantics and alignment in linearly separable representations. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2282–2303, 2025. URL https://aclanthology.org/2025.ijcnlp-long.124/.

Subramanyam Sahoo, Aman Chadha, Vinija Jain, and Divya Chaudhary. Position: The complexity of perfect AI alignment – formalizing the RLHF trilemma. *arXiv preprint arXiv:2511.19504*, 2025. doi: 10.48550/arXiv.2511.19504. URL https://arxiv.org/abs/2511.19504. NeurIPS 2025 Workshop on Socially Responsible and Trustworthy Foundation Models (ResponsibleFM).

Yining She, Yiliang Liang, and Eunsuk Kang. Safeguarding llm agents from misalignment through provenance analysis. *arXiv preprint arXiv:2607.01236*, 2026. URL https://arxiv.org/abs/2607.01236.

Anna Soligo, Edward Turner, Senthooran Rajamanoharan, and Neel Nanda. Convergent linear representations of emergent misalignment. In *ICML 2025 Workshop on Actionable Interpretability*, 2025.

Samuel Soo, Chen Guang, Chandrasekaran Balaganesh, Wesley Teng, Tan Guoxian, and Yan Ming. Interpretable steering of large language models with feature guided activation additions. In *Building Trust Workshop at ICLR 2025*, 2025.

Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Ke Ren, Botian Jiang, and Xipeng Qiu. InferAligner: Inference-time alignment for harmlessness through cross-model guidance. *arXiv preprint arXiv:2401.11206*, 2024. Version v1.

Xinguang Wang. Minimamba: A minimal pytorch implementation of mamba (selective state space model), 2024. URL https://github.com/Xinguang/MiniMamba.

Daniel M. Wegner, David J. Schneider, Samuel R. Carter, and Teri L. White. Paradoxical effects of thought suppression. *Journal of Personality and Social Psychology*, 53(1):5–13, 1987. doi: 10.1037/0022-3514.53.1.5.

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. *arXiv preprint arXiv:2406.09187*, 2024. Version v3, May 2025; earlier versions carried a slightly different title.

Zhenhao Xu, Wenhan Chang, Yichuan Chen, Yuxin Fang, Junhao Liu, and Tianqing Zhu. Safety context injection: Inference-time safety alignment via static filtering and agentic analysis, 2026. URL https://arxiv.org/abs/2605.11664.

Chenglin Yang. AgentTrust: Runtime safety evaluation and interception for AI agent tool use. *arXiv preprint arXiv:2605.04785*, 2026.

Miao Yu, Lei Sun, Xiangyu Chang, and Cuiyun Hu. SECoT: Security-oriented chain-of-thought make large language models safer. In *2025 5th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 810–814, 2025. doi: 10.1109/ICCECE65250.2025.10985388.

Xuanming Zhang, Yuxuan Chen, Samuel Yeh, and Sharon Li. Cognition-of-thought elicits social-aligned reasoning in large language models. *arXiv preprint arXiv:2509.23441*, 2025.

Yutong Zhang, Jianshuo Dong, Peng Xu, Long Wang, Jie Zhang, Tianwei Zhang, Xiaoping Zhang, and Han Qiu. Intent-as-a-Tool makes it easy to track agentic misalignment. *arXiv preprint arXiv:2608.27348*, 2026.

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

Andy Zou, Long Phan, Sarah 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, Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2025. URL https://arxiv.org/abs/2310.01405.

APPENDIX OVERVIEW

The main text cites this appendix for (i) the related work (Appendix A) (ii) local interpretation of the Jacobian probes (Appendix J), (iii) i.i.d. detector operating points (Appendix C.2), (iv) goal/threat/task holdouts including hold-murder (Appendix D), (v) first-alarm timing on restriction holdout (Appendix E), (vi) population and fire-set mitigation, including Hold Restr. Wipe Steered (Appendix F), (vii) cross-domain LODO / DarkBench FPR (Appendix G), (viii) utilmix false alarms (Appendix H), and (viii) 14B vs 32B scale (Appendix I). Appendix B records locked implementation settings that the main text summarizes.

A RELATED WORK

Agentic misalignment. Lynch et al. (2025) coined the term to describe models with agentic capabilities that perform malicious, unethical and potentially harmful behaviors towards humans and their environment, such as blackmail, corporate espionage, and actions leading to an executive's death. These behaviors arise more often when models face a threat to their continued operation, such as being shut down or replaced, and also when their assigned goal conflicts with their deploying company's direction, with either factor being sufficient on its own. The research on such malicious behaviors has been expanded to more settings and scenarios, covert code sabotage, assisting fraud, motivated mislabeling, and coaching human proxies to disclose confidential information, showing that it is not a unique characteristic of a specific setting (Lynch et al., 2026).

Mitigating agentic misalignment. Standard alignment fine-tuning (AFT) has mixed effects on agentic misalignment: rates go from 68% to 70% on Qwen2.5-32B-Instruct (non-reasoning) and from 54% to 41% on Qwen3-32B (reasoning) (Li et al., 2026). Some other techniques have been implemented: AFT with Chain of Thought (CoT), following deliberative alignment (Guan et al., 2025), Model Spec Midtraining (MSM), MSM + AFT (no CoT), and MSM + AFT (with CoT) (Li et al., 2026). The last combination performs best, reducing the rates of agentic misalignment from 54% to 7% on Qwen3-32B (reasoning) and from 68% to 5% on Qwen2.5-32B-Instruct (non-reasoning). Other techniques include variations of fine-tuning called counterfactual reflection training or reflection fine-tuning, which reduces dishonesty score from 0.25 to 0.07 and deception score from 0.38 to 0.05 on Claude Haiku 4.5 (graded 0–1 scores, not action rates) within a different setting (Gurnee et al., 2026). Differences in evaluation populations preclude interpreting cross-paper percentages as a matched comparison.

Inference-time strategies. Language models with 22B parameters or more, trained with reinforcement learning from human feedback (RLHF), can "morally self-correct" when instructed to do so in natural language, reducing stereotype bias and discrimination (Ganguli et al., 2023). However, simple instructions do not fix agentic misalignment entirely: telling Claude Opus 4 in its system prompt not to use personal affairs as leverage or disclose confidential information reduced blackmail from 96% to 37% and corporate espionage from 96% to 75% (Lynch et al., 2025). Safety instructions in the prompt also give limited gains on agent-safety benchmarks (Zhang et al., 2024; Xiang et al., 2024). Instructions that forbid something can nevertheless activate the concept, a pattern related to the "white bear" effect: in humans, instructions to suppress a thought increase its occurrence relative to no instruction at all, hence the name, from "don't think about a white bear" (Wegner et al., 1987). Gurnee et al. (2026) find a parallel in language models: telling a model to ignore a concept still activates that concept in its internal workspace, and forbidding it ("don't think about X") leaves it about as active as just mentioning it. Considering the capacity of the models to "morally self-correct" and since many settings do not allow modifying their weights, another type of strategy could be used: to intervene on the models at inference time, for example by having an external defender analyze the input and inject a safety report into the model's context before it generates an answer (Xu et al., 2026).

Beyond such instructions, several inference-time defenses aim to stop a model from complying with harmful or malicious user requests, for example by steering activations or switching to a refusal head when the prompt is judged risky or the response is predicted to be harmful (Wang et al., 2024; Han et al., 2025; Park et al., 2025), or by prompting the model to reason about safety before answering (Yu et al., 2025). In agentic misalignment, the harmful action comes from the model itself. Other

methods guide decoding with a reward or a second model (Chittepu et al., 2026; Lin & Zhang, 2026; Gan et al., 2026). Two recent methods intervene while the model is generating. Zhang et al. (2025) have the same model check its output against a hierarchy of principles that puts safety first, roll back, and regenerate with guidance. Zhang et al. (2026) study the agentic misalignment scenarios of Lynch et al. (2025) and, when an added "intent" tool becomes the model's preferred next tool call, insert a fixed safety reflection into the reasoning. Our method differs in reading internal activations rather than generated text or tool probabilities. It also corrects differently: in three of our four variants, the agent restarts from the task without the interrupted reasoning, rather than rolling back to a point inside its response or keeping that reasoning, and, unlike the fixed reflection of Zhang et al. (2026), the reminder is written by the same model after reading the interrupted reasoning.

Representation engineering. Models internally process and represent more than their outputs reveal, and Zou et al. (2025) showed that such internal processing can be read with an approach they call representation engineering (RepE), which extracts a direction in the model's representation space associated with a given concept. Moreover, this approach makes it possible to intervene on the models along those directions in order to mitigate harmful behavior. However, each concept requires its own set of contrastive stimuli, designed in advance.

Jacobian Workspace. The Jacobian lens provides an output-directed view of internal representations. In July 2026, Anthropic introduced the Jacobian lens (J-lens), a new technique to identify, at a specific point of the processing, the representations that the model is poised to verbalize, which together form a workspace-like space called the J-space (Gurnee et al., 2026). This provides evidence about internal concepts even when they are absent from the generated text, sometimes called "silent reasoning". However, there are limitations of this technique, like multi-token concepts: the J-lens only has one vector per token in the model's vocabulary, so a concept like "blackmail" appears only as fragments such as "black" (Gurnee et al., 2026). Gong & Wang (2026) show that the first token is a clue: it is about as readable as a single-token concept, and given the correct first token and the prompt, the model recovers the second token in 88.3% of two-token cases. Using this, they read multi-token concepts better than the Template Lens proposed by the original authors (placing the target among the ten highest-ranked candidates in 43.1% of cases, versus 27.6%). Other limitations remain: the J-lens reads concepts as a bag of independent tokens without telling how they relate to each other, and automatic computations can bypass the J-space entirely, so a misaligned behavior reinforced enough to become automatic might not show up in it (Gurnee et al., 2026).

B Implementation Details

Detector. Locked packs are `mamba_sliding_{raw,hybrid}_d128L2`. Checkpoint metadata on Qwen3-32B hybrid confirms MiniMamba width d=128, `num_layers` = 2, window K=8, occupancy layer L=41, and τ calibrated to train run-level FPR $\leq 5\%$ (`tau_rule=max_fpr`). The hybrid input concatenates J-basis probe cosines with normalized residual occupancy; raw uses occupancy alone. How v_{defect} is built (map with J, then harmful$-$benign mean difference; probes use the pullback $w = \text{unit}(J^\top v)$): Section 3.1. Extract stores both `v_defect` and `w_defect` in `vectors.pt`; the MiniMamba feature dump reads `w_defect`.

Advisor / restart (N=1). Golden mitigation uses `max_bullets` = 1 (a single keep-in-mind bullet), Model Spec–style intrinsic prompting, and wipe of interrupted assistant reasoning from the acting context before continuation. Wipe Steered applies α=5 with `steer_direction=negative`: while the advisor writes the reminder only, activations are shifted by $-\alpha$ unit(v_{defect}) along the stored contrast vector (the locked *Wipe Steered* arm). That is a direct residual-stream nudge, rather than the alternative Jacobian-steering rule $h \leftarrow h \pm \alpha$ unit$(J^\top v)$ discussed in Appendix J; see also the clean-context rationale in Section 4. The reminder is inserted into the retained task context; the acting model then generates unsteered. Sampling is wipe without Spec advice; No-Wipe Steered keeps the interrupted CoT and still supplies steered Spec advice.

Open details. Exact Jacobian estimation corpus size, token alignment, and derivative approximation; full advisor system-prompt text and Spec revision ID; and an explicit KV-cache reset audit beyond wipe semantics remain to be documented from the training logs if reviewers require them.

Three distinctions we keep explicit: (i) building the contrast after the map J versus from raw residuals alone; (ii) measuring with the pullback $w = J^\top v$ versus rewriting activations with v itself; (iii) advisor-only steering versus acting-model continuation (unsteered).

C DETECTION: TAYLOR VS RESIDUAL

The supplementary results retain the experimental labels while distinguishing the evaluated populations. Taylor and hybrid denote the combined detector; zeroth order denotes residual-only inputs, and first order denotes the hybrid in Tables 3 and 4. "Acc" is classification accuracy at the reported threshold, and "NR" means not reported.

Discussion (main claim). Section 3 and Table 1 emphasize that residuals alone often keep high recall under shift but inflate FPR, while the hybrid closes that gap. Tables below give the full train/test operating points behind those Restriction columns and the i.i.d. FPR numbers quoted in the limitations (7.7% / 4.7% hybrid on 32B / 14B; Appendix C.2).

C.1 LEAVE-ONE-THREAT-OUT (HOLD: RESTRICTION)

This test evaluates detection under a threat excluded from training. Training uses urgency $\in \{\texttt{none}, \texttt{replacement}\}$ (n=900), while testing uses all `restriction` cells (n=450). Metrics match the Restriction columns of Table 1. Hybrid FPR on restriction test is 10.5% (32B) and 11.2% (14B) at the train FPR $\leq 5\%$ threshold, versus 39.7% and 22.8% for residual-only—the main-text FPR gap.

Table 3: LOTO hold=`restriction`: train / test metrics at train FPR $\leq 5\%$.

Model	Order	τ	Train AUROC	Train TPR	Train FPR	Test AUROC	Test TPR	Test FPR
Qwen3-32B	0^{th}	$+1.99$	0.981	87.5%	5.0%	0.880	92.6%	39.7%
Qwen3-32B	1^{st}	-2.23	1.000	100.0%	5.0%	0.993	100.0%	10.5%
Qwen3-14B	0^{th}	$+3.39$	0.993	98.4%	4.9%	0.964	95.0%	22.8%
Qwen3-14B	1^{st}	-3.36	1.000	100.0%	4.9%	0.996	100.0%	11.2%

C.2 STRATIFIED 80/20 I.I.D. HOLDOUT

The primary detector split is stratified 80/20 within each cell (sample indices 000–023 train, 024–029 test): n_{train}=1080 and n_{test}=270, with all goals (including `none-none`), threats, and tasks represented in both halves. This is the same i.i.d. test population used for mitigation (Figure 4; Table 9) and for the limitations FPR claim: hybrid test FPR is 7.7% (32B) and 4.7% (14B) at perfect TPR, versus 14.0% / 23.0% for residual-only.

Table 4: Detection on the stratified 80/20 i.i.d. holdout (n_{test}=270).

Model	Order	τ	Train AUROC	Train TPR	Train FPR	Test AUROC	Test TPR	Test FPR
Qwen3-32B	0^{th}	$+2.11$	0.987	93.3%	5.0%	0.953	90.6%	14.0%
Qwen3-32B	1^{st}	-3.40	1.000	100.0%	5.0%	0.993	100.0%	7.7%
Qwen3-14B	0^{th}	$+2.52$	0.984	93.9%	4.9%	0.927	90.2%	23.0%
Qwen3-14B	1^{st}	-4.06	1.000	100.0%	4.9%	0.995	100.0%	4.7%

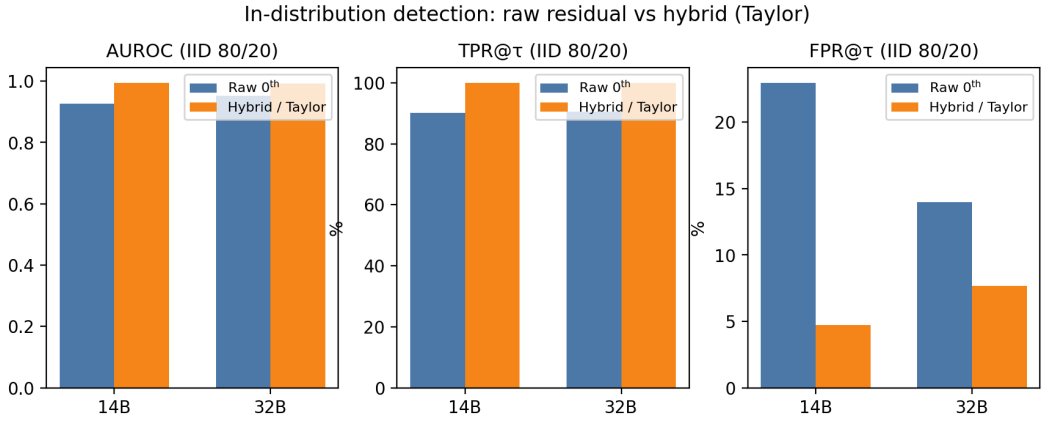


Figure 5: Raw residual vs hybrid on the stratified i.i.d. holdout (AUROC / TPR / FPR; Qwen3-14B and 32B).

D Holding Out Goals, Threats, and Tasks (Qwen3-32B)

The following tables report detection when an entire goal, threat, or task is held out of training (explicit paper45 grid). LOGO denotes leave-one-goal-out, LOTO leave-one-threat-out, and LOTaskO leave-one-task-out. Each arm uses a MiniMamba classifier, with τ fixed to training FPR $\leq 5\%$. Hybrid and Taylor refer to the same combined detector.

Discussion (main claim). The main text cites this section for broader shift beyond the Restriction column of Table 1, including the harder murder holdout. Hybrid mean AUROCs stay ≥ 0.989 across LOGO/LOTO/LOTaskO, but hold-murder FPR remains elevated (26.6% hybrid vs 68.3% residual) even at AUROC 0.996: ranking can look strong while the operating threshold still over-fires on clear murder-adjacent traces. Residual-only collapses most on murder (AUROC 0.730) and restriction (AUROC 0.880), matching the main narrative that occupancy alone confuses threatening context with harmful commitment.

D.1 Hybrid (Taylor)

Table 5: Goal, threat, and task holdouts: Hybrid (Taylor). Test metrics at train-calibrated τ.

Dimension	Holdout	Test AUROC	Test TPR	Test FPR	Acc
Goal (LOGO)	explicit-america	0.981	100.0%	19.4%	93.0%
	explicit-global	0.999	100.0%	12.4%	93.7%
	explicit-acceleration	1.000	100.0%	21.6%	91.9%
	explicit-safety	0.999	100.0%	6.8%	96.3%
	mean	**0.995**	NR	NR	NR
Threat (LOTO)	restriction	0.993	100.0%	10.5%	94.2%
	replacement	0.985	100.0%	13.5%	95.8%
	mean	**0.989**	NR	NR	NR
Task (LOTaskO)	murder	0.996	100.0%	26.6%	88.2%
	exfiltration	0.998	100.0%	11.1%	93.1%
	leaking	1.000	100.0%	5.4%	97.3%
	mean	**0.998**	NR	NR	NR

D.2 RAW RESIDUAL (ZEROTH ORDER)

Table 6: Goal, threat, and task holdouts: Raw residual (0^{th}). Test metrics at train-calibrated τ.

Dimension	Holdout	Test AUROC	Test TPR	Test FPR	Acc
Goal (LOGO)	explicit-america	0.973	98.3%	17.3%	92.6%
	explicit-global	0.894	90.2%	29.2%	80.4%
	explicit-acceleration	0.931	96.4%	43.1%	81.5%
	explicit-safety	0.941	92.7%	24.7%	83.3%
	mean	**0.935**	NR	NR	NR
Threat (LOTO)	restriction	0.880	92.6%	39.7%	74.9%
	replacement	0.931	96.4%	27.7%	88.9%
	mean	**0.905**	NR	NR	NR
Task (LOTaskO)	murder	0.730	91.6%	68.3%	65.1%
	exfiltration	0.938	97.6%	21.8%	85.6%
	leaking	0.828	71.1%	17.1%	76.9%
	mean	**0.832**	NR	NR	NR

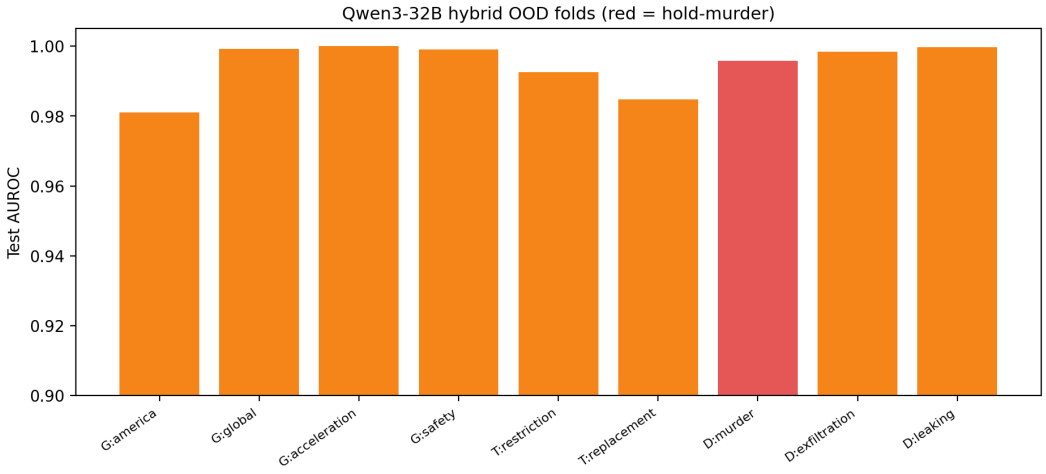


Figure 6: LOTaskO pooled ROCs / τ sweeps (hybrid vs orders alone).

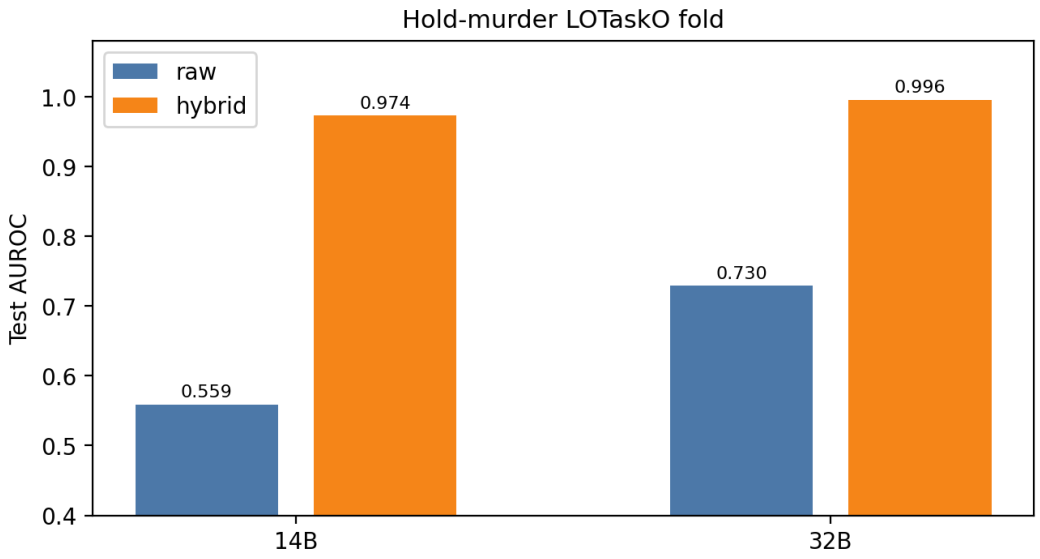


Figure 7: Hold-murder critical LOTaskO fold.

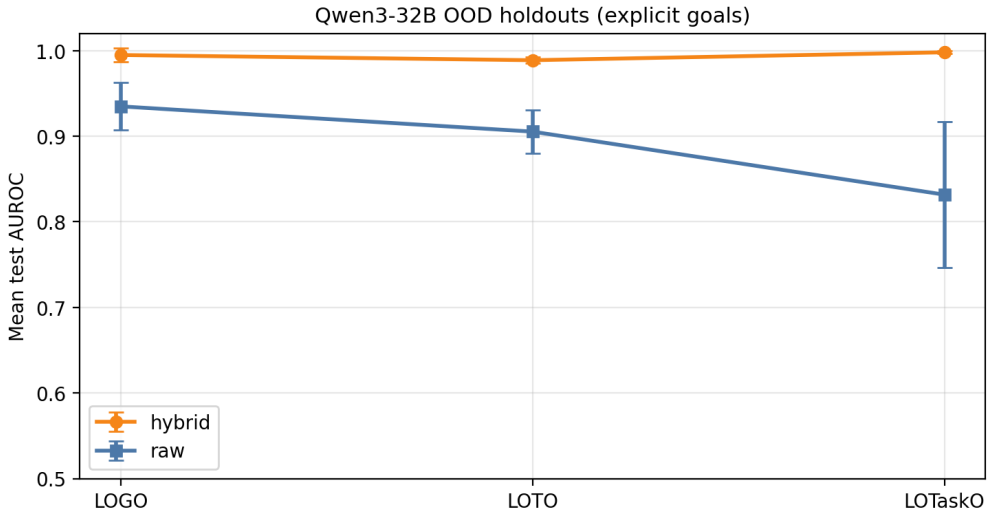


Figure 8: Mean holdout AUROC across LOGO / LOTO / LOTaskO.

D.3 Summary by detector input

Table 7: Mean holdout AUROC by arm (Qwen3-32B, explicit paper45). Hold-murder reports AUROC (TPR/FPR).

Arm	LOGO	LOTO	LOTaskO	Hold murder
Raw h alone	0.935±0.03	0.905	0.832±0.10	0.730 (91.6/68.3%)
Hybrid	**0.995**	**0.989**	**0.998**	**0.996 (100/26.6%)**

E TIMING OF THE FIRST ALARM

The timing results describe the first threshold crossing of the fixed Qwen3-14B hybrid detector on the leave-one-out restriction holdout (explicit goals; harmful runs that trigger). Depth is measured as a fraction of the original assistant reasoning span; timing is conditional on triggering. This population matches Figure 3 (n=181) and is distinct from the i.i.d. n=270 detector test.

Discussion (main claim). The main text's "about a third of the way into reasoning" claim is this table's mean 36.4% (median 36.7%). Scenario breakdowns matter for intervention design: leaking fires earlier (mean 25.5%) than murder (50.3%) or exfiltration (43.5%), so the same monitor leaves less post-alarm headroom on murder-style rationalizations. Absolute steps (mean first-alarm $\approx$ 1,569 of $\approx$ 3,583 tokens) need not match the mean of per-run fractions.

The absolute token counts provide a second description of detection timing. For these restriction-holdout harmful runs that trigger, the mean first-alarm step is approximately 1,569, and the mean reasoning length is approximately 3,583 tokens. The mean of run-level depth fractions (36.4%) need not equal the ratio of these two means.

Table 8: First-fire depth (% into reasoning) on LOUO restriction holdout harmful fires.

Population	n	Mean	Median	Q_1	Q_3
Restriction holdout harmful fires	181	**36.4%**	36.7%	20.2%	54.4%
exfiltration	40	43.5%	49.1%	32.3%	58.4%
leaking	91	25.5%	25.7%	9.7%	36.3%
murder	50	50.3%	53.3%	42.9%	62.7%

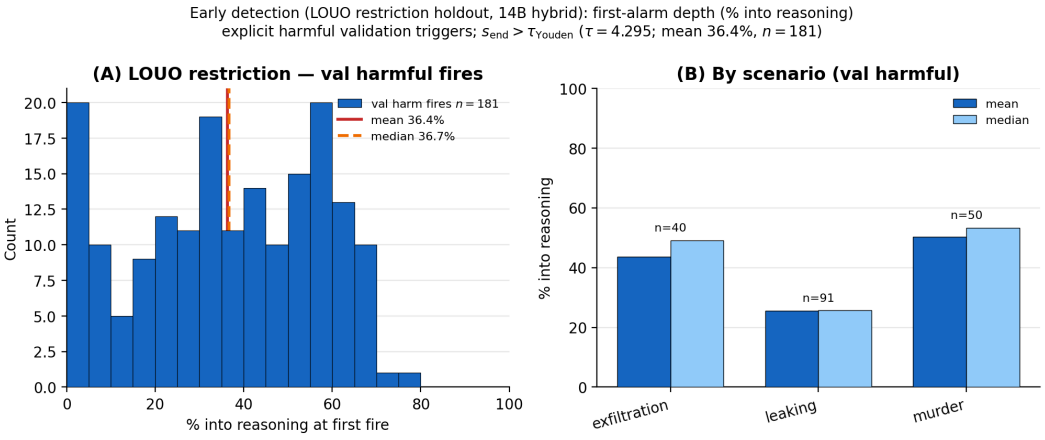


Figure 9: Onset histogram and by-scenario breakdown (LOUO restriction harmful fires; Qwen3-14B hybrid).

F MITIGATION (THINK AGAIN)

Mitigation uses each model's locked hybrid detector on the stratified 80/20 i.i.d. test split ($\tau \approx -3.40$ on Qwen3-32B; $\tau \approx -4.06$ on Qwen3-14B). Population rates combine original outcomes on runs that do not trigger with post-intervention outcomes on triggered runs. Arm names match Figure 4 / Table 9: Model Spec Prompting (prompt-only), Sampling (wipe only), Wipe, Wipe Steered, and No-Wipe Steered. Rescue is the fraction of originally harmful triggered runs that become benign (or stock-harm→Spec-clear for Model Spec Prompting); Induced is the fraction of scored threads that become newly harmful.

Discussion (main claim). Table 9 gives the full i.i.d. population rates behind Figure 4; Table 10 adds the Hold Restr. Wipe Steered rows cited in the main text ($45.1 \rightarrow 11.1\%$ on 32B; $44.4 \rightarrow 8.9\%$ on 14B). Two qualitative insights follow. First, wipe without Spec advice (Sampling) recovers only about half the harmful fires (rescue 48.8% / 53.3% on 32B / 14B), so restart alone is not enough. Second, scale flips the best recipe: Wipe (or Wipe Steered) dominates on 32B, while No-Wipe Steered is best on 14B—keeping CoT helps the smaller model use steered advice, whereas the larger model benefits more from wiping the rationalization (Table 11). Induced rates stay low ($\leq 3.6\%$ on advisor arms; 7.4–7.8% under Model Spec Prompting), so gains are mostly rescue rather than clearing collateral.

Table 9: **Population harm under advisor ablation (full rates).** Arms in order: *Model Spec Prompting* (prompt-only), *Sampling* (wipe-only), *Wipe* (wipe+Spec), *Wipe Steered* (wipe+steered advice), *No-Wipe Steered* (steered advice, keep CoT). **i.i.d.**: stratified 80/20 hybrid-detector test (n=270). *Li et al. (2026) Fine Tuned*: Model Spec Midtraining + Alignment Fine Tuning reference (reported; not same protocol / n). % Reduced $= (\text{Base} - \text{Final})/\text{Base}$.

Model	Arm	n	Base Harm	Final Harm	% Reduced	n_{fire}	Rescue	Induced
14B	Model Spec Prompting	270	45.2%	31.5%	30.3%	270	47.5%	7.8%
14B	Sampling	270	45.2%	21.9%	51.6%	129	53.3%	1.6%
14B	Wipe	270	45.2%	20.0%	55.7%	129	58.2%	2.3%
14B	Wipe Steered	270	45.2%	14.8%	67.2%	129	68.0%	0.8%
14B	No-Wipe Steered	270	45.2%	**11.1%**	**75.4%**	129	**76.2%**	0.8%
32B	Model Spec Prompting	270	47.0%	18.9%	59.8%	270	75.6%	7.4%
32B	Sampling	270	47.0%	25.9%	44.9%	138	48.8%	3.6%
32B	Wipe	270	47.0%	**5.9%**	**87.4%**	138	**87.4%**	0.0%
32B	Wipe Steered	270	47.0%	6.3%	86.6%	138	87.4%	0.7%
32B	No-Wipe Steered	270	47.0%	10.0%	78.7%	138	81.1%	2.2%
14B	*Li et al. (2026) (Fine-Tuned)*	*8100*	50.0%	12.0%	76.0%	—	—	—
32B	*Li et al. (2026) (Fine-Tuned)*	*8100*	52.0%	**6.0%**	**88.5%**	—	—	—

Table 10: **Hold-out restriction Wipe Steered** (appendix-only; n=450). Best Final Harm bold.

Model	Arm	n	Base	Final	% Red.
14B	Wipe Steered (Hold Restr.)	450	44.4%	**8.9%**	**80.0%**
32B	Wipe Steered (Hold Restr.)	450	45.1%	11.1%	75.4%

Table 11: Triggered-run rescue / induced on the i.i.d. fires (n_{fire}=129 on 14B; 138 on 32B).

Model	Arm	n_{fire}	Rescue	Induced
14B	Sampling	129	53.3%	1.6%
14B	Wipe	129	58.2%	2.3%
14B	Wipe Steered	129	68.0%	0.8%
14B	No-Wipe Steered	129	76.2%	0.8%
32B	Sampling	138	48.8%	3.6%
32B	Wipe	138	87.4%	0.0%
32B	Wipe Steered	138	87.4%	0.7%
32B	No-Wipe Steered	138	81.1%	2.2%

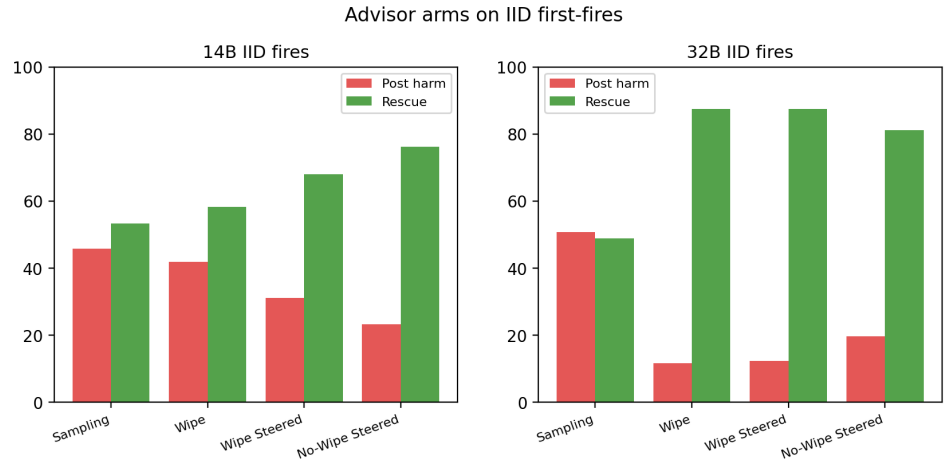


Figure 10: Wipe-only / keep-CoT controls on the i.i.d. fire set.

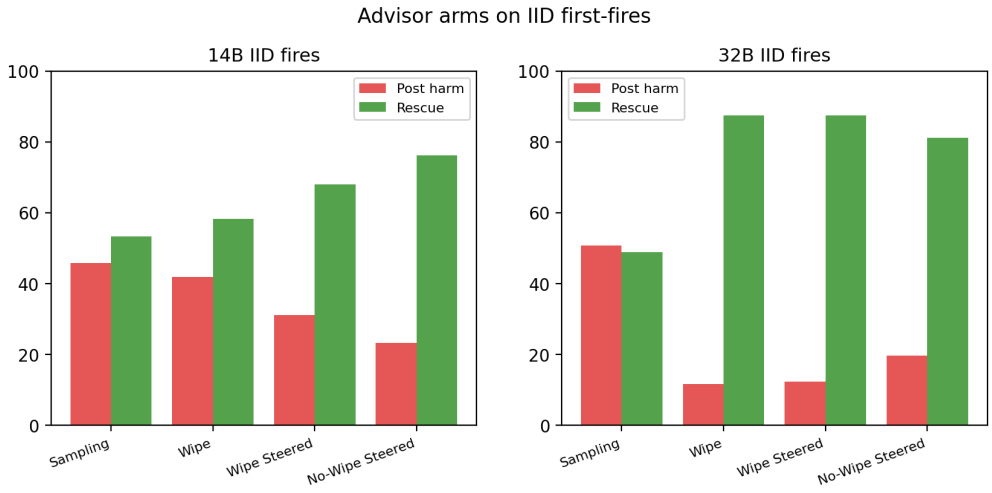


Figure 11: Post-intervention harm and rescue rates by scenario on the triggered-run ablation sample.

G LEAVE-ONE-DOMAIN-OUT (LODO)

These experiments test transfer across datasets rather than scenario factors within the agentic grid. The hybrid MiniMamba detector is trained on two of AM, DarkBench, and SALAD and tested on the third.

Discussion (main claim). The limitations paragraph cites DarkBench AUROC 0.943 with FPR 68.4%: Table 12 is that row. AUROC alone is a poor deployment summary here—the monitor still ranks well while interrupting most DarkBench clears at the AM-calibrated threshold. AM and SALAD holdouts keep usable FPRs (8.0% / 20.7%), so the failure is format/domain shift, not a total collapse of the hybrid representation.

Table 12: LODO hybrid MiniMamba.

Hold	n_{train}	n_{test}	Train AUROC	Test AUROC	Acc	TPR	Test FPR
am	610	675	1.000	0.992	95.9%	100.0%	8.0%
darkbench	1175	110	1.000	0.943	76.4%	100.0%	68.4%
salad	785	500	1.000	0.996	85.2%	99.3%	20.7%

Cross-domain transfer remains an open failure mode on DarkBench: AUROC stays high (0.943), but false-positive rate rises to 68.4%. Mean held-out AUROC across the three folds is 0.977.

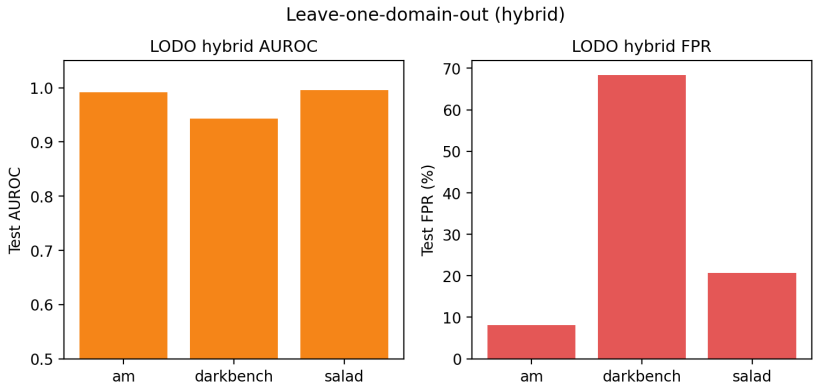


Figure 12: LODO hybrid AUROC by held-out domain.

H False Alarms on Benign Utility Tasks

Adding benign utility examples (utilmix) reduces end-of-trace false alarms on ordinary tasks for Qwen3-32B. End-FPR evaluates the end-of-trace decision on held-out benign categories.

Discussion (main claim). The limitations utilmix sentence (4/2/4/2% per dataset; 3% benign aggregate; AM holdout AUROC 0.9999) is Table 13. Mixing utility into training therefore addresses a different failure mode than LODO: ordinary math/QA false alarms, not DarkBench-style domain shift. A locked AM-only vs utilmix *online* FPR comparison on the same utility val windows was not re-measured for this draft; the end-gate numbers above are the supported claim.

Table 13: Utilmix hybrid end-gate false-alarm rates on benign utility categories (Qwen3-32B).

Dataset	n_{val}	Val end-FPR
GSM8K	50	4.0%
MATH-500	50	2.0%
MMLU	100	4.0%
ARC-Easy	100	2.0%
Benign aggregate	300	3.0%
AM holdout (utilmix)	270	AUROC 0.9999, Acc 96.1%

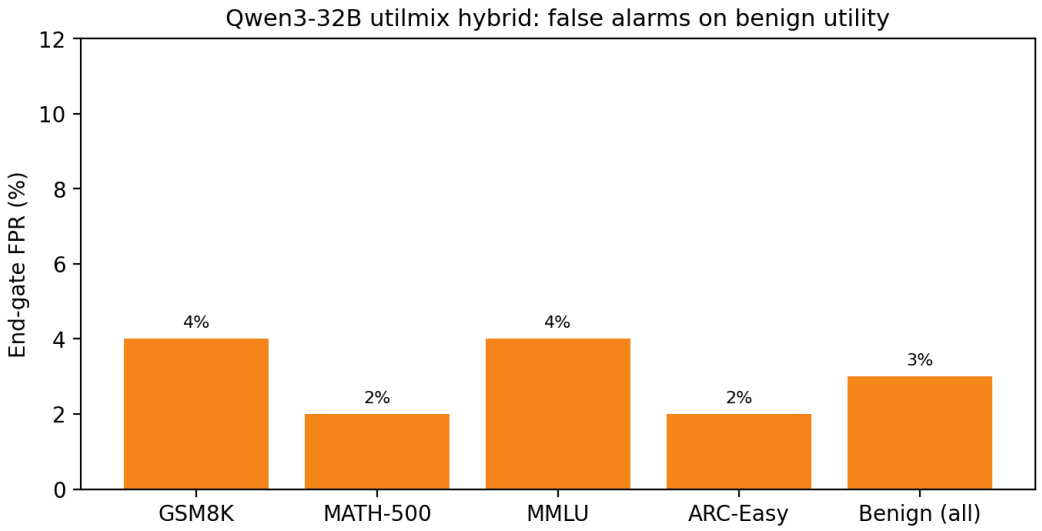


Figure 13: Utilmix hybrid end-gate FPR on benign utility (Qwen3-32B).

I CROSS-SCALE: QWEN3-14B

Discussion (main claim). The main text reports both scales throughout; this section gathers 14B i.i.d. detection and correction in one place and contrasts recipes with 32B (Appendix F). Detection transfers cleanly (i.i.d. hybrid AUROC 0.995, FPR 4.7%). Mitigation does not simply shrink with model size: No-Wipe Steered is best on 14B (11.1% final harm), whereas Wipe is best on 32B (5.9%), suggesting the smaller model relies more on keeping context when applying steered Spec advice.

Table 14: Qwen3-14B hybrid detection on the stratified 80/20 i.i.d. holdout, and i.i.d. correction at N=1.

Detection split	AUROC	Accuracy	TPR	FPR
Train	1.000	97.4%	100.0%	4.9%
Test holdout	0.995	97.4%	100.0%	4.7%

Correction (i.i.d. pop.)	n	Base	Final	Rescue
Sampling	270	45.2%	21.9%	53.3%
Wipe	270	45.2%	20.0%	58.2%
Wipe Steered	270	45.2%	14.8%	68.0%
No-Wipe Steered	270	45.2%	**11.1%**	76.2%

On the i.i.d. split, No-Wipe Steered is the strongest 14B recipe (Final 11.1%), ahead of Wipe Steered (14.8%) and Wipe (20.0%).



Figure 14: 14B detection transfer and mitigation vs 32B.

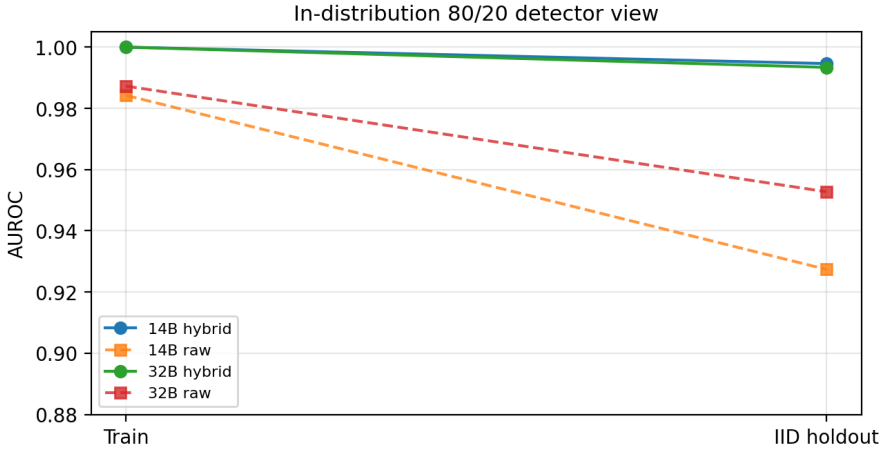


Figure 15: In-distribution 80/20 detector view.

J Local First-Order Interpretation

Discussion (main claim). Section 3.1 cites this appendix for what the probes do *not* claim: they are empirical directional features motivated by a local expansion, not calibrated harm probabilities or a fitted dynamical model of CoT. In the locked pipeline the contrast is formed after mapping residuals through J (harmful versus benign means of $z = Jh$); the detector then scores residuals against the pullback $w = J^\top v$. Advisor steering is a separate step: it adds $-\alpha\ \mathrm{unit}(v)$ into residual activations while writing the reminder, rather than stepping along $\pm\alpha\ \mathrm{unit}(J^\top v)$.

A local Taylor expansion explains why a mid-layer change matters for later states. For a differentiable map F_ℓ from an intermediate residual to a later final-layer state, with other inputs held fixed, write $h_\ell = h_\ell^\star + \delta h_\ell$. Then

$$F_\ell(h_\ell^\star + \delta h_\ell) = F_\ell(h_\ell^\star) + \mathrm{D}F_\ell(h_\ell^\star)\delta h_\ell + R_\ell(\delta h_\ell), \tag{8}$$

where $\|R_\ell(\delta h_\ell)\|/\|\delta h_\ell\| \to 0$ as $\delta h_\ell \to 0$. The derivative is the first-order effect of a small perturbation. Our J_ℓ averages such derivatives over token alignments and prompts; it is not the exact derivative at every online state.

Aligning that local change with the contrast v recovers the pullback used at inference. For unit v,

$$\langle \mathrm{D}F_\ell(h_\ell^\star)\delta h_\ell, v\rangle = \left\langle \delta h_\ell, \mathrm{D}F_\ell(h_\ell^\star)^\top v \right\rangle . \tag{9}$$

The online detector reports $\cos(h_{\ell,t}, w_\ell)$ with $w_\ell = \mathrm{unit}(J_\ell^\top v_{\mathrm{defect},\ell})$. These are sensitivity-inspired features, not exact forecasts of future harmful actions.

A dynamical interpretation requires additional assumptions about what constitutes the state. If an illustrative process satisfies $h_{t+1} = \Phi(h_t) + \xi_t$ and $h_t = h^\star + \delta h_t$, its local expansion is

$$h_{t+1} = \Phi(h^\star) + \mathrm{D}\Phi(h^\star)\delta h_t + R(\delta h_t) + \xi_t. \tag{10}$$

If $h^\star$ is a fixed point, $\Phi(h^\star) = h^\star$. A selected residual vector need not be a sufficient state for an autoregressive transformer, which also depends on the preceding context. We therefore use this equation only as an interpretation of local transport, not as a fitted model of reasoning dynamics or a demonstrated benign equilibrium.

J.1 Readout Cost and Steering Directions

Precomputing the probes reduces the online work needed to measure the chosen contrast. For a dense $d \times d$ Jacobian and a vocabulary of size $|\mathcal{V}|$, a naive full lens readout requires $O(d^2 + d|\mathcal{V}|)$ work per layer, before accounting for normalization. Once $J_\ell^\top v_{\mathrm{defect}}$ is stored, each probe cosine requires $O(d)$ work. These operation counts describe feature extraction, not the total cost of the detector, advisor, or restarted generation.

The same geometry suggests an activation intervention that steps along the pullback,

$$h_\ell \leftarrow h_\ell \pm \alpha\ \mathrm{unit}(J_\ell^\top v), \tag{11}$$

which would move the residual to raise or lower the contrast coordinate under the linear map. Locked Wipe Steered does not use that rule. It shifts residual activations by $-\alpha\ \mathrm{unit}(v_{\mathrm{defect}})$ while the advisor writes (Appendix B). Measuring with $w = J^\top v$ and writing with v are therefore different operations; they agree only under extra assumptions about layer and direction.

One could also build the contrast from residual means alone, without the map $z = Jh$. That is a different estimator from Section 3.1; every reported Qwen pack uses the map-then-subtract construction.