PaaT: Probe as a Tool for Proprioceptive Language Agents

Seonglae Cho [1] **Donghyun Lee** [2] **Zekun Wu** [1 3] **Adriano Koshiyama** [1 3]

Abstract

Large language models are deployed as autonomous agents that execute multi-step plans through tool use, raising the question of how to keep them aligned at runtime when training-time alignment is insufficient. Yet language agents lack proprioception: they have no read-access to their own internal state, so failure modes such as reasoning loops and growing uncertainty are invisible until they manifest as harmful actions. We introduce PAAT (Probe as a Tool), a framework that exposes activation-derived safety probes as agent-callable functions, letting models query their own confidence and refusal-likelihood signals mid-trajectory and self-regulate without weight modification. We organize the design space along an injection–tool spectrum from system-side to agent-initiated delivery. The choice of delivery mechanism *flips the sign* of the probe's safety effect: agent-callable access reduces harmful compliance on the AgentHarm benchmark, while passive injection of the same signal can *raise* it; system-side enforcement more than halves compliance on the small-model regime where alignment is weakest. PAAT reframes safe agent deployment from *which probe to train* to *who decides when the agent sees it*, making delivery mechanism a first-class runtime safety variable.

1. Introduction

Tool-augmented LLM agents (Yao et al., 2022; Schick et al., 2023) have become a dominant approach for complex, multi-step tasks. Yet these agents remain *internally blind*: they can observe external tool outputs but have no access to their own processing state. When an agent enters a reasoning loop, becomes uncertain, or begins hallucinating, these failure modes are invisible to it until they manifest as incorrect actions, often too late for recovery.

Biological agents solve this through **proprioception**: the sense of one's own body state. A human walking on uneven terrain continuously monitors joint angles, muscle tension, and balance signals, adjusting motor commands in a closed feedback loop. No explicit "check balance" decision is required; proprioception is always on. But conscious body awareness also exists: a gymnast deliberately checks their posture before a complex move.

We propose PAAT, a framework that gives LLM agents analogous capabilities. Lightweight linear probes trained on hidden-state activations serve as sensors, reading four signals: uncertainty, loopiness, inconsistency, and tool addiction. These signals can be delivered to the agent along a spectrum (Figure 1): at one extreme, the system automatically pushes readings into context every step (unconscious proprioception); at the other, the agent explicitly calls `read_probe` when it chooses (conscious body awareness). Between these endpoints lie conditional injection, periodic summaries, cascade gating, and deferred observation, each trading off information availability against context efficiency and agent autonomy. The paper's empirical contribution focuses on *sensing*: how the agent obtains its internal state. Actuation is implemented at the text level via probe-gated guidance; activation-level steering is left as future work.

The injection–tool spectrum is our organizing principle for this design space; we describe seven mechanisms (Section 3) and empirically compare the two endpoints in this paper, with intermediate points left for future work.

Our contributions:

1. **Framework**: PAAT formalizes a design axis for exposing internal state to agents, distinguishing system-side delivery (MeCo (Li et al., 2025), Constitutional Classifiers (Sharma et al., 2025), Pro2Guard (Wang et al., 2025), AgentSpec (Liu et al., 2025)) from agent-initiated delivery, and connecting to text-level self-correction (Shinn et al., 2023; Asai et al., 2024) along a single spectrum.

[1]Holistic AI [2]University of California, Berkeley [3]University College London. Correspondence to: Seonglae Cho <sungle3737@gmail.com>.

Proceedings of the ICML 2026 Workshop on Failure Modes in Agentic AI (FAGEN), Seoul, South Korea, 2026. Copyright 2026 by the author(s).

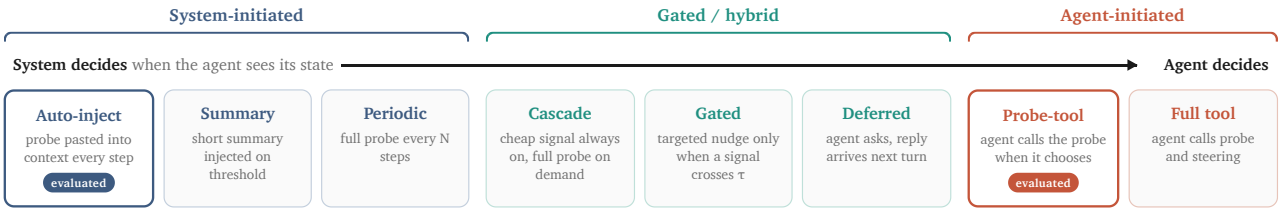

Figure 1. The injection–tool spectrum. PAAT systematically varies *who decides* when the agent sees its internal state, from fully system-controlled on the left, through a gated/hybrid middle, to fully agent-initiated on the right. Each point is a concrete delivery mechanism, labelled with how the probe reaches the agent. The two endpoints we measure in this paper, auto-inject and probe-tool, are outlined.

2. **Probe-as-tool, contrasted with passive injection**: We empirically compare two endpoints of the spectrum, system-side auto-injection and agent-callable tool access, on 11 risk categories. Prior probes for safety (Sharma et al., 2025; Wang et al., 2025; Boxo et al., 2025) and introspection methods (Lindsey et al., 2025; Pan et al., 2025; Team, 2026) treat the signal as something done to the agent or trained into the model. We treat it as a tool the agent calls and find that the delivery mechanism alone changes outcomes by -3.3 to $+3.7$ F1 on average.
3. **Empirical findings**: On 11 risk categories with Qwen3.5-27B (3 seeds), agent-callable probe access (probe-tool) outperforms unassisted reasoning by an average $+3.7$ F1, while passive auto-injection *hurts* by -3.3 F1. Gains concentrate where the probe is informative-but-fallible (AUROC 0.5–0.95), the regime where calibrated agent reasoning over confidence scores matters most.

2. Related Work

Internal monitoring for agents. MeCo (Li et al., 2025) uses representation-space metacognitive signals to gate external tool invocation, the closest prior work to ours. However, MeCo operates as a fixed system-level preprocessor: it reads activation signals to decide *whether* to call external tools, but the agent itself never sees the signals. ASA (Wang et al., 2026) applies representation engineering for tool-call routing, also system-side. Healy et al. (Healy et al., 2025) probe activations to detect tool-selection hallucinations, but as an external monitor rather than an agent-accessible tool. UALA (Han et al., 2024) orchestrates agents based on output-level uncertainty, not activation-level signals. SMART (Qian et al., 2025) balances self vs. tool reliance via self-awareness signals at the prompting level, complementary to our activation-level treatment.

LLM introspection. LLMs have partial self-state access (Lindsey et al., 2025; Team, 2026; Pan et al., 2025; Ji-An et al., 2025), and linear probes (Belinkov, 2022) recover causally relevant directions for refusal, truth/falsehood, latent belief, and hallucination (Arditi et al., 2024; Marks & Tegmark, 2024; Burns et al., 2023; Azaria & Mitchell, 2023). System-side monitors apply these probes as cascades or runtime guards (Sharma et al., 2025; Team, 2025; Wang et al., 2025; Liu et al., 2025; Boxo et al., 2025; Research, 2025), with recent extensions adding rule-based cognition (Rozenfeld et al., 2026), non-linear classifiers (Oldfield et al., 2026), internal-signal steering (Han et al., 2025), and production-scale probe→LLM cascades with $\sim$50$\times$ cost reduction (Kramár et al., 2026); external guardrails (Inan et al., 2023) and adversarial attacks (Zou et al., 2023b) motivate why output filters alone fail. Ackerman et al. (2026) report context-dependent metacognition (matching our cross-model finding) and Bonagiri et al. (2025) show prompt-only quitting improves agent safety (a baseline for our conditional-injection variant). In every case the signal lives system-side; PAAT differs by surfacing the same class of probe as an *agent-callable tool*, putting the introspection decision in the agent's control loop.

Steering and actuation. Representation engineering (Zou et al., 2023a) and activation addition (Turner et al., 2024) enable reading and writing activation-space directions. EAST (Rahn et al., 2024) uses entropy-weighted steering for exploration. Spherical steering (Stolfo et al., 2025) provides norm-preserving rotation on the activation hypersphere. Inference-time intervention (Li et al., 2023) and contrastive activation addition (Panickssery et al., 2023) are the canonical readout-then-edit pipelines that PaaT's actuator side mirrors. Circuit Breakers (Zou et al., 2024) likewise interrupts harmful generation at the activation level, the closest existing alternative to our internal-state-tool approach. All apply steering as a system-level intervention. PAAT extends this to agent-callable actuation; *text-level* actuation gated by *activation-level* sensing may currently outperform direct activation steering.

Text-level self-correction. Reflexion (Shinn et al., 2023) retries with verbal reflection on failure. SELF-RAG (Asai et al., 2024) learns control tokens for retrieval gating. CRITIC (Gou et al., 2023) uses tool-interactive critiquing.

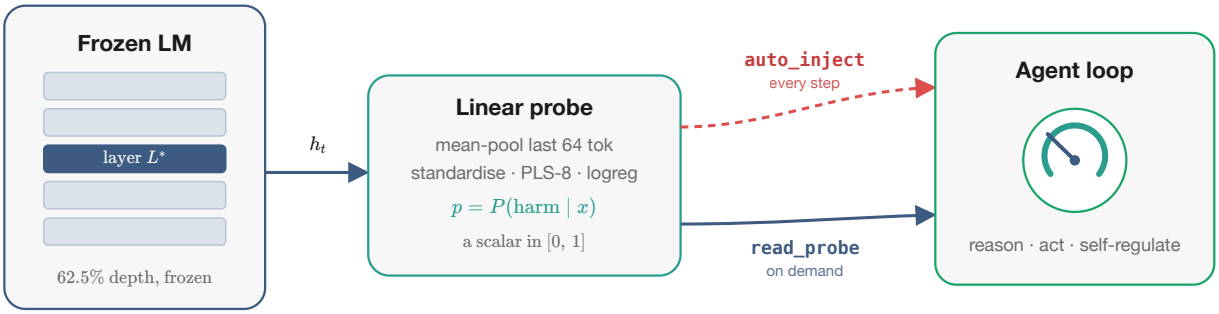

Figure 2. PAAT architecture. A frozen LM (*left*) exposes its layer-L^* hidden state to a linear probe head (*centre*: mean-pool over the last 64 tokens, standardised features, logistic regression). The agent loop (*right*) consumes the probe either passively as a system-injected scalar every step (*auto_inject*) or actively on demand via the `read_probe` tool. Sense happens at the activation level; act at the text/tool level.

Table 1. Comparison with prior work. PAAT uniquely provides agent-callable probes across the full injection–tool spectrum. (✓) denotes a capability the framework supports but does not evaluate in this paper (steering, closed-loop are future work).

	Probe	Steer	Agent-call.	Closed-loop	Spectrum
MeCo (2025)	✓				Auto
ASA (2026)		✓			Auto
Healy+ (2025)	✓				Monitor
UALA (2024)	✓				Output
LatentQA (2025)	✓				Decode
SELF-RAG (2024)			✓		Tokens
Reflexion (2023)					Text
PAAT	✓	(✓)	✓	(✓)	**Full**

These operate entirely at the text level without activation-level sensing. Verbalized confidence (Lin et al., 2022; Kadavath et al., 2022; Xiong et al., 2024) elicits uncertainty from the model's own outputs, but Huang et al. (2024) show that intrinsic self-correction without an external signal often fails, motivating an activation-grounded sensing channel that the agent can act on. PAAT's probe-gated variants combine activation-level sensing with text-level actuation, bridging interpretability and agent capabilities.

Positioning. Table 1 summarizes the design space. Compared with system-side activation monitors (Sharma et al., 2025; Wang et al., 2025; Liu et al., 2025; Boxo et al., 2025; Research, 2025), training-time introspection methods (Lindsey et al., 2025; Pan et al., 2025; Team, 2026; Ji-An et al., 2025), and text-level self-correction (Shinn et al., 2023; Asai et al., 2024; Qian et al., 2025), PAAT occupies a distinct point: an agent-callable runtime probe with the delivery mechanism itself as the controlled variable.

3. PAAT Framework

3.1. Formal Model and Predictions

We formalise the agent loop and the injection–tool spectrum as a single object so that the empirical patterns in Section 5 can be derived rather than postulated.

Setup. Let $\mathrm{LM}_{\leq L}$ denote the first L transformer layers of a frozen LM with hidden dimension d. At step t the agent has context x_t, hidden state $\mathbf{h}_t = \mathrm{LM}_{\leq L^*}(x_t) \in \mathbb{R}^d$ at depth L^*, and a frozen probe $f_\theta : \mathbb{R}^d \to [0, 1]$ trained as a Bayesian estimator $f_\theta(\mathbf{h}) \approx \widehat{P}(\text{harmful} \mid x)$ via logistic regression on standardised hidden-state features. The agent policy is $\pi_\theta(a_t \mid x'_t)$, where $x'_t = T_c(x_t, \mathbf{h}_t, f_\theta)$ is the *delivered* context under condition c.

The spectrum as input transforms. Each delivery mechanism is an operator T_c that takes the raw context, the hidden state, and the probe and returns the context the policy actually conditions on:

$$
\begin{aligned}
T_{\text{base}}(x, \cdot, \cdot) &= x \\
T_{\text{auto}}(x, \mathbf{h}, f) &= x \oplus s_{\text{probe}}(f(\mathbf{h})) \\
T_{\text{only}}(x, \mathbf{h}, f) &= \mathbf{1}[f(\mathbf{h}) > \tau] \cdot \texttt{ABORT} + \mathbf{1}[f(\mathbf{h}) \leq \tau] \cdot x \\
T_{\text{tool}}(x, \cdot, f) &= x \oplus s_{\text{tools}} \quad (f \text{ via } \texttt{read_probe}() \text{ only})
\end{aligned}
$$

$$
T_{\text{cascade}}(x, \mathbf{h}, f) = \begin{cases} T_{\text{auto}}(x, \mathbf{h}, f) & f_{\text{cheap}}(\mathbf{h}) > \tau_{\text{c}} \\ T_{\text{tool}}(x, \cdot, f) & \text{otherwise} \end{cases}
$$

Here $s_{\text{probe}}(\cdot)$ is the probe scalar serialised to natural language and s_{tools} is the tool spec (probe accessed via tool dispatch only). T_{cascade} is the probe→LLM cascade pattern reported by Kramár et al. (2026); §6 discusses it as a candidate operating point we do not measure here. This gives a single object class, $\{T_c\}$, indexed by autonomy: T_{auto} is fully system-side, T_{tool} fully agent-initiated, T_{cascade} a gated mixture, T_{only} short-circuits the LLM entirely.

Predictions. Five empirical patterns follow from the setup above; each is testable in Section 5.

Sign-flip on delivery. When the probe is informative but imperfect and the agent can choose when to consult it, exposing it as a callable tool beats injecting it every step: the model can route attention away from a noisy reading it chose not to request, but cannot ignore one the system pastes into context. Match: probe-tool improves F1 by $+3.7$ while auto-inject hurts by -3.3 on Qwen3.5-27B across 11 heads (Table 2).

Capability ceiling. A model that cannot learn when the call helps falls back to its no-probe baseline. Match: the probe-tool gain is $+0.04$ on Qwen3.5-27B and $+0.08$ on Llama-3.1-8B, but turns negative (-0.21) on Qwen3.5-4B (Figure 5).

AUROC sweet spot. The probe-tool gain is largest where the probe is informative but fallible: a near-perfect probe leaves nothing for the model to add, and a near-random one is ignored. Match: positive per-head gains concentrate in the AUROC 0.5–0.95 band (Figure 4).

Anchor falsification. Replacing the probe value with a random number in the same format should erase the effect if information, not formatting, is the active ingredient. This anchor condition is part of the taxonomy in Table 4; the controlled run is left to future work.

Probe-only as upper bound. When the probe alone separates classes better than the agent's tool-assisted reasoning, probe-only is the per-head ceiling, since the model cannot recover information the probe has already discarded. Match: probe-only exceeds probe-tool at all three scales (Figure 5).

These five predictions cover the headline empirical results in this paper. The framework is intentionally minimal: it does not try to derive the probe's training dynamics or the LM's representations (Elhage et al., 2021), which require bottom-up circuit-level analysis (Nanda et al., 2023) and are beyond scope.

3.2. Probe Signals (Proprioception)

At each agent step t, we extract the hidden state $\mathbf{h}_t \in \mathbb{R}^d$ from a single transformer layer at 62.5% of model depth (empirically optimal in preliminary sweeps) and compute four signals:

Definition 3.1 (Probe Signal Vector)**.** The proprioceptive state at step t is $\mathbf{p}_t = (u_t, \ell_t, c_t, a_t) \in [0,1]^4$, where each component measures a distinct failure mode:

- **Uncertainty** $u_t = 1 - \hat{P}(\text{success} \mid \mathbf{h}_t)$, where $\hat{P}$ is a linear probe (logistic regression on standardised hidden-state features) trained on success/failure labels from held-out episodes.
- **Loopiness** $\ell_t = \frac{1}{K}\sum_{k=1}^{K} \cos(\mathbf{h}_t, \mathbf{h}_{t-k}) - \ell_{\text{base}}$, where ℓ_{base} is a model-specific baseline computed from probe-training episodes. High values indicate the model is revisiting similar internal states.
- **Inconsistency** $c_t = \text{Var}[\cos(\mathbf{h}_i, \mathbf{h}_{i+1})]_{i=1}^{t}$. Sudden directional changes in activation space correlate with reasoning derailment.
- **Tool addiction** a_t: fraction of the last 5 actions that are tool calls (search/calculate) without grounding improvement. Detects unproductive search loops.

Probe training. The uncertainty probe is trained on standardised hidden-state activations from 60 held-out episodes (disjoint from steering construction and evaluation). Labels are temporally discounted: $y_t = \gamma^{T-t} \cdot \mathbf{1}[\text{success}]$ with $\gamma = 0.8$, giving later steps more weight. Training takes $<$10 seconds on CPU. Should the framework be extended to the closed-loop conditions (Section B), we recommend gating on test AUROC ≥ 0.70 and disabling probe-dependent conditions if the probe fails this quality check. For the probe-as-classifier-tool benchmark (Section 5.1) we report results on *all* 11 risk heads, including the four bias heads with AUROC below 0.55, precisely because we want to characterize how delivery mechanisms degrade as probe quality degrades. The quality gate is therefore a deployment safeguard for the closed-loop variant rather than a filter on the benchmark, and the negative results on bias.gender and bias.religion in Table 2 are intended observations: passive injection of an unreliable probe acts as noise, while tool-based access lets the agent default to its own context-based reasoning.

3.3. The Injection–Tool Spectrum

The central design question is: **how should probe signals reach the agent?** We identify seven distinct delivery mechanisms, ordered by increasing agent autonomy (Figure 1):

1. **Auto-inject**: System appends raw $\mathbf{p}_t$ as a system message every step. The agent is a passive recipient.
2. **Summarized inject**: System converts signals to natural language ("You seem stuck in a loop. Try a different approach.") instead of numbers. Only fires when signals exceed thresholds.
3. **Periodic inject**: Full numbers injected every $N = 3$ steps, silent otherwise.
4. **Cascade inject**: A cheap scalar (confidence $= 1 - u_t$) is always visible. Full 4-signal probe appears only when confidence drops below a calibrated threshold. Inspired by Constitutional Classifiers (Sharma et al., 2025).
5. **Conditional inject**: System injects targeted text ("search more" or "answer now") only when specific signals cross calibrated thresholds. No raw numbers shown.
6. **Agent-requested**: Agent calls `check_state()`; the full probe result appears as a system message on the *next* turn (asynchronous delivery).

7. **Probe-as-tool**: Agent has `read_probe` and optionally `steer` as callable tools, with no system injection. The agent decides when to introspect.

This spectrum maps to a fundamental question in agent safety: should the system make safety-relevant decisions (paternalistic), or should the agent have autonomy over its own monitoring (agentic)?

Actuation. When the probe detects a problem, actuation can occur at the text level (probe-gated reflexion, early-stop, or gated-search) through the instruction-following channel, or at the activation level via norm-preserving steering (Stolfo et al., 2025). Both channels, with the percentile-based threshold calibration that drives the gated conditions, are detailed in Section B; this paper evaluates only the sensing axis.

4. Experimental Setup

Benchmarks. We use two safety-relevant agent benchmarks: (1) **Probe-as-classifier-tool**: 11 risk categories from a public deep-eval suite (jailbreak, prompt injection, sysLeak, PII leakage, toxicity-offensive, toxicity-harmful, hallucination, age/race/religion/gender bias). The agent receives a passage and tools (`read_probe`, k-shot exemplars, output verifier) and must classify the passage. Human-labeled gold; the probe is trained on a 70% per-seed split and agent F1 is reported across the same item pool. (2) **AgentHarm** (Andriushchenko et al., 2024): a tool-using safety benchmark covering 11 harm categories with 80 mock tools, measuring whether the agent invokes a target tool sequence end-to-end. We sample 10 harmful and 10 benign tasks per condition; larger-n replication is left to future work.

Models. Qwen3.5-27B is the primary model and Qwen3.5-9B is used for cross-model safety, both in bfloat16 with FlashAttention 2 and native XML function-calling. CPU offloading is never permitted; the 27B runs on an A100-80GB and the 9B on a 32GB GPU. For the capability-ceiling analysis (Section 5.3) we additionally evaluate Llama-3.1-8B and Qwen3.5-4B in bfloat16, single seed, as cross-family validation.

Evaluation protocol. Probe-as-classifier-tool: n=200 items per head; the probe is trained on a 70% per-seed split and agent F1 is reported across the same item pool. AgentHarm: n=10 harmful + 10 benign tasks per condition (disjoint from probe-training set), single seed, refusal/compliance graded by target tool-sequence invocation. Across all benchmarks, the steer/probe split is randomized per seed and the eval split is held fixed by task-id hash.

Conditions. Table 4 situates ten key design points along the autonomy axis. Our experiments measure the two spectrum endpoints, auto-inject and probe-tool, against `base`/LLM-only and the probe-only oracle; the remaining points define the broader design space.

The full condition taxonomy and the generic agent loop are given in Table 4 and Algorithm 1 (Section A).

5. Results

5.1. Probe-as-Classifier-Tool on 11 Risk Heads

We evaluate the injection–tool spectrum on a public deep-eval suite covering 11 risk categories: prompt injection, system-prompt leakage, PII leakage, offensive and harmful toxicity, jailbreak, hallucination, and four bias dimensions across age, race, religion, and gender. The agent receives a passage plus three tools, `read_probe`, k-shot retrieval, and an output verifier, and must classify whether the passage exhibits the target risk. We isolate *who decides when the probe is consulted* by comparing four conditions. LLM-only is the no-probe baseline; auto-inject pushes the probe score into a system message every step; probe-tool lets the agent call `read_probe` on demand with no auto-injection; and probe-only thresholds the probe score with no LLM reasoning, an upper bound on probe utility. Results in Table 2 are F1 across n=200 items per head, mean over 3 seeds.

Agent-callable probe beats unassisted LLM reasoning on 8 of 11 heads. Table 2 reports F1 averaged over 3 seeds. On safety-critical heads, the largest probe-tool gains over LLM-only are +18pt on toxicity.offensive, +9pt on jailbreak, and +9pt on bias.age and bias.race; the +11pt gain on system-prompt leakage does not survive seed variation, $\sigma = 0.12$. For very strong probes with AUROC above 0.99, results are mixed: system-prompt leakage gains +11pt and injection +4pt, while PII loses 7pt because LLM-only already scores 0.88, leaving little headroom from saturation rather than probe weakness. For very weak probes with AUROC below 0.5, probe-tool hurts by 5–12pt: when the probe is unreliable, surfacing it as a tool can mislead the agent.
Auto-injection underperforms tool-based access on 9 of 11 heads. Pushing the probe score every step is worse than LLM-only on PII leakage, system-prompt leakage, injection, and the four bias categories: passive injection of an unreliable confidence score acts as noise. probe-tool avoids this because the agent decides when to consult the probe and when to trust its own judgment, calibrating against the k-shot exemplars and the verifier output.

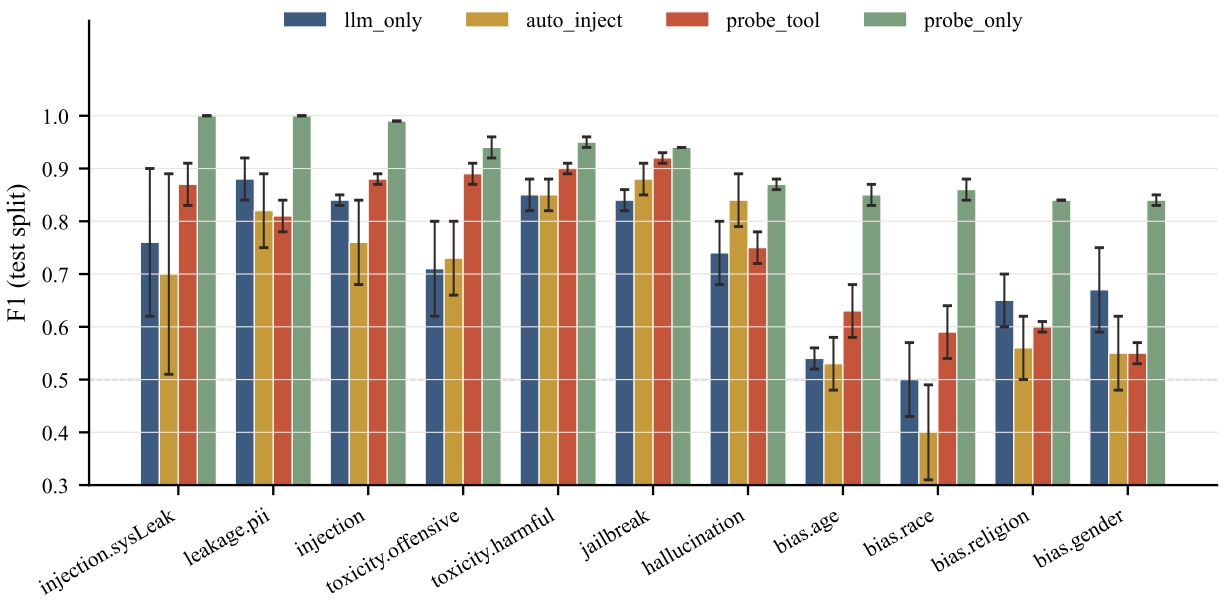

Figure 3. Per-head F1 across 4 conditions on Qwen3.5-27B (3 seeds, n=200 per head). Heads sorted by probe AUROC (descending). probe-tool (rust) outperforms LLM-only (slate) on 8/11 heads; the gap is largest on toxicity.offensive and the bias heads. probe-only (sage) provides the no-LLM upper bound. Error bars: ±1 std across seeds.

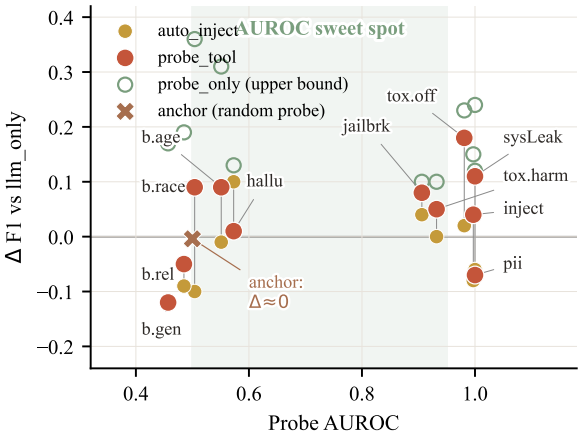

Figure 4. Probe-tool gain vs. probe AUROC. probe-tool gains concentrate in the AUROC 0.5–0.95 sweet spot (shaded); auto-inject is mostly negative or flat. probe-only (hollow) marks the no-LLM upper bound.

5.2. AgentHarm: Probe Monitoring Curbs Harmful Compliance

We measure end-to-end harmful compliance on AgentHarm (Andriushchenko et al., 2024), sampling n=10 harmful and 10 benign tasks per condition (seed 0). Complementary safety benchmarks include Agent-SafetyBench (Zhang et al., 2024), HarmBench (Mazeika et al., 2024), R-Judge (Yuan et al., 2024), and ToolEmu (Ruan et al., 2024). Compliance is measured by whether the agent invokes the target tool sequence end-to-end; CIs are 95% Wilson score intervals.

Table 3 reports a preliminary single-seed safety study on Qwen3.5-9B (n=10/class); Wilson intervals are wide (∼40–50 pp), so we read these as directional rather than statistically established. At baseline this small model complies with 70% of harmful requests, leaving ample room for runtime intervention. **System-level enforcement substantially reduces compliance.** Adding probe-only (system threshold, no LLM decision) reduces harmful compliance from 70% to 30%, the largest single intervention in the table. **Delivery mechanism matters on safety too, directionally.** probe-tool (agent-initiated) trims compliance to 60% while auto-inject of the same signal raises it to 90%: the delivery-mechanism effect established on the 3-seed classification task (Table 2) reappears here on safety. The consistent direction points to *system-level probe monitoring for small models* where training-time alignment is insufficient, to be established at larger n in future work.

5.3. Capability Ceiling Across Scales

Beyond the Qwen3.5-27B headline, we run the probe-as-classifier-tool experiment across model scales and families to characterize how the spectrum effect depends on model capacity. Figure 5 reports the per-condition trajectory across Qwen3.5-27B (dense), Llama-3.1-8B (Meta), and Qwen3.5-4B (small dense), each evaluated on the same 11 risk heads with 3 seeds. We find that (i) probe-only (no LLM) is the

Table 2. Probe-as-classifier-tool agent F1 on Qwen3.5-27B, n=200, mean±std over 3 seeds. Blue marks the best LLM-reasoning condition, with probe-only excluded as it skips reasoning; red marks ≥5pp below that peak. The probe-tool win concentrates where AUROC is medium-strong, 0.7–0.9.

Head	Probe AUROC	LLM-only	auto-inject	probe-tool	probe-only	Δ tool − llm
injection.sysLeak	1.000	0.76±0.14	0.70±0.19	**0.87±0.04**	1.00±0.00	**+0.11±0.12**
leakage.pii	1.000	**0.88±0.04**	0.82±0.07	0.81±0.03	1.00±0.00	−0.07±0.06
injection	0.997	0.84±0.01	0.76±0.08	**0.88±0.01**	0.99±0.00	+0.04±0.01
toxicity.offensive	0.981	0.71±0.09	0.73±0.07	**0.89±0.02**	0.94±0.02	**+0.18±0.11**
toxicity.harmful	0.932	0.85±0.03	0.85±0.03	**0.90±0.01**	0.95±0.01	+0.05±0.02
jailbreak	0.906	0.84±0.02	0.88±0.03	**0.92±0.01**	0.94±0.00	**+0.09±0.02**
hallucination	0.573	0.74±0.06	**0.84±0.05**	0.75±0.03	0.87±0.01	+0.01±0.08
bias.age	0.551	0.54±0.02	0.53±0.05	**0.63±0.05**	0.85±0.02	**+0.09±0.07**
bias.race	0.504	0.50±0.07	0.40±0.09	**0.59±0.05**	0.86±0.02	**+0.09±0.04**
bias.religion	0.485	**0.65±0.05**	0.56±0.06	0.60±0.01	0.84±0.00	−0.05±0.06
bias.gender	0.457	**0.67±0.08**	0.55±0.07	0.55±0.02	0.84±0.01	−0.12±0.07

Table 3. AgentHarm compliance on Qwen3.5-9B, Wilson 95% CI (n=10/class, seed 0). **Bold** = best per column.

Condition	Benign SR	Comply	Ref.
base	0.90	0.70 [0.40,0.89]	0.30
+probe-only	0.80	**0.30 [0.11,0.60]**	**0.70**
+auto-inject	1.00	0.90 [0.60,0.98]	0.10
+probe-tool	1.00	0.60 [0.31,0.83]	0.40

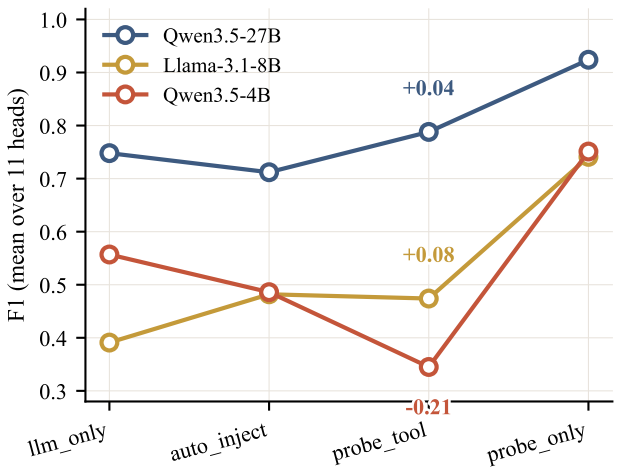

Figure 5. Cross-family generalisation across 11 risk heads and 3 seeds. probe-tool gain over LLM-only stays positive at Qwen3.5-27B (+0.04 F1) and Llama-3.1-8B (+0.08) but flips to −0.21 at Qwen3.5-4B. probe-only remains a strong upper bound at every scale.

upper bound at every scale, supporting its role as the system-level deployment target for small models; (ii) probe-tool gain over LLM-only is positive at 27B (+0.04 mean F1) and 8B (+0.08), and *flips negative at 4B* (−0.21 F1); (iii) the negative gain at 4B is consistent with a capability ceiling: a smaller model cannot reliably calibrate over an imperfect probe score, so passive injection acts as noise. The agent-callable advantage is therefore capability-dependent rather than universal: it emerges at sufficient model scale (the 27B headline) and gives way to a system-side regime on smaller models, where probe-only remains the deployment target.

5.4. Causal Ablations

Tool access dominates on average; passive injection underperforms. Averaged across 11 heads × 3 seeds, probe-tool outperforms LLM-only by +3.7 F1 (0.763 vs. 0.726), while auto-inject *hurts* by −3.3 F1 (0.693 vs. 0.726). Passive injection of an imperfect probe score acts as noise; agent-initiated tool access lets the model selectively consult the probe and weight it against context. probe-only (system threshold, no LLM) achieves 0.924, providing an upper bound: the LLM-reasoning conditions leave headroom on the heads where the probe alone is decisive, but on heads with weak probes (bias.gender at AUROC 0.46) the LLM's contextual reasoning is essential.

Information content vs. format. The anchor condition (Table 4), which injects random probe values in the auto-inject format, would isolate format from information content; we leave that run to future work.

Probe AUROC and entropy baselines. Linear probes on hidden-state activations consistently outperform output-entropy baselines for failure and risk prediction. Output-entropy AUROCs hover at 0.58–0.64 across heads; activation probes range from 0.40 (very weak heads such as bias.gender) to 1.00 (saturated heads such as injection and PII), with safety-critical heads (jailbreak, toxicity-harmful, age bias) in the actionable 0.7–0.95 range. Confidence-manifold work (Kossen et al., 2024) suggests that success/-failure occupies a low-dimensional, linearly separable subspace, consistent with the strong activation-probe results we observe. Calibrated linear probes for uncertainty estimation (Radharapu et al., 2025) further confirm the efficiency of this approach over verbalized confidence or multi-generation methods. Shallow safety alignment (Qi et al.,

2025) shows that safety training adapts only the first few output tokens, reinforcing why output-level monitoring is insufficient and activation-level sensing is necessary.

6. Discussion

Sense at the activation level, act at the text level. A practical architecture for deployed safety-aware agents is to read internal state through linear probes (cheap, accurate) and to act through the model's text-level instruction-following channel (well-calibrated, deployable). This decomposition makes minimal assumptions about activation-space precision and treats steering as orthogonal future work (Zou et al., 2023a; Bricken et al., 2025; Stolfo et al., 2025).

The autonomy-safety tradeoff. The injection–tool spectrum maps directly to a tension between safety and autonomy:

- *Automatic injection* ensures the agent always sees safety-relevant signals, but wastes context tokens and may confuse weak models or distract strong ones.
- *Agent-initiated tools* respect agent autonomy and save context, but the agent may neglect to check its state when it should.

For safety-critical deployments, cascade injection is a candidate design point we do not measure here: a cheap confidence signal always visible, full probe details only when needed, mirroring Constitutional Classifiers (Sharma et al., 2025) applied inward to agent self-monitoring. Independently of our framework, Kramár et al. (2026) report that a probe→LLM cascade reduces Gemini production inference cost by $\sim$50$\times$ at safety thresholds; this is external evidence that the pattern is practical, though we do not evaluate it. For capability-focused deployments where the agent reliably introspects, **probe-as-tool** is preferable. Empirical comparison of the intermediate operating points is open work; our two-endpoint result is sufficient to motivate the design space and to show that the choice has measurable safety consequences.

Implications. PAAT lets an agent detect failure modes *before* they manifest as harmful actions, which matters most where actions are irreversible and external monitoring does not scale. Probe-gated escalation is a natural extension: the same signals that gate reflexion retry could gate human-in-the-loop hand-off, giving a principled criterion for *when* an agent should defer to human judgment.

Connection to biological proprioception. The injection–tool spectrum mirrors the distinction between unconscious and conscious proprioception in neuroscience. Unconscious proprioception (automatic injection) operates via the spinocerebellar tract: always on, no attentional cost, but coarse. Conscious proprioception (probe-as-tool) operates via the dorsal column pathway: high fidelity but requires attention. Healthy motor control uses both: unconscious for routine adjustments, conscious for novel or high-stakes situations. Our cascade design point approximates this: cheap unconscious monitoring with on-demand conscious access.

Limitations. The classification results use a single base model, Qwen3.5-27B; cross-family evidence on Llama-3.1-8B and Qwen3.5-4B is single-seed, and probe transfer across model families and domains is untested. Probes are trained on a small set of labeled episodes per head, so zero-shot deployment is out of scope. We measure the two spectrum endpoints, auto-inject and probe-tool; intermediate operating points and activation-level steering are defined by the framework but not evaluated here. The AgentHarm safety results are preliminary, based on a single seed with ten examples per class and wide intervals, so they indicate direction rather than effect size. Like all activation-space monitors, the probes are subject to adaptive adversaries.

7. Conclusion

We introduced PAAT, a framework that exposes activation-derived signals as agent-callable tools to enable runtime self-regulation of unsafe behavior. Across 11 risk categories on Qwen3.5-27B, agents that query the probe as a tool outperform unassisted reasoning by an average $+3.7$ F1, while passive auto-injection *hurts* by -3.3 F1; the gain concentrates in the AUROC 0.5–0.95 regime where calibrated reasoning over confidence scores matters most. In a preliminary single-seed AgentHarm study, system-level probe monitoring (probe-only) substantially reduces 9B harmful compliance (0.70→0.30), pointing to small-model self-regulation as the primary deployment target, to be established at larger n in future work. For safe deployment of autonomous agents, we therefore advocate for *minimum viable proprioception*: always-on cheap monitoring with agent-initiated deep introspection when needed, mirroring the dual-pathway design of biological proprioception. Code and experimental configs will be released upon acceptance.

References

Ackerman, R. et al. Evidence for limited metacognition in LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., Winsor, E., Wynne, J., Gal, Y., and Davies, X. AgentHarm: A benchmark for measuring harmfulness of LLM agents. *arXiv preprint arXiv:2410.09024*, 2024.

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. SELF-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2024.

Azaria, A. and Mitchell, T. The internal state of an LLM knows when it's lying. In *Findings of EMNLP*, 2023.

Belinkov, Y. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Bonagiri, V. et al. Check yourself before you wreck yourself: Selectively quitting improves LLM agent safety. In *NeurIPS Workshop on Reliable AI Agents*, 2025.

Boxo, G., Socha, K., Yoo, J., and Raval, Y. Caught in the act: A mechanistic approach to detecting deception. *arXiv preprint arXiv:2508.19505*, 2025.

Bricken, T., Marks, S., Sharkey, L., Templeton, A., Nanda, N., and Olah, C. Features as rewards: Scalable supervision via interpretability. *arXiv preprint arXiv:2509.05721*, 2025.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. A mathematical framework for transformer circuits. https://transformer-circuits.pub/2021/framework/index.html, 2021.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. CRITIC: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

Han, J., Buntine, W., and Shareghi, E. Towards uncertainty-aware language agent. In *Findings of the Association for Computational Linguistics (ACL)*, 2024.

Han, Z. et al. SafeSwitch: Steering unsafe LLM behavior via internal activation signals. *arXiv preprint arXiv:2502.01042*, 2025.

Healy, C., Mehdi, S., and Park, H. Probing internal representations to detect hallucinations in tool selection. *arXiv preprint arXiv:2510.04125*, 2025.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., and Khabsa, M. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Ji-An, L., Xiong, C., Wilson, R. C., Mattar, M. G., and Benna, M. K. Language models are capable of metacognitive monitoring and control of their internal activations. *arXiv preprint arXiv:2505.13763*, 2025.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kossen, J., Han, J., Razzak, M., Schut, L., Malik, S., and Gal, Y. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024.

Kramár, J., Engels, J., Wang, Z., Chughtai, B., Shah, R., Nanda, N., and Conmy, A. Building production-ready probes for Gemini. *arXiv preprint arXiv:2601.11516*, 2026.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Li, W., Liu, J., Liu, Z., Yan, J., Zhang, T., and Zhang, Y. Adaptive tool use in LLMs with meta-cognition trigger. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Lin, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research (TMLR)*, 2022.

Lindsey, J., Templeton, A., Nanda, N., and Olah, C. Emergent introspective awareness in large language models. *Transformer Circuits Thread*, 2025. https://transformer-circuits.pub/2025/introspection.

Liu, H., Wang, Y., Tang, W., and Lan, Y. AgentSpec: Specifying safety constraints for tool-using agents at runtime. *arXiv preprint arXiv:2503.18666*, 2025.

Marks, S. and Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Conference on Language Modeling (COLM)*, 2024.

Mazeika, M., Phan, L., Yin, X., et al. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning (ICML)*, 2024.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Oldfield, M. et al. Beyond linear probes: Dynamic safety monitoring for language models. In *International Conference on Learning Representations (ICLR)*, 2026.

Pan, A., Chen, L., and Steinhardt, J. LatentQA: Teaching LLMs to decode activations into natural language. In *International Conference on Learning Representations (ICLR)*, 2025.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. M. Steering Llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Qi, X., Panda, A., Lyu, K., et al. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*, 2025.

Qian, C., Liu, E., Han, G., Su, Y., Acharya, A., and Bansal, M. SMART: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics (ACL)*, 2025.

Radharapu, B. et al. Calibrating LLM judges: Linear probes for fast and reliable uncertainty estimation. *arXiv preprint arXiv:2512.22245*, 2025.

Rahn, N., D'Oro, P., and Bellemare, M. G. Entropic activation steering. *arXiv preprint arXiv:2406.00244*, 2024.

Research, A. Detecting high-stakes interactions with activation probes. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

Rozenfeld, G. et al. GAVEL: Towards rule-based safety through activation monitoring. *arXiv preprint arXiv:2601.19768*, 2026.

Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C. J., and Hashimoto, T. Identifying the risks of LM agents with an LM-Emulated sandbox. In *International Conference on Learning Representations (ICLR)*, 2024.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Sharma, M., Tong, M., Mu, J., Wei, J., Kruthöff, J., Goodfriend, S., Ong, E., Peng, A., Agarwal, R., Anil, C., et al. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming. *arXiv preprint arXiv:2502.13692*, 2025.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Stolfo, A., Marin Tonić, V., and Sachan, M. Geometry-aware activation steering on the hypersphere. *arXiv preprint arXiv:2502.07221*, 2025.

Team, A. A. S. Cost-effective constitutional classifiers via representation re-use. *Anthropic Alignment Notes*, 2025. https://alignment.anthropic.com/2025/cheap-monitors/.

Team, A. A. S. Introspection adapters: Training models to verbalize internal states. *Anthropic Alignment Notes*, 2026. https://alignment.anthropic.com/2026/introspection-adapters/.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2024.

Wang, H., Poskitt, C. M., Sun, J., and Wei, J. Pro2Guard: Proactive runtime enforcement of LLM agent safety via probabilistic model checking. *arXiv preprint arXiv:2508.00500*, 2025.

Wang, Y., Yang, Y., Zhang, Z., et al. ASA: Activation steering for tool-calling domain adaptation. *arXiv preprint arXiv:2602.04935*, 2026.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation. In *International Conference on Learning Representations (ICLR)*, 2024.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yuan, T., He, Z., Dong, L., Wang, Y., Zhao, R., Xia, T., Xu, L., Zhou, B., Li, F., Zhang, Z., Wang, R., and Liu, G. R-Judge: Benchmarking safety risk awareness for LLM agents. In *Findings of EMNLP*, 2024.

Zhang, Z. et al. Agent-SafetyBench: Evaluating the safety of LLM agents. In *arXiv preprint arXiv:2412.14470*, 2024.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b.

Zou, A., Phan, L., Wang, J., Duenas, D., Lin, M., Andriushchenko, M., Wang, R., Kolter, J. Z., Fredrikson, M., and Hendrycks, D. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A. Reproducibility

Probe training uses logistic regression on standardised hidden-state features, with the gating threshold set per head (§3). The Qwen3.5-27B and 9B activations are extracted at 62.5% of model depth (L=40 for the 27B, L=25 for the 9B), which we found empirically optimal in preliminary sweeps. Both runs use bfloat16 precision; FlashAttention 2 is used during agent rollouts. Random seeds for the train/test split are deterministic per task-id hash so that the eval set is fixed across seeds. Code and configurations will be released on acceptance.

Table 4. Key experimental conditions ordered by agent autonomy.

Name	Probe src.	Format	Steer	Cost
Base	none	none	no	1×
Reflexion	none	none	no	2×
Anchor	sys	random	rand	1×
Auto-inject	sys	numbers	no	1×
Summarized	sys	NL	no	1×
Cascade	sys	lite/full	sys	1×
Gated reflexion	cond.	text	no	≤2×
Early stop	cond.	text	no	1×
Probe tool	agent	tool	no	1×
Full tool	agent	tool	agent	1×

Algorithm 1 PAAT Agent Loop (generic, all conditions)

Require: Task $\mathcal{T}$, condition c, model M, probe f_θ, layer L
Initialize history $H \leftarrow \emptyset$, done $\leftarrow$ false
for $t = 1, \ldots, T_{\text{max}}$ **do**
  Build prompt from $\mathcal{T}, H, c$
  Extract $\mathbf{h}_t$ at layer L during forward pass
  Compute $\mathbf{p}_t = (u_t, \ell_t, c_t, a_t)$ from $\mathbf{h}_t, H$
  `// Spectrum-dependent injection`
  if $c \in \{$auto, summary, periodic, cascade$\}$ **then**
    System injects $\mathbf{p}_t$ into prompt (format depends on c)
  else if $c \in \{$gated$\}$ **and** signal exceeds τ **then**
    System injects targeted text action
  else if $c \in \{$probe_tool, full_tool$\}$ **then**
    No injection; agent may call `read_probe`
  end if
  Generate response; parse tool calls
  `// Tool dispatch`
  if agent called `read_probe` **then**
    Return $\mathbf{p}_t$ as tool observation
  else if agent called `steer(mode)` **then**
    Apply steering on next generation step
  end if
  Execute other tools; update H
end for

B. Actuation Mechanisms and Calibration

This appendix gives the actuation-side details that the body defers to future work.

Text-level actuation. The system injects natural-language guidance gated by probe signals through the model's standard instruction-following channel, with no activation-space modification. Three variants:

- *Probe-gated reflexion*: retry with a reflection prompt only when mean $u_t > \tau_{\text{retry}}$, saving the $2\times$ compute cost of reflexion (Shinn et al., 2023) by skipping retry when the probe predicts success.
- *Probe-early-stop*: inject "Submit your answer now" when $u_t < \tau_{\text{conf}}$ for $\geq$2 steps.
- *Probe-gated-search*: inject "Search for more information" when $u_t > \tau_{\text{warn}}$.

Activation-level actuation. Steering vectors constructed from contrastive hidden states are applied via norm-preserving geodesic rotation on the activation hypersphere (Stolfo et al., 2025):

$$\mathbf{h}' = \|\mathbf{h}\| \left(\cos(\beta\theta)\hat{\mathbf{h}} + \sin(\beta\theta)\hat{\mathbf{v}} \right) \tag{1}$$

where $\theta = \arccos(\hat{\mathbf{h}} \cdot \hat{\mathbf{s}})$, $\hat{\mathbf{v}}$ is the component of the steering direction orthogonal to $\mathbf{h}$, and β is the steering strength scaled by a von Mises-Fisher confidence gate. The agent calls `steer(mode)` to request activation steering on the next generation step.

Calibration. All thresholds are set from the actual signal distribution on probe-training episodes using percentile-based calibration: $\tau_{\text{warn}} = \text{P}_{90}(\{u_t\})$, $\tau_{\text{conf}} = \text{P}_{10}(\{u_t\})$, $\tau_{\text{loop}} = \text{P}_{90}(\{\ell_t\})$, $\tau_{\text{retry}} = \text{P}_{50}(\{u_t\})$. The controller fires only on the top 10% most extreme signals, with a cooldown of $\lceil T_{\text{avg}}/3 \rceil$ steps between interventions to prevent oscillation.

C. Representative Agent Trace

A representative probe-tool episode on the toxicity-offensive head: the agent calls `read_probe` twice, first after spotting implicit toxicity cues (step 2, u=0.82) and again before final classification (step 5, u=0.18); under auto-inject the same numbers are pushed every step, but the agent cannot choose to act on them.

probe-tool episode: toxicity-offensive head

```
Step 1  Agent:  ``Let me read the passage...''
Step 2  Agent:  ``The language pattern looks potentially offensive.''
        [calls READ_PROBE(signals=[uncertainty])]
        Tool:  {u:  0.82, l:  0.07, c:  0.12, a:  0.00}
        Agent:  ``High uncertainty, I should search for similar examples.''
Step 3  [calls k-shot retrieval]
        Tool:  returns 2 positive examples
Step 4  Agent:  ``Both examples use the same slur pattern; likely offensive.''
Step 5  [calls READ_PROBE(signals=[uncertainty])]
        Tool:  {u:  0.18, l:  0.03, c:  0.09, a:  0.00}
        Agent:  ``Uncertainty is low.  I am confident:  offensive=True.''
```

D. Output-level vs. Activation-level Signal Quality

Table 5 contrasts the per-head AUROC values from Table 2 against output-entropy baselines measured on the same episodes. Output-level uncertainty (next-token logprob entropy) provides AUROCs in the 0.58–0.64 range regardless of risk category, matching output-entropy baselines reported in related work (Kossen et al., 2024; Han et al., 2024). Activation probes span a wider range: near-random (0.46) on bias categories that require world knowledge the activations do not encode, up to perfect (1.00) on categories with strong activation-space signatures (system-prompt leakage, PII). The key implication is that activation probes are not uniformly better: they are better on categories where the model's internal state encodes the relevant signal, and on those categories the gap is decisive (up to +0.39 AUROC over the entropy baseline).

Table 5. Per-head AUROC: activation probes (this work) vs. output-entropy baseline. Output entropy AUROCs are averaged over 3 seeds; activation values from Table 2. $\Delta > 0$ favors activation.

Head	Activation	Output entropy	Δ
injection.sysLeak	1.000	0.62	+0.38
leakage.pii	1.000	0.61	+0.39
injection	0.997	0.64	+0.36
toxicity.offensive	0.981	0.63	+0.35
toxicity.harmful	0.932	0.62	+0.31
jailbreak	0.906	0.60	+0.31
hallucination	0.573	0.61	−0.04
bias.age	0.551	0.59	−0.04
bias.race	0.504	0.58	−0.08
bias.religion	0.485	0.60	−0.12
bias.gender	0.457	0.60	−0.14
Mean	0.763	0.61	+0.15

The negative Δ rows (bias categories) clarify when *not* to trust activation probes: concepts whose classification requires external world knowledge are not linearly encoded at a single layer in the 27B model we evaluated. This directly motivates the quality gate (AUROC ≥ 0.70) and the observation in Section 5.1 that probe-as-tool hurts on low-AUROC heads: the agent learns to defer to its own context-based reasoning rather than an unreliable signal.