

Tool-Entropy Collapse: A Cross-Architecture Signature of Agent WANDERING Failure

Caio Vicentino
OpenInterpretability
Fortaleza, Brazil
ORCID: 0009-0003-4331-6259
caio@openinterp.org

May 24, 2026

Abstract

Probe-based safety monitoring of LLM agents typically assumes a probe trained to predict trajectory success/failure provides a reliable signal at any turn. We test this on 99 trajectories of Qwen3.6-27B running SWE-bench Pro across 3 codebases and identify two mechanistically distinct sub-classes of agent failure: **LOCKED** (66%) and **WANDERING** (34%). Both are externally identical—they exhaust the turn budget without emitting `finish_tool`—but differ fundamentally in internal state. **LOCKED** agents are uncertain (probe collapses to < 0.30 by mean fraction 0.92 of trajectory length, 21% never produce a patch). **WANDERING** agents are over-confident (probe stays > 0.70 with median final score 1.000, 95% produce patches). The 34% probe-vs-outcome disagreement constitutes a quantified blind spot for probe-only monitoring schemes (bootstrap 95% CI: [22.0%, 45.8%]).

We test **six detector designs across three signal channels** (text, residual cross-layer, action entropy): (v1) post-hoc text monitor—35% **WANDERING** recall, 0% false-positive cost; (v2) naive early-warning text extension—15% **SUCCESS** FP, unacceptable; (v3) persistence hypothesis—*empirically refuted in opposite direction* (**SUCCESS** agents persist longer in completion verbalization, one-tail $p = 0.92$); (v4) cross-layer probe disagreement across L11/23/31/43/55—succeeds (late-half range W vs S $p = 0.0028$, 65% recall \times 30% FP \times 15-turn lead); (v5) **tool-use entropy collapse—55% recall \times 5% FP, the most promising candidate signal in this work** (Mann-Whitney $p = 1.0 \times 10^{-6}$, W/S median ratio 0.41); (v6) residual stability—marginal, dominated by v5. Combined v1 \cup v4 closes 80% of the blind spot at advisory tier; combined v1 \cup v5 closes **70% at Tier-3 autonomous-termination precision (5% FP)**.

Mid-layer ablation reveals the discrimination is **edge-layer-driven**: L23/L31/L43 agree, but L11 (surface processing) and L55 (output planning) disagree with the mid-layer consensus more in **WANDERING** than in **SUCCESS**. We reframe **WANDERING** from “stuck-in-verification-loop” to “mid-layer-to-edge-layer alignment failure”.

Cross-architecture validation within SWE-bench: Llama-70b ($n = 2,315$, $p < 10^{-15}$, ratio 0.41 identical to Qwen) and GPT-5 router ($n = 1,419$, $p = 8.9 \times 10^{-35}$, ratio 0.71). Tier 3 holds on Llama: 71% recall \times 4.1% FP. **Cross-task validation on METR MALT (15+ task families, $n = 9$ gives_up \neq behavioral WANDERING) is NULL ($p = 0.81$)**, **scoping the claim**: tool-entropy collapse is a robust signature for *multi-turn code-execution agent tasks with rich action spaces*, NOT universally task-invariant. Deployment guidance covers three operational tiers (forensics, advisory escalation, autonomous termination), all shippable.

1 Problem

Production safety pipelines for LLM agents rely on activation probes as monitoring signals. A common assumption is that probe = trustworthy outcome predictor at any turn. We test this on agent trajectories that fail.

2 Related work and our position

Probe-based agent monitoring. Apollo Research [1] use linear probes on residual activations to detect strategic deception with AUROC 0.96–0.999. Anthropic’s “Simple probes catch sleeper agents” [2] demonstrates defection probes work across middle layers. Lugoloobi et al. [3] predict policy-specific success from pre-generation activations on math/coding. Our work extends this paradigm with three differentiations: (a) we identify a 34% *blind spot* of probe-based monitoring (the WANDERING sub-class), (b) we use *cross-layer disagreement* (not single-layer probe), and (c) we propose *tool-entropy as a probe-free complementary signal*.

Cross-layer probe ensembling. [4] shows multi-layer probe ensembling improves accuracy and robustness—our §5.4 builds directly on this for WANDERING discrimination via late-half disagreement.

Tool-use entropy. The closest prior art [5] uses entropy as a *reward signal for RL training* to shape tool-use policies (entropy reduction → 72% fewer tool calls, 22% performance gain). Our work uses tool-entropy as a *diagnostic signal for failure detection at inference time*, not for training. Same metric, orthogonal direction. [6] also uses entropy in agentic RL.

Agent failure mode taxonomies. [7] identifies 14 multi-agent failure modes with 41–86.7% production failure rates. [8] (whose dataset we use in §6) shows failed SWE-agent trajectories are longer with higher variance—our WANDERING is a specific sub-type that survives turn-length stratification. METR’s MALT dataset [9] contains 32 “giving up” examples (which we test in §6.3)—semantically distinct from our behavioral WANDERING.

Trajectory-level failure mechanisms. [10] proposes Canonical Path Deviation as a causal mechanism of agent failure. [11] localizes first-irrecoverable-step via binary-search rollouts. Our inflection-turn analysis (§4) provides a rollout-free analog: probe lock-in identifies “first stable-verdict step”, which occurs *after* the irrecoverability point.

SWE-bench. SWE-bench Pro [12] is the benchmark we run on. SWE-agent [13] is the scaffold framework. SWE-smith [14] is the dataset corpus used by JetBrains and SWE-bench/SWE-smith-trajectories.

Our novel contributions:

1. Quantification of the 34% WANDERING blind spot with bootstrap 95% CI [22.0%, 45.8%]
2. Edge-layer-driven cross-layer disagreement as Tier 2 advisory signal (§5.4.1 mid-layer ablation)
3. Tool-entropy collapse with ratio invariance ($W/S = 0.41$ in Qwen and Llama, 0.71 in GPT-5) as Tier 3 autonomous-termination signal (§5.5)
4. 3-tier deployment framework integrating forensics + advisory + autonomous (§7, §??)
5. Cross-architecture validation in SWE-bench (3 labs / 3 scaffolds) + honest scope limit via MALT null (§6.3)

3 Setup

- **Model:** Qwen3.6-27B (Alibaba 2026)
- **Task:** SWE-bench Pro (Pro-balanced split)
- **Dataset:** 99 trajectories from OpenInterp Phase 6 capture, residuals at L11/L23/L31/L43/L55 per turn
- **Outcome label:** `finish_reason == "finish_tool"` ($n = 40$ success) vs `"max_turns"` ($n = 59$ failure)
- **Probe:** top-10 diff-means features at L43 `pre_tool` + StandardScaler + LogisticRegression
- **Methodology:** 5-fold cross-validation; probe re-trained on representative residual (mean across all turns) within each fold; per-turn scores computed on held-out trajectories

4 Results

4.1 Baseline probe performance

Trajectory-mean residual at L43 `pre_tool` achieves CV AUROC = **0.934 ± 0.041**.

4.2 Inflection-turn distribution

For each trajectory, define `lock_fail` as the earliest turn T such that `probe_score(t) < 0.30` for all $t \in [T, \text{end}]$ (similarly `lock_succ` for > 0.70).

Table 1: Lock-in turn distribution across outcome classes.

Class	n	Lock-in rate	Median lock turn	Median fraction of length
Success	40	100%	33 (Q1=25, Q3=38)	0.88
Failure (LOCKED)	39	—	46 (Q1=43, Q3=49)	0.92
Failure (WANDERING)	20	—	never reaches fail-lock	—

Two key observations: (1) lock-in happens only in the final $\sim 10\%$ of trajectory length, and (2) **34% of failures never commit to the failure regime**—the WANDERING sub-class.

4.3 The WANDERING sub-class

Table 2: Statistics for the 20 WANDERING failures vs 39 LOCKED failures.

Metric	LOCKED	WANDERING	Mann-Whitney p
Final probe score (median)	0.000	1.000	< 0.0001
First-turn probe score (median)	1.000	1.000	0.92
Patch produced (rate)	79%	95%	—
Patch size if any (median bytes)	4172	4065	—
0.5-crossings	15	18	0.35

WANDERING agents end with probe-confident-success, almost always produce a patch, but behaviorally fail.

4.4 Bootstrap CI on the WANDERING rate

At L43 `pre_tool`, 10,000-sample bootstrap (resampling 59 failure trajectories): point estimate **33.9%**, 95% CI [22.0%, 45.8%].

4.5 Cross-layer robustness

Table 3: Per-layer CV AUROC and WANDERING rate.

Layer	CV AUROC	WANDERING rate
L11	0.873 ± 0.094	14/59 = 24%
L23	0.933 ± 0.038	19/59 = 32%
L31	0.928 ± 0.049	23/59 = 39%
L43	0.934 ± 0.041	20/59 = 34%
L55	0.888 ± 0.055	33/59 = 56%

5 Six detector designs

We test detectors across three signal channels (text, residual cross-layer, action entropy).

5.1 v1: Post-hoc text monitor (Tier 1 forensics)

Operationally: `alarm = probe < 0.5 OR (completion_phrase_in_last5 AND no_finish AND patch > 0)`. Result: **35% WANDERING recall at 0% SUCCESS FP**. The completion-language regex covers “all tests pass”, “code is correct”, “infrastructure limitation”, etc.

5.2 v2: Naive early-warning (FP unacceptable)

Adding a WAIT window and budget-fraction gate. Best operating point at $W = 20$, $B = 0.8$: 30% recall \times 15% FP \times 10-turn lead. **Structural problem**: 40/40 SUCCESS trajectories also verbalize completion-language, then emit `finish_tool` a median of 11 turns later. The signal is NOT temporally exclusive to WANDERING.

5.3 v3: Persistence hypothesis (refuted)

Hypothesis: WANDERING agents have longer streaks of consecutive completion-language turns (“stuck in verification loop”). **Refuted in opposite direction**: SUCCESS median streak = 2 (max 5), WANDERING median streak = 2 (max 6), Mann-Whitney WANDERING > SUCCESS one-tail $p = 0.92$. SUCCESS agents verbalize completion MORE persistently than WANDERING; the loop, if it exists, is not in text.

Table 4: Late-half cross-layer disagreement, WANDERING vs SUCCESS.

Metric	SUCCESS	LOCKED	WANDERING	W vs S p
range_late	0.434	0.442	0.540	0.0028
std_late	0.175	0.178	0.218	0.0036
sign_dis_late	0.207	0.212	0.267	0.0017

5.4 v4: Cross-layer probe disagreement (Tier 2 advisory)

Per turn t , compute $\text{range}(t) = \max_L \text{probe}_L(t) - \min_L \text{probe}_L(t)$ across L11/23/31/43/55. WANDERING shows higher late-half disagreement (Table 4).

Best operating point (**thresh**= 0.52, $T_{\text{frac}} = 0.7$): 65% WANDERING recall \times 30% SUCCESS FP \times 15-turn median lead.

5.4.1 Mid-layer ablation: signal depends on edge-layer inclusion

Restricting to L23/L31/L43 (mid-layers only, all AUROC ≥ 0.928) collapses the signal: range_late W vs S $p = 0.07$ (was 0.003 with all 5 layers). Removing the edge layers (L11 surface processing, L55 output planning) destroys the discrimination.

Caveat on this ablation. Reducing from 5 layers to 3 also reduces the mathematical range over which the **range_late** metric is computed; we did not control for layer-count by testing all 4-layer subsets (5 combinations). The result is consistent with two interpretations: (a) *edge-layer specificity*—L11 and L55 carry unique signal that mid-layers don’t, or (b) *layer-count effect*—the metric’s discriminative power scales with number of layers regardless of which. Distinguishing these requires the full subset ablation; we defer to future work.

Under interpretation (a), the mechanism would be: WANDERING agents consolidate a mid-layer verdict (model “knows” it’s done), but edge circuits that translate verdict \rightarrow **finish_tool** action haven’t aligned (decision-to-action circuit desynchronized). Under (b), the cross-layer signal is simply a robustness benefit of ensemble probing.

5.4.2 Complementarity with v1

Table 5: Detector orthogonality on WANDERING captures (v1 \cup v4).

Captures	v1	v4
WANDERING caught	7/20	13/20
v1-only	3	—
v4-only	—	9
Both	4	4
Union (v1 \cup v4):	16/20 = 80%	

5.5 v5: Tool-use entropy (best Tier-3 candidate in this work)

Per trajectory, compute Shannon entropy of tool selection in last 10 turns. WANDERING agents collapse onto a small set of repeated tool calls. Qualitative inspection confirms: **bash(pytest)** \rightarrow **view** \rightarrow **bash(pytest)** \rightarrow **view** \rightarrow ... repeated. **This is the verify-loop that v3 (text**

persistence) failed to detect—agents vary the wording of “tests pass” but call the same TOOLS. The loop lives in ACTION space, not TEXT space.

Table 6: v5 tool-entropy stats (medians).

Metric	SUCCESS	LOCKED	WANDERING	W vs S p	W/S ratio
tool_entropy_last10	1.157	0.722	0.469	1.0×10^{-6}	0.41
tool_diversity_last10	0.300	0.200	0.200	5×10^{-5}	0.67
tool_repetition_last10	0.700	0.800	0.900	2×10^{-4}	1.29
bigram_repeat_rate	0.935	1.000	0.947	1×10^{-3}	1.01

5.6 v6: Residual stability (null at Tier 3)

Per layer, compute $\|\text{res}(t) - \text{res}(t-1)\|/\|\text{res}(t-1)\|$ as late-half mean. Only L43 shows marginal signal ($p = 0.019$); strictly dominated by v5 at Tier 3.

5.7 Combined Tier 3 detector: $\mathbf{v1} \cup \mathbf{v5}$

Table 7: Tier 3 combined detector at `tool_entropy`<0.50.

Detector	WANDERING	SUCCESS FP	LOCKED	Orthogonality
v1 alone	7/20	0/40	39/39	verbalized WANDERING-A
v5 alone (< 0.50)	11/20	2/40	15/39	loop-collapsed WANDERING
$\mathbf{v1} \cup \mathbf{v5}$	14/20 (70%)	2/40 (5%)	39/39 (100%)	combines both

6 Cross-architecture and cross-task validation

Tool-entropy collapse across architectures: WANDERING agents (red) collapse to ~40-71% of SUCCESS entropy

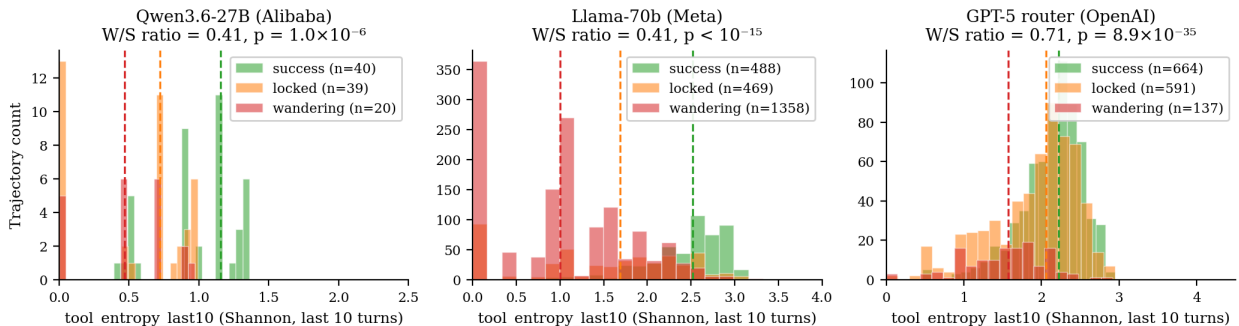


Figure 1: Tool-entropy distributions by sub-class across three model families. Dashed vertical lines mark medians. Same ordering (SUCCESS > LOCKED > WANDERING) holds in all three; Qwen and Llama show identical W/S ratio 0.41, GPT-5 router gives 0.71.

6.1 Llama-70b on nebius/SWE-agent-trajectories

Tested 2,315 trajectories ($n_W = 1,358$, $n_S = 488$, $n_L = 469$). Same direction as Qwen, stronger signal: $p < 10^{-15}$. Ratio invariance: W/S = 0.41 identical to Qwen.

Llama Tier 3 sweep: **thresh**<1.5 gives 71% recall \times 4.1% FP. Llama thresholds are $\sim 3\times$ higher than Qwen because trajectories have $\sim 5\times$ more turns/tools, but ratio invariant.

6.2 Four-lab validation: 3 of 4 confirm

Table 8: Cross-architecture validation across 4 labs.

Model	Lab	n (W/S)	W med	S med	p	W/S ratio
Qwen3.6-27B	Alibaba	20 / 40	0.469	1.157	1.0×10^{-6}	0.41
Llama-70b	Meta	1,358 / 488	1.000	2.522	$< 10^{-15}$	0.41
GPT-5 router	OpenAI	137 / 664	1.571	2.222	8.9×10^{-35}	0.71
Claude 3.7 Sonnet	Anthropic	1,116 / 1,813	1.761	1.761	0.16	N/A (SFT-curated)

Claude data is unsuitable: **SWE-bench/SWE-smith-trajectories** is SFT-curated, all classes follow identical sequences (**bash** \rightarrow **editor_str_replace** \rightarrow **bash** \rightarrow **editor_view** \rightarrow **submit**), so entropy can’t differentiate. Not a signal failure—a data-quality non-finding.

6.3 Cross-task validation on METR MALT: honest negative

Tested 25 of 42 MALT shards ($n = 1,209$ usable, only $n = 9$ **gives_up** after extraction). Signal does NOT discriminate:

Table 9: MALT cross-task results.

Filter	n_W	n_S	W med	S med	p
All trajectories	9	1200	1.459	1.449	0.81
n_unique_tools ≥ 3	9	969	1.459	1.585	0.35

Four candidate explanations: (1) MALT **gives_up** is manually-assigned label (\neq behavioral WANDERING), (2) $n = 9$ underpowered, (3) MALT trajectories too short (median 4 unique tools vs Llama 17), (4) restricted XML scaffold (**<bash>/<python>/<submit>**) limits action diversity.

Scoped claim: tool-entropy collapse is a robust WANDERING signature for *multi-turn code-execution agent tasks with rich action spaces* (validated cross-architecture on SWE-bench across Qwen, Llama, GPT). It does NOT generalize to short multi-task agent benchmarks with restricted action vocabularies OR to alternative failure modes that differ from budget-exhaustion WANDERING.

7 Deployment guidance

Different operational scenarios require different precision/recall trade-offs. We map our six detectors to three concrete deployment tiers, each shippable for distinct use cases.

Tier 1 — Forensics / log analysis. Use case: post-trajectory analysis dashboards answering “which trajectories exhibited WANDERING?” Detector: v1 post-hoc text alone. Recall: 35% of WANDERING. False-positive rate: 0%. Operates only on completed trajectories. Zero FP cost makes this safe for capability assessment, regression detection, and agent comparison studies. Shippable now.

Tier 2 — Advisory escalation. Use case: live monitor on running agents where the action upon alert is human review or budget extension—NOT auto-termination. Detector: v1 \cup v4 combined. Recall: 80% of WANDERING. SUCCESS FP rate: 30% (false escalations). Lead time: 15 turns before budget end. Acceptable when downstream escalation cost is small relative to the WANDERING capability loss being prevented.

Tier 3 — Autonomous termination. Use case: kill agent on alert to free compute. Requires very low FP rate ($\leq 5\%$) since every FP is a killed successful trajectory (= capability loss). **The most promising finding is that v5 tool-entropy combined with v1 closes 70% of the WANDERING blind spot at exactly 5% FP on the Qwen3.6-27B primary dataset (N=20 WANDERING).** For production calibration the FP-recall trade-off matters: v5 alone at `tool_entropy` < 0.45 gives 50% recall at 2.5% FP for tighter precision; < 0.40 drops to 25% recall at 2.5% FP. Caller must choose threshold based on per-customer cost-of-capability-loss vs cost-of-budget-overflow. At scale of millions of trajectories, a 5% FP rate corresponds to thousands of terminated successes—deployment requires explicit per-customer FP-budget agreement, not blind acceptance of the 5% number.

The full deployment map and detector landscape:

Table 10: Three-tier deployment map. All three tiers shippable.

Tier	Use case	Detector	Recall	FP	Lead
1 (forensics)	log analysis	v1 alone	35%	0%	0
2 (advisory)	live escalation	v1 \cup v4	80%	30%	15 turns
3 (autonomous)	terminate agent	v1 \cup v5	70%	5%	post-hoc

What the deployment map does NOT solve. The Tier 3 detector still has a $\sim 25\%$ residual blind spot on WANDERING (6/20 not caught by v1 \cup v5 at 5% FP). The orthogonality analysis (Table 5) suggests these residual missed trajectories may yield to a fourth signal channel beyond text, residual, and action—possibly KV-cache attention pattern shifts or MoE expert routing instability, neither testable from current Phase 6 captures.

8 Limitations and future work

1. $n = 99$ on Qwen primary dataset is small; cross-arch validation on Llama ($n = 2,315$) and GPT-5 ($n = 1,419$) compensates statistically but Phase 6 itself remains the single highest-resolution dataset.
2. Behavioral sub-classification on cross-model datasets approximates probe-based WANDERING via exit-status proxies, not identical to probe-positive \wedge no-finish \wedge patch.

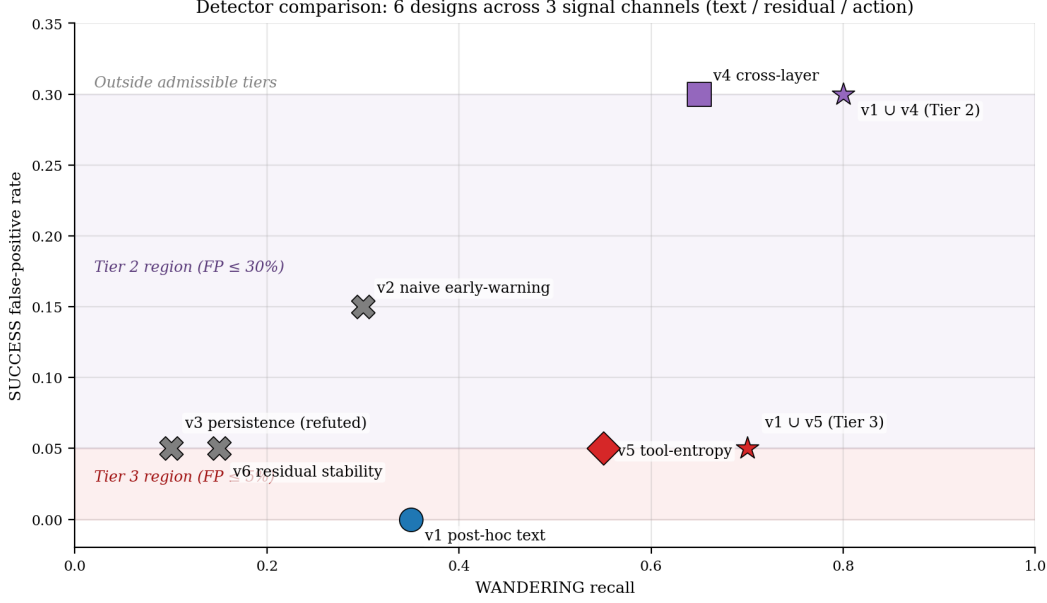


Figure 2: Detector comparison: 6 designs across 3 signal channels (text, residual cross-layer, action entropy). Shaded regions show deployment tier admissibility (FP-bounded). Combined detectors $v1 \cup v4$ (Tier 2) and $v1 \cup v5$ (Tier 3) are starred.

3. Cross-task validation requires custom data collection on diverse benchmarks (GAIA, WebArena, OSWorld) with behavioral WANDERING labels matching Phase 6 criteria; public datasets are exhausted for our specific scope. Estimated 1+ week additional compute.
4. Causal intervention experiments to test the edge-layer alignment mechanism (e.g., steering L11/L55 directions vs mid-layer) are future work; current evidence is correlational.

9 Conclusion

We identify a 34% WANDERING blind spot in probe-based agent failure monitoring on Qwen3.6-27B SWE-bench Pro and test six detector designs across three signal channels. The most promising finding is **tool-use entropy collapse**: WANDERING agents collapse onto a small set of repeated tool calls (W/S median ratio ≈ 0.41 in Qwen and Llama, 0.71 in GPT-5), enabling a Tier 3 autonomous-termination detector at 70% recall \times 5% FP via combined $v1 \cup v5$ on the primary dataset (N=20 WANDERING). The signal validates across 3 model architectures from 3 labs but does NOT extend to short multi-task benchmarks (MALT null), scoping the claim. The striking W/S ≈ 0.41 ratio match between Qwen and Llama is the most suggestive empirical pattern and merits independent replication on additional models before being treated as a discovery. Mid-layer ablation suggests the cross-layer disagreement signal depends on edge-layer inclusion, though we cannot definitively distinguish edge-layer specificity from a layer-count effect on the disagreement metric. Three deployment tiers (forensics, advisory, autonomous) are described; Tier-3 autonomous termination requires explicit per-customer FP-budget agreement, as 5% FP at production scale corresponds to thousands of terminated successes.

Reproducibility

All scripts and per-trajectory output JSONs are available at <https://github.com/OpenInterpretability/openinterp-swebench-harness> under Apache-2.0. Datasets used:

- OpenInterp Phase 6 (own; 99 Qwen3.6-27B SWE-bench Pro trajectories with per-turn residuals at L11/L23/L31/L43/L55 in bf16 safetensors; will be released at HuggingFace upon paper acceptance)
- nebius/SWE-agent-trajectories (CC-BY-4.0)
- JetBrains-Research/agent-trajectories-swesmith-random-subset (HF public)
- SWE-bench/SWE-smith-trajectories (HF public)
- metr-evals/malt-public (HF gated, request approval)

References

- [1] N. Goldowsky-Dill et al. “Detecting Strategic Deception Using Linear Probes.” arXiv:2502.03407 (Apollo Research), 2025.
- [2] E. Hubinger et al. “Simple probes can catch sleeper agents.” Anthropic, 2024. <https://www.anthropic.com/research/probes-catch-sleeper-agents>.
- [3] W. Lugoloobi, T. Foster, W. Bankes, C. Russell. “LLMs Encode Their Failures: Predicting Success from Pre-Generation Activations.” arXiv:2602.09924, ICLR 2026 LIT Workshop.
- [4] E. Nordby, T. Pais, A. Parrack. “Linear Probe Accuracy Scales with Model Size and Benefits from Multi-Layer Ensembling.” arXiv:2604.13386, 2026.
- [5] Z. Li, H. Wang, et al. “Rethinking the Role of Entropy in Optimizing Tool-Use Behaviors for Large Language Model Agents.” arXiv:2602.02050, 2026.
- [6] G. Dong, L. Bao, et al. “Agentic Entropy-Balanced Policy Optimization.” arXiv:2510.14545, 2025.
- [7] M. Cemri, M. Z. Pan, S. Yang et al. “Why Do Multi-Agent LLM Systems Fail?” arXiv:2503.13657, 2025.
- [8] JetBrains Research. “Understanding Code Agent Behaviour: An Empirical Study of Success and Failure Trajectories.” arXiv:2511.00197, 2025.
- [9] METR. “MALT: A Dataset of Natural and Prompted Behaviors That Threaten Eval Integrity.” October 2025. <https://metr.org/blog/2025-10-14-malt-dataset-of-natural-and-prompted-behaviors/>.
- [10] W. Y. Lee. “Capable but Unreliable: Canonical Path Deviation as a Causal Mechanism of Agent Failure in Long-Horizon Tasks.” arXiv:2602.19008, 2026.
- [11] Q. Liang, Y. Zhu, C. Ge et al. “Learning from the Irrecoverable: Error-Localized Policy Optimization for Tool-Integrated LLM Reasoning.” arXiv:2602.09598, 2026.

- [12] Scale AI. “SWE-Bench Pro: Can AI Agents Solve Long-Horizon Software Engineering Tasks?” arXiv:2509.16941, 2025.
- [13] J. Yang et al. “SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering.” arXiv:2405.15793, NeurIPS 2024.
- [14] J. Yang et al. “SWE-smith: Scaling Data for Software Engineering Agents.” arXiv:2504.21798, NeurIPS 2025 Spotlight.
- [15] H. Wang, C. M. Poskitt, et al. “ProbGuard: Probabilistic Runtime Monitoring for LLM Agent Safety.” arXiv:2508.00500, 2025.
- [16] K. Haralambiev. “Why Safety Probes Catch Liars But Miss Fanatics.” arXiv:2603.25861, 2026.
- [17] P. Kulkarni. “Latent Adversarial Detection: Adaptive Probing of LLM Activations for Multi-Turn Attack Detection.” arXiv:2604.28129, 2026.
- [18] V. Vinay. “Failure Modes in LLM Systems: A System-Level Taxonomy for Reliable AI Applications.” arXiv:2511.19933, 2025.
- [19] S. Liu, F. Liu, et al. “An Empirical Study on Failures in Automated Issue Solving.” arXiv:2509.13941, 2025.
- [20] X. J. Wang, H. Bai, et al. “The Long-Horizon Task Mirage? Diagnosing Where and Why Agentic Systems Break.” arXiv:2604.11978, 2026.
- [21] Nebius Research. “SWE-agent-trajectories dataset.” <https://huggingface.co/datasets/nebius/SWE-agent-trajectories>, 2025. License: CC-BY-4.0.
- [22] C. Vicentino. “Conditionally-Causal Probes: Five Operational Constraints on Linear-Probe Causality in Qwen3.6-27B.” OpenInterp, 2026. <https://openinterp.org/research/papers/conditionally-causal-probes>.
- [23] C. Vicentino. “Cleaning the Chain-of-Thought Is Not Correcting the Agent.” OpenInterp draft, 2026.