Failing Agents Leave Traces: Telemetry Signatures for Multi-Tier Agent Evaluation

Muthaheera Yasmeena Belgur Shamiullah
Amazon
muthahes@amazon.com

Abstract

Agent evaluation today depends on per-trace LLM-judge inference or human review, too expensive to run on every trace; production systems fall back to sampling a fraction. We find that failing agents leave a detectable behavioral signature in standard observability telemetry: disproportionate effort relative to outcome. We formalize four behavioral failure signatures from agent telemetry, validated on 4,671 traces spanning five agent domains: Compensatory Effort (17–159% more effort on each domain's primary signal, all five domains, $p<0.006$, the dominant signature everywhere tested); Output Inflation (+38% larger patches on SWE-rebench; direction and reliability vary by domain); Unfocused Orchestration (+5.98–12.3% higher token entropy, four of five domains); and Correlation Decoupling (effort-signal correlations shift structurally between pass and fail traces, $p<0.001$, three domains). A simple threshold check over these signals, with no judge, no label, and no model call, already catches 52.9–61.5% of failures. Across three human-labeled benchmarks, judges miss 6.5–18.1% of confirmed failures on the two directly comparable ones, and a trained telemetry classifier (AUROC 0.821) recovers 61.5–75.4% of each judge's misses; combining judge and classifier cuts the miss rate from 18.1% to 4.4%, a 4.1× reduction against the weakest of the eight judges tested (GPT-4o-mini); stronger judges start with fewer misses, so the reduction is correspondingly smaller.

1 Introduction

When an AI agent fails, does it fail silently, or does its execution trace reveal the failure?

We show empirically that it does: failing agents leave a detectable behavioral signature in their execution trace, disproportionate effort relative to outcome, rather than in the response itself. This signature groups into four behavioral failure signatures,

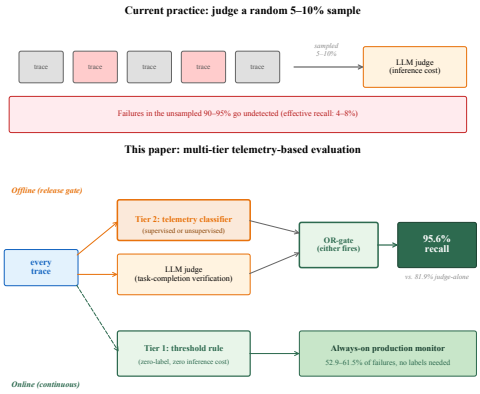

Figure 1: Overview. Top: current practice judges a random 5–10% sample (Pan et al., 2025), so most failures are never evaluated (effective recall ≈ sampling rate × judge recall ≈ 4–8%, illustrative). Bottom: a threshold check over telemetry signals scores *every* trace at zero inference cost, catching 52.9–61.5% of failures with no model call (Section 4); a trained classifier run in parallel with a judge recovers 95.6% of real failures vs. 81.9% judge-alone (Section 4).

each revealing a different way an agent's execution degrades while it continues pursuing its task goal: Compensatory Effort reveals the agent cannot achieve its goal despite increased effort (more tool calls, more tokens); Output Inflation reveals generated artifacts grow larger without improving; Unfocused Orchestration reveals a loss of coherent strategy (effort spreads diffusely across steps rather than concentrating); and Correlation Decoupling reveals structural shifts in how effort signals relate to each other, independent of their magnitude. Sections 2–3 formalize and validate each across five agent domains where its trace structure applies.

This principle has historical precedent: the PARADISE framework (Walker et al., 1997) established that task-level behavioral costs (dialogue turns, repairs, timeouts) predict user satisfaction in spoken dialogue systems, independent of task completion. We extend this to modern agentic sys-

tems, showing that the same class of behavioral cost signals, now observable from standard telemetry, indicates quality issues in LLM agents without requiring explicit evaluation.

All four signatures replicate across five structurally different agent domains (Section 3.1), survive a published task-difficulty confound (Mehtiyev and Assunção, 2026) (Section 3.3), and the multi-signal pattern holds after multiple-testing correction. Section 2 defines each signature; Table 4 and Figure 2 report the full per-domain results.

This further raises the central question: do these signals detect failures LLM judges already miss, or are they redundant with judge-based evaluation? Section 4 answers this: across three independent, human-labeled benchmarks with independent judge verdicts (AgentRewardBench, AJ-Bench, Counsel), the two signals are not redundant, and a telemetry classifier run alongside the judge recovers 61.5–75.4% of what each judge individually misses.

This finding has immediate practical value. Existing evaluation approaches, exhaustive per-trace LLM-judge inference, human review (which Pan et al. (2025) find 74% of production agents rely on primarily), or sampling only a fraction of traces, each trade coverage for cost. This motivates a specific deployment (Section 5) that avoids that trade-off: run the classifier in parallel with the judge, flagging on either signal, for high-stakes release gating; run it alone, as a continuous monitor between judge audits, for standing observability. Telemetry alone is not sufficient, however: the unsupervised threshold rule catches only 52.9–61.5% of failures without a judge (Section 4), and roughly 15% of human-confirmed errors show no effort elevation at all (the agent fails quietly), a structural blind spot for any effort-based method. This is why we recommend the parallel architecture over a telemetry-only deployment.

The result is an evaluation flywheel spanning both regimes: offline, the trained classifier and judge gate releases in parallel (OR-gate: flag a trace if *either* channel fires), cutting the miss rate from 18.1% to 4.4%; online, the zero-label threshold rule monitors every production trace at zero inference cost between judge audits (Section 5).

1.1 Datasets and Trace Structure

Validation covers 4,671 labeled traces spanning five agent domains; Table 1 summarizes each domain's trace structure and scale. Each trace records tool calls, token counts, turns, and generated artifacts: the raw signals the four signatures consume (effort counts, artifact sizes, per-step token distributions, and their pairwise relationships). Core signal extraction requires no access to the agent's internal reasoning, only standard observability telemetry (OpenTelemetry spans).

Dataset	Trace structure	n (p/f)
SWE-rebench	Agent reads issue, navigates repo, edits, tests, submits patch	2,000 (948/1,052)
AppWorld	Agent invokes APIs across services, reports via `finish()`	334 (314/20)
τ-bench	Simulated customer dialogue + tool use (booking, DB, policy)	851 (799/52)
Counsel	Human-reviewed errors on a disjoint τ-bench-derived pool	185 (54/131)
AgentRewardBench	Web-agent trajectories, independent human review	1,301 (355/946)

Table 1: The five validation domains (p/f = pass/fail).

Dataset	How pass/fail is determined
SWE-rebench	Binary `resolved` column from the SWE-bench test harness (Jimenez et al., 2024)
AppWorld	Agent's own `finish(status, answer)` call
τ-bench	Presence of a failed API span (OpenTelemetry status code = error)
Counsel	Independent human review confirming suspected errors
AgentRewardBench	Independent human review (Lù et al., 2025) of every trajectory

Table 2: Label source and derivation by dataset.

Data provenance and labeling. SWE-rebench traces are from Badertdinov et al. (2025), generated with the OpenHands agent (Wang et al., 2024). AppWorld (Trivedi et al., 2024) and τ-bench (Yao et al., 2024) traces span five LLMs and five agent harnesses. Counsel is an independent meta-evaluation dataset of human-confirmed agent errors (Pisupati et al., 2026). AgentRewardBench traces are from Lù et al. (2025). Pass/fail selection ranges from test-verified ground truth (SWE-rebench) to inferred agent output (AppWorld) to independent human review (Counsel, AgentRewardBench); Table 2 summarizes how

each dataset's pass/fail label is determined.

2 Four Behavioral Failure Signatures

If agent failures leave detectable traces in execution telemetry, those traces can serve as an evaluation signal without requiring a judge or a label. We define a *failure signature* as a statistically significant deviation in execution telemetry between successful and failed agent traces. Let $T(i) = \{s_1, \ldots, s_k\}$ be the telemetry signals for interaction i and $y(i) \in \{\text{pass}, \text{fail}\}$ the outcome. A failure signature exists when

$$P(y{=}\text{fail} \mid T \text{ deviates}) > P(y{=}\text{fail}). \quad (1)$$

Signal	Description	Derivation
tool_calls	Count of tool-invocation spans in the trace	Count of spans tagged as tool calls
total_tokens	Total token usage across the trace	Sum of token usage across all spans
patch_length / output_length	Size of the generated output artifact	Length of the final patch/answer artifact
n_spans	Count of execution spans in the trace	Count of all spans (OpenTelemetry)
token_entropy	How uniformly the agent distributes effort across spans	Shannon entropy of per-span token distribution
pairwise_ correlations	Whether relationships between signals shift when agents fail	Spearman ρ between signal pairs, Fisher z-test

Table 3: Telemetry signals used throughout (domain variants: `n_steps`/`assistant_turns`→`n_spans`; `tool_selection_entropy`→`token_entropy`).

Table 3 defines the six canonical signals. Four (`tool_calls`, `total_tokens`, `patch_length`/`output_length`, `n_spans`) are raw counts or sums already present as OpenTelemetry span attributes; `token_entropy` and pairwise correlations are lightweight statistics computed from those same attributes. No access to the agent's model weights, prompts, or chain-of-thought is required. Our validation traces are drawn from the benchmark datasets in Table 1, but the same underlying span attributes are already collected by production observability platforms (Langfuse, LangSmith, Datadog, Amazon Bedrock AgentCore), so practitioners can compute the four raw-count signals directly from their own deployed agents without additional instrumentation; `token_entropy` and pairwise correlations require per-span token counts, which most but not all platforms expose by default.

Compensatory Effort. Effort signals increase while the outcome stays failed: the agent that cannot complete a task does not stop, it keeps trying, and the trying is visible as more tool calls, tokens, or turns. Confirmed in all five domains by Mann-Whitney U tests on each domain's primary signal ($p{<}0.006$, the empirical maximum after correction; +17% to +159%; Section 3.1), the dominant signature by effect size everywhere tested.

Output Inflation. Generated artifacts grow larger without improving. Measured via Mann-Whitney U on artifact size (patch length, output tokens). Confirmed cleanly where a discrete artifact exists (SWE-rebench patches, +38%, $p{<}0.0001$; AgentRewardBench output tokens, +276.8%); its direction and testability vary sharply by domain, including one domain where it reverses (Section 3.1).

Unfocused Orchestration. Effort is distributed uniformly across steps (high token entropy) rather than concentrated where it matters. Confirmed in four of five domains (+5.98% to +12.3%); the fifth (AppWorld) is underpowered at $n{=}20$ failing traces rather than contradictory. Shannon entropy over k steps is bounded above by $\log_2(k)$, so step-count normalization is required before deployment (Section 3.1).

Correlation Decoupling. Relationships *between* signals shift when agents fail, independent of magnitude: failing agents lock into tighter effort-signal couplings. Measured via Spearman ρ computed separately on pass/fail subgroups, with the difference tested by Fisher z-transformation (e.g., SWE-rebench `tool_calls`×`patch_length` ρ:0.29→0.44, $p{=}0.0001$). Confirmed on SWE-rebench, AppWorld, and τ-bench ($p{<}0.001$); requires a matched pass/fail population sample (Section 3.1). This signature provides the strongest evidence against a pure task-difficulty confound (Section 3.3): difficulty increases signal *magnitude* but does not restructure signal *relationships*.

Together, these signatures share a property: they detect quality degradation in the execution trace rather than the response itself. Compensatory Effort reveals the agent cannot achieve its goal despite increased effort; Output Inflation

reveals artifacts grow without improving; Unfocused Orchestration reveals loss of coherent strategy; Correlation Decoupling reveals structural behavioral shifts. They differ, however, in how broadly they generalize: Compensatory Effort is the most consistent, confirmed in the same direction across all five domains, whereas Output Inflation, Unfocused Orchestration, and Correlation Decoupling are more domain-dependent (Output Inflation even reverses direction on Counsel) and should be interpreted per domain rather than assumed universal.

A boundary case within Counsel: under-investment. All four signatures above are *over*-effort patterns. Counsel's full human review surfaces the opposite: 20 of 131 human-confirmed real errors (15%) occur in traces *below* the population median on all three effort signals simultaneously (tool calls, assistant turns, and tool-selection entropy). This under-investment pattern, where the agent tries too little rather than too much, is structurally invisible to every signature validated here; we revisit it in Limitations, where the effort-based framing runs out.

3 Empirical Validation of Telemetry Based Failure Signatures

This section reports the per-domain results (Section 3.1), the statistical guardrails establishing that the observed effects are unlikely due to chance (Section 3.2), and a test of the strongest alternative explanation, task difficulty (Section 3.3). All findings are correlational: effort increase is a detectable symptom of failure, not a cause, and the paper's claims are detection claims throughout.

3.1 Results Across Five Domains

We test each signature independently on every domain: per-signal group comparisons (does the signal differ between passing and failing traces?) use Mann-Whitney U tests with Cohen's d effect sizes; correlation shifts (does the relationship *between* signals change on failure?) use Fisher z-tests on Spearman ρ; confirmed signals are then combined into a supervised classifier per domain (logistic regression, as a conservative lower bound on achievable performance). Each domain's *primary signal* is the telemetry signal with the largest confirmed effect for that domain: `tool_calls` for SWE-rebench and τ-bench, `total_tokens` for AppWorld and AgentRewardBench, `n_spans` (assistant turns) for Counsel; Figure 2 labels each bar with its measured signal so effect sizes are compared on the correct unit. Table 4 and Figure 2 summarize the results across all five domains.

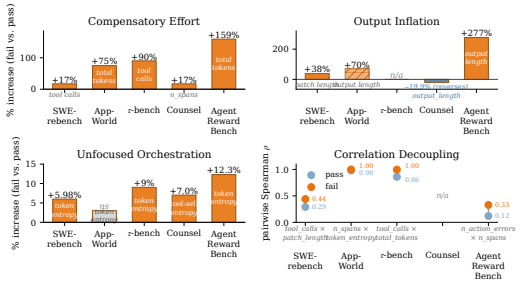

Figure 2: All four failure signatures validated across five domains. Compensatory Effort (top-left) is confirmed everywhere (+17% to +159%); the remaining three vary by domain. Each bar labels its measured telemetry signal; the bottom-right panel shows the pass-vs.-fail correlation shift for Correlation Decoupling.

Signature	SWE-rebench	AppWorld	τ-bench	Counsel	Agent-RewardBench
Compensatory Effort	+17%	+75%	+90%	+17%	+159%
Output Inflation	+38%	+70%	n/a	−19.9%	+277%
Unfocused Orchestration	+5.98%	ns	+9%	+7.0%	+12.3%
Correlation Decoupling	confirmed	confirmed	confirmed	†	partial

Table 4: Cross-domain results (fail-vs.-pass % on primary signal). ns: not significant (AppWorld entropy, p=0.34, underpowered at n_{fail}=20). n/a: not testable (τ-bench Output Inflation: harness confound). †: tested but excluded after a confound check (Counsel Correlation Decoupling: judge-flagged sampling violates the matched-population requirement).

Appendix C reports each domain's complete per-signal breakdown, multi-signal analysis, and root-cause interpretation. The cross-domain picture: Compensatory Effort is the one signature confirmed in all five domains ($p<0.006$, the dominant signature by effect size everywhere tested), consistent with a common mechanism across the domains tested: an agent that cannot complete its task keeps spending effort without achieving its goal. The remaining three signatures vary by domain, reflecting differences in trace structure, artifact type, and sampling design.

Key findings per domain:

SWE-rebench (n=2,000, test-verified labels): the cleanest validation, all four signatures confirmed by Mann-Whitney U on resolved vs. unresolved traces, including Output Inflation (+38% larger patches) and Correlation Decoupling (ρ:0.29→0.44, Fisher z). A multi-signal classifier reaches AUROC 0.660; full ROC curves appear in Figure 3 (Section 4). Observation: unresolved

traces enter a characteristic edit-test-fail loop, producing patches 23× larger while consuming 2× the tokens (Appendix D).

AppWorld (n=334, n_{fail}=20, labels from the agent's `finish()` call): the weakest domain due to only 20 failing traces. Compensatory Effort confirmed (+75% tokens, Mann-Whitney U), but entropy and tool-call differences do not reach significance, consistent with underpowering rather than absence of effect. Observation: `tool_calls` actually *decreases* for failures (-33%, p=0.066), suggesting failing agents stall early rather than retry.

τ-bench (n=851, labels from failed API spans): three signatures confirmed by Mann-Whitney U (pooled across three sub-domains), with Correlation Decoupling (Fisher z) replicating in 2 of 3 sub-domains (retail and telecom; airline underpowered at n_{fail}=6). No classifier is trained on this domain. Observation: 98.1% of failed API spans are terminal in their trace, ruling out a retry-driven confound for the +90% tool-call increase.

Counsel (n=185, human-reviewed labels, Bonferroni-corrected): Compensatory Effort and Unfocused Orchestration confirmed on human-confirmed errors (Mann-Whitney U, all $p<0.006$). A 3-feature classifier reaches AUROC 0.656. Output Inflation reverses direction (-19.9%, responses are *shorter* when failing). Observation: 15% of confirmed errors (20/131) fall *below* the population median on all effort signals simultaneously, an under-investment pattern invisible to over-effort signatures.

AgentRewardBench (n=1,301, independent human review): the largest human-labeled domain, all four signatures measurable (Mann-Whitney U). A 7-feature classifier (logistic regression) reaches AUROC 0.821 (0.814 under 5-fold CV). Entropy is step-normalized (d=0.67); Correlation Decoupling is partial (1 of 6 signal pairs survives a step-cap control via Fisher z). Observation: over half of all trajectories hit the 31-step cap, inflating naive correlation shifts; only `n_action_errors`×`n_spans` reflects genuine behavioral coupling after excluding capped traces.

3.2 Statistical Methodology

The per-domain findings above could be artifacts of underpowering, multiple testing, or post-hoc family selection. We assess each potential confound below; every claimed signature remains significant under correction.

Statistical power. Table 5 reports the primary-signal effect size and achieved power for all five domains. Three are well powered (>99%–100%); AppWorld (77%, n=20 failing traces) and Counsel (79%, judge-flagged sample) carry wide confidence intervals that bound how much weight their estimates can carry.

Dataset	d	95% CI	Power
SWE-rebench	0.56	[0.47, 0.65]	>99%
AppWorld	0.63	[0.17, 1.08]	77%
τ-bench	0.66	[0.38, 0.95]	100%
Counsel	0.45	[0.13, 0.77]	79%
AgentRewardBench	1.07	[0.94, 1.20]	>99%

Table 5: Three domains are well powered (>99%); AppWorld and Counsel carry wide confidence intervals due to limited failing traces. Cohen's d on each domain's primary effort signal.

Multiple testing correction. Running many hypothesis tests inflates the chance that some appear significant by luck; every independent test underlying a claimed signature is therefore submitted to correction, and every one survives it. The 13 tests from the three original domains (SWE-rebench, AppWorld, τ-bench) form one Benjamini–Hochberg false-discovery-rate (FDR) family: 11 of 13 survive at adjusted $p<0.05$, and the two that do not (AppWorld `n_spans`, `token_entropy`) are reported non-significant and excluded from our claims. AppWorld `tool_calls` (p=0.066, wrong direction) was never claimed as evidence for any signature.

Correction for Counsel and AgentRewardBench. These later domains are treated as separate replication families: Counsel clears Bonferroni ($\alpha/3$=0.017; all $p<0.006$); AgentRewardBench's entropy test clears α=0.05 as a one-test family ($p<0.000001$). As a robustness check, pooling all 17 tests into a single FDR family leaves every replication test significant. The net result: no claimed signature in this paper depends on which correction family we chose.

3.3 Controlling for Task Difficulty

Mehtiyev and Assunção (2026) show that effort–failure correlation reverses once task difficulty is controlled (Related Work). We test this on SWE-rebench using an independent difficulty proxy (historical resolve rate). Stratifying into four quartiles, our multi-signal classifier retains AUROC 0.631–0.647 within the two class-balanced middle

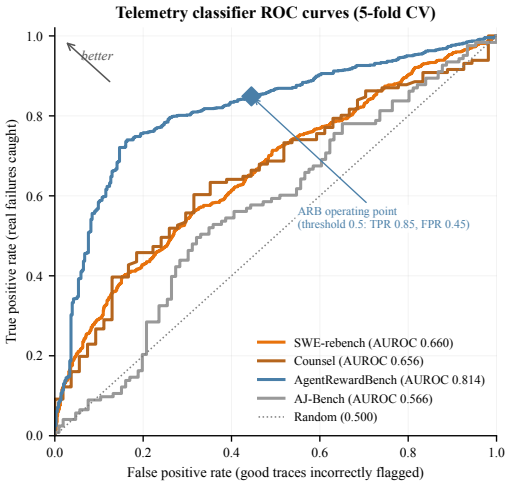

Figure 3: ROC curves for the telemetry classifier on four domains (5-fold cross-validated); closer to the top-left corner means stronger separability, the diagonal is chance. The marked point on the AgentRewardBench curve is the threshold-0.5 operating point (TPR 0.85, FPR 0.45) used in the OR-gate of Section 4.1. AgentRewardBench discriminates strongest (0.814), followed by SWE-rebench (0.660) and Counsel (0.656); AJ-Bench (0.566) is near chance: signal strength varies substantially by domain.

tiers: the signal survives difficulty control. The clearest difficulty-independent evidence is Correlation Decoupling: under a pure difficulty confound, correlation *structure* should be identical across pass/fail groups; we observe the opposite (ρ:0.29→0.44, p=0.0001), which a magnitude-only confound cannot produce.

4 Telemetry Classifier vs. LLM Judges

Telemetry alone is insufficient. Using the unsupervised $p75$ threshold rule (flag if any signal exceeds the population 75th percentile, no labels required), union coverage is only 52.9–61.5% across the three natural-population domains (Table 6). Roughly half of real failures produce no elevated-effort signature — the agent fails without visibly struggling. A second evaluation channel is required.

Dataset	n (fail)	Union recall	Gap
SWE-rebench	1,052	52.9%	47.1%
AppWorld	20	55.0%	45.0%
τ-bench	52	61.5%	38.5%

Table 6: The zero-label threshold rule catches 52.9–61.5% of real failures with no model and no labels, leaving 38.5–47.1% undetected.

We cross-tabulate a trained telemetry classifier against eight LLM judges on two human-labeled benchmarks (AgentRewardBench, n=946 failures, seven judges; AJ-Bench, n=123 failures, one judge), asking: does the judge already catch what telemetry catches, or are they blind to different failures? The two channels are blind to *different* failures. Each judge misses 6.5–18.1% of confirmed failures; the telemetry classifier recovers 61.5–75.4% of those misses. Running both in parallel (OR-gate) raises recall from 81.9% to 95.6%, a 4.1× miss-rate reduction. Telemetry detects effort-based failures (the agent struggles visibly but the judge overlooks it); judges detect content-based failures (the agent answers confidently wrong with normal effort). Neither alone covers both. Below, Section 4.1 presents the primary evidence and Section 4.2 tests generalization across judges and benchmarks.

4.1 Primary Evidence: AgentRewardBench

Measuring what a judge misses requires traces carrying independent human ground truth, an LLM-judge verdict, and telemetry simultaneously. AgentRewardBench (Lù et al., 2025) meets this requirement: n=1,301 (946 failures, 355 successes) with eight judge verdicts spanning five model families. We train a supervised classifier (logistic regression, chosen for interpretability; any supervised or unsupervised model scoring these features can fill the same role) with features standardized via z-scoring (decision threshold 0.5) on seven telemetry features: `n_spans`, `total_input_tokens`, `total_output_tokens`, `total_retries`, `n_action_errors`, `avg_reasoning_len`, and `unique_actions`. This achieves AUROC 0.821 (5-fold CV: 0.814); `total_tokens` alone reaches 0.787 without any model (Appendix A), but removing token features drops CV AUROC only to 0.810, so the remaining five features carry independent signal. All OR-gate results are essentially unchanged under cross-validated scores. Figure 3 shows full ROC curves per domain; Figure 4 shows the precision-recall trade-off.

What does each quadrant look like? All 946 traces below are failures confirmed by AgentRewardBench's independent human annotators, who reviewed every trajectory against its task goal. We cross-tabulate each trace by (a) whether the classifier flags it at the 0.5 probability threshold and (b) whether the GPT-4o-mini judge verdict says

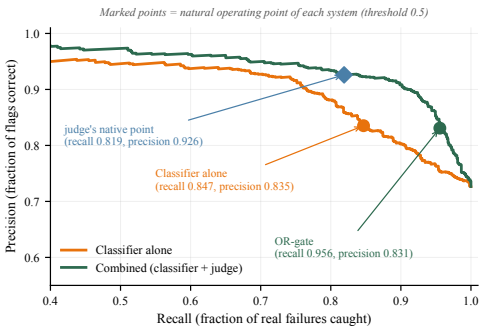

Figure 4: Combining telemetry with the judge trades precision for recall along a favorable frontier. On AgentRewardBench (n=1,301), the combined curve (green) dominates the classifier alone (orange) at every recall level, and matches the judge's own precision (0.926) at its native recall (0.819); pushing recall to 0.956 lowers precision from 0.926 to 0.831. The gain is a recall/precision trade-off, not a precision-neutral improvement.

"Unsuccessful." The 2×2 breakdown (Figure 5A) reflects two distinct failure modes visible in the telemetry:

- *Classifier catches, judge misses* (129 traces): high-effort failures, median 160K tokens and 31 steps (the dataset's step cap). A confound check: retraining the classifier without `n_spans` (removing the step-count feature entirely, 6 remaining features), the OR-gate recall is 96.4% and recovery 80.1%, both equal to or above the headline result. The step-cap is not inflating the classifier's value; the remaining token and error features carry the signal independently (the improvement on removal is consistent with `n_spans` adding noise via ceiling censoring while the bivariate correlation shift it participates in, Section C.5, reflects a genuine structural relationship visible below the cap).
- *Judge catches, classifier misses* (103 traces): low-effort failures, median 21K tokens and 5 steps, where the agent stopped early but was confidently wrong. The judge detects the content error; telemetry sees no anomalous effort.
- *Neither catches* (42 traces): low-effort, low-risk-score failures (median 26K tokens, 5 steps) where the agent failed quietly with no effort spike and no content flag, the irreducible blind spot at this operating point.

The remaining 672 traces (71.0%) are caught by both channels. The pattern clarifies *why*

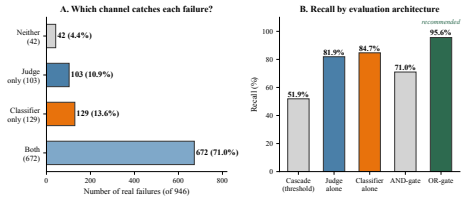

Figure 5: A: Neither evaluation channel subsumes the other: of 946 human-confirmed failures, 13.6% are caught only by the classifier and 10.9% only by the judge. B: The parallel OR-gate reaches 95.6% recall, dominating every single-channel and the cascade alternative (51.9%).

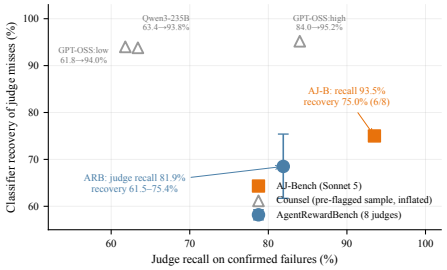

Figure 6: Classifier recovery of judge misses: 61.5–75.4% across seven LLM judges and two benchmarks (programmatic judge excluded at 42.9%; Counsel's 93–95% inflated by pre-flagged sampling).

the two are complementary: telemetry catches effort-based failures (loops, retries); judges catch content-based failures (confidently wrong with normal effort). Neither alone covers both (Figure 5A). A cascade (pre-filter before the judge) is not viable: the $p75$ rule recovers only 51.9% of real errors on Counsel, so a pre-filter would drop nearly half before the judge sees them.

Figure 5B compares architectures. The OR-gate combines two binary flags per trace: the classifier's failure probability thresholded at $p \geq 0.5$, and the judge's binary verdict; a trace is escalated if *either* fires (`flag` = `classifier` ∨ `judge`). This reaches 95.6% recall, against 81.9% judge-alone, 84.7% classifier-alone, and 71.0% AND-gate (requires *both*). The combined operating point matches the judge's own precision (0.926) at the judge's native recall (0.819) and dominates the classifier-alone curve at every recall level (Figure 4); pushing recall to 0.956 is a recall/precision trade-off, lowering precision from 0.926 to 0.831 rather than leaving it unchanged. At that operating point, 51.5% of good traces are also flagged; Section 5 discusses this trade-off, and Table 8 reports its sensitivity to failure prevalence.

4.2 Generalization: Across Judges, AJ-Bench, and Counsel

The primary evidence uses one judge (GPT-4o-mini) on one dataset. We test whether the non-redundancy generalizes along two axes. *Across judges*: repeating on all eight AgentRewardBench judges, the classifier recovers 61.5–75.4% of each LLM judge's misses (median 66.4%, n=7 LLM judges; the eighth, a programmatic-reward judge, is excluded from this range at 42.9% because it operates on functional correctness signals orthogonal to telemetry); stronger judges leave a smaller gap (r=−0.64, n=7, not significant), and a frontier-judge check (Claude Sonnet 5, 150-trajectory subsample, 89.3% accuracy) confirms the pattern. *Across benchmarks*: we repeat on two further datasets (Table 7): AJ-Bench (Shi et al., 2026), a structurally unrelated database/filesystem benchmark (n=229, 123 real failures, no overlap with AgentRewardBench), with Claude Sonnet 5 as judge (Bedrock model `us.anthropic.claude-sonnet-5`, verdicts collected 2026-07-17 and frozen at collection) and a three-feature classifier (`tool_calls`, `total_tokens`, `n_spans`); and Counsel (n=185, Section 2), whose three judge models were each meta-reviewed by humans, giving a per-judge miss count on human-confirmed errors.

Dataset / Judge	Recall	Precision	Telemetry Classifier Recovery
AgentRewardBench (n=946 failures)			
GPT-4o-mini	81.9%	92.6%	75.4%
GPT-4o (no ax-tree)	86.5%	92.1%	63.3%
Claude 3.7 Sonnet	88.1%	90.7%	66.4%
Llama 3.3 70B	86.7%	92.0%	66.7%
Qwen 2.5-VL 72B	83.0%	93.9%	66.5%
AER	87.0%	90.0%	61.8%
AERV	87.6%	89.3%	61.5%
Functional (prog.)	96.3%	85.5%	42.9%
AJ-Bench (n=123 failures)			
Claude Sonnet 5	93.5%	59.0%	75.0%
Counsel† (n=131 errors)			
GPT-OSS-120B:low	61.8%	—	94.0%
GPT-OSS-120B:high	84.0%	—	95.2%
Qwen3-235B	63.4%	—	93.8%

Table 7: Per-judge recall, precision, and telemetry classifier recovery of judge misses across three datasets. †Counsel: pre-flagged sample, so recall is Error Confirmation Rate and recovery (93–95%) is inflated.

Figure 6 plots judge recall against classifier recovery across all three datasets. The pattern holds: regardless of how accurate the judge is, the classifier recovers a substantial fraction of its misses (61.5–75.4% on the two directly comparable datasets). AJ-Bench's judge flags excessively (precision 59.0% vs. AgentRewardBench's 92.6%), so its 93.5% recall leaves only 8 misses; the classifier recovers 6 of 8 (75.0%, directionally consistent but not independently well-powered at n=8). The OR-gate on AJ-Bench reaches 99.2% recall but at 57.3% precision, a trade-off acceptable only in high-stakes screening where missed failures dominate false-alarm costs. The classifier is weaker on this domain (cross-validated AUROC 0.566 vs. 0.821 on web agents): telemetry signal strength varies by benchmark. Counsel's recovery (93–95%) is inflated by the classifier's aggressive operating point (Table 7 caption). Better judges do not eliminate the telemetry signal's value; they change *which* failures it recovers, not *whether* it recovers them.

5 Deployment and Conclusion

Multi-tier evaluation design. Two tiers form an evaluation flywheel, both at zero runtime LLM-inference cost (feature extraction, thresholding, and monitoring infrastructure still apply). *Tier 1 (unsupervised, always-on)*: flag traces exceeding the population $p75$ on any applicable signal; catches 52.9–61.5% of failures with no labels, surfacing candidates for Tier 2. *Tier 2 (offline, gates releases)*: telemetry classifier + judge in parallel (OR-gate), 81.9%→95.6% recall. The classifier may be supervised (logistic regression here) or unsupervised (e.g., isolation forest) where labels are unavailable; the threshold (0.5 here) should be tuned per team's quality bar (Figure 4). The flywheel closes as Tier 1's candidates become review labels that improve Tier 2, whose coverage in turn refines Tier 1's thresholds.

Conclusion. Failing agents do leave traces. Four behavioral failure signatures, detectable from standard observability telemetry without any model call, generalize across five structurally different agent domains and complement LLM judges on a dimension judges cannot cover alone. Combined in a parallel OR-gate, telemetry and judge together raise failure-detection recall from 81.9% to 95.6%, a 4.1× reduction in missed failures relative to the weakest judge tested (GPT-4o-mini; the reduction is smaller for stronger judges, which miss less to begin with), with the telemetry classifier adding zero runtime LLM-inference cost beyond the judge already in use.

Related Work

Agent evaluation and behavioral signals. Pan et al. (2025) find 74% of production agents rely on human evaluation; Ndzomga (2026) applies Item Response Theory for cost-efficient benchmarking; ours needs no model inference. Mehtiyev and Assunção (2026) show trajectory-length/failure correlation reverses under difficulty control (9,374 trajectories); it holds for single signals here but the multi-signal effect survives. David and Gervais (2025) find tool usage and cost correlate negatively with success in penetration-testing agents, consistent with Compensatory Effort.

Tool use and evaluator validity. Schick et al. (2023) established tool-calling; Vaghasiya et al. (2026) audit judges, finding an 18.5% evaluator-human misalignment rate. We bypass the judge: tool-use *pattern* predicts outcome directly (AUROC 0.821).

LLM judges, drift, and monitoring. Zheng et al. (2023) established judge/human-preference correlation and Lightman et al. (2024) process reward models; Krumdick et al. (2025) show judges fail on unanswerable questions, which our OR-gate closes. Lù et al. (2025) provide our main benchmark, Badave et al. (2026) confirm hallucination detection needs execution-quality flags, and Shi et al. (2026) propose Agent-as-a-Judge (AJ-Bench). Shukla (2025), Rath (2026), and Wang (2026) monitor agents; across five domains, telemetry recovers the failures judges miss.

Limitations

Behavioral telemetry does not detect hallucination. Every signature is an effort-outcome imbalance measure; a fluent, confidently wrong response with normal effort produces no detectable deviation. Counsel's under-investment boundary case is one concrete instance: 20 of 131 confirmed errors (15%) sit below the population median on all three effort signals, and the trained classifier's 4 false negatives at its 0.5 threshold average roughly half the median effort on every signal. Badave et al. (2026) independently confirm on a different domain that hallucination detection requires execution-quality flags, not raw telemetry.

The AgentRewardBench entropy effect is confounded with step count; we report the corrected number. Raw token entropy yields d=1.196, but entropy and `n_spans` correlate at r=0.96 (Shannon entropy over k spans is bounded above by $\log_2(k)$), so much of this effect restates the separately-reported `n_spans` Compensatory Effort result. The step-normalized effect (d=0.668) is smaller but still significant and is the number we treat as primary throughout.

τ-bench's label and Compensatory Effort signal could be mechanically related; we tested this and did not find the mechanism. τ-bench's failure label is a failed API span, and Compensatory Effort is measured via `tool_calls`; if a failed call triggered a visible retry, part of the effect could be a labeling artifact. We checked: 51 of 52 failed spans (98.1%) are terminal in their trace; excluding tool-call spans immediately following a failed span leaves the effect numerically unchanged (+90%, d=0.66). This does not rule out circularity in a harness with visible retry loops, but here the effect is not driven by retries inflating the count.

Scope of detection. Low-effort traces are undetectable (AUROC ≈0.50 when effort is uniformly low). Compensatory Effort and Unfocused Orchestration operate per-trace; Correlation Decoupling is system-level. Effort ≠ failure always: hard tasks legitimately require more effort, and the multi-signal approach retains but does not eliminate this confound (Section 3.3). All findings are correlational.

Generalization. No cross-dataset prediction beyond the transfer test in Appendix A: only total token consumption transfers without retraining; practitioners should expect to re-derive most domain-specific features.

OR-gate false-positive cost. At the recommended operating point, 51.5% of genuinely good traces (183/355) are flagged for review; this review burden is acceptable only in high-stakes release gating where missed regressions dominate false-alarm costs. Production deployments with lower tolerance for review volume would need threshold tuning (Figure 4 shows the full precision-recall trade-off) or a triage stage.

Sensitivity to failure prevalence. The precision figures for the OR-gate, classifier, and judge (Section 4) are computed at AgentRewardBench's 72.7% failure base rate; production deployments typically run far lower (single-digit to low-tens-of-

	FPR	Precision at failure prevalence			
Operating point		72.7%	10%	5%	1%
Judge alone	17.5%	0.926	0.343	0.198	0.045
Classifier alone	44.5%	0.835	0.175	0.091	0.019
OR-gate	51.5%	0.831	0.171	0.089	0.018

Table 8: Projected precision as the failure base rate falls, holding each system's recall and false-positive rate fixed at its AgentRewardBench operating point (basis n=355 successes; the p=72.7% column reproduces the paper's reported precision). At fixed recall and FPR, only precision erodes as prevalence falls.

percent, illustrative). Recall and FPR are, by construction, independent of the success-to-failure ratio; precision is not. Holding each system's (recall, FPR) fixed and re-projecting (Table 8), the OR-gate's 83% precision falls to ≈17% at 10% prevalence and 9% at 5%—about five to ten false alarms per true failure caught—while the judge alone, at a lower FPR (17.5% vs. 51.5%), degrades more gracefully (34% at 10%). This is a calibration projection: it assumes the operating point's recall and FPR, measured on AgentRewardBench, hold on a differently-distributed production stream, which is not guaranteed—if production failures skew lower-effort, real precision could be lower still. This reinforces the deployment split: the OR-gate is for release gating, not always-on monitoring, which is Tier 1's role. Tier 1's healthy-trace flag rate grows with the number of signals OR'd and should be calibrated per deployment; we report its recall (Table 6) but no single-number precision, as a label-free rule has no per-trace ground truth.

Evidence strength. Severity-weighted recall is untested: AgentRewardBench provides only binary labels, and if the classifier's unique catches are systematically lower-severity than the judge's, the OR-gate's practical value is smaller than the headline recall suggests, the most important open question for future work. Frontier-judge evidence is limited on both datasets: the AgentRewardBench subsample has only 5 misses in 150 trajectories, and AJ-Bench's frontier judge (Claude Sonnet 5) leaves only 8 misses because it flags most traces as failures; the classifier recovers 6 of 8 (75.0%), directionally consistent with the main result but not independently well-powered. Judge verdicts on AJ-Bench are frozen from a single collection run; live judge behavior varies across runs, which is itself a reproducibility hazard for judge-based evaluation that telemetry signals do not share.

Label quality. SWE-rebench uses test-verified labels (gold standard); AppWorld uses `finish()`-output inference (n=20 failures, least reliable, and this small failing-trace count limits every AppWorld estimate in this paper, not only the non-significant ones: even the confirmed Compensatory Effort effect carries a wide 95% CI, d=0.63 $[0.17, 1.08]$); τ-bench uses failed-API-span presence, and its Output Inflation test is additionally blocked by a harness-instrumentation gap unrelated to sample size (Section 2); AgentRewardBench uses independent human review, the most reliable source, where all four signatures show a measurable effect, though Output Inflation's artifact proxy and five of Correlation Decoupling's six signal pairs require controlling for a step-count censoring confound before the effect is trustworthy (Section C.5). Counsel's review targets judge-flagged spans, so its fail rate is not a natural base rate, and the same sampling bias is why we do not report Correlation Decoupling there even though a naive scan finds a marginal pair.

Ethical Considerations

All datasets are public research benchmarks (SWE-rebench, AppWorld, τ-bench, AgentRewardBench) or an independent meta-evaluation dataset released for research use (Counsel); no production or proprietary telemetry is used. The intended application is system health monitoring and quality screening, not automated decisions about individual users or a wholesale replacement of human/judge evaluation where stakes are high (Section 4.2). Behavioral monitoring could theoretically be repurposed for worker surveillance; our method analyzes system-level distributions, not individual human performance, and we caution against repurposing it for employee monitoring without consent and governance. With per-trace classifier AUROC ranging 0.566–0.821 across tested domains and an irreducible 4.4% blind spot even combined with a judge, substantial false positives/negatives exist; deployment should use these signals for screening and escalation, not autonomous quality verdicts. Counsel's raw traces remain governed by their original release terms (Pisupati et al., 2026) and are not redistributed.

References

Harshada Badave, Santosh Borse, Andrea Gomez, Harshitha Narahari, Sara Carter, Vishwa Bhatt, Ais-

hani Rachakonda, Shuxin Lin, and Dhaval Patel. 2026. Beyond final answers: Auditing trajectory-level hallucinations in multi-agent industrial workflows. *arXiv preprint arXiv:2605.24219*.

Ibragim Badertdinov et al. 2025. Swe-rebench: An automated pipeline for task collection and decontaminated evaluation of software engineering agents. *arXiv preprint arXiv:2505.20411*.

Isaac David and Arthur Gervais. 2025. Multi-agent penetration testing ai for the web. *arXiv preprint arXiv:2508.20816*.

Carlos E. Jimenez et al. 2024. SWE-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. No free labels: Limitations of LLM-as-a-judge without human grounding. *arXiv preprint arXiv:2503.05061*.

Hunter Lightman et al. 2024. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. 2025. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories. *arXiv preprint arXiv:2504.08942*.

Tural Mehtiyev and Wesley Assunção. 2026. Beyond resolution rates: Behavioral drivers of coding agent success and failure. *arXiv preprint arXiv:2604.02547*.

Franck Ndzomga. 2026. Efficient benchmarking of ai agents. *arXiv preprint arXiv:2603.23749*.

Melissa Z. Pan et al. 2025. Measuring agents in production. *arXiv preprint arXiv:2512.04123*.

Sashank Pisupati, Henry Broomfield, Eujeong Choi, Antonia Calvi, Charlie Wang, Roman Engeler, Max Bartolo, and Patrick Lewis. 2026. Counsel: A meta-evaluation dataset for agentic tasks. *arXiv preprint arXiv:2606.21627*.

Abhishek Rath. 2026. Agent drift: Quantifying behavioral degradation in multi-agent LLM systems over extended interactions. *arXiv preprint arXiv:2601.04170*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wentao Shi, Yu Wang, Yuyang Zhao, Yuxin Chen, Fuli Feng, Xueyuan Hao, Xi Su, Qi Gu, Hui Su, Xunliang Cai, and Xiangnan He. 2026. Aj-bench: Benchmarking agent-as-a-judge for environment-aware evaluation. *arXiv preprint arXiv:2604.18240*.

Manish Shukla. 2025. Adaptive monitoring and real-world evaluation of agentic ai systems. *arXiv preprint arXiv:2509.00115*.

Harsh Trivedi et al. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *arXiv preprint arXiv:2407.18901*.

Jay Vaghasiya, Vishvesh Bhat, Muhammad Ahmed Mohsin, and Asad Aali. 2026. Benchmarking the benchmarks: A validity audit of tool-calling evaluation. *arXiv preprint arXiv:2607.02577*.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chunxiao Wang. 2026. Nautilus compass: Black-box persona drift detection for production llm agents. *arXiv preprint arXiv:2605.09863*.

Xingyao Wang et al. 2024. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*.

Shunyu Yao et al. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Appendix: Cross-Domain Feature Transfer

Does a classifier fit on SWE-rebench's telemetry schema generalize to a domain it was never trained on, without retraining? Our 7-feature classifier (AUROC 0.660 under 5-fold cross-validation on SWE-rebench) maps only `total_tokens` cleanly onto AgentRewardBench's schema; `patch_length`/`avg_observation_length` have no analog, `truncation_ratio` is constant, and the rest require lossy proxies. Applied via these proxies it yields AUROC 0.788; `total_tokens` alone, with no model, already achieves 0.787, so the apparent multi-signal transfer is attributable to a single signal. A classifier trained fresh on AgentRewardBench's own labels reaches AUROC

0.821 (+0.033), confirming retraining still adds value where one signal transfers.

B Appendix: Statistical Notation

p: probability of the observed effect under the null hypothesis (Mann–Whitney U for group comparisons, **Fisher z-test** for correlation differences); we treat $p<0.05$ as significant after correction. **AUROC**: 0.5=random, 1.0=perfect. **Cohen's d**: standardized effect size (0.2/0.5/0.8 = small/medium/large). **Spearman's ρ**: rank correlation, -1 to $+1$. **Recall/precision**: fraction of real failures flagged / fraction of flagged traces that are real failures. **p75**: 75th percentile of a signal over the full observed trace population (all traces, not pass-only, so it needs no labels), used as a per-trace threshold. **FDR**: expected proportion of false positives among rejected nulls under Benjamini–Hochberg correction.

C Appendix: Per-Domain Validation Detail

Complete per-domain analysis summarized in Section 3.1 and Table 4.

Having established the statistical methodology above, we now validate the four failure signatures across five agent domains. For each dataset, we describe the setup, measure which signatures are observable on a per-signal basis, test for multi-signal patterns, interpret the underlying root cause, and summarize what was and was not confirmed.

C.1 Coding Agents (SWE-rebench)

Dataset and Setup. Modality: single-agent, text-to-code. The agent reads an issue description, navigates the codebase, and produces a patch. Complexity: multi-step reasoning (median 61 steps per trajectory), repository-scale code navigation, test-verified correctness. Scale: 2,000 trajectories (Badertdinov et al., 2025) (SWE-bench task format (Jimenez et al., 2024)) generated with the OpenHands agent (Wang et al., 2024), 948 resolved / 1,052 unresolved. Source: `nebius/SWE-rebench-openhands-trajectories` (HuggingFace), via a third-party curated random 2,000-trajectory subset.

Per-Signal Analysis. Each telemetry signal is compared independently between resolved and unresolved traces using Mann–Whitney U tests (Table 9). `tool_calls` and `total_tokens` (Compensatory Effort) and `patch_length` (Output Inflation) are all significant at $p<0.0001$ and survive FDR correction. Unfocused Orchestration, `token_entropy` 4.80 vs. 5.09 (+5.98%[1], $p\approx0$), is extracted separately from the raw per-message trajectory logs (not present in the pre-aggregated feature set used for the other three signals) and cross-validated against the labeled sample's `resolved` field at 100% match before use. Per-signal detection uses the $p75$ thresholds defined in Appendix B: flag traces where `tool_calls`, `total_tokens`, or `patch_length` exceed that percentile.

Signal	Res.	Unres.	Δ
tool_calls	58.9	68.8	+17%
total_tokens	43,369	51,105	+18%
patch_length	9,241	12,795	+38%

Table 9: SWE-rebench (all $p<0.0001$). tool_calls/tokens=Compensatory Effort, patch_length=Output Inflation.

Multi-Signal Analysis. Correlation Decoupling (Fisher z-test): inter-signal relationships shift between resolved and unresolved groups. `tool_calls`×`patch_length`: ρ:0.29→0.44 (p=0.0001); `tokens`×`observation_length`: ρ:0.20→0.05 (p=0.0007). A combined classifier (logistic regression, as a simple baseline, on seven features derived from the signals in Table 3: `tool_calls`, `total_tokens`, `patch_length`, `assistant_turns`, `avg_observation_length`, `num_observations`, `truncation_ratio`) reaches AUROC 0.660 (62.9% recall vs. 33.9% for the best single threshold; top-quartile flagging achieves 75% precision at 36% recall), and 0.631–0.647 within independent difficulty tiers (Section 3.3), inconsistent with a pure task-difficulty explanation.

Root Cause Interpretation. Failing agents make more tool calls and produce larger patches without resolving the issue, and tool usage/patch size become more tightly coupled, while succeeding agents keep these signals loosely correlated. Table 12 (Appendix) contrasts two real traces from the same agent on comparable repositories: the resolved trace (tool_calls=52, total_tokens=25,498,

[1]Computed from full-precision means (4.8001, 5.0872) before rounding for display; recomputing the delta from the rounded 4.80/5.09 gives +6.04%, the same display-rounding artifact noted for the AgentRewardBench entropy figure (Section C.5).

patch_length=856 chars) reads target files, makes a focused edit, validates, and submits; the unresolved trace (tool_calls=74, total_tokens=49,411, patch_length=19,982 chars) enters a repeated edit-test-fail loop, producing a patch roughly 23× larger while consuming about twice the tokens, a concrete instance of Compensatory Effort and Output Inflation at the level of individual steps.

Validation Summary. Per-signal: Compensatory Effort (+17% tool calls, +18% tokens) and Output Inflation (+38% patches) confirmed ($p<0.0001$); Unfocused Orchestration confirmed (+5.98% entropy, $p\approx 0$). Multi-signal: Correlation Decoupling confirmed ($p=0.0001$), and the combined classifier achieves AUROC 0.660 on the full dataset (0.631–0.647 within independent difficulty tiers). Statistical power is high: $d=0.56$, 95% CI $[0.47, 0.65]$, power >99% (Table 5).

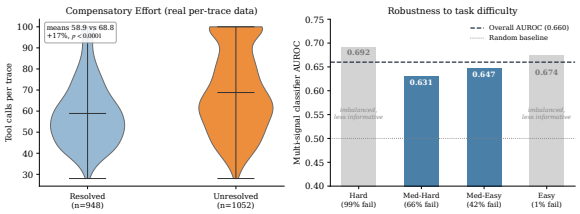

Figure 7: SWE-rebench. Left: real per-trace tool_calls distributions, resolved vs. unresolved (means 58.9 vs. 68.8, +17%, $p<0.0001$). Right: the multi-signal classifier retains discriminative power within the two class-balanced middle difficulty tiers (0.631/0.647, blue), inconsistent with a pure task-difficulty explanation (multi-signal adds little within tiers, Section 3.3); the extreme tiers (grey) are near-degenerate in class balance and less informative as difficulty controls.

C.2 Multi-Tool Personal Assistants (AppWorld)

Dataset and Setup. Modality: multi-tool API orchestration. The agent invokes APIs across services (calendar, email, shopping, etc.) and reports completion via `finish(status, answer)`. Complexity: 5 harnesses, 6 models. Scale: 406 traces; a trace is excluded as ambiguous if the agent never called `finish()` (72 of 406; an earlier pass used a different, less conservative exclusion criterion that excluded 97, but we report the single-rule result here since it is reproducible end-to-end from the raw traces), leaving $n=334$ (314 success, 20 fail).

Per-Signal Analysis. `total_tokens` 991,689 vs. 1,731,529 (+75%, $p=0.0001$, Compensatory Effort, $d=0.63$, 95% CI $[0.17, 1.08]$); `n_spans` 20.2 vs. 19.3 (−4%, $p=0.463$); `tool_calls` 26.7 vs. 17.9 (−33%, $p=0.066$); `token_entropy` 3.80 vs. 3.92 (+3%, $p=0.340$). Only `total_tokens` is significant; the rest are **not** and do not survive FDR correction; all four estimates share the same limitation, only 20 failing traces, so even the confirmed `total_tokens` effect carries a wide confidence interval.

Multi-Signal Analysis. Correlation Decoupling: `spans×entropy` tightens significantly (ρ:0.980→0.999, $p<0.0001$), indicating more steps mechanically produce more distributed effort in failing traces. Output Inflation: the agent's `finish(status, answer)` payload gives a length-of-artifact proxy, but only 116 of the 334 sessions submit a non-empty literal answer (the rest call `finish()` with empty arguments, valid for action-only tasks with no literal answer to report); on this reduced, non-random subsample, mean answer length is 70.1% longer when failing ($p=0.032$, $n=15$ fail/101 pass), directionally consistent with Output Inflation but not a clean full-sample result.

Root Cause Interpretation. The one clean per-trace signal, `total_tokens`, plus the correlation shift together suggest a mechanical relationship: on failing traces, the agent takes more steps, and each additional step both consumes more tokens and reinforces the tokens/entropy coupling as effort spreads across a longer trace. The small failing-trace count ($n=20$) makes it hard to separate this from broader noise; `n_spans`, `tool_calls`, and `token_entropy` all move in a directionally plausible way but do not clear significance individually.

Validation Summary. Per-signal: Compensatory Effort confirmed on `total_tokens` alone (+75%, $p=0.0001$); Unfocused Orchestration not confirmed ($p=0.340$, likely underpowered rather than absent). Multi-signal: Correlation Decoupling confirmed ($p<0.0001$); Output Inflation suggestive on a caveated 116/334 subsample (+70.1%, $p=0.032$). Statistical power is moderate: $d=0.63$, 95% CI $[0.17, 1.08]$, power 77% (Table 5) a reliable finding, but with more uncertainty than the other domains.

C.3 Customer Service Agents (τ-bench)

Dataset and Setup. Modality: simulated customer dialogue with tool use (booking, database

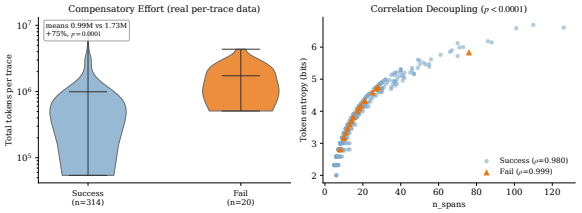

Figure 8: AppWorld. Left: real per-trace `total_tokens` distributions (log scale); the one signal that clears significance at n=20 fail (+75%, p=0.0001). Right: real spans×entropy scatter; the coupling tightens from ρ=0.980 (success) to ρ=0.999 (fail).

lookups, policy actions). Complexity: 4 harnesses, 3 models; three sub-domains of varying task structure: airline (196), retail (469), telecom (186). Scale: 851 traces (Yao et al., 2024), 799 completed / 52 did not.

Per-Signal Analysis. Compensatory Effort dominates with the largest effect sizes across all domains tested in this paper (p<0.0001 for `tool_calls/tokens`, p=0.0004 for `n_spans`); see Table 10 for the full per-signal breakdown, including Unfocused Orchestration (`token_entropy`, +9%, p=0.0057).

Signal	Succ.	Fail	Δ
tool_calls	5.7	10.8	+90%
total_tokens	115,842	289,584	+150%
n_spans	10.52	12.04	+14%
token_entropy	3.04	3.32	+9%

Table 10: τ-bench (n=799/52; all p<0.006, most p<0.0001). tool_calls/tokens/n_spans=Compensatory Effort; entropy=Unfocused Orchestration.

Multi-Signal Analysis. Correlation Decoupling shifts significantly in the pooled sample (`tool_calls`×`tokens` ρ:0.859→0.996, p<0.0001; the other two pairs shift comparably) and replicates fully in the retail and telecom sub-domains (all three signal pairs significant); airline replicates only partially (1 of 3 pairs, n=6 failing traces, likely underpowered for the other two). Output Inflation is not testable on the public release: three of the four harnesses do not capture the agent's own message spans (only the user-simulator's), and the one harness that does, `claude_code`, contains 51 of the 52 total failures, confounding harness choice with the failure label; within `claude_code` alone the effect is not significant (p=0.212).

Root Cause Interpretation. Failing agents invoke more tool calls and consume more tokens before the trace ends at a failed API call; elevated effort precedes and coincides with the eventual failure rather than a purely reactive retry loop (98.1% of failed spans are terminal in their trace; Limitations).

Validation Summary. Per-signal: Compensatory Effort (+90% tool calls, +150% tokens) and Unfocused Orchestration (+9% entropy) confirmed. Multi-signal: Correlation Decoupling confirmed in the pooled sample and 2 of 3 sub-domains; Output Inflation blocked by a harness-instrumentation confound, not a data-availability gap. Statistical power is high: d=0.66, 95% CI $[0.38, 0.95]$, power 100% (Table 5).

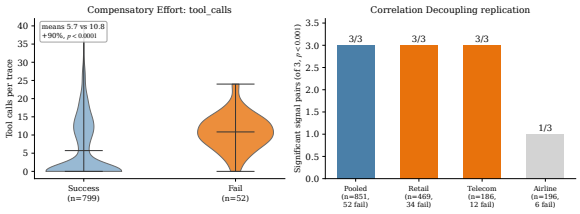

Figure 9: τ-bench. Left: real per-trace tool_calls distributions (means 5.7 vs. 10.8, +90%, p<0.0001). Right: Correlation Decoupling replication across sub-domains: all three signal pairs shift significantly in the pooled sample, retail, and telecom; airline replicates 1 of 3 pairs, likely underpowered at n=6 failing traces.

C.4 Independent Meta-Evaluation Replication (Counsel)

Dataset and Setup. Modality: human-reviewed errors on a τ-bench-derived dialogue pool, independent of the τ-bench validation above. Complexity: judge-flagged span sampling (every trace has ≥1 flagged span, not a random population draw). Scale: n=185 (131 human-confirmed real errors, 54 without) (Pisupati et al., 2026), trace pool and agent population entirely disjoint from τ-bench.

Per-Signal Analysis. Compensatory Effort and Unfocused Orchestration are both confirmed (Table 11): `assistant_turns` +17.1% (p=0.0053), `unique_tools_invoked` +12.0% (p=0.0014), and `tool_selection_entropy` +7.0% (p=0.0027), an independent fourth-domain replication of the same over-effort pattern found in SWE-rebench, AppWorld, and τ-bench.

Multi-Signal Analysis. Output Inflation is measurable here (final assistant response length) but reverses direction: responses are 19.9% *shorter*,

Signal	Δ	p
assistant_turns	+17.1%	0.0053
unique_tools_invoked	+12.0%	0.0014
tool_selection_entropy	+7.0%	0.0027

Table 11: Counsel (n=131 error / 54 no-error). Rows 1–2=Compensatory Effort; row 3=Unfocused Orchestration. Output Inflation is tested but reverses direction (-19.9% response length, p=0.0153; Section 2); Correlation Decoupling is excluded after a sampling-bias confound check (judge-flagged sampling does not preserve matched pairing).

not longer, when failing (p=0.0153), the opposite of what the signature predicts. Correlation Decoupling is not applicable: Counsel's 185 traces are a judge-flagged sample, not a population draw, and the one marginally significant pair found in an unadjusted scan (p=0.019) collapses to non-significance (p=0.38) once trace length, the variable most tied to how a trace gets sampled, is partialled out.

Root Cause Interpretation. Counsel's review also surfaces a boundary case invisible to any of the four signatures as defined: a subset of human-confirmed real errors occurs in traces with *below*-median tool calls, assistant turns, and tool-selection entropy, an under-investment pattern (the agent tries too little, not too much) rather than an over-effort one. We revisit this in Limitations.

Validation Summary. Per-signal: Compensatory Effort and Unfocused Orchestration confirmed, an independent fourth-domain replication. Multi-signal: Output Inflation tested and reversed (-19.9%, p=0.0153); Correlation Decoupling excluded after a sampling-bias confound check. Counsel's independent human-review labels are among the most reliable in this paper, but its judge-flagged sampling limits what multi-signal analysis can validly test.

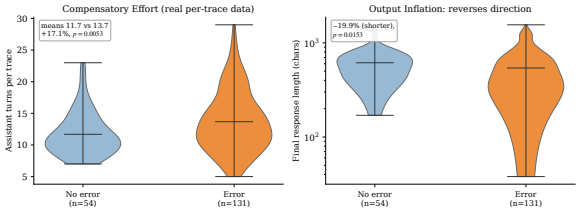

Figure 10: Counsel. Left: real per-trace assistant_turns distributions (means 11.7 vs. 13.7, +17.1%, p=0.0053). Right: Output Inflation reverses direction on this domain, responses are 19.9% shorter, not longer, when failing.

C.5 Web Navigation Agents (AgentRewardBench)

Dataset and Setup. Modality: web navigation with independent human review. We use AgentRewardBench (Lù et al., 2025): 1,301 deduplicated web-agent trajectories with full, independent human review, excluding one "Unsure"-labeled trace (946 fail / 355 success). This fifth domain doubles as the judge-comparison benchmark of Section 4; here we validate the signatures on its telemetry. A classifier trained on its own labels reaches AUROC 0.821, used throughout Section 4 (cross-domain feature transfer from SWE-rebench is limited; Appendix A).

Per-Signal Analysis. Compensatory Effort: `total_tokens` +159.0%, `n_spans` +120.9% (both $p<10^{-50}$). Unfocused Orchestration (raw): 4.20 vs. 2.75 bits (+52.9%[2], $p<0.000001$, d=1.196, 95% CI $[1.066, 1.326]$), **but confounded with step count** (r=0.96 with `n_spans`, since entropy over k spans is bounded above by $\log_2(k)$): normalizing by the step-count ceiling gives a smaller but still significant effect (+12.3%, $p<10^{-51}$, d=0.668), comparable to Unfocused Orchestration's other effect sizes rather than an outlier. We report the raw figure for consistency with Tables 9–11 and treat the normalized figure as primary elsewhere. Output Inflation (`total_output_tokens`) also shows a large, robust effect (+276.8%, $p<10^{-58}$) that survives excluding step-capped trajectories.

Multi-Signal Analysis. Correlation Decoupling requires more care: a naive scan across six signal pairs finds dramatic shifts, but five of six are an artifact of the same step cap, over half of all trajectories sit exactly at a hard 31-step limit, and that cap is itself correlated with the failure label; excluding capped trajectories collapses these five deltas to near-zero. One pair survives the exclusion: `n_action_errors`×`n_spans` (ρ:0.120→0.328, p=0.0045), as summarized in Table 4.

Root Cause Interpretation. The step-count confound recurring across both Unfocused Orchestration and Correlation Decoupling on this domain points to a shared mechanism: AgentRe-

[2]Computed from full-precision means (4.1996..., 2.7462...) before rounding for display; recomputing the delta from the rounded 4.20/2.75 gives +52.7%, a display-rounding artifact rather than a different result.

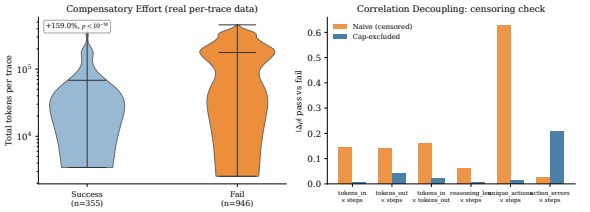

Figure 11: AgentRewardBench. Left: real per-trace total_tokens distributions, log scale (n=355/946, +159.0%, $p<10^{-50}$). Right: naive Correlation Decoupling shifts (orange) collapse toward zero once step-capped trajectories are excluded (blue), except for `n_action_errors`×`n_spans`, which survives the exclusion.

wardBench's step cap (31) truncates long trajectories, and truncation correlates with failure, so any signal that is itself bounded by step count (entropy) or that mechanically strengthens with more steps (most signal-pair correlations) will show an inflated effect unless the cap is controlled for. The one Correlation Decoupling pair that survives (`n_action_errors`×`n_spans`) is consistent with error-driven retry behavior coupling more tightly with step count specifically in failing trajectories, a genuine behavioral signal rather than a censoring artifact.

Validation Summary. Per-signal: Compensatory Effort (+159.0% tokens, +120.9% steps) and Output Inflation (+276.8% output tokens) confirmed, both robust; Unfocused Orchestration confirmed after step-count normalization (+12.3%, primary figure). Multi-signal: Correlation Decoupling mostly a censoring artifact (5 of 6 pairs), but one pair (`n_action_errors`×`n_spans`) survives confound removal (p=0.0045).

D Appendix: Illustrative Examples

Concrete examples of the raw telemetry the signatures are computed from, for readers unfamiliar with OpenTelemetry-style instrumentation.

D.1 Example Telemetry Span

Each trace is a sequence of spans. Below is a real span from an AppWorld personal-assistant trace (DeepSeek-V3.2 model), showing the attributes from which `tool_calls` and `total_tokens` are extracted directly, with no access to the model's internal chain-of-thought:


```
{
 "span_id": "4c638ed5174d3184",
 "name": "chat azure/DeepSeek-V3.2",
 "start_time": "...T06:42:42.29Z",
 "end_time":   "...T06:42:47.68Z",
 "attributes": {
  "gen_ai.usage.input_tokens": 113422,
  "gen_ai.usage.output_tokens": 42,
  "gen_ai.response.finish_reasons":
    ["tool_calls"] },
 "status": {"code": 1, "message": ""} }
```


`tool_calls` counts spans whose `finish_reasons` includes "tool_calls"; `total_tokens` sums input+output tokens across spans; `token_entropy` is the Shannon entropy of the per-span token distribution.

D.2 Example: Resolved vs. Unresolved SWE-rebench Trace

Two real SWE-rebench traces from the same agent (OpenHands) on comparable repositories show how Compensatory Effort and Output Inflation manifest at the step level; illustrative of Table 9, whose tests use the full 2,000-trace population.

Step	Resolved	Unresolved
1	file_read (target)	file_read (scan dir)
2	file_read (test)	str_replace_editor
3	bash (run tests)	bash (tests) → FAIL
4	str_replace_editor	str_replace_editor
5	bash → PASS	bash (tests) → FAIL
...	submit at step 52	loop continues to 74

Table 12: Resolved (tool_calls=52, tokens=25,498, patch=856 chars) vs. unresolved (tool_calls=74, tokens=49,411, patch=19,982 chars): a 23× larger patch from an edit-test-fail loop.