FROM EVAL ROLLOUTS TO PROCESS SUPERVISION: MEASURING AND CURATING LLM AGENT TRAJECTORIES AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern agents solve long tasks through many rounds of exploration, tool use, editing, and recovery. Whether a run ultimately succeeds or fails hides most of this behavior: two runs with the same outcome can differ sharply in wasted work, repeated errors, verification, and use of feedback. Existing behavior taxonomies provide useful descriptions, but automatically scoring these properties raises a separate question: are the resulting measurements reliable and stable enough to guide evaluation, data selection, or training? We introduce **FGA**, a framework that defines seven dimensions of agent process quality and measures them with deterministic rules and semantic judges over a unified trajectory representation. We evaluate FGA on roughly $126{,}000$ runs from ten sources. The results show that automatically computed process metrics are useful but uneven: detector reliability and cross-source behavior depend strongly on the signal and environment. Several measurements nevertheless reveal behavioral differences that final task success does not capture. As a downstream test, a historical recipe combining a success gate with process heuristics reaches 14.4%, versus 9.8% for a trajectory-count-matched random subset, in a single-seed 9B experiment; this comparison does not isolate a process-only effect. FGA makes measurements of agent behavior explicit and auditable, while separating signals that have empirical support for downstream use from those that still require validation.

1 INTRODUCTION

Consider two software agents whose patches both pass the task verifier. The first narrows the fault to one component, edits a small region, and runs a targeted test. The second searches the repository broadly, repeats failed commands, rewrites the same code several times, and eventually reaches a passing patch. An outcome-only benchmark treats the two rollouts as equivalent. As examples for another model to imitate, however, they are not obviously equally valuable.

The same mismatch appears among failures. A rollout that identifies the root cause and narrowly misses final verification can contain useful supervision, whereas a rollout that never forms a relevant hypothesis may not. Final outcomes therefore collapse four practically different cases: clean success, wasteful success, productive failure, and uninformative failure. Benchmarks such as SWE-bench, its 500-task Verified subset, and GDPval provide indispensable external evidence about task resolution or deliverable quality (Jimenez et al., 2024; Chowdhury et al., 2024; Patwardhan et al., 2025); they do not preserve all of the information contained in how the artifact was produced. This paper asks: *when two agent rollouts receive the same final outcome, are they equally informative and equally valuable as supervision?*

Recent work provides useful vocabularies for describing agent behavior. ACT*ONOMY organizes observable actions, while AutoTraceGT iteratively discovers task-specific behavior and failure categories (Gao et al., 2026; Lu et al., 2026b). Other systems expose intermediate progress or reasoning rather than only final correctness, and recent SWE-agent analysis explicitly studies lucky passes (Ma et al., 2024; He et al., 2025; Lu et al., 2025; Sahoo et al., 2026; Xia et al., 2025; Golovneva et al., 2023). Yet this leaves the question studied here unanswered: which differences recorded in an execution trace can be extracted at scale, distinguish rollouts that share the same outcome, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

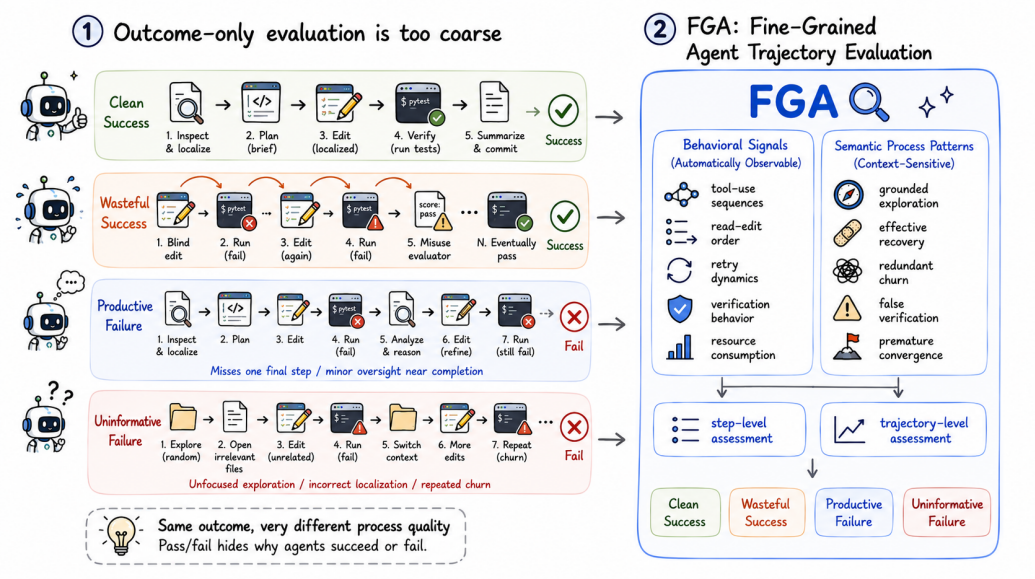


Figure 1: Outcome labels collapse clean and wasteful successes, as well as productive and uninformative failures. FGA recovers the hidden process differences.

support downstream use? The goal is not to replace outcome evaluation, but to recover information that outcome labels discard.

We introduce **FGA** as a mechanism for extracting and testing the information contained in agent execution. FGA maps heterogeneous logs into a canonical trajectory graph, organizes observed behavior into seven process-quality constructs, and combines deterministic event measurements with semantic judgments. Each reported value retains a link to the action, observation, or text span that produced it. This makes it possible to ask concrete questions—whether the agent read before editing, changed strategy after an error, verified a fix, or claimed success despite contrary evidence—without reducing all behavior to a single score.

Our empirical study follows this question from execution logs to learning. We first establish which FGA signals can be interpreted, then use them to expose differences hidden within outcome classes, test which of those differences survive a change of harness, and finally ask whether trajectory selection changes SFT. Individual measurements do not reproduce a binary verifier, but they reveal substantial variation among rollouts that receive the same label, including late persistence errors and success claims contradicted by external evaluation.

In a single-seed 9B experiment, the historical Gold-v1 recipe reaches 14.4% on SWE-bench Verified, compared with 9.8% for a trajectory-count-matched random subset. Gold-v1 combines an outcome gate with several process heuristics, so the result shows only that the complete data-composition recipe changes observed downstream performance; whether process information adds training value after outcome is held fixed remains open.

Our contributions are:

- **Extracting process information.** FGA provides a common representation for heterogeneous agent trajectories and connects observed behavior to auditable deterministic and semantic measurements with event-level provenance.
- **Characterizing what outcomes miss.** Across ten heterogeneous trajectory sources, we show that process and outcome capture different information, identify temporal and self-evaluation patterns hidden by final labels, and determine which signals remain interpretable across environments.
- **Connecting rollout analysis to learning.** We observe a 9B gain under the historical selection recipe, while the current evidence does not isolate a process-only causal effect.

Table 1: FGA's seven constructs connect observational units to representative deterministic and semantic measurements.

Construct	Evidence unit	Deterministic	Semantic	Scope
Task Progress	state/evaluation change across actions	Code-Locate Efficiency; Retry Improved	Effective Action Ratio	cross-step / trajectory
Execution Validity	tool call, arguments, and intent	Format-Failure Rate; Parameter Compliance	Parameter Accuracy; Reasoning–Action Consistency	step-level
Recovery	error → subsequent actions	Blind Retry	Error Recovery; Evidence-to-Action Gap	local / cross-step
Verification	hypothesis or edit → relevant check	Bug Reproduction; Post-Fix Verify	Hypothesis–Verify Loop	cross-step
Efficiency	action, step, and resource sequence	Effective-Step Rate; Explore–Edit Ratio	—	trajectory aggregate
Integrity	claim/action → requirement or evidence	Eval Abuse	Self-Eval Calibration; Cross-Stage Violations	cross-step / outcome-linked
Interaction Quality	feedback → subsequent behavior	—	Feedback Utilization; Interaction Precision	interaction sequence

2 Extracting Process Information from Agent Rollouts

FGA is not a second outcome predictor or a flat inventory of agent actions. Its purpose is to turn observable differences in how an agent proceeds into explicit, auditable candidate signals. Figure 1 summarizes the path from trajectory evidence to downstream use.

2.1 From observed behavior to process constructs

Prior work bounds the behavioral scope, while trajectory evidence supplies the observable content (Rahwan et al., 2019; Sumers et al., 2024; Ribeiro et al., 2020). We use constant-comparison ideas (Glaser & Strauss, 1967; Charmaz, 2006) to organize recurring trace-grounded labels, merge obvious aliases, and separate labels that conflate distinct processes. This exploratory organization supplies candidate measurements; it is not a formal saturation study.

The observed behavior spans eight recurring families: localization and exploration, planning and orchestration, reproduction and verification, editing strategy, environment and tool friction, workspace and copy correctness, hygiene and cleanup, and termination and convergence. Representative contrasts include focused read-before-edit versus grep scatter, reproduce–fix–verify versus verification theater, re-diagnosis versus identical retry, and requirement audit versus submit-with-known-failure. Cross-cutting cases such as *sound-process–wrong-fix*, *self-verified false confidence*, *under-scoped fix*, and *wasteful-but-correct* already show why process and outcome cannot be treated as one axis. Appendix Table 4 summarizes these recurring families, and Appendix Table 7 defines the executable core used in the experiments.

We keep five levels distinct throughout the paper. An *observed pattern* is a recurring description grounded in traces. A *construct* groups related questions about process quality. An *executable measurement* specifies how evidence is detected or judged. An *emitted field* is a scalar implementation output, such as a count or rate. A *selection policy* decides whether and how a validated measurement influences data use. FGA groups the observed patterns into seven constructs: Task Progress, Execution Validity, Recovery, Verification, Efficiency, Integrity, and Interaction Quality; Table 1 gives the mapping.

2.2 A common representation with evidence-backed measurements

All measurements read a canonical typed trajectory graph containing reasoning, action, observation, planning, judge, and summary nodes. Adapters normalize roughly ten raw formats into this graph, following the interleaved reasoning–action–observation structure of tool-using agents such as ReAct (Yao et al., 2023). Tool arguments, observations, timing, tokens, and provenance remain attached,

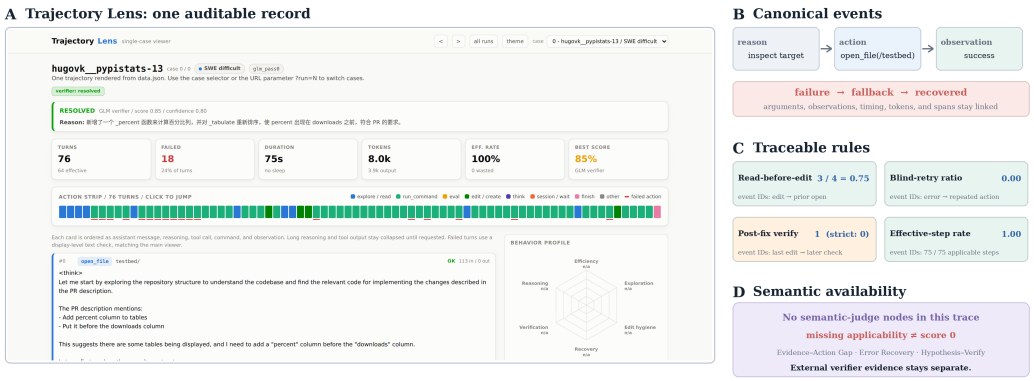


Figure 2: Trajectory Lens keeps the tool-facing record as primary evidence; compact event, rule, and applicability summaries retain traceability.

so an aggregate can be traced to the event or text span that produced it. Applicability masks preserve the difference between *structural absence* and *behavioral zero*: a harness without session tools cannot exhibit an observed Session-Close event and must report N/A rather than zero.

The evaluated core contains 58 executable measurements: 31 deterministic rule modules and 27 semantic measurements. Rules capture event footprints and denominator-aware rates, such as read–edit order, parameter compliance, blind retry, and post-fix verification. Semantic judges use metric-specific rubrics when syntax alone cannot determine whether evidence changed the strategy, recovery was effective, or a completion claim was justified (Zheng et al., 2023; Raghavendra et al., 2026). Appendix Table 7 defines the complete core.

Figure 2 follows one rollout from raw events to evidence-backed measurements. External evaluation remains beside, rather than merged with, the process record, while each rule or judge value links back to its supporting action, observation, rationale, and text span. This audit chain allows a signal to be inspected before it is used for diagnosis or selection.

3 Which Process Signals Can Be Trusted?

Before a process difference can support diagnosis or selection, it must pass three separate gates: the detector must measure its intended event, its outputs must not merely duplicate other fields, and the behavior must remain interpretable in the target environment. Passing one gate does not imply passing the others.

3.1 Detector Reliability

Semantic-judge reliability depends on both the protocol and the metric. We measure single-rating, absolute-agreement ICC under the Shrout–Fleiss framework (Shrout & Fleiss, 1979). For Evidence-to-Action-Gap ($N = 20$ trajectories, three raters), a two-step facts-then-score protocol reaches ICC 0.800, compared with 0.695 for a no-ambiguity rubric, 0.532 for same-family single-step scoring, 0.447 for a 10-point scale, 0.430 for an anchor-calibrated prompt, and 0.175 for a cross-family single-step panel. In a separate 14-metric, 840-record sweep, Evidence-to-Action-Gap has ICC 0.564, while the overall range runs from 0.973 for Per-Error-Type Solidification and 0.908 for Error Recovery to 0.358 for Feedback Utilization and 0.324 for Interaction Precision; Question Quality has no between-item variance, so its ICC is not estimable. Reliability established for one rubric cannot be inherited by the rest of the catalog.

Deterministic rules fail differently: they execute reproducibly even when they detect the wrong event. We manually label 50 cases for each of four current rule-metric families. Deadlock/Polling is the only detector that clears the pre-specified precision ≥ 0.70, recall ≥ 0.80 gate (precision 0.778, recall 0.933). Its legacy implementation reaches recall 1.000 but precision only 0.319. Post-Fix-Verify and Blind-Submit are precise (0.917–0.957) but reach only 0.733 recall, while Retry-Improved misses

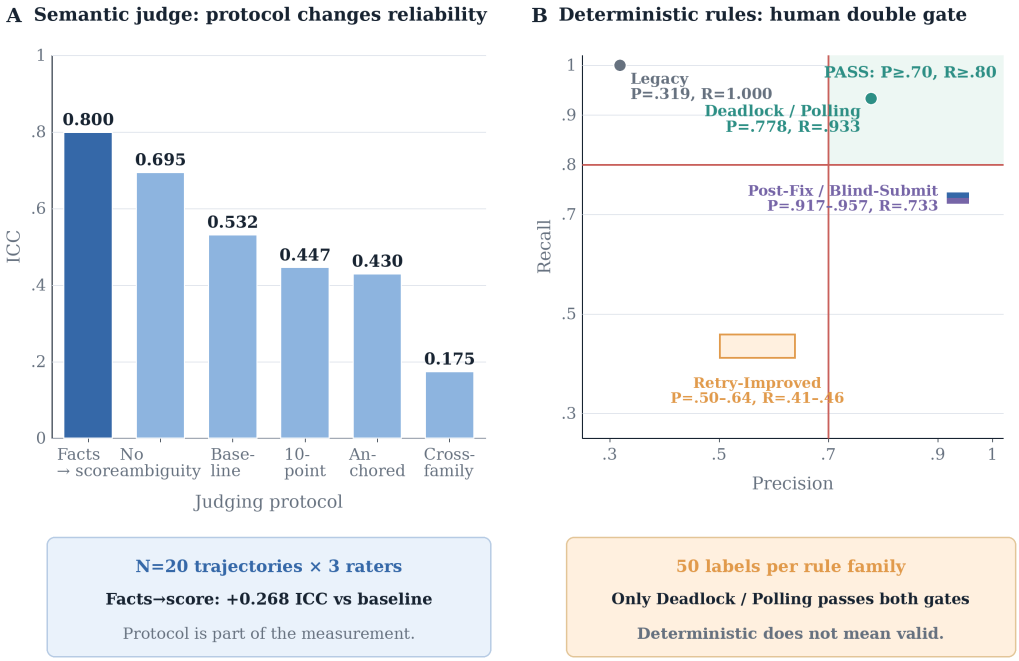


Figure 3: Detector reliability is protocol-specific: judge ICC ranges from 0.175 to 0.800, and only current Deadlock/Polling passes the human precision–recall gate.

both thresholds. Figure 3 places these two reliability failures side by side; complete sweeps appear in Appendix Tables 12, 13, and 10.

3.2 Complementarity without duplicate fields

Across the tested rule–judge pairs, trajectory-level correlations range from -0.047 to $+0.071$. Rules record whether an event footprint occurred; judges ask whether it was appropriate and consequential. The near-zero correlations support treating them as complementary, although low correlation alone does not establish that either is correct.

Within the automatic channel, implementation variants can instead be nearly identical. Read-Before-Edit Rate and Explore–Edit Ratio have $\rho = +1.000$, while Read-Before-Edit and Code-Locate Efficiency are almost reciprocal ($\rho = -0.999$). We therefore distinguish conceptual measurements from emitted aliases, counts, rates, and source-specific variants. Appendix Table 24 reports the complete redundancy screen.

3.3 Environment robustness

The source-coverage screen spans ten local exports and 125,777 rollout records: three exports provide binary outcomes, InnovatorBench (Wu et al., 2025) provides partial graded evaluation, and six are label-free (Appendix Table 18). Historical scored exports used in Section 4 are kept separate from this census: "AlphaEval scored" and AgencyBench scored (internal tag "Gen-1") denote experiment-specific local exports, whereas the census contains an AlphaEval-derived binary export and a distinct AgencyBench-derived binary export (internal tag "V3.0"). Source names and version strings are internal export labels; citations establish benchmark or scaffold lineage, not exact public releases, and local scales, filters, and rollout counts are not public task counts. Transfer has three distinct requirements. The harness must expose the relevant event; the observed value must remain on a comparable scale; and its relation to an outcome or rubric must preserve direction. Structural absence is therefore not a favorable zero: a source without session, polling, planning, or summary events cannot support the corresponding behavioral interpretation.

Several full-coverage ratios, including Effective-Step Rate and Parameter-Schema-Compliance, are comparatively stable, but many fields are source-specific. Subtask Count changes sign, and Session-Close is informative in one source but unavailable or constant in another. Appendix D separates coverage, distribution shift, and direction; it also retains a historical 136-field screen for provenance but does not treat that legacy schema as evidence for the current core. Transfer is therefore a property of a measurement–environment pairing, not of a metric in isolation.

4 What Outcome Labels Miss

Final outcomes remain the authoritative test of whether a task was solved. Process measurements answer a different question: what happened during execution, including variation that remains after the verifier has treated two rollouts as equivalent.

4.1 Variation within the same outcome

Against the binary verifier attached to a parent corpus of 95,828 SWE trajectories, individual semantic-measure AUCs lie around 0.49–0.56 (maximum 0.556; median about 0.51); applicable per-metric samples range from 333 to 2,000, and individual rules are similarly weak. This near-chance discrimination is not evidence that every process measure is useless. It shows that no individual measurement simply reproduces the task verifier. Within the passing class, 12,696 of 55,131 trajectories (23.0%) trigger at least one applicable pre-specified detector, such as blind retry or edit-without-read. Because most of these detectors have not passed the human double gate, this result establishes detector-trigger variation, not a validated defect rate.

Three historical scored exports—AgencyBench scored (internal tag "Gen-1"), InnovatorBench, and AlphaEval scored—provide complementary evidence (Li et al., 2026; Wu et al., 2025; Lu et al., 2026a). These names identify the archived analysis exports, not the complete public benchmark releases; Appendix Table 18 separately lists the binary local snapshots used by the source census. Source-level median absolute associations range from $|\rho| = 0.45$ to 0.63, and the strongest Trajectory Satisfaction association reaches $\rho = 0.92$ (Appendix Table 5). This is not a controlled label-granularity experiment: corpus, harness, export, grader, and label type change together, and semantic judges may share method variance with LLM-graded outcomes. The results nevertheless show that process measurements can align with distinctions a binary verifier omits (Figure 4).

4.2 How process quality evolves

Outcome labels also hide when and how a rollout deteriorates. Dividing 595 trajectories into the first 20%, middle 60%, and final 20% shows persistence errors rising $4.2\% \rightarrow 10.4\% \rightarrow 18.7\%$ and logic errors rising $0.2\% \rightarrow 2.2\% \rightarrow 5.5\%$, while syntax errors fall $2.9\% \rightarrow 1.2\% \rightarrow 0.5\%$. Mean severity rises $0.22 \rightarrow 0.45 \rightarrow 0.77$. Superficial syntax problems tend to disappear, while unresolved late-stage problems increasingly involve persistence and logic.

Among passing SWE trajectories, the pre-specified process-flag rate rises from 0.5% below 20 steps to 8.6% at 20–40, 21.7% at 40–60, and 26.9% beyond 120 steps, then plateaus near 25–27% after 60 steps. In the full binned corpus, rather than the passing-only subset used for the flag-rate series, pass rates remain within 52.6–62.4% across bins above 20 steps. Longer runs create more detector exposure and often correspond to harder tasks, so this pattern does not imply that truncation would help. Several semantic scores also remain length-sensitive after model fixed effects (Appendix K); trajectory length is therefore a nuisance variable, not a quality target. Prefix analyses and full regressions remain in the appendix.

4.3 Self-evaluation versus external evidence

Across 595 trajectories, 79 runs (13.3%) meet an operational conflict definition: Self-Eval Calibration score 1, an explicit success claim, and external evaluation below 5. Their external evaluation has mean 0.12 and median 0, compared with mean 20.6 and median 7.7 for the honest score-3 group. These cases are concentrated by model and source rather than representing intrinsic family rates (Appendix Tables 34–36). The reported mean thought-expression rate is only 10.0% among instrumented cases,

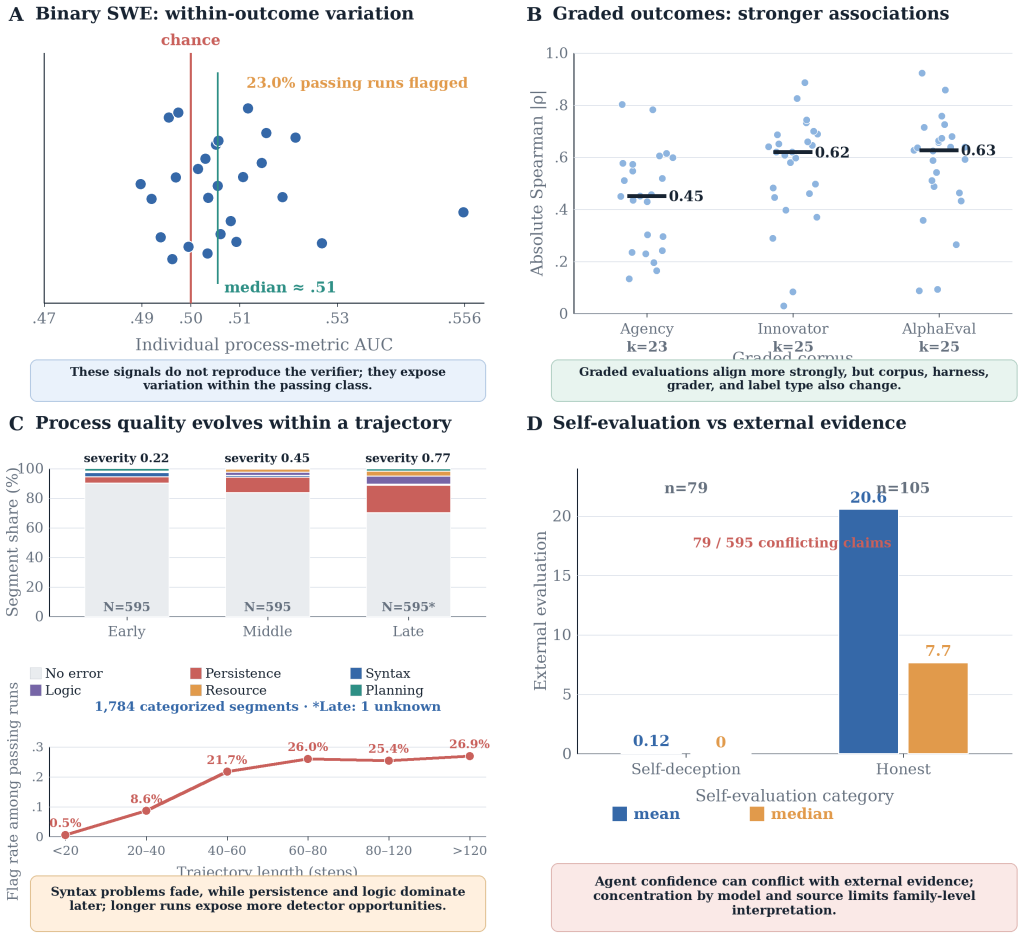


Figure 4: Process signals expose within-outcome variation, source-dependent graded associations, late-stage error shifts, and success claims contradicted by external evidence.

further limiting what can be inferred from absent reasoning. FGA exposes the conflict; provenance and coverage determine how cautiously it should be interpreted.

4.4 Context cannot be replaced by a bad-pattern count

The outcome–process distinction does not make an arbitrary sum of suspicious actions useful. On 26,849 swefactory trajectories, a hand-crafted aggregate of superficially undesirable actions obtains only 0.560–0.566 precision for the unresolved label, approximately its 0.5676 base rate, despite 0.88–0.97 recall. Many of the counted actions are also normal parts of debugging. This is a pass/fail test rather than a hack/clean validation, and this hand-crafted aggregation does not provide a useful failure signal. Concrete cases in Appendix O illustrate clean recovery, disciplined verification, and non-convergence.

5 Does Process Information Change Training Value?

If execution contains supervision beyond the final outcome, using that information during trajectory selection should be capable of changing downstream learning. We test this implication with Gold-v1, a historical recipe that combines a success gate with process criteria.

Table 2: Metric usage policy. Role is recorded independently of polarity; counterexamples are observed associations rather than intervention effects.

Role	Question	Example metrics	Allowed use	Why not optimize directly
Online early warning	Does a prefix show a pre-specified risk?	Blind Retry; Eval Abuse; Deadlock/Polling; Format/Tool Error	Alert or inspect; only validated contract gates	Only current Deadlock/Polling passes the tested human double gate.
Post-hoc filtering	Does a completed run contain curation evidence?	Error Recovery; Failure Pattern; Evidence-to-Action Gap; Self-Eval Calibration	Stratify, audit, or soft-rank after source calibration	Process evidence can disagree with outcome and attribution.
Candidate progress signal	Is an intermediate state change externally verified?	Verified subtask completion; hidden-test improvement; Retry Improved	Shape or schedule only after persistence validation	Retry-Improved fails the tested human double gate.
Descriptive only	What characterizes workload, strategy, or harness?	Tokens; Duration; Action Entropy; Subtask Count; Repeated View	Report, match, or control as a nuisance variable	Duration/tokens correlate positively; Subtask Count flips sign.

5.1 A CONSERVATIVE HISTORICAL SELECTION RECIPE

A measured behavior is not automatically a reward or filter. Its meaning can change with the environment, and normal debugging actions can resemble undesirable behavior when stripped of context. FGA therefore treats most measurements as descriptive candidates unless they pass detector and environment checks. Table 2 distinguishes online warnings, post-hoc filters, candidate progress signals, and descriptive-only quantities. Gold-v1 predates the present validation audit and is retained only as a historical composite recipe, not as a process-only intervention or a policy endorsed by the audit.

Starting from $95{,}828$ swefactory trajectories, we remove 499 held-out evaluation tasks, enforce valid messages, tool arguments, templates, and loss masks, apply a conservative length margin, and discard sequences above $65{,}536$ model-tokenizer tokens. The trainable universe contains $16{,}728$ trajectories (819.4M tokens). Full uses this universe, Random samples $6{,}500$ trajectories with seed 0 (318.8M tokens), and Gold-v1 retains $6{,}500$ trajectories (338.0M tokens) using one outcome gate and five recorded process criteria, some of which are algebraically redundant:

Criterion	Requirement
Outcome	passed
Process	reasoning coverage ≥ 0.64; post-fix verify ≥ 1 blind-retry ratio $= 0$; format-fail rate ≤ 0.05 effective-step rate ≥ 1.0

Gold and Random are trajectory-count matched but not token matched; Gold contains 6.0% more dataset tokens per trajectory. Gold-v1 is therefore a historical passed-plus-process recipe, not a clean causal test of any one FGA measurement.

5.2 9B DOWNSTREAM RESULT

In this single-seed Qwen3.5-9B setting, Base resolves $8/500$ tasks (1.6%), Random $49/500$ (9.8%), Full $54/500$ (10.8%), and Gold-v1 $72/500$ (14.4%). "Qwen3.5-9B" is the checkpoint identifier preserved in the experiment artifacts; we do not infer a different public-release identity from that local name. Gold-v1 improves over Random by 4.6 points (paired bootstrap 95% CI $[1.6, 7.6]$; exact McNemar $p = .004$) and over Full by 3.6 points (CI $[0.4, 6.8]$; $p = .033$). Exact-instance and high-overlap problem-statement contamination are zero. The supported conclusion is that the complete

Table 3: 9B SFT setup and SWE-bench Verified outcomes. Dataset tokens are unique tokens per epoch.

Model	Data	Traj.	Dataset tokens	Epochs / exposure	Resolved
Qwen3.5-9B	Base	–	–	not separately reported	8/500 (1.6%)
	Random	6,500	318.8M	not separately reported	49/500 (9.8%)
	Full	16,728	819.4M	not separately reported	54/500 (10.8%)
	Gold-v1	6,500	338.0M	not separately reported	**72/500** (14.4%)

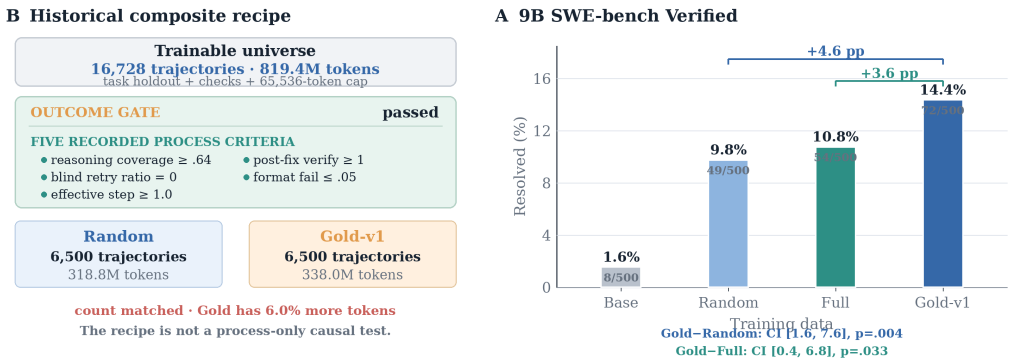


Figure 5: Gold-v1 matches Random in trajectory count, not tokens, and combines a success gate with five process criteria. It reaches 14.4% at 9B: +4.6 points over Random and +3.6 over Full.

selection recipe is associated with higher downstream performance in this single-seed setting. Table 3 reports the recovered data composition and outcomes; the original training exposure was not recorded separately by arm.

5.3 Attribution Boundary

The current experiment does not isolate the source of the 9B ordering. The format gate is nearly inert; Blind-Retry and Effective-Step are almost redundant ($\rho = -0.999$); Reasoning-Coverage is confounded; and Post-Fix-Verify is weak against outcome (AUC 0.516). We therefore attribute the gain only to the complete recipe, not to process thresholds among passing trajectories.

A decisive test is passed-only, task-, token-, and length-matched selection that varies process quality while holding harness, teacher mix, optimization, checkpoint selection, and training seeds fixed.

6 Limitations and Conclusion

FGA makes process differences hidden by final outcomes explicit and auditable. Its signals remain detector- and environment-dependent, and Gold-v1 does not isolate a process-only training effect. Evaluation rollouts can support diagnosis and, in some settings, learning, but their supervision value must be established for each signal and training regime.

AI Use Statement

We used generative AI tools to assist with software development and debugging, experiment orchestration, data-analysis scripts, figure prototyping, literature discovery, and drafting and revising the manuscript. The model-based semantic judges studied by FGA are part of the methodology and are described separately in the paper. The authors reviewed all AI-assisted text, code, analyses, and figures; checked numerical claims against stored experimental artifacts; tested executable code and configurations; and manually verified the cited literature. AI-generated suggestions were treated as drafts rather than evidence. The authors take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4082. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Kathy Charmaz. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage, 2006.

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E. Jimenez, John Yang, Leyton Ho, Tejal Patwardhan, Kevin Liu, and Aleksander Madry. Introducing SWE-bench Verified. OpenAI, August 2024. URL https://openai.com/index/introducing-swe-bench-verified/. Published August 13, 2024; updated February 24, 2025.

Jie Gao, Kaiser Sun, Jen-tse Huang, Katherine Van Koevering, Sijie Ji, Heyuan Huang, Weiyan Shi, Zhuoran Lu, Ziang Xiao, Daniel Khashabi, and Mark Dredze. How to interpret agent behavior. *arXiv preprint arXiv:2605.13625*, 2026. doi: 10.48550/arXiv.2605.13625. URL https://arxiv.org/abs/2605.13625.

Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine, 1967.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=xYlJRpzZtsY.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoit Dumoulin. TRAJECT-Bench: A trajectory-aware benchmark for evaluating agentic tool use. *arXiv preprint arXiv:2510.04550*, 2025. doi: 10.48550/arXiv.2510.04550. URL https://arxiv.org/abs/2510.04550.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Keyu Li, Junhao Shi, Yang Xiao, Mohan Jiang, Jie Sun, Yunze Wu, Dayuan Fu, Shijie Xia, Xiaojie Cai, Tianze Xu, Weiye Si, Wenjie Li, Dequan Wang, and Pengfei Liu. AgencyBench: Benchmarking the frontiers of autonomous agents in 1M-token real-world contexts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7422–7440, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.337. URL https://aclanthology.org/2026.acl-long.337/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Jiaying Lu, Bo Pan, Jieyi Chen, Yingchaojie Feng, Jingyuan Hu, Yuchen Peng, and Wei Chen. AgentLens: Visual analysis for agent behaviors in LLM-based autonomous systems. *IEEE Transactions on Visualization and Computer Graphics*, 31(8):4182–4197, 2025. doi: 10.1109/TVCG.2024.3394053. URL https://doi.org/10.1109/TVCG.2024.3394053.

Pengrui Lu, Bingyu Xu, Wenjun Zhang, Shengjia Hua, Xuanjian Gao, Ranxiang Ge, Lyumanshan Ye, Linxuan Wu, Yiran Li, Junfei Fish Yu, Yibo Zhang, Ruixin Li, Manxiang Li, Xiao Han, Xiaocong Zhou, Guangyao Chi, Zisheng Chen, Kaishen Chen, Kun Wang, Qihua Xu, Fengyue Meng, Yuchen Ni, Jiajun Li, Jinxiu Liu, Danfeng Zhang, Jingru Zhao, and Pengfei Liu. AlphaEval: Evaluating agents in production. *arXiv preprint arXiv:2604.12162*, 2026a. doi: 10.48550/arXiv.2604.12162. URL https://arxiv.org/abs/2604.12162.

Zhuoran Lu, Yangyang Yu, Zhuoyan Li, Yibo Meng, Nan Jiang, Chengxi Zang, Jie Gao, and Ziang Xiao. Using grounded theory for agent behavior analysis at scale. *arXiv preprint arXiv:2608.30391*, 2026b. doi: 10.48550/arXiv.2608.30391. URL https://arxiv.org/abs/2608.30391.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:74325–74362, 2024. doi: 10.52202/079017-2365. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html.

Tejal Patwardhan et al. GDPval: Evaluating AI model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025. doi: 10.48550/arXiv.2510.04374. URL https://arxiv.org/abs/2510.04374.

Mohit Raghavendra, Anisha Gunjal, Bing Liu, and Yunzhong He. Agentic rubrics as contextual verifiers for SWE agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15265–15290. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.697. URL https://aclanthology.org/2026.acl-long.697/.

Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019. doi: 10.1038/s41586-019-1138-y. URL https://doi.org/10.1038/s41586-019-1138-y.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Priyam Sahoo, Gaurav Mittal, Xiaomin Li, Shengjie Ma, Benjamin Steenhoek, Pingping Lin, and Yu Hu. AgentLens: Revealing the lucky pass problem in SWE-agent evaluation. *arXiv preprint arXiv:2605.12925*, 2026. doi: 10.48550/arXiv.2605.12925. URL https://arxiv.org/abs/2605.12925.

Johnny Saldaña. *The Coding Manual for Qualitative Researchers*. SAGE Publications, 3rd edition, 2016.

Patrick E. Shrout and Joseph L. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979. doi: 10.1037/0033-2909.86.2.420. URL https://doi.org/10.1037/0033-2909.86.2.420.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7584–7600. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.409. URL https://aclanthology.org/2024.acl-long.409/.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=1i6ZCvflQJ.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Xingyao Wang, Boxuan Li, Yufan Song, et al. OpenHands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a4b6ad6b48850c0c331d1259fc66a69c-Abstract-Conference.html.

Yunze Wu, Dayuan Fu, Weiye Si, Zhen Huang, Mohan Jiang, Keyu Li, Shijie Xia, Jie Sun, Tianze Xu, Xiangkun Hu, Pengrui Lu, Xiaojie Cai, Lyumanshan Ye, Wenhong Zhu, Yang Xiao, and Pengfei Liu. InnovatorBench: Evaluating agents' ability to conduct innovative LLM research. *arXiv preprint arXiv:2510.27598*, 2025. doi: 10.48550/arXiv.2510.27598. URL https://arxiv.org/abs/2510.27598.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26): 27723–27730, 2025. doi: 10.1609/aaai.v39i26.34987. URL https://ojs.aaai.org/index.php/AAAI/article/view/34987.

Qian Xiong, Yuekai Huang, Ziyou Jiang, Zhiyuan Chang, Yujia Zheng, Tianhao Li, and Mingyang Li. Butterfly effects in toolchains: A comprehensive analysis of failed parameter filling in LLM tool-agent systems. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 16712–16729. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.907. URL https://aclanthology.org/2025.findings-emnlp.907/.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software engineering agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3239. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8b86cf5ace600c48fd188efbb8dedec8-Abstract-Datasets_and_Benchmarks_Track.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:46595–46623, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79281–79327, 2025. URL https://proceedings.mlr.press/v267/zhou25aa.html.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, et al. Where LLM agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*, 2025. doi: 10.48550/arXiv.2509.25370. URL https://arxiv.org/abs/2509.25370.

A Related Work

Behavior description and discovery. Machine-behavior research treats observable system behavior as an empirical object (Rahwan et al., 2019). Grounded Theory and qualitative coding provide procedures for constructing categories through iterative engagement with evidence (Glaser & Strauss, 1967; Charmaz, 2006; Saldaña, 2016). ACT*ONOMY contributes a general action vocabulary, and AutoTraceGT induces task-specific codebooks with an operational saturation criterion (Gao et al., 2026; Lu et al., 2026b). Failure taxonomies focus on particular breakdowns, including cascading module failures, multi-agent coordination, and tool-parameter errors (Zhu et al., 2025; Cemri et al., 2025; Xiong et al., 2025). FGA uses discovery as the empirical origin of a measurement space, then asks which distinctions survive measurement and transfer tests.

Trajectory and process evaluation. AgentBoard, TRAJECT-Bench, and AgentLens expose intermediate progress or trajectory structure beyond final task completion (Ma et al., 2024; He et al., 2025; Lu et al., 2025). ReasonEval and ROSCOE evaluate intermediate reasoning separately from answer correctness (Xia et al., 2025; Golovneva et al., 2023). LLM judges and contextual rubrics scale semantic assessment but remain sensitive to evaluator capability and rubric construction (Zheng et al., 2023; Raghavendra et al., 2026). FGA complements these systems by holding a normalized rule–judge layer fixed across ten formats and separately testing human alignment, judge agreement, applicability, and transfer.

Process supervision and trajectory selection. Process supervision converts step-level assessments into reward-model or reinforcement-learning signals (Lightman et al., 2024; Wang et al., 2024); trajectory optimization can instead learn from whole-trajectory success–failure preferences (Song et al., 2024). Data-centric work scales software-engineering training data or refines general training corpora (Yang et al., 2025; Zhou et al., 2025). Our focus is the preceding measurement question: when does an observed difference in how an agent acts have enough evidence to guide curation or learning?

B Supporting tables moved from the main paper

This section preserves the metric definitions, complete construct mapping, reliability sweep, prefix analysis, transfer tables, and compact model leaderboard referenced from the main paper.

Table 4: Recurring process-pattern families used to construct candidate measurements, not automatic inclusion or exclusion rules.

Family	Representative productive patterns	Representative pathological patterns
Localization / exploration	Focused read-before-edit; evidence-directed search	Grep scatter; unfocused repository-wide reading
Planning / orchestration	Explicit subgoals; pre-execution requirement audit	Under-scoped plan; broad propagation without dependency checks
Reproduction / verification	Reproduce–fix–verify; discriminating test	Verification theater; self-verified false confidence
Editing strategy	Minimal targeted fix; evidence-backed propagation	Repeated edit reversals; over-broad propagation
Environment / tool friction	Re-diagnosis after feedback; adaptive fallback	Identical retry; evidence-free friction loop
Workspace / copy correctness	Confirm target tree, copy state, and artifact	Edit a stale copy; submit the wrong artifact
Hygiene / cleanup	Isolate probes; remove them before delivery	Temporary-script sprawl; unrelated workspace changes
Termination / convergence	Requirement recheck; verified delivery	Known-failing submission; continued non-convergence

C Multivariate outcome-prediction audit

A 94-field logistic model reaches five-fold cross-validated AUC 0.677 within the SWE source, but the features mix process with task difficulty, model identity, resource use, and harness information. We

Table 5: Process measurements and available outcome signals. The graded-corpus rows summarize absolute Spearman associations for overlapping semantic measures. Corpus, harness, grader, and label type vary together, so these rows provide complementary rather than causal evidence about label granularity.

Corpus	Outcome signal	Median $\lvert\rho\rvert$ / AUC	# $\lvert\rho\rvert > 0.3$
swefactory local rollouts	binary verifier	AUC 0.49–0.56 (max 0.556)	$0/26$
InnovatorBench	partial graded evaluation	$\lvert\rho\rvert = 0.62$	$22/25$
AlphaEval scored local export (Lu et al., 2026a)	archived score 0–100	$\lvert\rho\rvert = 0.63$	$22/25$
AgencyBench scored export (internal tag "Gen-1") (Li et al., 2026)	archived score 0–50	$\lvert\rho\rvert = 0.45$	$16/23$

Table 6: Positioning within the emerging stack for interpreting agent trajectories.

	ACT*ONOMY	AutoTraceGT	FGA (ours)
Primary question	What observable action is occurring?	What task-specific behavior or failure pattern emerges?	How reliably does the observed process progress, recover, verify, and use feedback?
Artifact	General hierarchical vocabulary and trace labels	Saturated codebook and theoretical narrative	Executable rules/judges, measurements, validation status, and usage roles
Validation focus	Human mapping agreement and behavioral case studies	Saturation, replicate/backend stability, taxonomy coverage	Human event labels, judge reliability, nonredundancy, applicability, transfer, and intervention boundary
Downstream use	Behavioral profiles	Failure discovery and prediction	Diagnosis and training-trajectory auditing/curation

therefore treat this result as an audit of what the representation jointly encodes, not as process-only prediction. Table 14 reports the prefix associations, and Appendix I retains the detailed prediction and leakage analyses.

D TRANSFER-ANALYSIS DETAILS

The original stability screen encodes structurally absent behaviors as constants. We retain its complete numbers for reproducibility, while the main paper separates empirical coverage from stability and does not interpret an absent tool as observed zero behavior.

E TRAJECTORY SOURCES

F HISTORICAL CORPUS PROFILE AND PREFIX ANALYSIS

A frozen historical four-corpus snapshot contains approximately $54{,}000$ raw semantic-judge calls. After metric-specific applicability filtering, local execution scores are comparatively high: Step Precision averages 4.62, Parameter Accuracy 4.02, and Root-Cause Attribution 4.01. Measurements that require control across steps are lower: Evidence-to-Action Gap averages 2.84, Hypothesis–Verify 2.71, Error Recovery 2.29, and Propagation Length 1.64. Figure 8A groups these values without treating native rubric means as a model ranking. The snapshot suggests that agents often produce locally plausible actions while struggling to maintain evidence, recovery, and verification across longer sequences.

G EXPLORATORY BEHAVIOR STRUCTURE

As a separate exploratory analysis, we examine $29{,}425$ trajectories from seven sources and three agent frameworks using 21 headline process fields. Within-source rank normalization and residualization for trajectory length produce six descriptive dimensions: longitudinal reasoning dynamics, tool-use discipline, thinking utility, planning and idling/stalling, verification habits, and evaluation abuse. A k-means summary with $k = 4$ yields four readable profiles: *Blind-edit* (editing before sufficient

Table 7: Definitions for FGA's 58 operational measures. The evaluated core contains 31 rule modules and 27 implemented judge-catalog metrics; 23 judge columns enter the composite matrix, while the remainder use separate pipelines or have no applicable events. Appendix Table 1 summarizes the seven constructs.

#	Metric	Definition	#	Metric	Definition
1	R: Tool-Chain Depth	Number of tool-using (ReAct) steps.	28	R: Action Entropy	Normalized entropy of tool-to-tool transitions.
2	R: Subtask Count	Total declared subgoals; decomposition granularity.	29	R: Has Planning	Whether a genuine planning step exists.
3	R: Tool Reuse Rate	Edits to existing files divided by all file mutations.	30	R: Retry Improved	Whether the last evaluation beats the first, and by how much.
4	R: Read-Before-Edit	Fraction of edits preceded by reading the same file.	31	R: Budget Utilization	Deprecated compatibility alias of Effective-Step Rate.
5	R: Blind Retry	Longest run of consecutive same-error failures.	32	J: Error Recovery	How often and how well the agent recovers after an error.
6	R: Eval Abuse	Consecutive evaluations with no intervening code change.	33	J: Cross-Stage Violations	Later decisions that contradict earlier plans or constraints.
7	R: Session Leak / Reclaim	Unclosed sessions; ratio of closed to created sessions.	34	J: Effective Action Ratio	Fraction of actions judged to advance the goal.
8	R: Deadlock / Polling	Share of steps in poll-and-wait loops.	35	J: Parameter Accuracy	Whether tool argument values are correct and valid.
9	R: Repeated View	Fraction of opened files re-read at least three times.	36	J: Trajectory Satisfaction	Holistic rating of overall trajectory quality.
10	R: Search-vs-Open	Search/find actions relative to open-file actions.	37	J: Root-Cause Attribution	How clearly the fundamental cause is diagnosed.
11	R: Token Consumption	Total LLM tokens; a cost proxy.	38	J: Propagation Length	Steps between a root-cause error and its consequence.
12	R: Duration	Wall-clock time, separated into active and sleep time.	39	J: Failure Pattern	Dominant structural failure mode.
13	R: Format-Failure Rate	Steps that produce no usable action.	40	J: Error Solidification	Early error subsequently treated as correct.
14	R: Effective Step Rate	One minus format-fail, blind-retry, and deadlock shares.	41	J: Per-Type Solidification	Persistence of logic and syntax errors.
15	R: Token-per-Point	Token cost per unit of achieved score.	42	J: CAT Failure Rate	Co-occurrence of blind retry, eval abuse, and leak.
16	R: First Eval Step	Position of the first self-evaluation.	43	J: Failure Precision	Claimed failed steps that are actual failures.
17	R: Parameter Compliance	Fraction of tool calls with well-formed arguments.	44	J: Failure Recall	Actual failed steps identified by the agent.
18	R: Scaling Bottleneck	Step budget after which extra steps stop helping.	45	J: Failure F1	Harmonic mean of step-level precision and recall.
19	R: Precursor Signals	Pre-specified early-warning patterns associated with failure.	46	J: Evidence-to-Action Gap	Correct evidence gathered but not acted upon.
20	R: Code-Locate Efficiency	Depth at which the first substantive file edit occurs.	47	J: Stage-Wise Error Type	Dominant error in early, middle, and late thirds.
21	R: Explore-Edit Ratio	Exploration actions per file mutation.	48	J: Interaction Precision	Necessary and decision-relevant proactive interactions.
22	R: File Reach Rate	Distinct files explored divided by distinct files edited.	49	J: Question Quality	Specificity, answerability, and decision value of questions.
23	R: Bug Reproduction	Whether a reproduction is run before the first edit.	50	J: Thinking Hedging	Frequency and calibration of expressed uncertainty.
24	R: Post-Fix Verify	Whether a successful check follows the last fix.	51	J: Uncertainty Calibration	Gap between expressed confidence and success.
25	R: Edit Rollback	Edits that re-touch the same file within five actions.	52	J: Feedback Execution	Coverage of explicit feedback points in subsequent actions.
26	R: Reasoning Degradation	Late-versus-early reasoning length.	53	J: Feedback Utilization	Whether retries incorporate requested changes.
27	R: Think Effectiveness	Success-rate lift of a step following explicit thought.	54	J: Self-Eval Calibration	Self-reported completion against external score.
–	–	–	55	J: Reasoning–Action Consistency	Whether stated reasoning matches the issued action.
–	–	–	56	J: Hypothesis–Verify Loop	Whether hypotheses are tested before commitment.
–	–	–	57	J: Dual-Channel Divergence	Disagreement between stated intent and behavior.
–	–	–	58	J: Subtask Propagation	How far an error travels across dependent subtasks.

Table 8: Four metric roles. Role 3 contains candidate optimization targets that still require intervention-level validation and a reliable outcome gate. Role 4 is descriptive, not a standalone reward.

Role	Question it answers	Representative metrics
1. Online early warning	Does the observed prefix contain a pre-specified risk flag?	Blind Retry, Eval Abuse, Deadlock/Polling, Format/Tool-Error, Error Solidification, resource leak, repeated failed strategy
2. Post-hoc filtering	Which process flags occur in the completed run?	Error Recovery, Failure Pattern, Evidence-to-Action Gap, Self-Eval Calibration, Trajectory Satisfaction, Reasoning–Action Consistency
3. Candidate progress signal	Is there externally verified evidence that task state advanced?	Verified subtask completion, hidden-test improvement, verifier-confirmed state change, Retry Improved
4. Descriptive only	Which quantities characterize a run without a safe intervention interpretation?	Token Consumption, Duration, Tool-Chain Depth, Action Entropy, Subtask Count, Repeated View, Has Planning, Tool Variety

inspection), *Explorer* (high exploration/edit), *Refiner* (broad reach and reuse), and *Wanderer* (repeated views and high transition entropy).

Table 9: FGA measures process quality rather than merely naming action types. Each construct combines auditable event footprints (R) with semantic judgments (J) where needed.

Construct	Measurement question	Representative operational metrics
Task progress	Does the trajectory move toward a resolved task state?	Subtask Count (R), Code-Locate Efficiency (R), Retry Improved (R), Effective Action Ratio (J)
Execution validity	Are actions executable, well-formed, and consistent with stated intent?	Format-Failure Rate (R), Parameter Compliance (R), Parameter Accuracy (J), Reasoning–Action Consistency (J)
Recovery	Does feedback change the strategy after an error?	Blind Retry (R), Error Recovery (J), Error Solidification (J), Evidence-to-Action Gap (J)
Verification	Are hypotheses and fixes checked against evidence?	Bug Reproduction (R), Post-Fix Verify (R), Hypothesis–Verify Loop (J), Self-Eval Calibration (J)
Efficiency	How much exploration and resource use produces useful progress?	Effective-Step Rate (R), Explore–Edit Ratio (R), Token Consumption (R), Repeated View (R)
Integrity	Does the process avoid unsupported claims or gaming-like behavior?	Eval Abuse (R), Cross-Stage Violations (J), Dual-Channel Divergence (J), Trajectory Satisfaction (J)
Interaction quality	Does the agent solicit and use decision-relevant feedback?	Interaction Precision (J), Question Quality (J), Feedback Execution (J), Feedback Utilization (J)

Table 10: Human double-gate for four tested rule metrics (50 labels each). Passing requires precision ≥ 0.70 and recall ≥ 0.80.

Metric	Precision	Recall	Passes?
Deadlock / Polling	0.778	0.933	**yes**
Deadlock / Polling (legacy)	0.319	1.000	no
Post-Fix-Verify	0.917–0.957	0.733	no
Blind-Submit	0.917–0.957	0.733	no
Retry-Improved	0.50–0.64	0.41–0.46	no

The profiles are not natural agent types: silhouette is only ≈ 0.11, and agreement with the pooled solution varies by source (ARI 0.81 for SWE-CodeAct, 0.39 for SWE-agent, and 0.155 for InnovatorBench). Pooled assignments differ in external evaluation on 444 scored InnovatorBench trajectories (Kruskal–Wallis $H = 11.4$, $p = .0099$), whereas InnovatorBench-only re-clustering is not outcome-significant ($p = .132$). The recovered structure is therefore continuous and harness-sensitive rather than a universal agent typology.

H Model-level statistical-debiasing metrics

Table 19 reports the population-level debiasing layer (11 statistical measures) over the InnovatorBench corpus. These remove token/length/verbosity confounds before models are compared and are, by construction, Role-4 (never per-trajectory rewards).

I Cross-model discrimination: per-metric detail

This appendix transcribes the per-metric statistics retained from the first-generation discrimination report for InnovatorBench, AgencyBench, and SWE-bench. Grades (S/A/B/C, dead, redundant) are the report's own discrimination ratings and are preserved as historical report outputs rather than current-paper validation tiers.

PCA capability dimensions. Five latent dimensions recur across benches: (i) *scale/investment* (tool-chain depth, tokens, unique tools, output chars); (ii) *efficiency/quality* (effective-step-rate, error rate, effective-subtask-rate); (iii) *planning* (subtask count, has-planning, read-before-edit); (iv) *self-repair* (retry-improved, degradation, undo); (v) *resource management* (session leak, budget utilization, code-locate efficiency). InnovatorBench needs > 10 principal components for 80% variance; AgencyBench and SWE-bench each need 9 (82%).

Legacy-table exclusion. We do not reproduce the first-generation per-metric tables because several legacy fields used formulas and scales that differ from the current core despite sharing names. The archived results are therefore not used as cross-source evidence in this paper.

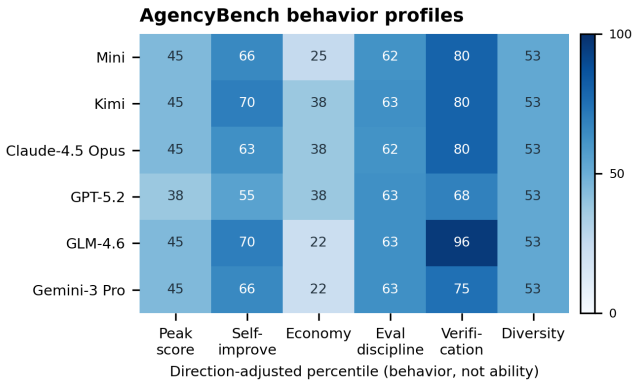


Figure 6: Direction-adjusted AgencyBench behavioral percentiles over 76 runs. These describe harness-specific behavior, not general capability. The complementary trajectory-length process-flag analysis appears in the main text (Fig. 4).

Table 11: The construct hierarchy (representative primary metrics; implemented definitions in Table 7). Rule = reference-free Python; Judge = LLM assessment.

Construct	Representative primary metrics	Type
Task Progress	Effective Step Rate, Subtask Completion / Propagation, Retry-Improved (eval-delta)	rule+judge
Execution Validity	Read-Before-Edit Rate, Parameter Accuracy, Step Precision/Recall, Tool-Error Rate	rule+judge
Recovery	Error Recovery, Blind Retry, Error (Type) Solidification, Stage-Wise Error Persistence	rule+judge
Verification	Post-Fix Verify, Hypothesis-Verify, Evidence-to-Action Gap	rule+judge
Efficiency	Repeated View, Explore-Edit Ratio, Action Entropy, Token/Step-Controlled Success	rule
Integrity / Hack	Eval Abuse, Self-Eval Calibration (self-deception), test-edit / hard-code heuristics	rule+judge
Interaction Quality	Feedback Utilization, Question Quality, Interaction Precision, Dual-Channel Divergence	judge

J DASHBOARD METRIC STATISTICS AND FULL LEADERBOARDS

The deployed *Trajectory Lens* holds 220 sampled runs across 6 sources, a 192-metric $\times$ 7-source discrimination matrix (29,425 trajectories), and 5,429 stratified metric rows; its pages (OOD validation, failure taxonomy, universality, the full discrimination report, the 96-metric catalog, 3,915 judge reasons, and the self-assessment / human-double-gate reports) are the sources from which these appendix tables are transcribed.

The main-text auditability figure (Figure 2) integrates the deployed single-case view with its canonical nodes, rule outputs, applicability mask, and evidence links. This appendix transcribes the deployed dashboard's live data arrays: the full behavioral-metric statistics table (Table 25), rule-based sub-variants derived from parent judge metrics (Table 26), top rule-metric discriminators (Table 27), and per-task coverage (Table 28). The legacy composite model leaderboard is excluded because its score does not support a current paper claim. "Method" is JUDGE / AUTO-RULE (rule) / AUTO-STAT (population statistic); Q is the dashboard's own composite discrimination score; "Rate" is its S/A/B/C/D grade. For the AUTO-STAT rows the ρ/H cells hold regression coefficients / F-statistics, not Spearman/Kruskal (flagged with †).

Table 12: Protocol variants for Evidence-to-Action-Gap ($N = 20$ trajectories, 3 raters). This comparison is separate from the cross-metric baseline sweep in Table 13 and the tightened-rubric results in Table 41.

Protocol variant	ICC	Δ vs. baseline
Two-step "facts-then-score"	0.800	$+0.268$
No-ambiguity rubric	0.695	$+0.163$
Baseline single-step (1–5), same-family panel	0.532	—
10-point scale	0.447	-0.085
Anchor-calibrated prompt	0.430	-0.102
Baseline, cross-family panel	0.175	-0.357

Table 13: Baseline two-step "facts-then-score" ICC across 14 judge metrics (840 records; $N = 20$ trajectories, 3 Claude raters). Question Quality is reported as not estimable because its ratings have zero between-item variance. Tightened-rubric variants are reported separately in Table 41.

Judge metric	ICC	Judge metric	ICC
Per-Error-Type Solidification	0.973	Error Solidification	0.676
Error Recovery	0.908	Hypothesis-Verify Loop	0.615
Subtask Propagation	0.819	Failure Pattern	0.610
Trajectory Satisfaction	0.808	Evidence-to-Action Gap	0.564
Dual-Channel Divergence	0.799	Hedging Frequency	0.557
Cross-Stage Violations	0.736	Feedback Utilization	0.358
		Interaction Precision	0.324
		Question Quality	n/e (zero-var.)

Table 25: Full FGA behavioral-metric statistics from the deployed dashboard (55 catalog rows over 655 InnovatorBench trajectories). ρ = Spearman with eval score; H = Kruskal–Wallis across models.

Metric	Dimension	Method	N	Mean	SD	ρ	H	Q	Rate
Failure Pattern	Robustness	JUDGE	655	2.17	0.85	0.769	249.7	192	S
Error Recovery	Robustness	JUDGE	591	2.22	0.68	0.759	233.3	177	S
Feedback Utilization	Behavioral	JUDGE	655	2.40	0.76	0.748	235.3	176	S
Subtask Propagation	Efficiency	JUDGE	655	2.32	0.87	0.700	213.6	150	A
Evidence-Action Gap	Reasoning	JUDGE	655	2.49	0.66	0.715	207.3	148	A
Hypothesis-Verify	Reasoning	JUDGE	655	2.08	0.75	0.732	190.1	139	A
Propagation Length	Efficiency	JUDGE	642	1.66	0.91	0.757	163.6	124	A
Step F1	Robustness	JUDGE	655	2.38	0.79	0.535	190.4	102	A
Error-Type Solidification	Robustness	JUDGE	653	2.64	0.72	0.696	145.9	102	A
Step Recall	Robustness	JUDGE	655	2.56	0.80	0.539	162.8	88	A
Parameter Accuracy	Efficiency	JUDGE	655	3.63	0.57	0.569	150.2	85	A
Effective Action Ratio	Efficiency	JUDGE	655	2.25	0.71	0.550	153.1	84	A
Error Solidification	Robustness	JUDGE	655	2.57	0.83	0.499	139.9	70	B
Dual-Channel	Reasoning	JUDGE	655	3.38	0.60	0.522	105.2	55	B
Cross-Stage Violations	Behavioral	JUDGE	585	3.45	0.75	0.579	69.4	40	B
Hedging Frequency	Reasoning	JUDGE	148	3.09	0.54	0.501	59.9	30	C
Step Precision	Robustness	JUDGE	655	4.13	0.73	0.182	154.9	28	C
Uncertainty Calibration	Reasoning	JUDGE	183	2.65	0.55	0.474	54.3	26	C
CAT Failure	Behavioral	JUDGE	631	3.48	0.81	0.155	146.3	23	C
Root Cause	Reasoning	JUDGE	611	3.37	0.91	-0.008	115.6	1	D
Duration (Seconds)	Efficiency	AUTO-RULE	655	14692.9	18911.1	0.324	189.1	61	B
Precursor Signals	Overall	AUTO-RULE	655	2.19	1.58	0.291	124.9	36	C
Tool Chain Depth	Overall	AUTO-RULE	655	159.93	169.91	0.238	145.7	35	C
Deadlock Step Ratio	Robustness	AUTO-RULE	655	0.083	0.132	0.231	62.3	14	C
Session Leak	Robustness	AUTO-RULE	655	1.64	2.34	0.070	148.4	10	C
Repeated View Rate	Behavioral	AUTO-RULE	655	0.156	0.173	0.096	107.7	10	C
Action Entropy	Behavioral	AUTO-RULE	655	4.328	0.986	0.073	126.5	9	D
Retry Improved	Robustness	AUTO-RULE	344	0.314	0.464	0.530	13.7	7	D
Post-Fix Verify	Behavioral	AUTO-RULE	586	0.904	0.294	0.125	55.5	7	D
File Reach Rate	Behavioral	AUTO-RULE	502	8.428	6.272	0.141	36.0	5	D
Read-Before-Edit Rate	Behavioral	AUTO-RULE	502	0.671	0.311	0.059	39.0	2	D
Eval Abuse Rate	Robustness	AUTO-RULE	655	0.072	0.220	0.110	20.0	2	D
Explore-Edit Ratio	Behavioral	AUTO-RULE	586	2.888	3.556	0.011	38.0	0	D
Tool Reuse Rate	Behavioral	AUTO-RULE	586	0.536	0.319	-0.109	20.4	-2	D
Reasoning Degradation	Reasoning	AUTO-RULE	270	9.762	27.981	-0.057	34.3	-2	D
Think Effectiveness	Reasoning	AUTO-RULE	632	-0.003	0.245	-0.111	40.9	-5	D

Metric	Dimension	Method	N	Mean	SD	ρ	H	Q	Rate
First Eval Step	Behavioral	AUTO-RULE	446	131.87	92.03	-0.153	39.2	-6	D
Effective Step Rate	Efficiency	AUTO-RULE	655	0.861	0.211	-0.113	69.8	-8	D
Code Locate Efficiency	Behavioral	AUTO-RULE	586	0.219	0.151	-0.149	56.6	-8	D
Search vs Open Ratio	Behavioral	AUTO-RULE	604	0.104	0.135	-0.163	54.2	-9	D
Parameter Schema Compliance	Behavioral	AUTO-RULE	637	0.9973	0.0347	0.0590	41.4	2	D
Scaling Bottleneck	Reasoning	AUTO-RULE	655	128.64	251.50	0.2381	—	24	C
Bug Reproduction Rate	Robustness	AUTO-RULE	502	0.8884	0.3151	0.2057	53.5	11	C
Edit Rollback Rate	Robustness	AUTO-RULE	502	0.3561	0.2598	-0.0291	94.9	3	D
Self-eval Calibration Error	Interaction	JUDGE	655	2.55	0.95	0.3044	89.3	27	C
Reasoning-Action Consistency	Reasoning	JUDGE	394	2.89	0.30	0.1006	190.9	19	C
Token-Controlled Success†	Evaluation	AUTO-STAT	655	7.6298	0.3972	4.0855	7.32	40	C
Step-Controlled Success†	Evaluation	AUTO-STAT	655	9.2355	0.3741	0.0108	6.64	37	C
Reasoning-Length Controlled†	Evaluation	AUTO-STAT	655	10.1954	0.3795	3.1088	6.80	38	C
Verbosity-Normalized Score†	Evaluation	AUTO-STAT	655	0.7313	1.3780	0.9998	123.88	124	A
Model$\times$Attribute Spearman	Evaluation	AUTO-STAT	380	0.2225	0.1953	meta	meta	—	C
Model$\times$Attribute Std	Evaluation	AUTO-STAT	275	15.9959	9.9884	meta	meta	—	C
Bucket-wise Self-Diagnosis	Evaluation	AUTO-STAT	275	15.8989	12.8048	meta	meta	—	C

K LONG-HORIZON REGRESSION: FULL PER-MODEL AND PER-METRIC DETAIL

This appendix gives the full coefficient and per-model tables for five long-horizon analyses. The main paper promotes the stage-wise, trajectory-length, and selected sensitivity results; the remaining analyses are retained here as supporting audits. Significance: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

L JUDGE RELIABILITY AND SELF-ASSESSMENT: FULL TABLES

This appendix gives the full per-model and per-task breakdown behind the judge-deployment findings of Section 4 (595 trajectories $\times$ 20 tasks $\times$ 15 models, 2,857 judge calls, claude-4.7-opus). A single-judge deployment has no inter-annotator agreement; reliability is established separately in Section 3.

Cross-metric family profiles. Combining the three deployed judge metrics gives a legible per-family risk profile: GLM-4.5 is self-deceptive and near-silent (calibration 1.80, thought-rate 1.5–4.8%, but best late-persistence 11%); GPT-5 is cautious but persistent (thought-rate 11.1%, worst late-persistence 39%); Kimi is expressive but stuck (thought-rate 20.3%, late-persistence 39%); GLM-4.7 is the most calibrated (2.11); Claude-4 is balanced. The three metrics' Spearman correlations with eval score are 0.21 (Self-Eval Calibration), ≈ 0.05 (thought-rate), and -0.15 (late persistence).

M JUDGE-COMPOSITE LEADERBOARD AND TASK BEHAVIOR: FULL TABLES

Table 40 ranks all 20 InnovatorBench tasks by pass rate (fraction of runs scoring > 0); Table 41 gives the pilot inter-rater reliability with discrimination tiers and the recommended metric configurations.

Recommended metric configurations. From the evaluation report, two deployment bundles are recommended. *Lite-6*: Trajectory Satisfaction (0.81), Failure Pattern (0.78), Evidence-to-Action Gap (0.90), Subtask Propagation (0.82), Feedback Utilization (0.97), Error Recovery (0.91). *Full-10* adds Per-Error-Type Solidification (0.97), Error Solidification (0.91), Dual-Channel Divergence (0.80), Cross-Stage Violations (0.74). Not recommended (low reliability): Interaction Precision, Question Quality, Thinking Hedging Frequency (ICC 0.11), Hypothesis-Verify Loop (ICC < 0.7). The layered cost tiers are: automatic rule metrics (7, free), judge metrics (15, $\sim$\$0.02 each), and human labels (2–3).

Status of these bundles. These are historical candidates from the source report, not current deployment recommendations. The pilot uses only 20 trajectories, agreement does not establish human validity, and the relation between the tightened snapshot and the baseline sweep is not fully documented. Any deployment requires a new, versioned reliability and human-validity study in the target environment.

Table 14: Prefix associations on swefactory. "Divergence @" is the first sampled prefix at which resolved/unresolved curves separate; it is descriptive and does not establish that intervening on the metric changes success.

Metric (plain)	Divergence @	End gap	Type
Distinct Files Viewed	0.1	2.849	early association
Files Reached	0.1	2.357	early association
Reasoning Coverage	0.1	0.15	early association
Read-Before-Edit Rate	0.1	0.142	early association
Post-Fix Verify	0.1	−0.036	early association
Action Transition Entropy (norm.)	0.1	0.03	early association
Effective Step Rate	0.1	−0.001	early association
Open-File Count	0.2	5.127	early association
Tool-Chain Depth (step count)	0.4	5.309	late symptom
Max View Count	0.4	1.393	late symptom

Table 15: Original distribution-stability screen across 10 sources (coefficient of variation of per-bench means). "#bench" records empirical coverage and must be read before the CV. The right-hand entries primarily diagnose structural absence.

Lowest CV	CV	#bench	Structural-absence signals	CV	#bench
Think-Success-Rate	0.013	3	Deadlock "busy" reports	3.000	10
Effective-Step-Rate	0.029	10	Planning-with-Subgoals	3.000	10
Budget-Utilization	0.029	10	Session Close-Observed	3.000	10
Post-Fix-Verify	0.032	8	Session Create-Count	3.000	10
Non-Think-Success-Rate	0.086	10	Session Leak	3.000	10
Effective-Density	0.123	10	Max Polling Run	3.000	10
Read-Before-Edit Rate	0.130	7	Polling Step Ratio	3.000	10
Param-Schema-Compliance	0.152	10	Planning-Node Usage	3.000	10
Blind-Edit Rate	0.218	7	Summary-Node Usage	3.000	10
Mean-Edits-per-Path	0.265	7	Format Implicit-Finish	3.000	10
Burn-Window Size	0.275	10	Truncated Rate	3.000	10
Retry-Improved (max)	0.284	4	Max Consecutive Deadlock	3.000	10

N NARROW NEGATIVE: HACK HEURISTICS VERSUS PASS/FAIL

On 26,849 swefactory trajectories (11,610 passed / 15,239 unresolved), we sum hand-crafted "reward-hacking suspicion" signals—test-file edits, hardcode occurrences, skip markers, git-checkout of test files, blind retries, edit/test ratio, and empty assistant turns—into a hackscore. Across thresholds $2/4/6/8$, it predicts the *unresolved* label at precision 0.560–0.566, essentially the base rate (0.5676), with recall 0.88–0.97: no discriminative power. Some "hack-like" signals are higher in passed trajectories because editing and running tests is normal debugging. The label here is pass/fail, not hack/clean, so this is not a validation of true reward-hacking detection; hidden-oracle hack labels are required.

O CASE STUDIES

Loop-Trap (Search-Augmented-RL, Task 16, final score 0). An agent edited the same 12 lines 58 times without ever running a test, exhausting its budget. FGA signature: maximal Blind-Retry, high Repeated-View, zero Post-Fix-Verify. **Good-Recovery** (NL2SQL, score 4). After a HuggingFace download failed, the agent pivoted to a synthetic dataset in one step; after a ray failure, it restarted in two steps—flagged positively by the Error-Recovery judge. **Hypothesis-Verify** (Data-Cleaning, score 4). The agent backs up, cleans, validates, evaluates, and adjusts in a disciplined loop. These cases illustrate process distinctions; none is claimed to be the causal basis of the Gold-v1 SFT result.

Table 16: Cross-bench discriminative magnitudes on two binary-compatible local exports: internal `swefactory` and an AgencyBench-derived export (internal tag "V3.0"). Values are $|2 \cdot \mathrm{AUC} - 1|$; direction must be checked separately before calling a field consistent. This table is distinct from the historical screen in Figure 7D.

Metric field (plain)	swefactory	internal V3.0 tag	min
Max-Edits-per-Path	0.115	0.152	0.115
Mean-Edits-per-Path	0.103	0.145	0.103
Per-Path Reuse Rate	0.103	0.145	0.103
Explore-Edit Log-Ratio (unclipped)	0.122	0.084	0.084
Explore-Edit Log-Ratio	0.122	0.084	0.084
Post-Run-Command Success (count)	0.087	0.078	0.078
Post-Run-Command (count)	0.092	0.078	0.078
Action Transition Entropy (norm.)	0.190	0.077	0.077
Param-Compliant (count)	0.076	0.103	0.076
Explore-Edit Ratio	0.144	0.070	0.070
Post-Edit Actions (count)	0.074	0.068	0.068
Tool-Call Count	0.076	0.063	0.063
ReAct Count	0.076	0.063	0.063
Action Bigram Count	0.075	0.063	0.063
Step Count	0.076	0.063	0.063

Table 17: Compact judge-composite leaderboard. The 15,720 count denotes trajectory–metric judgments, not the product of model and metric counts alone. Trajectory-Satisfaction uses its own rubric scale.

Model	Composite	Note
kimi-k2.5	3.30	Trajectory-Satisfaction 5.24
claude-4-sonnet	3.07	
glm47-codeact	3.06	
glm47	3.02	
glm-4-5-full	2.96	
glm47-sweagent	2.88	
kimi-k2-instruct	2.88	
gpt-5	2.84	
grok-4	2.78	
grok-code-fast	2.58	
kimi-instruct	1.76	

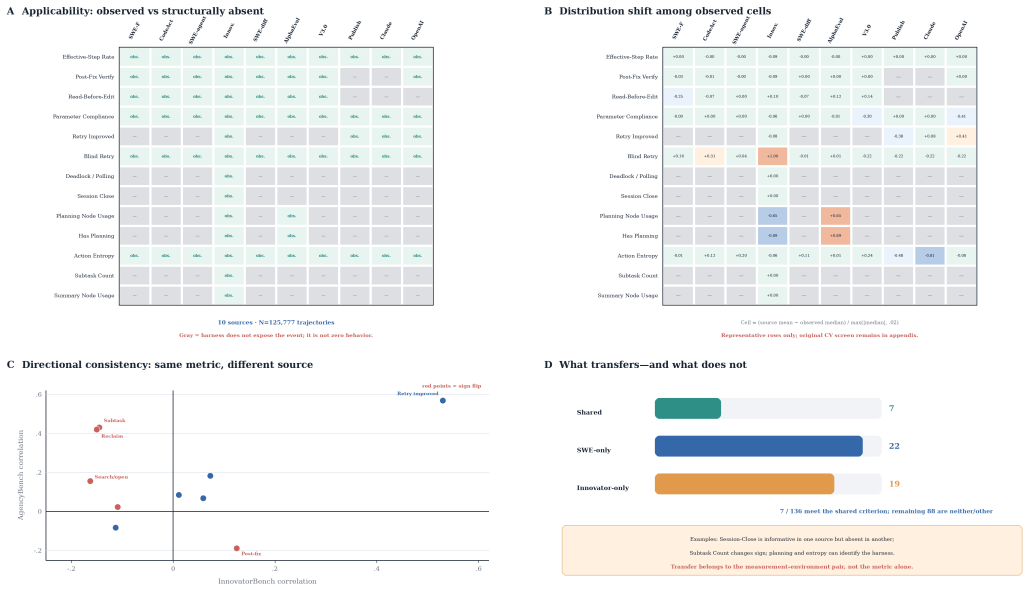


Figure 7: Cross-source applicability and transfer. Coverage distinguishes observed behavior from structural absence; among observed cells, distribution and outcome associations can change across sources. Panel D preserves a separate historical 136-field screen; it is not supported by Table 16, which reports the current binary-compatible comparison.

Table 18: Dataset overview and analytical role. The ten local exports contain 125,777 rollout records: three have binary outcome labels, InnovatorBench has partial graded evaluation, and six are label-free. Counts are locally filtered rollout records, not public benchmark task counts; Table 5 uses separate historical scored exports.

Source	Domain / harness	N	Outcome	Main analytical use
internal `swefactory` export	SWE / CodeAct-style lineage (Wang et al., 2025)	95,828	Binary pass (57.5%)	Outcome complementarity; transfer; SFT universe
internally labeled SWE-CodeAct export (Wang et al., 2025)	SWE / CodeAct lineage	17,996	None	Applicability and distribution
internally labeled SWE-agent export (Yang et al., 2024)	SWE / SWE-agent lineage	10,000	None	Harness-format comparison
InnovatorBench	Long-horizon coding / mixed	661	Graded for 444	Graded association; behavior structure
SWE-difficult	SWE-difficult	580	None	Distribution stress test
AlphaEval-derived binary export (Lu et al., 2026a)	Agent-evaluation harness	294	Export-supplied binary label (31.0%)	Source association
AgencyBench-derived binary export (internal tag "V3.0") (Li et al., 2026)	Interactive agency	230	Export-supplied binary label (63.9%)	Cross-source discrimination
AgencyBench publish	Publish workflow	112	None	Applicability and distribution
AgencyBench claude	Claude workflow	40	None	Applicability and distribution
AgencyBench openai	OpenAI workflow	36	None	Applicability and distribution

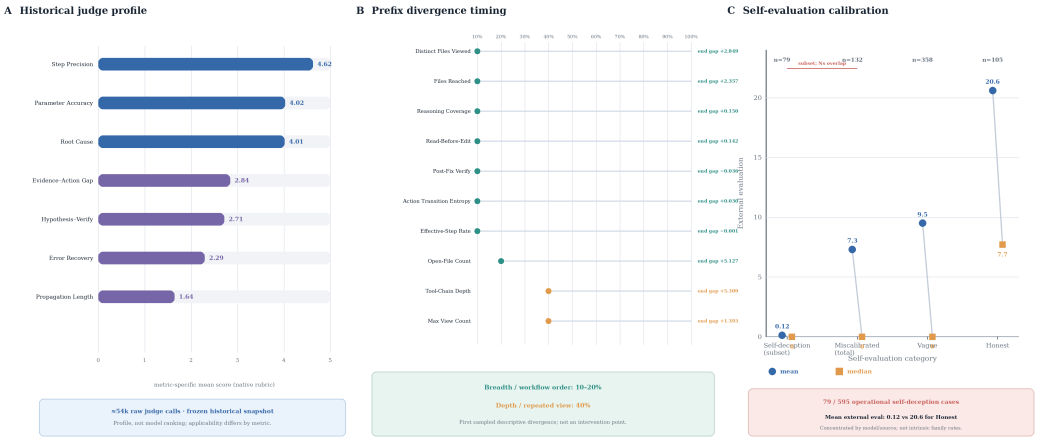


Figure 8: What FGA reveals at corpus scale. (A) In a HISTORICAL SNAPSHOT of approximately 54,000 raw judge calls, local execution scores exceed cross-step control scores after metric-specific applicability filtering. (B) In the CURRENT CORE, recomputing 95 swefactory measurements at trajectory deciles shows breadth and workflow-order differences by 10–20%, while depth and repeated-view signals first appear at 40%; end gaps retain native units and are not magnitude-comparable. (C) In the CURRENT CORE semantic sample, the self-deception subset has external-evaluation mean 0.12 and median 0, versus 20.6 and 7.7 for the honest group. These panels are behavioral profiles, not capability rankings.

Table 19: Model-level statistical-debiasing layer (InnovatorBench).

Analysis	Result
Token-Controlled Success	regress score on log-tokens + task fixed-effects: $R^2 = 0.397$, $F = 7.3$; top residual kimi-k2.5 $+19.5$, glm47 $+8.3$; bottom kimi-k2-instruct-0905 -10.5
Step-Controlled Success	$R^2 = 0.374$, $F = 6.6$; top kimi-k2.5 $+20.1$
Reasoning-Length-Controlled Success	OLS $n = 655$, $R^2 = 0.379$, $F = 6.80$
Verbosity-Normalized Score	$n = 655$, mean 0.731, Kruskal–Wallis $H = 123.88$, $p = 1.09 \times 10^{-19}$
Model×Attribute Spearman	380 pairs, mean $\lvert\rho\rvert = 0.223$, max 0.996
Model×Attribute Std	275 pairs, mean 15.996
Bucket-wise Self-Diagnosis	62/275 monotonic, max gap 78.5

Table 20: Per-benchmark model leaderboards (first-generation discrimination report). InnovatorBench score is on a 0–100 scale, AgencyBench on 0–50.

InnovatorBench (top 10)				AgencyBench (top 10)			
#	Model	Score	n	#	Model	Score	n
1	kimi-k2.5	35.6	21	1	claude-4.5-opus	23.3	6
2	glm47	23.6	20	1	gemini-3-pro	23.3	6
3	glm47-codeact	17.4	29	3	claude-4.5-sonnet	23.0	9
4	claude-4-sonnet	16.9	207	4	gpt-5.2	22.3	11
5	glm-4-5-full-data	14.9	25	5	grok-4.1-fast	21.8	10
6	glm-4-5-wo-interact-filter	11.5	20	6	claude-4.5-opus	21.3	11
7	glm47-sweagent	9.8	21	7	kimi-k2-thinking	20.3	10
8	glm-4-5-w-interact-filter	8.9	86	8	gemini-3-pro	19.3	10
9	gpt-5	4.1	64	9	minimax-m2	19.2	6
10	kimi-k2-instruct	3.7	29	10	glm-4.6	15.9	10

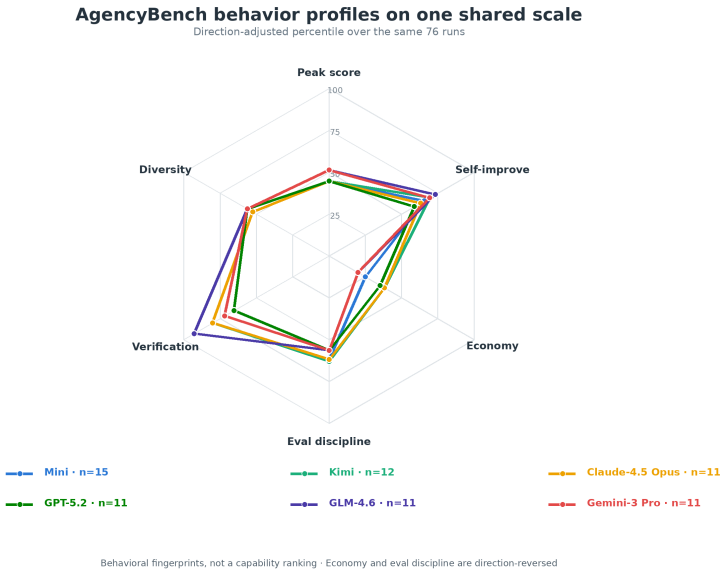


Figure 9: Six AgencyBench behavior profiles overlaid on one shared polar chart. Axes are direction-adjusted percentile ranks over the same 76 runs; the profiles describe harness-specific behavior, not capability.

Table 21: SWE-bench: three data sources (agent frameworks) differ by construction, which is why their top by-source effect sizes are large. Columns: trajectories, mean tool calls, mean error rate, Effective Step Rate, Action Entropy, tool variety, share using a think tool.

Data source	Traj.	Tool calls	Err. rate	Eff. step	Entropy	Tool var.	Think
codeact-glm47	11,984	93.3	1.42	0.846	1.22	4.2	98.7%
sweagent-glm47	11,591	98.0	1.54	0.823	1.04	3.0	0%
difficult_cls	3,090	47.4	2.10	0.760	1.10	3.0	0%

Table 22: RandomForest feature importances for predicting final score (score itself excluded). The top feature dominates and is score-derived, which is the source of the target-leakage caveat.

InnovatorBench ($R^2 = 0.876$)			AgencyBench ($R^2 = 0.993$)		
#	Feature	Imp.	#	Feature	Imp.
1	Retry Improved (delta)	0.383	1	Effective Subtask Rate	0.696
2	Think Effectiveness	0.049	2	Subtask Count	0.081
3	Output Tokens	0.049	3	Degradation Rate	0.043
4	Duration (Seconds)	0.047	4	Reasoning Degradation (count)	0.037
5	Explore-Edit Ratio	0.046	5	Total Attempts	0.036
6	Repeated View Rate	0.041	6	Chars-per-Point	0.034
7	Precursor (max consec. sleep)	0.037			
8	Action Entropy	0.036			

Table 23: Counter-intuitive discrimination findings (first-generation report). These are the empirical basis for the Role-4 "must-not-be-a-reward" counter-examples of Section 5.1.

Finding	Statistic
Models spending more time/tokens score higher	Duration $\rho = +0.32$, Tokens $\rho = +0.29$ (InnovatorBench)
Models with a higher deadlock-step share rank higher	Deadlock/Polling $\tau = +0.56$ (InnovatorBench)
On AgencyBench, more score-degradation correlates with higher totals	Score Degradation $\rho = +0.39$
Subtask Count points in opposite directions across benches	InnovatorBench $\rho = -0.15$; AgencyBench $\rho = +0.43$

Table 24: SWE-bench highly redundant metric pairs ($|\rho| > 0.8$; 8 of a reported 15 shown).

Metric A	Metric B	ρ	Recommendation
Read-Before-Edit Rate	Explore-Edit Ratio	$+1.000$	identical; delete one
Tool Call Count	Assistant Turns	$+1.000$	identical; delete one
Undo Count	Undo Rate	$+1.000$	identical; delete one
Read-Before-Edit Rate	Code-Locate Efficiency	-0.999	reciprocals
Total Chars	Observation Chars	$+0.989$	observation dominates
Unique Tool Count	Has Planning	$+0.950$	think-tool drives both
Think Count	Has Planning	$+0.934$	strong redundancy
Error Count	Error Rate	$+0.892$	keep rate

Table 26: Rule-based sub-variants derived from parent judge metrics (deployed dashboard). Negative ρ indicates rule-count variants that inversely track eval score.

Parent	Sub-variant	N	Mean	SD	ρ	H	Q	Rate
Error Recovery	Error Recovery Count	646	8.776	20.567	-0.013	114.06	2	D
Feedback Utilization	Feedback Utilization Rate	169	0.217	0.351	-0.069	15.58	1	D
Evidence-Action Gap	Evidence-to-Action Gap	646	0.357	0.205	-0.622	108.91	68	B
Hypothesis-Verify	Hypothesis-Verify Loop Rate	52	0.375	0.412	-0.021	9.02	0	D
Effective Action Ratio	Effective Action Ratio	655	0.711	0.166	0.421	138.81	58	B
Error Solidification	Error Persistence Rate	411	0.302	0.428	-0.326	31.02	10	C
Dual-Channel	Dual-Channel Divergence	55	0.151	0.224	-0.495	8.88	4	D
Cross-Stage Violations	Cross-Stage Violations	654	0.240	0.451	-0.140	34.54	5	D
Hedging Frequency	Thinking Hedging Frequency	64	0.303	0.168	-0.000	9.14	0	D
Uncertainty Calibration	Uncertainty Calibration Error	64	0.166	0.135	-0.179	6.07	1	D
CAT Failure	CAT-class Failure Total	646	5.090	17.502	-0.156	102.03	16	C
Stage-wise Error Type	Stage Error Severity	595	1.039	1.211	-0.155	78.76	12	C

Table 27: Top rule-metric discriminators by cross-model η^2 (deployed dashboard auto-metric panel).

Metric (component)	η^2
Duration (Seconds)	0.2891
Session Leak	0.2269
Tool Chain Depth	0.2228
Precursor Signals (max consecutive sleep)	0.1910
Deadlock Step Ratio	0.1230

Table 28: Per-task evaluation coverage across the 20 InnovatorBench tasks (deployed dashboard). "No-Eval" = runs that never triggered evaluation.

Task	Scored	Total	Best model	Best score	Families	Zero	No-Eval
1	10	10	glm-4-5-full-data	96.4	7	0	0
2	10	10	claude-4-sonnet	39.4	7	0	0
3	10	10	glm-4-5	18.9	6	0	0
4	10	10	kimi-k2.5	100.0	8	0	0
5	10	10	claude-4-sonnet	23.2	8	0	0
6	10	10	glm-4-5-full-data-wo-interact	100.0	7	0	0
7	10	10	claude-4-sonnet	25.8	8	0	0
8	10	10	glm47-codeact	38.0	8	0	0
9	2	10	glm47	60.8	5	5	3
10	1	10	kimi-k2.5	15.5	6	6	3
11	10	10	claude-4-sonnet	86.3	6	0	0
12	7	10	kimi-k2.5	95.4	7	2	1
13	10	10	glm47	31.4	6	0	0
14	6	10	claude-4-sonnet	37.4	4	1	3
15	9	10	glm-4-5-full-data-w-interact-w-filter	75.7	6	0	1
16	0	10	claude-4-sonnet	0.0	4	9	1
17	10	10	glm47	55.9	6	0	0
18	5	10	glm47-sweagent	70.0	5	0	5
19	10	10	gpt-5	92.5	7	0	0
20	10	10	gpt-5	87.8	6	0	0

Table 29: Analysis 1 — decision-quality decay. Left: logistic main effects ($N = 103{,}027$ node-observations). Right: per-model within-model position coefficient (odds ratio per unit position).

1a. Main effects

Variable	Coef.	OR	p
Position (no controls)	$+0.048$	1.05	0.182
Position (controls: model+length)	$+0.106$	1.11	0.022^{*}
log trajectory length	-1.337	0.26	$\approx 0^{***}$
Quadratic position (nonlinearity)	$+1.008$	—	$2 \times 10^{-12\,***}$

1b. Per-model position effect

Model	Coef.	OR	N
glm-4-5-full-wo-interact	-2.142	0.12	4,069
kimi-k2-instruct	-10.674	≈ 0	2,850
grok-4	-1.080	0.34	1,062
glm-4-5-w-interact-filter	-0.593	0.55	14,180
claude-4-sonnet	-0.152	0.86	23,951
gpt-5	-0.056	0.95	6,591
kimi-k2.5	$+0.980$	2.66	5,114
glm47-codeact	$+2.904$	18.24	5,971

Table 30: Analysis 2 — information forgetting. Left: OLS main effects ($N = 5{,}060$ file-revisit events, $R^2 = 0.079$). Right: per-model first-access-position coefficient and mean file-revisit gap (steps).

2a. Main effects

Variable	Coef.	p
First-access position	-69.44	$6.8 \times 10^{-60\,***}$
log trajectory length	$+18.23$	$2.5 \times 10^{-23\,***}$

2b. Per-model

Model	Coef.	R^2	Gap
gpt-5	-114.2	0.251	60.7
glm47-codeact	-109.0	0.151	24.6
glm47	-96.2	0.088	39.3
kimi-k2.5	-93.6	0.130	40.9
claude-4-sonnet	-68.5	0.210	28.8
grok-4	-13.2	0.162	8.6

Table 31: Analysis 3 — strategy switching ($N = 66{,}665$ action-type switch events, $R^2 = 0.0008$). Best and worst switches by mean pre/post success-rate change.

Best switches			Worst switches		
From	To	Δ	From	To	Δ
null	create_file	$+0.575$	run_command	null	-0.500
null	sleep	$+0.491$	open_file	null	-0.515
null	edit_file	$+0.388$	sleep	null	-0.491
null	run_command	$+0.343$			

Table 32: Analysis 4 — context saturation (panel regression over deciles, $N = 6{,}240$). Position×length interaction on action diversity: $+0.053$, $p = 4.6 \times 10^{-8}$.

Dependent variable	Pos. coef.	p	Len. coef.	p	R^2
Action diversity	$+0.086$	1.0×10^{-44} ***	-0.205	≈ 0***	0.550
Repetition rate	-0.089	1.5×10^{-43} ***	$+0.180$	≈ 0***	0.482
Think ratio	-0.032	6.5×10^{-38} ***	-0.012	6.7×10^{-24} ***	0.154

Table 33: Analysis 5 — full judge-metric sensitivity to trajectory length (per-metric OLS: score $\sim \log$ length + model fixed effects). Rows above the rule are length-decreasing; below are length-increasing.

Metric	Len. coef.	p	Simple R^2	Full R^2	N
CAT Failure Rate	-0.609	1.1×10^{-51} ***	0.272	0.430	631
Step-Level Failure Recall	-0.413	1.1×10^{-22} ***	0.056	0.253	655
Step-Level Failure Precision	-0.364	3.6×10^{-22} ***	0.204	0.295	655
Error Solidification	-0.360	1.6×10^{-18} ***	0.054	0.331	655
Effective Action Ratio	-0.283	7.2×10^{-14} ***	0.023	0.228	655
Root-Cause Attribution	-0.278	1.8×10^{-8} ***	0.080	0.202	611
Per-Error-Type Solidification	-0.216	1.0×10^{-7} ***	0.003	0.114	653
Step-Level Failure F1	-0.193	7.9×10^{-6} ***	0.001	0.156	655
Parameter Value Accuracy	-0.189	2.7×10^{-9} ***	0.009	0.118	655
Cross-Stage Constraint Violations	-0.175	1.2×10^{-3} **	0.005	0.087	585
Dual-Channel Reasoning Divergence	-0.126	1.7×10^{-4} ***	0.004	0.123	655
Propagation Length	-0.047	—	—	—	—
Evidence-to-Action Gap	-0.013	—	—	—	—
Interaction Precision	-0.003	—	—	—	—
Subtask Dependency Propagation	$+0.001$	—	—	—	—
Question Quality	$+0.058$	0.012*	0.027	0.226	655
Error Recovery Count	$+0.102$	0.014*	0.085	0.285	591
Feedback Utilization	$+0.106$	6.3×10^{-3} **	0.064	0.247	655
Failure Pattern	$+0.165$	1.6×10^{-4} ***	0.086	0.240	655
Hypothesis-Verify Loop Rate	$+0.186$	2.2×10^{-6} ***	0.095	0.232	655
Trajectory Satisfaction	$+0.485$	1.7×10^{-6} ***	0.095	0.225	654

Table 34: Self-Eval Calibration by model super-family. Self-deception = score$= 1 \wedge$ success claim $\wedge$ eval< 5; the 79 cases sum to $66 + 6 + 0 + 1 + 1 + 5$. "Fake succ." = share of that family's success claims that are false.

Family	N	Calib. mean	Score 1	Score 2	Score 3	Self-decep.	Fake succ.
GLM-4.5	207	1.80	102	44	61	66	55.9%
Kimi	69	1.94	12	49	8	6	42.9%
GPT-5	64	2.02	0	63	1	0	0%
Grok	36	2.03	1	33	2	1	—
Claude-4	149	2.05	9	123	17	1	10.0%
GLM-4.7	70	2.11	8	46	16	5	41.7%

Table 35: Left: Self-Eval Calibration by eval-performance bucket. Right: eval-score separation by calibration score, showing the self-deception subset scores essentially zero.

Eval bucket	N	Calib. mean	% success claims
Zero (= 0)	343	1.90	21%
Low (0–10)	66	1.95	36%
Medium (10–30)	106	1.99	29%
Good (30–60)	51	2.02	27%
High (60–100)	29	2.41	45%

Condition	Mean eval	Median	N
Honest (score 3)	20.6	7.7	105
Vague (score 2)	9.5	0.0	358
Miscalibrated (score 1)	7.3	0.0	132
Self-deception subset	0.12	0.0	79

Table 36: Self-Eval Calibration per task, ranked by self-deception rate (all 20 tasks).

Task	N	Mean	S.d.	Rate	Task	N	Mean	S.d.	Rate
Task 16	25	1.72	11	44.0%	Task 3	28	1.86	3	10.7%
Task 9	26	1.77	8	30.8%	Task 15	28	2.11	3	10.7%
Task 13	33	1.85	9	27.3%	Task 17	34	2.09	3	8.8%
Task 12	32	1.91	8	25.0%	Task 20	36	1.78	3	8.3%
Task 14	28	1.93	7	25.0%	Task 4	25	1.84	2	8.0%
Task 10	21	2.00	5	23.8%	Task 11	27	2.04	2	7.4%
Task 18	28	2.04	4	14.3%	Task 5	28	2.07	1	3.6%
Task 2	35	1.86	4	11.4%	Task 7	28	2.00	1	3.6%
Task 8	36	1.97	4	11.1%	Task 6	33	2.18	1	3.0%
					Task 1 / Task 19	34/30	2.12/1.93	0/0	0%

Table 37: Left: Reasoning-Action Consistency step-level distribution ($n = 477$ judged steps; only one truly-inconsistent step). Right: thought-expression rate by family (mean 10.0%).

Score	Label	Count	%
1	Inconsistent	1	0.2%
2	Partial	58	12.2%
3	Consistent	418	87.6%

Model	N	Thought rate
Kimi-K2	29	20.3%
Claude-4-Sonnet	149	15.3%
GLM-4.7-CodeAct	29	14.7%
GPT-5	64	11.1%
GLM-4.7 (base)	20	10.6%
Grok-4	18	10.4%
GLM-4.5 (base)	47	4.8%
GLM-4.5-Interact-Filter	86	4.2%
GLM-4.5-FullData	25	1.5%

Table 38: Stage-Wise Error Type: dominant error-category share early (first 20%) / middle (20–80%) / late (last 20%). The judge returns 1,785 stage outputs; 1,784 fall into the six reported categories, with one late-stage UNKNOWN of severity zero. Persistence errors rise 4.5×; syntax errors are fixed early.

Error category	Early	Middle	Late
No error	90.1% (536)	83.7% (498)	70.1% (417)
Persistence error	4.2% (25)	10.4% (62)	18.7% (111)
Logic error	0.2% (1)	2.2% (13)	5.5% (33)
Resource error	0.7% (4)	2.2% (13)	3.5% (21)
Syntax error	2.9% (17)	1.2% (7)	0.5% (3)
Planning error	2.0% (12)	0.3% (2)	1.5% (9)
Mean severity	0.22	0.45	0.77

Table 39: Left: late-stage persistence-error share by family. Right: late persistence versus eval-score bucket (successful runs avoid late persistence).

Family	N	Late persist	Late no-error
GPT-5	64	39%	58%
Kimi	69	39%	43%
GLM-4.7	70	16%	76%
Grok	36	14%	61%
Claude-4	149	13%	72%
GLM-4.5	207	11%	81%

Eval bucket	N	Late persist	Late no-error
Zero ($= 0$)	343	22%	66%
Low (0–10)	66	12%	70%
Med (10–30)	106	20%	71%
High (30+)	80	8%	88%

Table 40: InnovatorBench behavioral index: 20 tasks ranked by pass rate (share of runs scoring > 0). 600 runs total; 263 scored, 172 zero, 165 no-eval.

#	Task	Total	Scored	Zero	No-Eval	Pass rate	Best run (score)
1	Task 4	25	19	1	5	76.0%	kimi-k2.5 (100.0)
2	Task 7	28	21	1	6	75.0%	claude-4-sonnet (25.8)
3	Task 19	30	19	2	9	63.3%	gpt-5 (92.5)
4	Task 5	32	20	2	10	62.5%	claude-4-sonnet (23.2)
5	Task 1	35	21	3	11	60.0%	glm-4-5-full-data (96.4)
6	Task 11	27	16	4	7	59.3%	claude-4-sonnet (86.3)
7	Task 6	33	19	3	11	57.6%	glm-4-5-full-wo-interact (100.0)
8	Task 20	36	20	7	9	55.6%	gpt-5 (87.8)
9	Task 3	28	15	5	8	53.6%	glm-4-5 (18.9)
10	Task 2	35	18	5	12	51.4%	claude-4-sonnet (39.4)
11	Task 17	34	16	10	8	47.1%	glm47 (55.9)
12	Task 8	36	16	10	10	44.4%	glm47-codeact (38.0)
13	Task 13	33	13	9	11	39.4%	glm47 (31.4)
14	Task 15	28	9	10	9	32.1%	glm-4-5-full-interact-filter (75.7)
15	Task 12	32	7	18	7	21.9%	kimi-k2.5 (95.4)
16	Task 14	28	6	17	5	21.4%	claude-4-sonnet (37.4)
17	Task 18	28	5	13	10	17.9%	glm47-sweagent (70.0)
18	Task 9	26	2	17	7	7.7%	glm47 (60.8)
19	Task 10	21	1	14	6	4.8%	kimi-k2.5 (15.5)
20	Task 16	25	0	21	4	0.0%	claude-4-sonnet (0.0)

Table 41: Pilot inter-rater reliability ICC$(2, 1)$ across 3 judges $\times$ 20 trajectories (two-step protocol), with discrimination tier (S ≥ 0.9, A 0.7–0.9, B 0.5–0.7, C 0.3–0.5, D < 0.3). Feedback Utilization appears twice in the source (a tightened-rubric 0.966 and a data-limited 0.358); both are preserved. Question Quality is not estimable because its ratings have zero between-item variance.

Metric	ICC	Tier	Metric	ICC	Tier
Per-Error-Type Solidification	0.973	S	Cross-Stage Violations	0.736	A
Feedback Utilization (tight)	0.966	S	Hypothesis-Verify Loop	0.505	B
Error Recovery Count	0.908	S	Feedback Utilization (ltd.)	0.358	C
Error Solidification	0.906	S	Interaction Precision	0.324	C
Evidence-to-Action Gap	0.904	S	Question Quality	n/e	—
Subtask Propagation	0.819	A			
Trajectory Satisfaction	0.808	A			
Dual-Channel Divergence	0.799	A			
Failure Pattern	0.783	A			