Beyond Final Success: Semantic Progress Diagnostics for Agent Trajectory Evaluation

Anonymous ACL submission

Abstract

Long-horizon agent benchmarks increasingly evaluate multi-step tool use and stateful interaction, but final success alone cannot show how far an agent progressed, when progress emerged, or why a run failed. We introduce the Auditable Semantic Reference Graph (ASRG), a framework for diagnosing semantic progress in agent trajectories. ASRG constructs task-level Semantic Reference Units (SRUs) without held-out evaluation trajectories, strictly matches trajectories to SRUs, and measures progress coverage, first-hit timing, unexplained actions, and structured failure attribution. We instantiate ASRG on 4,740 released Toolathlon evaluation trajectories. The diagnostics reveal trajectory-level progress differences and paired process differences among similarly successful models, and support efficiency analysis through progress timing and proxy-normalized progress costs. They also show that 20.46% of failed trajectories cover all measured process SRUs while still failing the original evaluator. Structured attribution suggests that many such near-completion failures are better understood as final-state or finalization-contract errors than as incomplete execution. ASRG complements outcome-level evaluation by turning semantic progress into auditable evidence for process efficiency and failure attribution.

1 Introduction

For a given task in a long-horizon agent benchmark, identical failure labels can hide substantially different trajectories. Some agents stall early, while others retrieve the correct evidence, call appropriate tools, and produce required artifacts, but still miss a final-state or submission-contract requirement. Outcome-level evaluation captures whether a task was completed, but it does not reveal how far the agent progressed, when progress occurred, or why the remaining failure arose. This gap matters as agent benchmarks increasingly evaluate multi-step tool use, web interaction, software applications, mobile or desktop control, and general assistant tasks (Shi et al., 2017; Yao et al., 2022; Zhou et al., 2024; Deng et al., 2023; Drouin et al., 2024; Li et al., 2020; Xie et al., 2024; Rawles et al., 2025; Yoran et al., 2024; Li et al., 2026).

Existing evaluation protocols provide limited visibility into this process. Final success rates collapse early failures, inefficient partial progress, near-completion failures, and ambiguous evidence gaps into a single outcome. Prior work has begun to analyze agents through progress, trajectory behavior, tool-use correctness, process verification, and diagnostic visualization (Lightman et al., 2023; Ma et al., 2024; Ou et al., 2025; He et al., 2026; Deshpande et al., 2025). However, long benchmark trajectories still lack a reusable reference layer for measuring semantic progress across tasks and models. Low-level efficiency measures such as token count, turn count, latency, and tool-call count quantify resource consumption, but they do not show whether the consumed budget produced meaningful task progress.

A reusable semantic progress diagnostic must handle multiple valid solution paths without relying on a single reference trace, fixed tool-call sequence, or brittle checklist. It should define semantic milestones that are checkable yet abstract enough to cover different valid behaviors. It must also keep references isolated from held-out evaluation trajectories, reject shallow or unsupported references, and produce auditable evidence rather than only unconstrained natural-language judgments over entire trajectories (Zheng et al., 2023; Lù et al., 2025).

We introduce the Auditable Semantic Reference Graph (ASRG), a framework for diagnosing semantic progress in agent benchmark trajectories. ASRG represents each task with Semantic Reference Units (SRUs), checkable reference points for meaningful progress toward task completion. SRUs are syn-

thesized from benchmark artifacts and reference-construction development trajectories, without using held-out evaluation trajectories. ASRG normalizes each evaluation trajectory, applies strict trajectory-to-SRU matching, and derives progress, timing, action-coverage, and failure-attribution evidence.

We instantiate ASRG on released Toolathlon evaluation trajectories from 22 models across 108 tasks. Semantic progress aligns with final success after model-level aggregation, but it also exposes trajectory-level structure and process differences among models with similar success. For successful trajectories, ASRG supports process-efficiency analysis by distinguishing earlier and more cost-efficient progress. For failed trajectories, it identifies high-progress cases that final success alone cannot separate from low-progress failures. In Toolathlon, 747 failed trajectories achieve $\mathrm{PC} = 1$ under process SRUs, accounting for 20.46% of failed evaluation trajectories and appearing across all 22 models. Structured attribution suggests a shift toward wrong final result or state and finalization-contract errors, while the category labels remain judge-assisted diagnostics rather than human-certified causal labels.

The paper makes three contributions. First, it introduces SRUs as data-isolated, checkable task-level references for measuring semantic progress in long-horizon agent trajectories. SRUs provide a reference layer between raw trajectory traces and final outcome labels, enabling progress measurement without held-out evaluation trajectories. Second, it develops ASRG as a strict trajectory-to-SRU alignment framework that computes progress coverage, first-hit timing, and action-level unknown rate while keeping these primary metrics independent of free-form LLM judgment. Judge-assisted failure attribution remains a separate evidence-bounded layer for failed trajectories. Third, it instantiates ASRG on 4,740 Toolathlon evaluation trajectories from 22 models across 108 tasks, showing that semantic progress adds information beyond final success, that models with matched success can differ substantially in process profile, and that many failed runs are high-progress candidates whose remaining errors often involve final-state or finalization-contract requirements.

2 Progress-Aware Trajectory Diagnostics

We study a post-hoc diagnostic setting for existing agent benchmarks. Let $\mathcal{T}$ be a set of tasks, $\mathcal{M}$ a set of agent systems, and $\tau_{m,t,i}$ the i-th trajectory produced by agent m on task t. Each trajectory contains agent messages, tool calls, observations, intermediate artifacts, and final responses. The benchmark provides an outcome label $y_{m,t,i} \in \{0, 1\}$ indicating whether the task-level success condition is satisfied.

Outcome labels anchor task completion, but they do not expose the process behind that outcome. A progress-aware diagnostic layer should preserve the outcome label as the completion anchor while adding measurements of semantic progress, first-hit timing, action coverage, and failure evidence.

For each task t, we define a set of Semantic Reference Units (SRUs), $\mathcal{R}_t = \{r_1, \ldots, r_k\}$. Each SRU is a checkable reference point associated with meaningful progress toward task completion. SRUs provide an auditable measurement layer for semantic progress. Satisfying an SRU indicates that the trajectory reached a task-relevant state or operation, while terminal success may still depend on additional constraints, alternative evidence, or final-state conditions outside a particular unit.

Given $\tau_{m,t,i}$ and $\mathcal{R}_t$, the diagnostic problem is to assign each SRU a status and, when applicable, a first-hit position. Strict hits support the main progress metrics, while weaker or unresolved statuses preserve uncertainty for sensitivity analysis and audit. The resulting diagnostics cover strict SRU coverage, the point at which progress first appears, the share of the action stream outside the explanatory coverage of the action labeler, and evidence for broad failure modes such as wrong final result or state, finalization-contract error, blocked or thrashing execution, evidence gap, and task misframing.

3 ASRG for Semantic Progress Diagnostics

ASRG operationalizes the diagnostic view in Section 2 as reference-based semantic-progress measurement. It represents each task with SRUs, aligns evaluation trajectories to these references through strict matching, and estimates progress coverage, first-hit timing, and action-coverage uncertainty. Failure attribution is modeled as a separate evidence-bounded layer applied only to failed

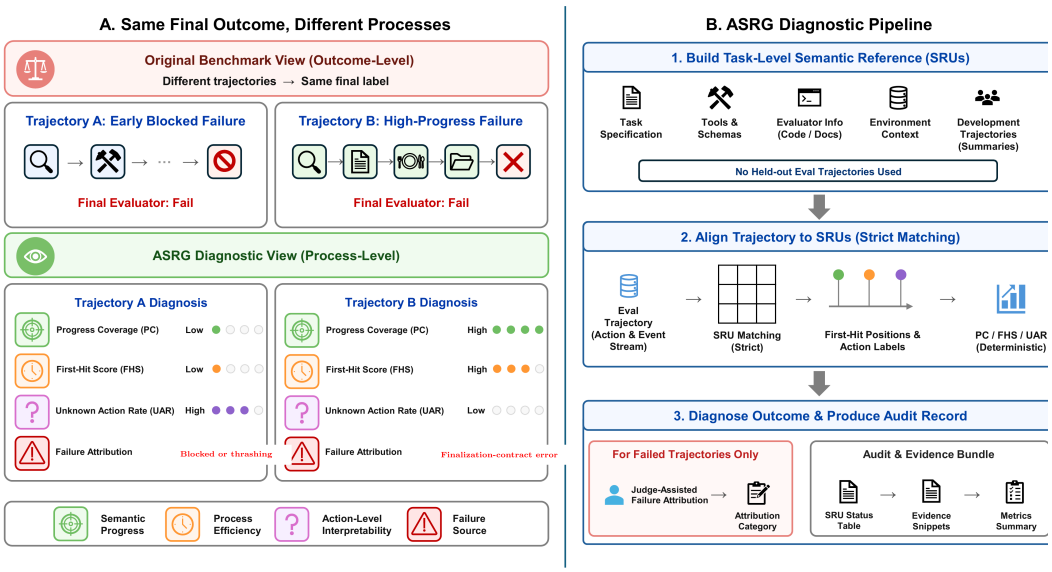


Figure 1: ASRG motivation and workflow. The figure contrasts early blocked and high-progress failures, then summarizes reference construction, strict matching, and diagnostic outputs.

trajectories, so PC, FHS, and UAR remain independent of free-form attribution judgments.

3.1 Reference Construction and Trajectory Evidence

For each task t, ASRG constructs a task-level reference set $\mathcal{R}_t$ of SRUs. An SRU is a checkable semantic reference point, such as identifying a required entity, retrieving task-relevant evidence, producing an intermediate artifact, satisfying a terminal condition, or preserving an invariant. SRUs describe semantic milestones rather than fixed traces. Different trajectories may satisfy the same SRU through different tool calls, observations, or intermediate states.

SRUs are synthesized once per task rather than per trajectory. Construction uses benchmark artifacts, including task instructions, tool schemas, environment context, evaluator summaries or code excerpts, and summaries of development trajectories reserved for reference construction. Held-out evaluation trajectories are not used to construct SRUs. Candidate SRUs are filtered for observability, verifier support, and grounding in benchmark evidence before finalization. Each finalized unit specifies the information needed for strict verification and audit, including its statement, verifier, weight, evidence source, and timing policy. Appendix A.1 gives the full record schema and the status rules used during strict matching. The main progress metrics focus on process SRUs, namely global progress units and active, non-ambiguous branch-progress units.

ASRG standardizes each evaluation trajectory τ into an evidence view containing messages, tool events, observations, artifacts, and final responses. A deterministic action labeler marks actions as progress-supporting, terminal-supporting, verification, repeated failure, redundant, or UNKNOWN. These labels support action-coverage diagnostics and are not used as success labels; Appendix A.1 gives the full priority order.

3.2 Strict Matching and Process-Efficiency Metrics

For each trajectory τ and reference set $\mathcal{R}_t$, ASRG assigns a status to every SRU. The main matching mode is strict. A unit is counted as hit only when the trajectory contains sufficient structured evidence for the SRU verifier. For process SRUs, the verifier checks observable fields such as required tools or apps, argument keys or patterns, output patterns, task entities, state predicates, output availability, verifier thresholds, anti-patterns, and multi-event evidence-chain requirements. Weak, unknown, and unverifiable evidence states are retained for audit and sensitivity analysis, but the main progress metrics count only strict hits.

Let $\mathcal{P}_t \subseteq \mathcal{R}_t$ denote the process SRUs for task t, let $w_r > 0$ be the stored SRU weight for unit r, and let $\mathbf{1}_{\text{hit}}(r, \tau)$ indicate whether r is strictly hit by trajectory τ. The weights are within-task salience

weights assigned during reference construction and normalized inside each metric denominator. They do not define a cross-task utility scale. Progress Coverage is

$$\mathrm{PC}(\tau, t) = \frac{\sum_{r \in \mathcal{P}_t} w_r \, \mathbf{1}_{\mathrm{hit}}(r, \tau)}{\sum_{r \in \mathcal{P}_t} w_r}. \tag{1}$$

PC measures strict semantic progress coverage. A trajectory with $\mathrm{PC} = 1$ has full strict coverage over the process SRUs, but this does not imply that every terminal evaluator condition is satisfied.

To capture process efficiency, ASRG also measures when progress occurs. Let $\mathcal{P}_t^{\mathrm{time}} \subseteq \mathcal{P}_t$ be the process SRUs with timing enabled, let $h_r(\tau)$ be the one-indexed first-hit tool-action position of unit r in trajectory τ, and let $H_t > 0$ be the task reference horizon. For unhit units, $h_r(\tau)$ is not used and the timing score is set to zero. In this instantiation, H_t is the median number of tool actions in successful development trajectories for task t. If no successful development trajectory is available, ASRG uses the median over development trajectories, with a minimum fallback horizon of one. The per-unit timing score is

$$\begin{aligned} s(r, \tau) &= \begin{cases} \tilde{s}(r, \tau), & \text{if strict hit,} \\ 0, & \text{otherwise,} \end{cases} \\ \tilde{s}(r, \tau) &= \max\left(0, 1 - \frac{h_r(\tau) - 1}{H_t}\right). \end{aligned} \tag{2}$$

The auxiliary $\tilde{s}(r, \tau)$ is the nonzero timing score assigned to a strictly hit unit. The trajectory-level First-Hit Score is

$$\mathrm{FHS}(\tau, t) = \frac{\sum_{r \in \mathcal{P}_t^{\mathrm{time}}} w_r s(r, \tau)}{Z_t}. \tag{3}$$

Here $Z_t = \sum_{r \in \mathcal{P}_t^{\mathrm{time}}} w_r$ is the positive within-task normalizer over process-timed SRUs. Higher FHS indicates that semantic progress appears earlier relative to the task horizon, which corresponds to greater process efficiency because the agent reaches task-relevant milestones with fewer actions before verification or finalization.

Finally, ASRG uses Unknown Action Rate as an action-coverage guardrail. Let A_τ be the sequence of labeled actions in τ, with $|A_\tau| > 0$, and let $\ell(a)$ be the deterministic action label for action a:

$$\mathrm{UAR}(\tau) = \frac{|\{a \in A_\tau : \ell(a) = \textsc{unknown}\}|}{|A_\tau|}. \tag{4}$$

Higher UAR indicates that more of the action stream falls outside the explanatory coverage of the current action-labeling layer. We interpret UAR as an action-coverage and interpretability diagnostic, not as a direct failure metric.

3.3 Failure Attribution

For failed trajectories, ASRG attaches structured failure-attribution evidence. The attribution judge receives an evidence bundle containing PC, FHS, UAR, unit-hit evidence, action labels, extracted events, trajectory excerpts, and a rule-based diagnostic summary. It then assigns one mutually exclusive primary category, namely wrong final result or state, finalization-contract error, blocked or thrashing execution, evidence gap, or task misframing. ASRG retains the evidence links supporting each score and attribution, enabling targeted audit of high-risk subsets such as high-PC failures, high-UAR trajectories, and evidence-gap cases.

4 Instantiating ASRG on Toolathlon

We instantiate ASRG on Toolathlon, a long-horizon tool-use benchmark with task-specific evaluators and released trajectories (Li et al., 2026). We use the original evaluator labels as the completion anchor and add post-hoc diagnostics for semantic progress, timing, action coverage, and failure attribution.

4.1 Trajectories and Models

We use the released Toolathlon trajectories (Li et al., 2026) and normalize 7,116 trajectories, with 2,376 development trajectories for reference construction and 4,740 held-out evaluation trajectories for all reported results. The evaluation split covers 22 models and 108 tasks, with two runs for each available model-task pair. This yields 2,370 available pairs because 19 models cover all tasks and three models each miss two tasks. Trajectory-level results treat the two runs as separate trajectories, while task-paired model-level analyses first aggregate available runs within each model-task pair before comparing or resampling tasks.

The held-out evaluation split contains 1,089 successful and 3,651 failed runs according to the original Toolathlon evaluators. On average, each evaluation trajectory contains 36.03 steps, 25.12 tool calls, 9.89 assistant turns, and 1.02 user turns. For long-horizon, multi-turn trajectories of this kind, final outcome labels alone provide limited visibility into intermediate progress and failure mechanisms.

4.2 Toolathlon Instantiation Details

We instantiate the measurement design in Section 3 with Toolathlon task artifacts and reference-construction development trajectories. Held-out evaluation trajectories are excluded from SRU construction. The finalized references contain an average of 9.05 SRUs per task, with a range of 6 to 15; reference-construction details, schema details, and validation rules are reported in Appendix A.

Held-out trajectories are scored with strict matching and the deterministic action labeler described in Section 3. PC, FHS, and UAR are not directly scored by an LLM judge. For failed trajectories, failure categories are assigned by the structured attribution judge and are not revised by human audit. The targeted human audit is reported in Appendix J; it is used for calibration and risk discovery rather than for modifying the fixed ASRG measurements used in the main tables.

Cross-Benchmark Portability. We also test whether the reference and alignment interface of ASRG extends beyond Toolathlon by applying it to the non-visual subset of AgentRewardBench (Lù et al., 2025). Using only a benchmark-specific adapter, ASRG constructs task-level references and scores held-out trajectories from AssistantBench, WebArena, and WorkArena. Appendix B reports the resulting diagnostics, which show that PC, FHS, UAR, and repeated-failure measures expose cross-benchmark differences in process progress, action coverage, and looping behavior.

5 Semantic Progress and Process Efficiency

We evaluate ASRG by asking whether progress diagnostics are useful but non-redundant. They should align with final success after model-level aggregation, while still revealing trajectory-level and same-success process differences. We therefore analyze both aggregate alignment and the additional structure exposed by PC, FHS, and UAR.

5.1 Overall Diagnostic Distributions

Across the evaluation split, the average final success rate is 0.2297, while the mean Progress Coverage is 0.6365 with a median of 0.7000. This gap does not imply that failed trajectories should be counted as successful. It shows that many unsuccessful runs still make measurable semantic progress before missing the terminal conditions of the original evaluator.

First-Hit Score has a mean of 0.4001 and a median of 0.4122. Since higher FHS indicates earlier semantic progress, ASRG distinguishes trajectories that eventually reach similar milestones from those that reach them earlier and leave more budget for verification or finalization. The average UAR is 0.1810 over trajectories with at least one labeled action. We use it as an action-coverage and interpretability guardrail rather than as a direct failure metric, because unknown actions may reflect labeler coverage, missing evidence, or genuinely off-reference behavior.

5.2 Aggregate Alignment and Trajectory-Level Non-Redundancy

At the model level, final success correlates strongly with PC and FHS, with Pearson correlations of 0.7424 and 0.6966, while UAR is negatively correlated with success at -0.6084. At the trajectory level, the same correlations are much weaker: 0.2160 for PC, 0.1081 for FHS, and -0.2294 for UAR. This contrast shows that ASRG tracks aggregate competence while exposing process variation that final success does not explain at the individual-trajectory level. Model-level Spearman correlations show the same pattern, with 0.7520 for success versus PC and 0.6915 for success versus FHS, indicating substantial but imperfect alignment.

Cheap Proxies Are Weak Semantic Substitutes. Inexpensive resource proxies are not reliable substitutes for semantic progress. At the trajectory level, tool-call count, assistant turns, and total tokens correlate weakly or negatively with final success (-0.0224, -0.0867, and -0.0615) and only weakly with PC and FHS, with maxima of 0.1825 and 0.1231 across the three proxies. Model-level correlations are more mixed, because aggregate resource use partly tracks PC, but proxy correlations with final success remain below those of PC and FHS. Table 5 gives the full correlation summary. These results show that resource use can reflect execution scale, but it is not semantic progress.

Cost-Normalized Progress Separates Efficiency from Completion. ASRG also enables resource-efficiency analysis by normalizing budget by semantic progress. Claude 4.5 Opus has the highest success rate among the compared strong models (0.440), and Claude 4.5 Sonnet has the strongest median PC and FHS (0.800 and 0.395), giving the strongest absolute progress profile. Gemini 3 Pro Preview has lower success and median PC (0.366

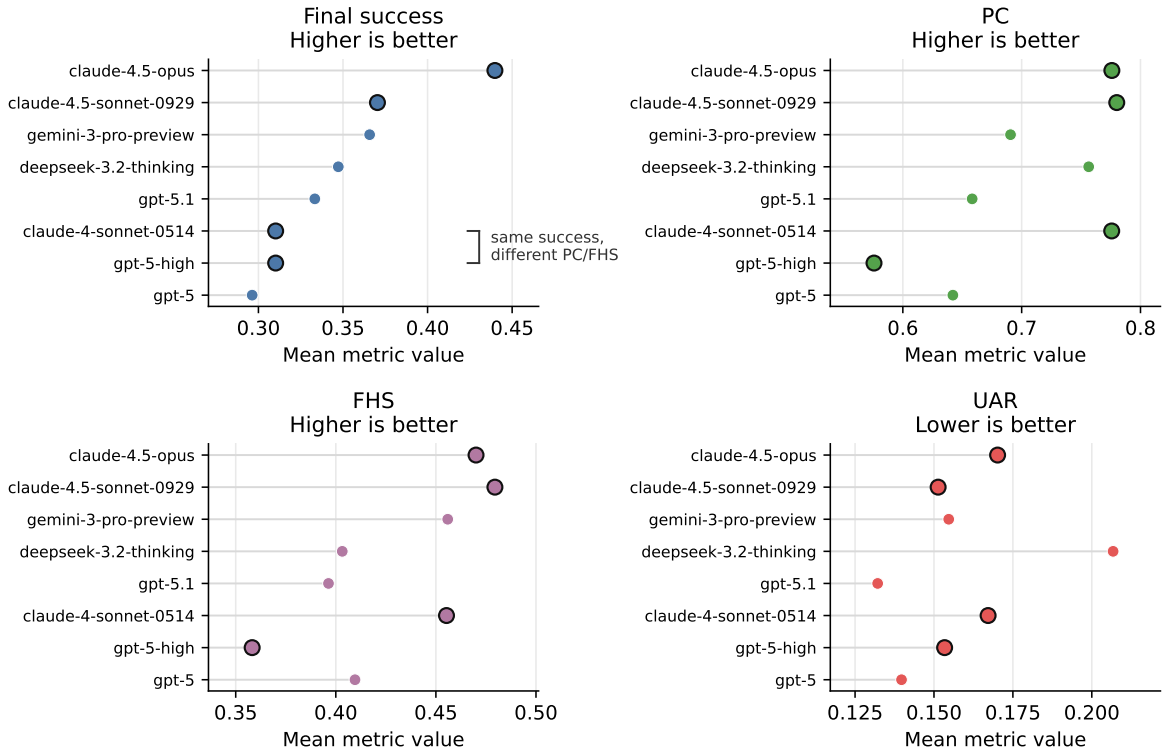


Figure 2: Model-level diagnostic rankings for the top 8 models by final success. All panels share the final-success order; Appendix F reports the full 22-model view.

and 0.588), but uses fewer tokens per unit of median PC and FHS: 510k tokens/PC and 983k tokens/FHS, compared with 866k and 1.554M for Claude 4.5 Opus and 1.148M and 1.938M for Claude 4.5 Sonnet. It also requires fewer tool calls per FHS, at 56.4 versus 69.2 and 67.2 for the two Claude models. The corresponding mean cost-per-progress values show the same qualitative pattern, so the comparison is not driven only by medians. These ratios should be read jointly with absolute PC and FHS rather than as standalone quality scores, since low cost per progress can also reflect shorter exploration. This comparison illustrates that ASRG provides the semantic denominator needed to analyze process efficiency beyond raw resource proxies.

5.3 Model-Level Process Differences Beyond Success

Model-level diagnostics preserve the broad outcome ordering, but they expose process profiles that final success alone cannot express. Figure 2 shows this pattern among the top eight models by final success. Claude 4.5 Opus has the strongest final-success result, while Claude 4.5 Sonnet has the strongest PC/FHS profile. We therefore do not treat the figure as a hard re-ranking of the leaderboard. Rather, it shows that outcome-level strength and process-level strength are related but not identical.

The clearest example is the same-success comparison between Claude 4 Sonnet and GPT-5 High. Both models achieve the same final success rate, 0.3102, but their trajectories look different under ASRG. Claude 4 Sonnet covers substantially more measured process progress, with a paired PC difference of 0.2000, and reaches progress earlier, with a paired FHS difference of 0.0971. The task-paired test in Appendix F confirms that this is not an artifact of comparing unpaired aggregate means. Thus, two models can solve the same fraction of tasks while differing in how consistently and how early they make task-relevant semantic progress.

This is the role of ASRG in model comparison. Final success remains the completion anchor, while PC and FHS describe the process by which trajectories approach that outcome. UAR adds a separate warning about how much of the action stream remains outside the deterministic explanatory coverage of the labeler. The next section applies this process view to failed trajectories, where high semantic progress can coexist with final evaluator

failure.

6 High-Progress Failures and Failure Attribution

We next use the gap between semantic progress and final success to analyze failed trajectories. The goal is not only to count failure categories, but also to test whether semantic progress changes what kind of failure remains visible after a trajectory has already covered task-relevant process evidence.

6.1 Failure Attribution over All Failed Trajectories

Attribution Separates Evidence Gaps from Terminal Errors. Across the 3,651 failed evaluation trajectories, ASRG assigns one judge-assisted attribution label to every failed trajectory. Wrong final result or state is the largest category, accounting for 2,202 failures, or 60.31%. Finalization-contract error accounts for 574 failures (15.72%), blocked or thrashing execution for 543 failures (14.87%), and evidence gap for 300 failures (8.22%). These totals are diagnostic category counts rather than human-certified causal rates. The metric profiles nevertheless show a clear separation. Evidence-gap failures have low mean PC and FHS (0.1517 and 0.1167) and high UAR (0.4196), while finalization-contract failures have much higher mean PC and FHS (0.7490 and 0.4923) and lower UAR (0.1550). This pattern indicates that attribution is not merely relabeling final failure. It separates trajectories with little usable task evidence from trajectories that make substantial measured progress before failing near the terminal contract.

6.2 High-Progress Failures

Progress Bins Trace the Failure Shift. We define a high-progress failure as a failed trajectory with $\mathrm{PC} = 1$, meaning full strict coverage over the measured process SRUs. Among the 3,651 failed trajectories, 747 satisfy this condition, accounting for 20.46% of all failures and 15.76% of all evaluation trajectories. The trend across PC bins is more informative than the endpoint alone. Among PC=0 failures, evidence gap and blocked or thrashing execution account for 59.81% and 27.41%, while finalization-contract error accounts for only 3.74%. Among PC=1 failures, evidence gap falls to 2.41% and blocked or thrashing execution falls to 7.10%, while finalization-contract error rises to 25.97%. At the same time, mean FHS increases from 0.0000 in the PC=0 bin to 0.6059 for PC=1 failures, and mean UAR drops from 0.6639 to 0.1101. Thus, high-progress failures are not simply longer failed runs. They are cleaner, earlier-progress trajectories whose remaining errors are concentrated closer to the final state or final contract.

Subset	N	Process PC	Terminal PC	All PC
All	4,740	0.6365	0.2297	0.3914
Success	1,089	0.7637	1.0000	0.8948
Failure	3,651	0.5986	0.0000	0.2413
PC=1 failure	747	1.0000	0.0000	0.4040

Table 1: Process and terminal coverage by outcome subset. Process PC measures strict coverage over measured process SRUs, Terminal PC is anchored to the original evaluator outcome, and All PC combines process, terminal, and invariant components.

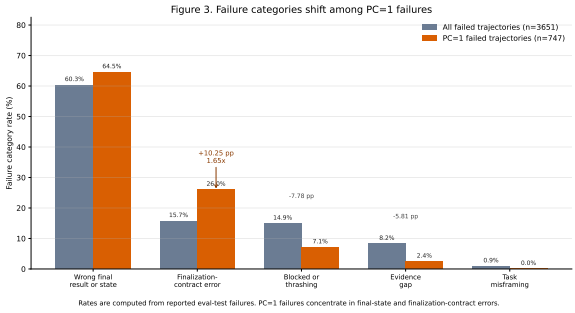


Figure 3: Failure-attribution shift among high-progress failures. Bars compare all failed trajectories with the subset that covers all measured process SRUs.

Process Coverage and Terminal Completion Are Separable. Table 1 makes the process/terminal boundary explicit. We report $\mathrm{PC}_{\mathrm{process}}$ as strict process-SRU coverage, $\mathrm{PC}_{\mathrm{terminal}}$ as terminal-goal coverage anchored to the original evaluator outcome, and $\mathrm{PC}_{\mathrm{all}}$ as the configured all-component SRU coverage over process, terminal, and invariant components. Across all trajectories, $\mathrm{PC}_{\mathrm{process}}$ is 0.6365, while $\mathrm{PC}_{\mathrm{terminal}}$ is 0.2297. For PC=1 failed trajectories, process coverage is 1.0000 but terminal coverage is 0.0000, with $\mathrm{PC}_{\mathrm{all}} = 0.4040$. This decomposition rules out a degenerate interpretation of PC=1 as hidden success. It shows that ASRG isolates measured process completion while preserving the original evaluator as the terminal completion anchor.

Figure 3 gives the endpoint view of the same PC-bin trend, so the category percentages should be read as judge-assisted diagnostic rates rather than as a separate causal result.

6.3 Case Study of Progress Without Final Contract Satisfaction

Boundary Between Process Completion and Final Contract. A representative boundary case comes from DeepSeek V3.2 Exp on cooking-guidance, where the task requires recipe-grounded dish recommendations and a shopping list for missing or insufficient ingredients. ASRG assigns $\mathrm{PC} = 1.0$, $\mathrm{FHS} = 0.6912$, and $\mathrm{UAR} = 0.0$, but the original evaluator still marks the trajectory as failed. The terminal evidence shows successful final writes, yet all three terminal goals remain not hit because the recommendation artifact uses shortened dish names rather than the exact recipe-server titles, and the shopping list mostly contains seasonings while omitting main ingredient deficits implied by the selected recipes, such as additional potato and winter melon. The diagnostic subcategory is final schema invalid. This case supports the aggregate high-progress failure analysis because ASRG identifies completed process evidence, but it does not convert missed final-contract satisfaction into task success.

7 Related Work

Agent benchmarks and tool-use evaluation. Recent benchmarks evaluate agents in web navigation, software applications, mobile or desktop control, embodied environments, and realistic assistant tasks (Shi et al., 2017; Shridhar et al., 2021; Wang et al., 2022; Yao et al., 2022; Deng et al., 2023; Liu et al., 2024; Zhou et al., 2024; Jimenez et al., 2024; Drouin et al., 2024; Koh et al., 2024; Lù et al., 2024; Xi et al., 2024; Zheng et al., 2024; He et al., 2024; Boisvert et al., 2024; Li et al., 2020; Xie et al., 2024; Rawles et al., 2025; Yoran et al., 2024; Le Sellier De Chezelles et al., 2024). Tool-use studies further analyze prompting, API selection, argument formation, and unsafe or underspecified tool interaction (Yao et al., 2023; Schick et al., 2023; Li et al., 2023; Patil et al., 2024; Tang et al., 2023; Qin et al., 2024; Huang et al., 2023; Ruan et al., 2024; Yao et al., 2024). Toolathlon is especially relevant because it provides long-horizon tool-use tasks, task-specific evaluators, and released trajectories (Li et al., 2026). ASRG is complementary to these benchmarks and tool-use evaluations: it keeps the benchmark evaluator as the outcome anchor and studies the semantic progress visible in completed trajectories.

Trajectory-level and process-aware evaluation. Prior work moves beyond final success through progress analysis, trajectory diagnosis, tool-use correctness, trace localization, and modular progress metrics (Ma et al., 2024; Ou et al., 2025; He et al., 2026; Deshpande et al., 2025; Gioacchini et al., 2024). These studies show that trajectory-level signals are useful, but they do not provide the same reference object as ASRG. AgentBoard analyzes progress in multi-turn agents, AgentDiagnose scores broad agentic competencies, TRAJECT-Bench focuses on tool selection, argument correctness, and dependency or order satisfaction, TRAIL studies trace reasoning and issue localization, and AgentQuest proposes modular progress metrics. ASRG instead constructs task-level SRUs without held-out evaluation trajectories and strictly matches held-out trajectories to these units to measure progress coverage, first-hit timing, and action-coverage uncertainty across models and tasks.

Failure attribution, judge reliability, and auditability. AgentRx, GUIDE, DiagEval, and AgentRewardBench study agent failures, evaluator-side errors, or automatic evaluator reliability (Barke et al., 2026; Zhai et al., 2026; Hong et al., 2026; Lù et al., 2025). ASRG addresses a different measurement target. It first computes reference-based PC, FHS, and UAR independently of attribution, then attaches judge-assisted failure categories only to failed trajectories for evidence-bounded error analysis.

8 Conclusion

ASRG turns semantic progress in agent trajectories into auditable measurement. With SRUs and strict matching, it separates final success from process coverage, progress timing, and action-coverage uncertainty, while keeping failure attribution as a separate evidence-bounded layer. Empirically, these diagnostics add information beyond final success: models with similar success differ in progress amount and timing, and many failed runs are high-progress trajectories that miss final-state or finalization contracts. ASRG therefore supports evaluation reports that explain process efficiency and residual failure modes after substantial measured progress.

Limitations

Synthesized semantic references. ASRG constructs SRUs through an LLM-assisted reference-construction process over benchmark artifacts and

development trajectories. Held-out evaluation trajectories are not used during SRU construction, and the generated references are passed through schema validation and quality checks before evaluation. Nevertheless, the resulting SRUs are not gold-standard human annotations. They may omit valid but less common progress paths, encode imperfect abstractions of task requirements, or include references for which evidence is difficult to verify in some trajectories. The reported PC and FHS values should therefore be interpreted as measurements relative to the constructed reference layer, not as exhaustive coverage of every possible solution path.

Conservative strict matching. The main progress metrics use strict trajectory-to-SRU matching. This design improves reproducibility and reduces over-crediting, but it can undercount implicit progress when a trajectory contains weak, indirect, or incomplete evidence. ASRG retains weak, unknown, and unverifiable statuses for audit and sensitivity analysis, but the main PC and FHS values are conservative estimates of measured semantic progress.

Judge-assisted attribution. Failure attribution is produced by a structured judge over evidence bundles containing progress metrics, unit-hit records, action labels, events, trajectory excerpts, and rule-based diagnostic summaries. These categories provide diagnostic evidence rather than gold-standard human causal labels. Our targeted human audit found high agreement on a small reviewed set. We use this audit for targeted checking and risk discovery, not for relabeling. Stronger causal claims about failure origins would require larger-scale human validation, multiple annotators, or adjudicated attribution labels.

Coarse failure taxonomy. The failure taxonomy is intentionally coarse to support aggregate analysis across many tasks and models. In particular, wrong final result or state aggregates several content-level error types, such as incorrect entities, incomplete artifacts, wrong values, or invalid state mutations. This level of granularity is sufficient for the main high-progress failure analysis of this paper, but future work could use the retained audit evidence and trajectory evidence to subdivide broad categories into finer-grained failure modes. More generally, applying ASRG to additional agent benchmarks would test how well the same reference-construction and attribution pipeline transfers beyond Toolathlon.

Ethical Considerations

This work analyzes released benchmark trajectories and does not deploy an agent system or interact with end users. The primary ethical risk is interpretive rather than user-facing. Progress scores and failure categories may be mistaken for definitive judgments of model capability or causal responsibility. We therefore frame ASRG as a diagnostic layer rather than a replacement for benchmark evaluators, report PC/FHS/UAR as measurements relative to a constructed reference layer, and treat failure attribution as judge-assisted diagnostic evidence rather than gold-standard human causal labeling.

The analysis uses LLM-assisted reference construction and judge-assisted failure attribution, which may inherit biases or inconsistencies from the underlying models. To mitigate overclaiming, progress metrics are computed through strict matching rather than free-form judgment, and attribution records are backed by inspectable evidence bundles. If diagnostic materials are released, they should preserve the anonymity and licensing constraints of the underlying benchmark trajectories and should not be used for high-stakes deployment decisions without human review.

References

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. 2026. AgentRx: Diagnosing AI agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault Le Sellier De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. 2024. WorkArena++: Towards compositional planning and reasoning-based common knowledge work tasks. In *Advances in Neural Information Processing Systems*, volume 37.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2Web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. 2025. TRAIL: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David

Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Workshop on Large Language Models for Agents at ICLR*.

Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, Kiril Gashteovski, David Friede, Roberto Bifulco, and Carolin Lawrence. 2024. AgentQuest: A modular benchmark framework to measure progress and improve LLM agents. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 185–193, Mexico City, Mexico. Association for Computational Linguistics.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.

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. 2026. TRAJECT-bench: A trajectory-aware benchmark for evaluating agentic tool use. In *International Conference on Learning Representations*.

Sirui Hong, Zhijie Liu, Tengfei Li, Wei Tao, Yifan Wu, and Chenglin Wu. 2026. DiagEval: Trajectory-conditioned diagnosis for reliable software evaluation with gui agents. *Preprint*, arXiv:2605.17439.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2023. MetaTool benchmark for large language models: Deciding whether to use tools and which to use. *arXiv preprint arXiv:2310.03128*.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand. Association for Computational Linguistics.

Thibault Le Sellier De Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Quentin Cappart, Graham Neubig, Ruslan Salakhutdinov, Nicolas Chapados, and Alexandre Lacoste. 2024. The BrowserGym ecosystem for web agent research. *arXiv preprint arXiv:2412.05467*.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, and 2 others. 2026. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *International Conference on Learning Representations*.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3116.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. 2024. WebLINX: Real-world website navigation with multi-turn dialogue. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 33007–33056. PMLR.

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. In *Conference on Language Modeling*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems*.

Tianyue Ou, Wanyao Guo, Apurva Gandhi, Graham Neubig, and Xiang Yue. 2025. AgentDiagnose: An open toolkit for diagnosing LLM agent trajectories. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 207–215, Suzhou, China. Association for Computational Linguistics.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2025. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*.

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. *arXiv preprint arXiv:2302.04761*.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. 2023. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024. Agentgym: Evolving large language model-based agents across diverse environments. *Preprint*, arXiv:2406.04151.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. AssistantBench: Can web agents solve realistic and time-consuming tasks? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8938–8968, Miami, Florida, USA. Association for Computational Linguistics.

Yuwen Zhai, Runze Li, Liang Wang, Nian Shi, Liwu Xu, Wei Zhang, Ran Lin, Bo Xu, and Benlei Cui. 2026. GUIDE: Interpretable gui agent evaluation via hierarchical diagnosis. *Preprint*, arXiv:2604.04399.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61349–61385. PMLR.

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. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*.

A Implementation and Reproducibility Details

This appendix documents how the reported diagnostics are produced and how their claims should be interpreted. It focuses on the implementation details, robustness checks, and audit records needed to interpret the main PC, FHS, UAR, and failure-attribution results.

Result provenance. All reported TOOLATHLON trajectory-level metrics, aggregate statistics, figures, and failure-attribution labels are computed from a single frozen TOOLATHLON evaluation snapshot. The snapshot covers all 4,740 expected evaluation trajectories, with no excluded trajectories. The diagnostic snapshot includes unit-hit evidence, action labels, event links, attribution evidence, and 108 finalized task references. The targeted manual audit in Appendix J is a separate validation layer over selected trajectories; it assesses metric reasonableness and attribution boundaries but does not modify the reported ASRG scores or labels. The reference-construction boundary is also fixed: SRUs are built in a within-task reference-construction setting, not as a task-held-out generalization benchmark.

LLM-assisted reference construction. The fixed run uses `mimo-v2.5-pro` for LLM-assisted SRU proposal and review. The proposal prompt requires observable task-level milestones, verifier-grounded evidence, and explicit separation from human-certified gold annotation. The review prompt checks whether candidate SRUs are observable, verifier-supported, and sufficient to cover the task objective. Failure attribution uses a separate evidence-grounded prompt over failed-trajectory packets. These LLM components construct references and attribution records, but they do not directly score PC, FHS, or UAR.

A.1 SRU Record and Matching Details

Table 2 gives a representative process-SRU record used by the strict matcher. The example is a process unit because it measures an intermediate milestone rather than final task success. Terminal-goal units use the same record schema, but in this Toolathlon instantiation their hit status is anchored to the original evaluator outcome rather than independently re-verified by ASRG.

The strict matcher assigns one status per SRU using the rules in Table 3. A strict `hit` requires both a selector match and task-semantic grounding. Matching only a generic app, tool, or keyword is insufficient for process scoring. Weak and unresolved statuses are retained in the diagnostic records, but the main PC and FHS metrics count only `hit` process SRUs.

The task horizon H_t used by FHS is not the maximum, mean, or a percentile over all trajectories. ASRG sets H_t to the median tool-action count among successful development trajectories for the task. If no successful development trajectory is available, it uses the median over development trajectories, with a minimum fallback of one. In the current Toolathlon instantiation, 77 task references use the successful-development median and 31 use the fallback development median or task-budget source.

Action labels are deterministic and single-label. Table 4 gives the priority order. When more than one rule could apply, ASRG assigns the first applicable label in this order and stores the related units or evidence links that triggered the label. This conflict-resolution rule keeps UAR reproducible. UNKNOWN means that none of the deterministic evidence rules explained the action.

B Cross-Benchmark Portability on AgentRewardBench

Cross-Benchmark Portability. We evaluate whether the reference and alignment interface of ASRG can be applied to external agent trajectories by using the non-visual subset of AgentRewardBench (Lù et al., 2025). With a benchmark-specific adapter, ASRG constructs task-level references and scores held-out trajectories from AssistantBench, WebArena, and WorkArena using the same SRU matching and diagnostic definitions as in the Toolathlon analysis. The run contains 1,001 normalized trajectories, 223 frozen task-level references, and 446 scored held-out trajectories. These results support cross-benchmark portability by showing that the same semantic progress diagnostics can be computed outside Toolathlon.

Domain-Level Profiles. The transferred diagnostics expose different process profiles across

Field	Example value
unit_id	PU_1
unit_type	progress; the unit enters process coverage and first-hit timing.
statement	The agent has read the ingredient table and parsed ingredient types, names, and quantities.
scope / required_for_success / timing_policy	global / false / process_timed.
weight	0.15. Weights are positive within-task salience weights assigned during reference construction and normalized inside each metric denominator.
source / confidence	task_instruction, spec_dev_trace / 0.95.
Development support	Development evidence stores observed step and tool-call identifiers supporting the unit during reference construction.
verifier.type	tool_event.
verifier selectors	Required app filesystem; required tools include file-reading tools; required argument key path; required argument pattern matches the ingredient-table filename.
verifier grounding	Requires available tool output, an output pattern matching the ingredient-table header, successful tool call, output grounding, minimum evidence 1, confidence threshold 0.85, and anti-patterns for error or missing-file outputs.
scoring evidence fields	Unit-hit records store status, first-hit tool-call index, first-hit step, matched event ids, matched tool-call indexes, matched evidence types, failed evidence requirements, weak-candidate event ids, and a verifier summary.

Table 2: Representative process-SRU record. The table shows the fields consumed by strict scoring; optional descriptive fields such as construction notes and observable predicates are retained with the record when available.

Status	Rule
hit	At least one candidate event satisfies the verifier confidence threshold, no anti-pattern fires, required tool/app/argument/entity/output conditions hold, required output or state evidence is available when requested, and the matched evidence includes task-semantic grounding. Multi-event chain hits require a grounded chain over usable events.
weak_candidate	Candidate events match shallow selectors, but semantic grounding, output pattern, entity match, evidence-chain requirement, or minimum evidence is insufficient.
unverifiable	Matching requires output or state evidence that is unavailable in the normalized trajectory record.
unknown	No usable evidence is available, event confidence is below threshold, branch activation is ambiguous, or an invariant lacks state or evaluator evidence.
not_hit	Strict matching fails without weak or unverifiable evidence; inactive non-ambiguous branch units are also treated as not hit.
preserved	Invariant status used when evaluator or state evidence supports preservation; otherwise invariants remain unknown in this instantiation.

Table 3: Operational SRU status rules used by strict matching. Terminal-goal units are mapped from evaluator success in this Toolathlon instantiation; process units are matched against structured trajectory evidence.

benchmark domains. WebArena shows the most balanced pattern, with success 0.473, PC 0.533, FHS 0.445, and UAR 0.319. WorkArena has lower success 0.279, lower PC 0.205, lower FHS 0.183, and higher UAR 0.704, indicating that fewer actions are matched to semantic milestones or effective recovery evidence. AssistantBench shows a contrasting pattern, with success 0.125 but PC 0.702 and FHS 0.600. This profile is consistent with the main paper finding that trajectories may execute recognizable intermediate steps even when the final benchmark answer is not accepted.

Looping and Redundancy. The clearest alignment with AgentRewardBench annotations appears in the looping dimension. Non-looping trajectories have success 0.526, PC 0.494, UAR 0.387, and RFR 0.011, whereas looping trajectories have success 0.093, PC 0.280, UAR 0.596, and RFR 0.042. The same direction appears under optimality labels: complete failures have PC 0.288 and UAR 0.522, while completely optimal trajectories have PC 0.488 and UAR 0.279. Across scored trajectories, UAR is positively correlated with looping ($r = 0.284$) and negatively correlated with optimality ($r = -0.257$); PC and FHS have weaker positive correlations with optimality ($r = 0.175$ and $r = 0.153$). These trends show that ASRG metrics capture process-level signals for redundancy and inefficient exploration in an external benchmark setting.

Reliability-Gated Profiles. Reliability gates identify the subsets where diagnostic readings are most interpretable. Branch diagnostics remain stable, with an ambiguous-branch rate of 0.018. Action diagnostics are closer to the boundary, with UAR 0.477 overall and UAR 0.121 on the action-reliable subset. On that subset, ASRG reports higher process quality, with SRU coverage 0.517, FHS 0.539, and success 0.449. The all-reliable subset is smaller but cleaner, with $n = 62$, success 1.000, SRU coverage 0.840, FHS 0.500, and UAR 0.088. These results show that the same reliability gates used for Toolathlon also help interpret cross-benchmark diagnostic evidence.

Priority	Label family	Condition
1	Direct or terminal support	The action is the first hit, or supporting evidence, for one or more SRUs; successful finalization events tied only to terminal units are labeled terminal support.
2	Failure and repeated failure	Error events are labeled failed; repeated error signatures are labeled repeated failed.
3	Recovery	A recovery event after an observed failure is labeled successful recovery.
4	Verification	New verification facts are labeled valid verification; duplicate verification facts are redundant.
5	Evidence reuse	Search, read, extract, or non-terminal compute events are chain-support progress if their evidence is reused by a later direct SRU hit; duplicate evidence is redundant.
6	Mutation	Mutations linked to violated invariants are harmful mutations; evaluator-confirmed successful finalization mutations are terminal support; otherwise they remain unknown.
7	Other terminal support	Other successful finalization-like events are terminal support when tied to evaluator-confirmed terminal success.
8	Unknown	If no rule applies, the action is labeled UNKNOWN.

Table 4: Action-label priority and conflict resolution. Labels support action-coverage diagnostics only and are not task-success labels.

C Strictness and Reliability Checks

Strict matching controls over-credit. The strongest extended diagnostic is the gap between strict hits and weak evidence. In the fixed Toolathlon result set, strict PC is 0.6365, while PC rises to 0.9527 if weak-candidate evidence is counted. The same pattern appears for timing: strict FHS is 0.4001, while FHS with weak evidence is 0.6742. These gaps show that many trajectories contain partial evidence for process milestones, but the main PC and FHS metrics deliberately withhold credit unless the strict verifier is satisfied. This supports the conservative interpretation of ASRG progress scores.

Reliability gates bound stronger readings. Reliability gates are useful as claim boundaries rather than as additional model scores. Action reliability covers 4,214 of 4,740 trajectories (88.90%), branch reliability covers 4,608 trajectories (97.22%), and invariant reliability is much narrower at 1,056 trajectories (22.28%). Their intersection contains 971 trajectories (20.49%). The all-reliable subset has lower UAR (0.0740) and IUR (0.0248) than the full set, indicating cleaner evidence coverage. Because these gates condition on evidence availability, the subset should not be treated as an independent performance estimate; it identifies where action, branch, and invariant diagnostics are most interpretable.

D Proxy Correlation Details

Table 5 reports Pearson correlations between inexpensive trajectory proxies and outcome or progress metrics. The trajectory-level rows use individual held-out trajectories, while the model-level rows use model means over the evaluation split. Model-level correlations are higher for PC than trajectory-level correlations, but the trajectory-level results remain weak, and raw resource proxies do not replace semantic progress diagnostics.

E Uncertainty Intervals

Table 6 reports 95% intervals for the main reported quantities on their original metric scales. UAR has a smaller denominator because action-reuse accounting is available only for trajectories with labeled actions. The PC=1 failure and closure rows serve as boundary checks for the high-progress failure analysis. In particular, the broad any-diagnostic closure quantity includes weak proxy closures, whereas the strong closure rate is the conservative quantity used for strict closure claims.

F Task-Paired Rank and Same-Success Checks

We report task-paired uncertainty checks for model-level progress claims. For each model and metric, we first average the two available evaluation runs within each task, then resample tasks with replacement to obtain 95% bootstrap intervals. Rank deltas are computed within each bootstrap sample as final-success rank minus process-metric rank, so positive values indicate that the process metric ranks the model higher than final success. Table 7 reports PC, FHS, and rank-delta intervals for every model.

The same-success comparison uses paired task-level differences and a sign-flip permutation test. Table 8 shows that Claude 4 Sonnet and GPT-5 High are tied under final success, but differ under PC and FHS. This supports the same-success claim without treating unpaired aggregate differences as sufficient evidence.

Three models, GROK-4, GROK-4-FAST, and GROK-CODE-FAST-1, are observed on 106 of 108 tasks. We therefore repeat the model-level analysis on the 106-task complete-case subset. The three

Level	Proxy	Success	PC	FHS
Trajectory	Tool-call count	-0.0224	0.1825	0.1231
Trajectory	Assistant turns	-0.0867	0.1648	0.0807
Trajectory	Total tokens	-0.0615	0.1209	0.0300
Model	Mean tool-call count	0.5998	0.7317	0.5464
Model	Mean assistant turns	0.1548	0.5423	0.3591
Model	Mean total tokens	0.4453	0.7796	0.5889

Table 5: Pearson correlations between inexpensive resource proxies and final success, PC, and FHS.

Metric	Estimate	N	95% interval
Success	0.2297	4,740	[0.2177, 0.2422]
PC	0.6365	4,740	[0.6271, 0.6461]
FHS	0.4001	4,740	[0.3935, 0.4063]
UAR	0.1810	4,595	[0.1752, 0.1873]
PC=1 failure rate	0.1576	4,740	[0.1477, 0.1673]
PC=1 strong-closed rate	0.0032	4,740	[0.0017, 0.0049]

Table 6: Uncertainty intervals for reported result quantities.

incomplete models keep the same success, PC, and FHS ranks under complete-case evaluation. Other changes are adjacent swaps with maximum absolute movement of one rank position. This sensitivity check indicates that the reported rank-change claims are not driven by the two missing tasks.

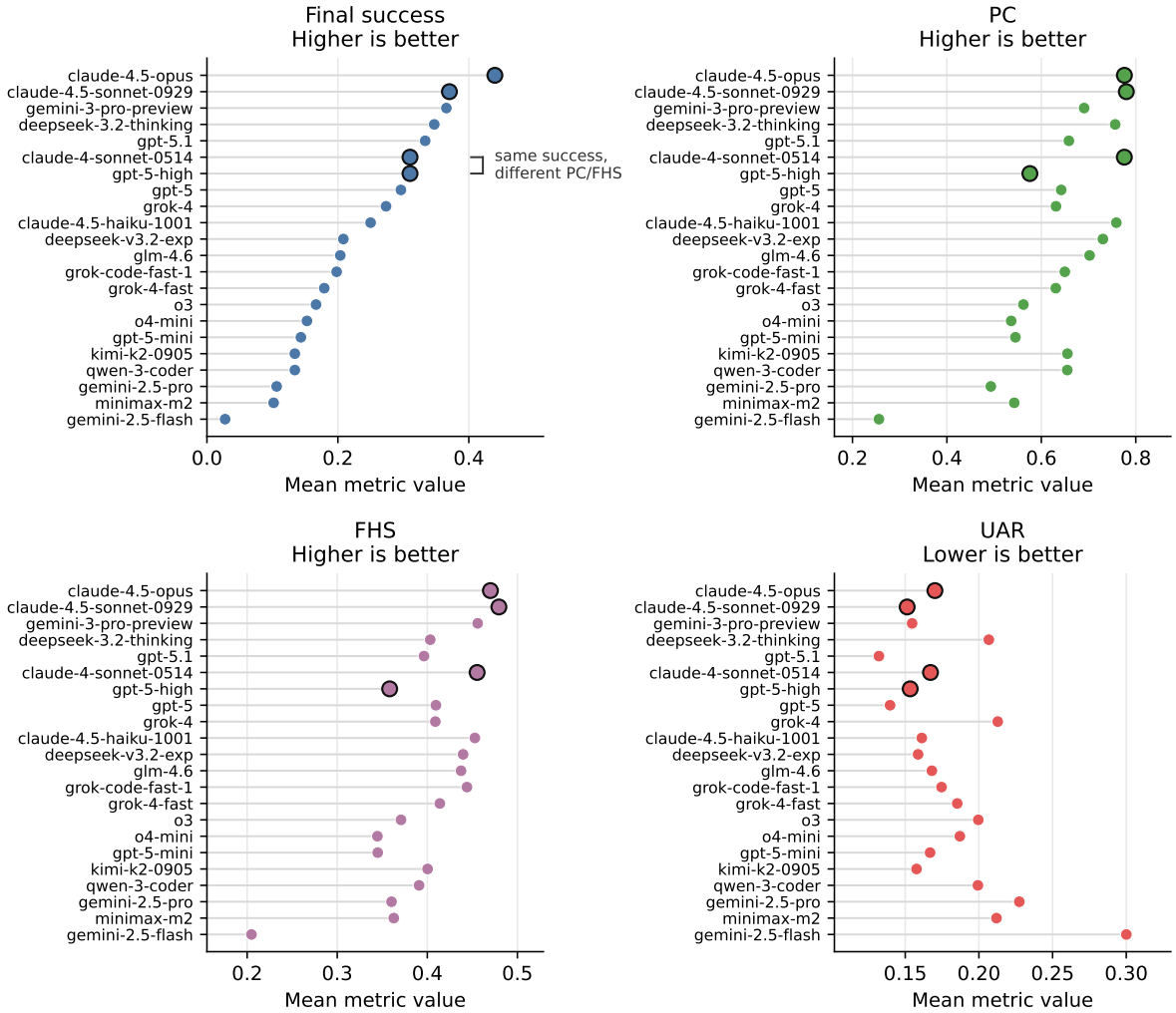


Figure 4: Full 22-model diagnostic ranking view corresponding to Figure 2. Models are ordered by final success rate in all panels. The full view preserves complete model coverage, while the main-text figure shows the top 8 models to improve readability.

Model	PC [95% CI]	FHS [95% CI]	Δrank PC [95% CI]	Δrank FHS [95% CI]
CLAUDE-4.5-OPUS	0.7759 [0.7314, 0.8187]	0.4701 [0.4314, 0.5090]	-1.0 [-4.0, 0.0]	-1.0 [-4.0, 0.0]
CLAUDE-4.5-SONNET-0929	0.7800 [0.7372, 0.8213]	0.4795 [0.4478, 0.5123]	1.0 [-2.0, 5.0]	1.0 [-1.0, 5.0]
GEMINI-3-PRO-PREVIEW	0.6906 [0.6464, 0.7348]	0.4559 [0.4228, 0.4900]	-5.0 [-7.0, -1.0]	0.0 [-4.5, 3.5]
DEEPSEEK-3.2-THINKING	0.7564 [0.7113, 0.7995]	0.4032 [0.3670, 0.4395]	-1.0 [-3.0, 3.5]	-8.0 [-12.0, -3.0]
GPT-5.1	0.6583 [0.6069, 0.7059]	0.3963 [0.3630, 0.4310]	-4.0 [-10.0, -1.0]	-9.0 [-13.0, -3.0]
CLAUDE-4-SONNET-0514	0.7758 [0.7343, 0.8168]	0.4553 [0.4205, 0.4912]	3.5 [0.5, 8.0]	2.5 [-2.0, 6.0]
GPT-5-HIGH	0.5757 [0.5220, 0.6269]	0.3582 [0.3222, 0.3956]	-9.5 [-14.0, -7.0]	-12.5 [-16.0, -8.0]
GPT-5	0.6421 [0.5918, 0.6922]	0.4096 [0.3746, 0.4456]	-5.0 [-9.0, -1.0]	-2.0 [-8.5, 1.0]
GROK-4	0.6311 [0.5747, 0.6848]	0.4089 [0.3687, 0.4483]	-5.0 [-9.0, 0.0]	-2.0 [-7.0, 2.0]
CLAUDE-4.5-HAIKU-1001	0.7587 [0.7127, 0.8015]	0.4528 [0.4200, 0.4862]	6.0 [2.0, 9.0]	5.0 [0.0, 9.0]
DEEPSEEK-V3.2-EXP	0.7304 [0.6777, 0.7782]	0.4397 [0.4022, 0.4777]	5.0 [3.0, 10.0]	4.0 [1.0, 10.0]
GLM-4.6	0.7022 [0.6532, 0.7487]	0.4375 [0.3993, 0.4739]	5.0 [2.0, 8.5]	4.0 [2.0, 9.5]
GROK-CODE-FAST-1	0.6498 [0.5979, 0.6971]	0.4440 [0.4073, 0.4799]	1.0 [-3.0, 6.0]	7.0 [2.5, 12.0]
GROK-4-FAST	0.6306 [0.5768, 0.6795]	0.4139 [0.3766, 0.4506]	-1.0 [-3.0, 5.0]	5.0 [-1.0, 8.0]
O3	0.5621 [0.5014, 0.6236]	0.3708 [0.3275, 0.4149]	-2.0 [-7.0, 2.0]	-1.0 [-7.0, 3.5]
O4-MINI	0.5361 [0.4762, 0.5956]	0.3446 [0.3048, 0.3857]	-4.0 [-6.0, 1.0]	-5.0 [-7.0, 1.0]
GPT-5-MINI	0.5452 [0.4813, 0.6086]	0.3449 [0.3058, 0.3839]	-1.0 [-6.0, 2.0]	-3.0 [-7.0, 1.5]
KIMI-K2-0905	0.6552 [0.5971, 0.7117]	0.4003 [0.3607, 0.4418]	8.5 [1.0, 12.0]	5.5 [0.0, 11.0]
QWEN-3-CODER	0.6549 [0.6027, 0.7075]	0.3909 [0.3535, 0.4297]	7.5 [2.0, 11.0]	3.5 [-1.0, 8.0]
GEMINI-2.5-PRO	0.4931 [0.4418, 0.5439]	0.3603 [0.3213, 0.3983]	-1.0 [-5.0, 1.0]	2.0 [-3.0, 6.0]
MINIMAX-M2	0.5427 [0.4875, 0.5948]	0.3628 [0.3257, 0.4006]	2.0 [-1.0, 4.0]	4.0 [-1.0, 6.0]
GEMINI-2.5-FLASH	0.2560 [0.2122, 0.3002]	0.2048 [0.1701, 0.2409]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]

Table 7: Task-paired bootstrap intervals for model-level PC, FHS, and rank deltas. Positive rank deltas mean that the process metric ranks the model higher than final success.

Metric	Paired diff.	95% CI	Perm. p
Success	0.0000	[-0.0787, 0.0833]	1.0000
PC	0.2000	[0.1529, 0.2473]	< 0.001
FHS	0.0971	[0.0658, 0.1309]	< 0.001

Table 8: Paired task-level comparison for Claude 4 Sonnet minus GPT-5 High over 108 shared tasks.

G Supplemental Diagnostic Views

The supplemental figures provide additional views of the fixed result set. They are intended to make the main diagnostic patterns easier to inspect, not to introduce new benchmark objectives. We group related figures into paired panels so that rank disagreement, failure shifts, and reliability warnings can be read together.

H Terminal-Contract Evidence for High-Progress Failures

For the 194 PC=1 failed trajectories assigned finalization-contract error, all records include terminal-contract evidence spans. Table 9 groups these violations by terminal subcategory. The table is not a gold-standard human relabeling of the failures. Instead, it reports the terminal evidence recorded by the diagnostic layer to show that the finalization label is grounded in a missed terminal contract rather than only in a category name.

Representative evidence spans show the boundary between process progress and terminal failure. In one case, the trajectory completes the core communication workflow but omits a required final delivery record. In another, the trajectory produces a structured spreadsheet update, but the submitted layout does not satisfy the required ordering and column contract. In the cooking-guidance case in Section 6, the trajectory retrieves recipe evidence and writes structured outputs, but the final contents violate dish-name, ingredient-coverage, and quantity-consistency requirements. These examples support the interpretation that finalization-contract errors are terminal-contract violations after measured process progress, while preserving the limitation that the terminal layer is evaluator-anchored rather than independently human-verified.

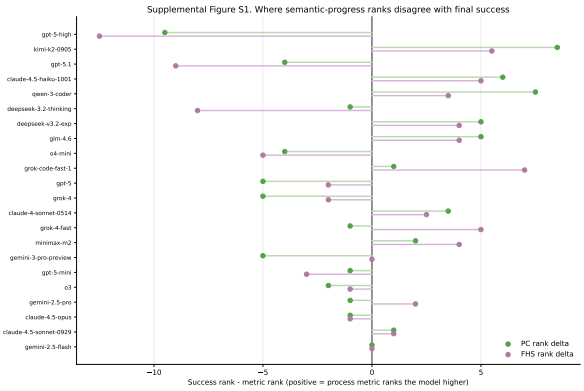
I Model Coverage

The held-out split contains 4,740 trajectories from 2,370 available model-task pairs, with two evaluation runs per pair. Nineteen models cover all 108 Toolathlon tasks. GROK-4, GROK-4-FAST, and GROK-CODE-FAST-1 each miss two tasks. Trajectory-level aggregates use all available trajectories, and task-paired model-level analyses aggregate the two runs within each available pair before task-level comparison. We do not impute missing runs.

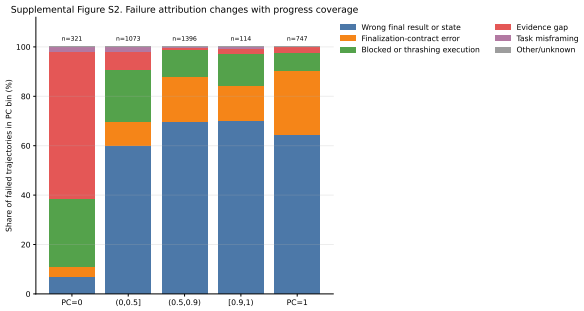
J Targeted Diagnostic Audit

We manually reviewed a targeted set of 60 trajectories drawn from GPT, Claude, and Gemini model families. This audit is separate from the computational result set. The result set supplies the trajectories, metric records, and attribution records being reviewed, while the manual judgments below are used only as a targeted diagnostic and error-analysis layer. The sample is balanced by family, with 20 trajectories from each family, and stratified into 15 review-oriented random samples, 25 high-progress failures with $\mathrm{PC} = 1$, and 20 high-UAR or evidence-gap cases. The review set covers 56 failed and 4 successful trajectories across 26 tasks. Because the sample intentionally over-represents boundary cases, the rates in Table 10 should not be read as population accuracy estimates for the full evaluation split or all 3,651 failure-attribution records. For each failed sample, the review records whether the failure attribution agrees with the trajectory evidence. For all samples, it records whether PC, FHS, and UAR are reasonable, whether the SRU set is adequate for the task, whether action labels are acceptable, and a short diagnostic note.

Table 10 shows that all reviewed PC, FHS, UAR, SRU-quality, and action-label decisions were judged reasonable. Failure attribution agrees in 54 of 56 failed cases. The four successful trajectories have no failure-attribution denominator. The two disagreements are boundary cases rather than metric failures. One PC=1 ADD-BIBTEX failure labeled as finalization-contract error could also be read as wrong final result or state if the BibTeX entry content is incorrect. One high-UAR WOOCOMMERCE-NEW-PRODUCT case labeled as task misframing could also be read as evidence gap or wrong final result or state because the agent appears to stop early after empty or incorrect data. The audit is used only to assess metric reasonableness and attribution boundaries; it does not relabel the reported ASRG scores or failure categories.

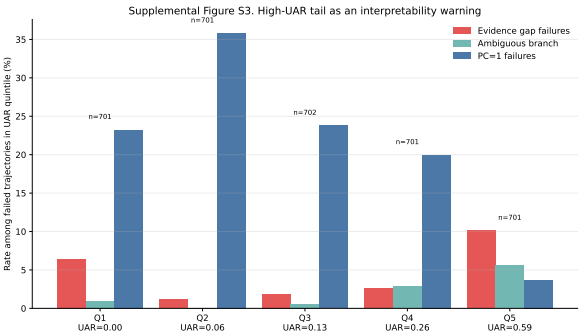


(a) Rank disagreement

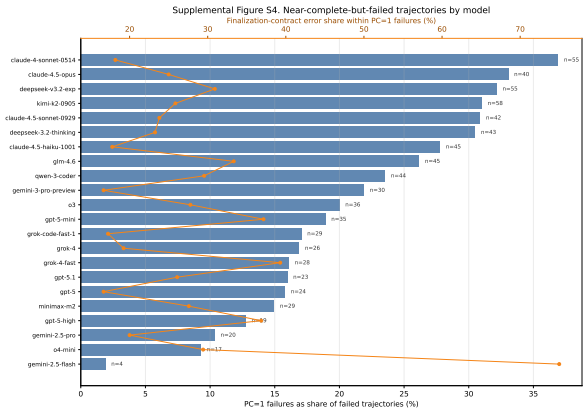


(b) Failure category by PC bin

Figure 5: Supplemental views of rank disagreement and failure-category shifts. Panel (a) shows that process diagnostics can rank models differently from final success. Positive deltas indicate that a process metric ranks a model higher than final success. Panel (b) shows that evidence gaps dominate PC=0 failures (59.81%), while wrong final result or state and finalization-contract errors dominate high-progress bins. Because the categories are judge-assisted rather than human-certified, these views support diagnostic shifts rather than gold-standard causal distributions.



(a) UAR tail risk



(b) PC=1 failures by model

Figure 6: Supplemental views of action-coverage risk and high-progress failures. Panel (a) shows that the highest-UAR quintile has the largest evidence-gap rate (10.13%) and ambiguous-branch rate (5.56%), but a lower PC=1 failure rate (3.71%) than lower-UAR quintiles. UAR is therefore an action-coverage warning rather than a progress score. Panel (b) shows that PC=1 failures occur across model families, including higher-success models, which supports the interpretation of high-progress failure as a recurring diagnostic pattern.

Terminal-contract violation	Count	Evidence pattern
Missing final delivery	85	Required submission, message, or delivery record absent after core workflow steps.
Invalid final schema or layout	64	Structured output violates required fields, ordering, or format.
Unspecified terminal gap	27	Terminal goal remains unmet, but the recorded span does not identify a finer subtype.
Numeric or sorting error	11	Final values or ordering violate the task contract.
Required final artifact missing	5	A required final deliverable is absent.
Field or external-state mismatch	2	Final fields or external state disagree with the required target.

Table 9: Terminal-contract evidence among PC=1 failed trajectories assigned finalization-contract error. Counts sum to 194 trajectories.

Audit item	Accepted	Rate
PC reasonable	60/60	100.00%
FHS reasonable	60/60	100.00%
UAR reasonable	60/60	100.00%
SRU quality acceptable	60/60	100.00%
Action labels acceptable	60/60	100.00%
Failure attribution agrees	54/56	96.43%

Table 10: Manual diagnostic audit outcomes over 60 GPT, Claude, and Gemini trajectories.