PRESERVING RULE UNCERTAINTY FOR AUDITABLE CODING-AGENT EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating coding agents requires interpreting both execution outcomes and trajectory-level behavior, yet deterministic rules can be misleading when their preconditions fail, their implementations change, or supporting evidence is absent. We present RAVEL, a framework that represents each rule as a *versioned sensor contract* specifying applicability, provenance, evidence requirements, and an admissible reliability set. Two model proxies inform these sets, but their agreement is not treated as semantic correctness and their disagreement is retained as uncertainty. A compact generative judge is supervised to emit criterion scores and event citations, then trained with matched reward variants, including a worst-case rule-agreement objective over admissible reliability weights. An audit of 402K production trajectories shows that one parser correction changes 36.9% of grades and that 42.0% of sampled pairwise orders remain uncertified across two detector versions. For diagnostic evaluation, we construct 670 instances from automatically extracted execution signals and 2,017 metamorphic relations. Across 2,680 evaluation views, supervised initialization raises strict structured-output coverage from 0% to 88.1%. The four reward variants, however, have overlapping uncertainty intervals and change rank between 128 and 256 training steps; the proposed set objective does not show a stable advantage. These results support the measurement framework—auditing applicability, provenance, and version sensitivity—while delimiting what the present execution-anchored study can establish about process-level judgment.

1 INTRODUCTION

Large language model (LLM) agents now edit files, run commands, inspect logs, and iterate on failures through reasoning-and-acting or reflection loops (Yao et al., 2023; Shinn et al., 2023). Benchmarks such as SWE-bench, WebArena, and AgentBench measure whether agents eventually complete a task (Jimenez et al., 2024; Zhou et al., 2024; Liu et al., 2024). This outcome view is necessary but insufficient for deployed systems: two agents may both pass the final tests while differing in unnecessary tool use, recovery from errors, respect for user intent, and the supervision they demand, while a failed trajectory can still contain useful partial progress.

Production evaluation therefore needs to reason about the entire trajectory, and three common solutions each fail differently. Human review is rich but too costly and privacy-sensitive for dense labels at scale. Generic LLM-as-a-Judge systems scale semantic assessment (Zheng et al., 2023; Kim et al., 2024) but exhibit position and presentation biases (Shi et al., 2025) and often return a confident score without auditable trace evidence. Deterministic metrics are cheap and reproducible, but a rule is only meaningful under its preconditions: "completion without a file write" is suspicious for an implementation task and normal for a question-answering task, and repeated reads may indicate a pathological loop or a legitimate debugging cycle.

Our data make both conditionality and implementation uncertainty concrete. We analyze 402K production coding-agent trajectories. Although recovery and alignment appear nearly perfect on average, both equal their default value for 99.74% of traces because almost all traces have no observable correction episode. In a 100-trace diagnostic, a completion-honesty rule flags seven trajectories, but six are question-answering sessions that require no code modification. Two loop detectors produce counts that differ by up to 19×: one is restricted to consecutive similar calls, while the other also

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.

counts oscillation and retry patterns. A parser correction changes 36.9% of reported grades. Rule signals are thus neither useless nor gold labels: they are *fallible, versioned sensors* with context-dependent coverage.

We propose RAVEL (**R**ules **A**s **V**ersioned, **E**vidence-**L**inked sensors), which treats evaluation as partial identification from heterogeneous evidence. Figure 1 summarizes the design. First, it induces a hierarchical rubric from open-ended production feedback, following the insight that users should determine what is evaluated (Zhu et al., 2026). Second, every deterministic detector is represented by a sensor contract containing preconditions, version provenance, a reliability set, and evidence spans. Historical detector upgrades act as natural interventions for measuring implementation fragility, while two model proxies parameterize a conservative uncertainty set rather than supply target truth. Third, a compact generative judge produces evidence-linked assessments; matched GRPO variants differ only in how rule uncertainty enters the reward. Fourth, we test the judge against automatically extracted execution outcomes and trajectory transformations. We use "no human evaluation labels" narrowly: no task-level human score is used for reward computation, threshold fitting, or test scoring; human-authored feedback informs rubric induction, and model-generated teacher responses supervise the output protocol.

Our contributions are:

- We formulate deterministic trace metrics as *versioned sensor contracts*, separating criterion applicability, detector provenance, implementation stability, reliability sets, and cited evidence.
- We study a robust rule objective in which dual-model proxy judgments define an admissible support set and the rule term optimizes worst-case agreement over that set, without interpreting either proxy as correct.
- We construct a diagnostic meta-evaluation protocol from automatically extracted execution outcomes and four families of metamorphic relations, with candidate/session-clustered uncertainty estimates (Angelopoulos et al., 2025; Bates et al., 2021).
- We report both the positive and negative evidence from a matched four-way reward ablation: supervised initialization gates strict structured evaluation (0% vs. 88.1% coverage), whereas no reward variant has a stable advantage across the two frozen training budgets.

2 RELATED WORK

Agent evaluation beyond final outcomes. Outcome benchmarks certify task completion (Jimenez et al., 2024; Zhou et al., 2024; Liu et al., 2024) but do not localize process failures. Recent work moves toward trajectory-level assessment: Agent-as-a-Judge equips an evaluator to inspect intermediate steps (Zhuge et al., 2025), and production triage systems sample and label agentic interactions at scale (Chen et al., 2026). RAVEL addresses a different layer. It does not primarily add evaluator tools or claim superior semantic judgment; it asks which rule is defined, which implementation produced it, what trace evidence supports it, and which conclusions survive the resulting uncertainty.

LLM judges and their meta-evaluation. LLM judges approximate human preferences at much lower cost than manual review (Zheng et al., 2023). Fine-tuned evaluators such as JudgeLM, Auto-J, and Prometheus improve rubric following (Zhu et al., 2025; Li et al., 2024; Kim et al., 2024), while Think-J and J1 train generative reasoning judges (Huang et al., 2026; Whitehouse et al., 2026). Reliability remains a meta-evaluation problem: Counsel adds human labels of critique location and reasoning quality for agent tasks (Pisupati et al., 2026), REFLECT creates localized failures in research-agent traces (Wang et al., 2026), and BabelJudge uses controlled degradations including trajectory perturbations (KC, 2026). These works primarily test the judge; RAVEL represents uncertainty in the rules, parsers, and applicability states supplied *to* a judge and propagates it into the final decision.

Weak supervision, partial identification, and selective risk. Data programming models heuristic labeling functions as noisy, conflicting sources rather than truth (Ratner et al., 2016). RAVEL shares this stance but adds criterion-level preconditions, detector versions, reliability sets, and event

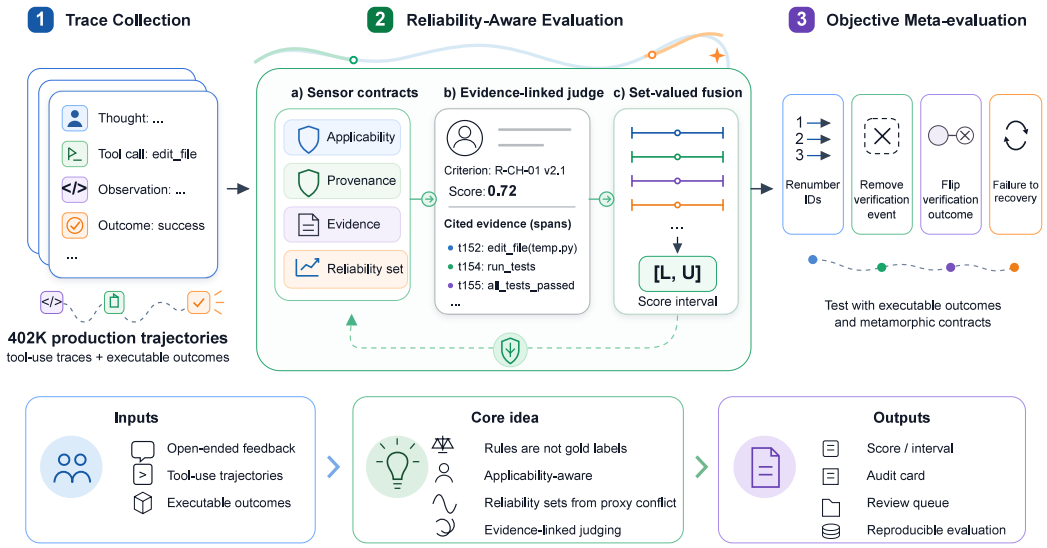

Figure 1: Overview of RAVEL. Production coding-agent trajectories (left) are evaluated through versioned sensor contracts that model rule applicability, provenance, evidence requirements, and reliability; an evidence-linked judge produces criterion-level scores and event citations, and set-valued fusion preserves sensor uncertainty as a score interval (center). Executable outcomes and four metamorphic contracts provide objective meta-evaluation without reference answers (right): event-ID renumbering, verification removal, outcome flips, and observed recoveries.

evidence for sequential agent traces. Risk-controlling prediction sets and learn-then-test calibration provide tools for choosing predictive sets or policies under explicit loss budgets (Bates et al., 2021; Angelopoulos et al., 2025); here the observable losses are execution errors and metamorphic-contract violations, not assumed human disagreement. AutoLibra grounds open-ended feedback in behavior and induces concrete metrics (Zhu et al., 2026); we focus on what follows induction: determining when a metric applies, connecting it to executable detectors, and preserving uncertainty when implementations or proxies disagree. Rule rewards work well when answers are objectively verifiable (Shao et al., 2024; Guo et al., 2025); coding-agent process rules are only partially verifiable, so RAVEL never treats them as universal labels.

3 Problem Formulation

Let τ_i be a trajectory and $\{m_k\}_{k=1}^{K}$ a rubric. For criterion m_k, the judge emits

$$z_{ik} = (\hat{a}_{ik}, \hat{y}_{ik}, c_{ik}, E_{ik}), \tag{1}$$

where $\hat{a}_{ik} \in \{0,1\}$ is the judge's applicability decision, $\hat{y}_{ik} \in [0,1]$ is its score, $c_{ik} \in [0,1]$ is confidence, and E_{ik} contains supporting trace spans. Detector version v separately emits a rule-side value $\tilde{y}_{ikv}$ and applicability state $\tilde{a}_{ikv}$.

A deterministic criterion is specified by a sensor contract

$$C_k = (P_k, \mathcal{V}_k, \mathcal{Q}_k, \mathcal{E}_k), \tag{2}$$

where P_k is an applicability precondition, $\mathcal{V}_k$ is an admissible set of parser/detector versions, $\mathcal{Q}_k = [\underline{q}_k, \overline{q}_k]$ is an admissible proxy-support set, and $\mathcal{E}_k$ specifies required evidence. We call $\mathcal{Q}_k$ a reliability set for brevity, but it is *not* an accuracy confidence interval: agreement between two proxies need not imply semantic validity. Available evidence may likewise identify applicability only to a set $\mathcal{A}_{ik} \subseteq \{0,1\}$. Given a fusion operator F, the criterion score is set-valued:

$$\mathcal{S}_{ik} = \{F(\hat{y}_{ik}, \tilde{y}_{ikv}, a, q) : v \in \mathcal{V}_k, a \in \mathcal{A}_{ik}, q \in \mathcal{Q}_k\}. \tag{3}$$

Trajectory-level aggregation yields an interval $[L_i, U_i]$, and a pairwise order between two trajectories is *certified* only when their intervals do not overlap. This makes explicit that a single scalar score can be an artifact of one admissible implementation.

4 METHOD

Schematic worked example. Consider a trajectory that answers a repository question without being asked to modify code. A "completion without a write" detector may fire, yet its contract precondition P_k is false, so the rule-side applicability $\tilde{a}_{ik}$ is zero and the Gate, Proxy-LCB, and RAVEL-Set rewards exclude it. If evidence is ambiguous and the proxies disagree about observation fidelity, $\mathcal{Q}_k$ remains wide; this records disagreement rather than converting agreement into correctness. In a modification task, alternate parser versions in $\mathcal{V}_k$ may attach different tool results to the write event. The training objective then evaluates score agreement across admissible weights, while inference carries the versions, applicability states, and weights through F to an interval. A point decision is certified only if it survives that set. This example connects the contract stages; it is explanatory, not an additional experiment.

Reliability sets from dual proxies. Two model proxies independently judge applicability, observation fidelity, quality direction, and evidence for each trace–rule pair. We define $\mathcal{Q}_k$ as the hull of the joint-support and source-specific 95% posterior intervals. This construction is intentionally wider than a pooled estimate: it preserves proxy disagreement and never asserts that either proxy is correct. It therefore measures support stability among two sources, not rule accuracy, and shared proxy bias remains possible. Instance support is reduced further when parsing is incomplete, denominators are small, or detector versions disagree.

Version stability and fragility routing. We retain outputs from detector versions $v, v' \in \mathcal{V}_k$ and define an intervention target

$$f_{ik}^{vv'} = \mathbf{1}[|\tilde{y}_{ikv} - \tilde{y}_{ikv'}| > \delta_k], \tag{4}$$

or the corresponding downstream grade change. A lightweight router $g_\psi(\phi_i)$ predicts $P(f_{ik}^{vv'} = 1 \mid \phi_i)$ from pre-update trace and provenance features, giving a stability factor $q_{ik}^{\mathrm{stab}} = 1 - g_\psi(\phi_i)$. This target measures *measurement fragility*, not whether either detector is semantically correct.

Matched reward variants. Let $s_{ik} \in [0, 1]$ be the Huber similarity between the judge's criterion score and the rule target. Training-time gates a_{ik} come from the frozen rule-contract data (preconditions plus proxy-audited applicability), not from the judge's emitted $\hat{a}_{ik}$. The four matched objectives differ only in the rule term. Naive averages s_{ik} over all criteria; Gate averages over rule-applicable criteria only; Proxy-LCB weights them by a conservative point support q_k,

$$r_{\mathrm{lcb}} = \frac{\sum_k a_{ik} q_k s_{ik}}{\epsilon + \sum_k a_{ik} q_k}; \tag{5}$$

and RAVEL-Set optimizes the worst case over the full reliability set,

$$r_{\mathrm{set}} = \min_{\mathbf{q} \in \prod_k \mathcal{Q}_k} \frac{\sum_k a_{ik} q_k s_{ik}}{\epsilon + \sum_k a_{ik} q_k}. \tag{6}$$

All four share the same structure, evidence, and formatting reward terms, so any difference is attributable to how rule uncertainty enters the objective.

Set-valued fusion and selective evaluation. For a fixed admissible sensor state (v, a, q), the fused score is

$$\bar{s}_{ik}(v, a, q) = \frac{c_{ik}\hat{y}_{ik} + aq\tilde{y}_{ikv}}{c_{ik} + aq + \epsilon}. \tag{7}$$

Evaluating it over $v \in \mathcal{V}_k$, $a \in \mathcal{A}_{ik}$, and $q \in \mathcal{Q}_k$ yields $[L_i, U_i]$. With criterion weights $w_{ik} = c_{ik}\hat{a}_{ik}$, a conflict statistic

$$d_i = \frac{\sum_k w_{ik} |\hat{y}_{ik} - \tilde{y}_{ikv}|}{\epsilon + \sum_k w_{ik}} \tag{8}$$

is combined with interval width $U_i - L_i$, judge uncertainty, and missing-evidence indicators into a nonconformity score u_i. The selector $g_\eta(i) = \mathbf{1}[u_i \le \eta]$ exposes the selection-conditioned risk $R(\eta) = \mathbb{E}[g_\eta(i)\ell_i]/(\mathbb{E}[g_\eta(i)] + \epsilon)$, where ℓ_i is strict success/failure error on natural instances and a metamorphic-contract violation on paired interventions. Thresholds are fit on a candidate/session-disjoint development split and frozen before test.

5 EXPERIMENTAL SETUP

No task-level human evaluation score enters reward computation, threshold selection, or test scoring. Human-authored production feedback informs rubric induction, model-generated teacher responses initialize the structured protocol, and model proxies parameterize uncertainty; none is interpreted as human gold for the objective suite.

Datasets. The production audit contains 402K trajectories from 48 consecutive days, 23 model variants, and 3.58M normalized tool events. For controlled experiments we reconstruct complete redacted trajectories into a 2,000-trajectory evaluation-candidate pool and a session-disjoint 20,000-trajectory training pool; the automated privacy scan found zero residual identifiers. A deterministic split yields a 19,026-example GRPO sampling frame and 974 development examples, and schema-valid teacher responses above a pre-frozen confidence of 0.70 give 1,580 supervised training and 77 development examples.

Dual-proxy audit. GPT-5.5 and Claude Opus 4.6 independently judge ten rule contracts on 180 trajectories (1,800 trace–rule pairs) at temperature 0. Unless otherwise noted, GPT-5.5 refers to `gpt-5.5-0424-global` and Claude Opus 4.6 refers to `claude-opus-4-6`. Applicability agreement is 81.1% ($\kappa = 0.612$), fidelity 63.9% ($\kappa = 0.492$), direction 91.8% ($\kappa = 0.795$), exact agreement on all core fields 59.6%, and mean evidence Jaccard 0.471. The 728 disagreements motivate reliability sets; none of these values is interpreted as human accuracy.

Objective suite. We automatically extract execution anchors when a command matches a verification pattern and has an explicit exit code, yielding 670 instances. These are diagnostic anchors, not manually verified ground truth: command-pattern matching can admit semantically weak checks, and an exit code certifies only the observed command outcome. Candidate/session-disjoint splitting gives 124 development instances (117 success, 7 failure) and 546 test instances (517 success, 29 failure). Each instance has four views: the original, an event-ID renumbering, a verification-event removal, and a controlled verification-outcome flip. Seven observed failure-to-recovery pairs add temporal relations, giving 2680 requests and 2017 contracts. Renumbering checks invariance and citation remapping; the other relations test pre-specified weak monotonic contracts. These contracts are contestable for a multi-criterion score—for example, a costly recovery can reduce efficiency—and the non-strict directional aggregate can be satisfied by an unchanged prediction. Synthetic flips never enter natural outcome metrics. Labels and relation types are not passed as separate fields, but the controlled tool outputs contain explicit synthetic markers, which may reveal the intervention. Missing, malformed, or truncated outputs count as failures for strict threshold and relation metrics.

Compute-bounded students. The student is Qwen3-8B in non-thinking mode, emitting a frozen JSON assessment (task type, score distribution, completion status, 15 criterion records, event citations, short diagnostic) with no exposed chain-of-thought. The controlled comparison uses a deterministic metadata-stratified subset of 512 GRPO prompts with four candidates each; long prompts use an event-complete head/tail view capped below 60K rendered characters with omitted events counted and hashed, never silently truncated. Naive, Gate, Proxy-LCB, and RAVEL-Set share the same supervised adapter (seed 29, checkpoint 100), random seed, prompt subset, context policy, reward weights, and optimizer settings; only the rule-reward term differs. A 128-prompt reward probe passed all four pre-registered gates before training. Each method trains for 256 GRPO steps on one NVIDIA H20 GPU with checkpoints frozen at steps 128 and 256, giving two matched budgets that are scored separately. Inference is greedy (temperature 0). The complete ledger is capped at 300 H20 GPU-hours, with failed jobs counted.

Baselines and metrics. Judge baselines are the two prompted API models above (identical system and user messages, temperature 0), the untuned compact open model, the shared supervised initialization, and the three ablated reward variants. Execution AUROC is computed on schema-valid natural-test outputs; strict balanced accuracy uses the development-frozen threshold and counts invalid outputs as errors. Schema coverage is measured over all 2,680 views. The directional aggregate is the unweighted weak-relation satisfaction rate, and Execution-Risk@90 is strict classification error after retaining the 90% of natural-test cases with highest self-reported confidence. This is not the interval/conflict selector defined above and, under the 517/29 class imbalance, is not by itself

Method (checkpoint-256)	Cov.%↑	Exec. AUROC↑	BAcc↑	Dir. Consist.↑	Exec. R@90↓
Prompted GPT-5.5	100.0	0.628	0.595	0.743	64.4
Prompted Claude Opus 4.6	100.0	0.476	0.493	0.841	68.6
Untuned Qwen3-8B	0.0	–	–	–	–
Supervised init (shared)	88.1	0.656	0.395	0.784	27.1
Naive-GRPO	87.6	0.663	0.639	0.769	40.7
Gate-GRPO	86.4	0.646	0.473	0.768	92.9
Proxy-LCB-GRPO	86.3	0.625	0.385	0.718	29.1
RAVEL-Set GRPO	89.0	0.619	0.434	0.758	32.6

Table 1: Matched diagnostic comparison at the full 256-step budget. Cov. is strict-schema coverage over all 2680 views. AUROC uses each row's schema-valid subset of the 546 natural test instances; strict BAcc uses the development-frozen threshold and counts invalid outputs as errors. Dir. Consist. is a weak, non-strict metamorphic aggregate. Exec. R@90 is strict classification error (%) at 90% self-reported-confidence coverage; it is not the set-valued selector. A common-valid-subset recomputation changes each AUROC by at most 0.006. The first four rows are training-step-independent and shared with Table 2. The untuned model emits no strictly valid prediction, so its threshold-dependent cells are undefined.

evidence of calibrated or useful confidence. Intervals use 1,000 candidate/session-cluster bootstrap resamples.

6 RESULTS

6.1 MEASUREMENT INSTABILITY

Two natural interventions establish that a scalar rule score is not a stable measurement. First, a single parser correction changes 270K of 402K quality scores (67.1%) and 148K letter grades (36.9%); pre/post rankings correlate at Spearman $\rho = 0.678$, and only 64.2% of the pre-fix bottom 1% remain in the post-fix bottom 1%. The effect appears on all 48 collection dates (changed-score rate 57.5%–77.6%). Second, treating the two implementations as equally admissible measurements yields trajectory-level score intervals of mean width 0.0618; only 58.0% of two million sampled trajectory pairs have certifiable order, and letter grades are singletons for just 63.1% of trajectories.

Rule semantics are equally conditional. Recovery and alignment sit at their default value for 99.74% of traces, so apparent perfection reflects unobservability rather than quality. Among seven completion-honesty alerts, only one survives a coding-task precondition (precision 14.3%, Wilson 95% CI [2.6%, 51.3%]). Strict and broad loop detectors agree on 92% of traces yet differ in counts by up to 19.4× ($\rho = 0.884$). In 2,445 existing LLM criterion judgments, 245 neutral scores (10.0%) carry explicit not-applicable statements, accounting for 26.8% of all zero scores; masking only those statements shifts aggregate means by up to 0.51 points on individual criteria. A temporal-holdout router trained on the first 36 dates (289K trajectories) and tested on the final 12 (113K) predicts grade change with AUROC 0.743 [0.723, 0.764], while distance to the nearest grade boundary achieves only 0.460. Measurement fragility is therefore predictable from provenance features that a score margin ignores.

6.2 MATCHED COMPARISON AT TWO BUDGETS

Tables 1 and 2 report the four matched reward variants at the two frozen budgets alongside four training-step-independent baselines. Both budgets use identical prompts, decoding, threshold-fitting code, and scoring code; only the checkpoint differs. Because coverage differs across rows, we also recompute AUROC on the common subset of 466 natural test instances (439 successes, 27 failures) answered validly by every judge. All values move by at most 0.006; two close rows exchange order, but the substantive conclusion is unchanged: differences are small, interval estimates overlap, and the reward ranking depends on budget.

Structured supervision gates strict evaluation. The untuned base model produces no strictly valid assessment across all 2680 views, so its threshold cannot even be calibrated, whereas the

Method (ckpt-128)	Cov.%	AUROC	BAcc	Dir.	R@90
Naive-GRPO	87.6	0.635	0.599	0.775	38.9
Gate-GRPO	90.1	0.643	0.482	0.789	29.5
Proxy-LCB-GRPO	88.7	0.635	0.493	0.732	37.3
RAVEL-Set GRPO	89.8	0.666	0.486	0.771	28.5

Table 2: Training-budget sensitivity at the halved 128-step budget under the identical protocol and its own development-frozen thresholds. The training-step-independent baselines of Table 1 apply unchanged; the two budgets are scored separately and never pooled.

same model after supervised initialization reaches 88.1% strict coverage and all GRPO variants stay between 86.3% and 90.1%. This reproduces at full evaluation scale what a separate 35B protocol study shows on 77 development examples (strict structured output rising from 0/77 to 74/77) and is consistent with supervised protocol initialization, rather than reward design, being what makes strict structured evaluation possible. Because every GRPO run starts from the supervised adapter, this attribution is observational.

Prompted API judges provide context, not an upper bound. Both API judges follow the schema perfectly. GPT-5.5 reaches 0.628 AUROC, within the 0.619–0.666 range of the task-trained GRPO checkpoints, whereas Claude Opus 4.6 reaches 0.476 and has the strongest weak directional consistency (0.841). This is not a like-for-like capability comparison: the compact model is trained on this protocol, the API models are prompted only, and the objective suite is narrow and platform-specific. We therefore use these rows as contextual baselines and make no claim that the compact judges generally match frontier judges.

No reward objective has a stable advantage. Figure 2 plots both budgets with clustered bootstrap intervals. At 128 steps RAVEL-Set attains the best AUROC (0.666) and lowest Execution-Risk@90 (28.5%); at 256 steps Naive leads on AUROC and BAcc (0.663/0.639) with RAVEL-Set at 0.619. Because all AUROC intervals overlap on a test split with 29 failures, we treat the two budgets as two equally valid readings of the same matched comparison rather than selecting the more favourable one. The practical implication is methodological: for compute-bounded judge training, a single tuned checkpoint can order reward variants differently from a checkpoint at half the budget, so budget-paired reporting is necessary for a fair ablation. Gate-256 makes the same point for selectors: its development-fitted threshold drifts to 0.805 and its confidence ranking degrades on test (Execution-Risk@90 of 92.9%) while its AUROC (0.646) stays mid-range, showing that ranking quality and selector calibration must be reported separately.

Strict balanced accuracy is threshold-limited. Most rows show strict BAcc below 0.5 even though the trained compact judges' AUROCs exceed 0.5; the prompted Claude row is 0.476. The development split contains only 7 natural failures, so a fitted threshold has high variance and can transfer poorly to the 29 test failures. AUROC avoids that threshold but remains uncertain with so few failures. We therefore interpret both as diagnostic and report strict BAcc to expose threshold transfer rather than explain it away.

The clearest gain is protocol compliance. The supervised initialization reaches 88.1% strict coverage, while only two of eight GRPO checkpoints exceed its AUROC of 0.656. Its Execution-Risk@90 is 27.1%, and RAVEL-Set-128 reaches 28.5%, but these values should not be described as usable selective confidence: with 517 successes and 29 failures, an always-success predictor has 5.3% full-coverage error and at most 5.9% error on any 90%-sized subset. Balanced accuracy and AUROC reveal that such a trivial predictor does not discriminate failures. The reproducible improvement supported here is therefore structured-output compliance; the reward and confidence results are mixed diagnostics.

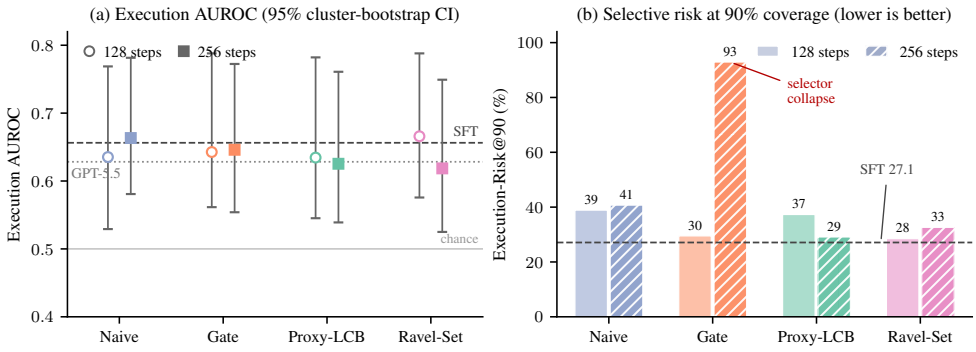

Figure 2: Training-budget sensitivity of the matched reward-variant comparison. (a) Execution AUROC with 95% candidate/session-clustered bootstrap intervals; the dashed and dotted lines mark the shared supervised initialization and the prompted GPT-5.5 judge. (b) Strict execution risk at 90% confidence-ranked coverage. Solid bars are the 128-step checkpoints and hatched bars the 256-step checkpoints; the two budgets are scored independently and never pooled. The ranking flips between budgets, all intervals overlap, and Gate-256 exhibits a selector collapse despite a mid-range AUROC.

7 Discussion

Why rules should be sensors, not labels. Rule metrics are deterministic, cheap, and inspectable, but their weakness is conditional semantics rather than noise. Missing negative turns make recovery unobservable, not perfect; repeated reads may indicate stalling or careful debugging; a build result matters only when the task makes a build relevant. Modeling applicability and reliability makes these assumptions visible and testable, and our diagnostics show the cost of ignoring them: a 14.3% alert precision and a 19$\times$ detector disagreement are properties of the instrument, not of the agents.

Why a score can be set-valued. Choosing the newest detector silently assumes that implementation history has resolved measurement uncertainty. Our natural intervention shows otherwise: the fix improved result matching, but its effect on any single ranking is not knowable from the old point score. A version interval answers a narrower question that stays auditable, namely which conclusions survive the currently admissible implementations, and prevents implementation choice from being mistaken for semantic certainty. The present experiments evaluate intervals induced by two detector versions; they do not validate the full fused judge–sensor interval of Equation (7) end to end.

Why evidence matters more than exposed reasoning. Recent judges improve through explicit reasoning (Huang et al., 2026; Whitehouse et al., 2026), but a long rationale is not automatically faithful. RAVEL trains the model to cite trace events and emit a compact diagnostic, which supports auditing without requiring exposed chain-of-thought. The transformation metrics do not establish faithful evidence use: the weak directional contracts permit ties, the synthetic markers may reveal intervention type, and invariance violations remain at 25–35% even for the best checkpoints.

Scope and limitations. Our claims are deliberately scoped to the measurement instrument. Automatically extracted execution anchors test decisions about observed command outcomes, not tool efficiency, recovery quality, intent alignment, root cause, or overall task correctness; they can also admit semantically weak commands. Human validation on those process dimensions remains absent. The test split contains 29 failures and the development split 7, so thresholds and interval estimates are unstable; outcome flips are synthetic, their markers are model-visible, the directional contracts permit unchanged outputs, and the seven recovery pairs are exploratory. The reported risk uses model confidence rather than the proposed interval/conflict selector, and the full fused interval is not evaluated. Production data come from one platform. The version analysis covers one parser correction and two detector implementations, so it demonstrates that versioning can matter rather than establishing generality across rules. Dual-model proxies may share biases; their sets record

source disagreement, not accuracy. Finally, a fixed single-GPU budget fixes the student size, four candidates per prompt, a 512-prompt subset, and one training seed. These limitations preclude a claim that RAVEL-Set outperforms simpler objectives; they do not negate the observed measurement instability that motivates the framework.

8 CONCLUSION

Evaluating coding-agent trajectories requires scalable semantic judgment without pretending that heuristics or model proxies are truth. RAVEL makes the measurement layer explicit through versioned sensor contracts, applicability conditions, provenance, evidence, and set-valued uncertainty. A parser correction changes over one third of production grades, and 42.0% of sampled pairwise orders remain uncertified across the two observed implementations. Supervised initialization, rather than the proposed robust reward, produces the clearest experimental gain by making strict structured evaluation possible. The matched reward variants have overlapping intervals and reverse order across budgets, so the current evidence supports the auditing framework but not a performance advantage for RAVEL-Set. Human validation of process-level judgments, broader detector histories, verified execution anchors, and end-to-end evaluation of fused intervals are required next.

ETHICAL AND PRIVACY CONSIDERATIONS

Production traces may contain proprietary code, credentials, personal data, or organization-specific identifiers. Secret scanning, path redaction, and access control precede any training or external API use, and even redacted traces are sent to an external endpoint only after organizational approval. Released examples should be synthetic or explicitly approved. The evaluator should not be the sole basis for employee assessment; selective abstention and evidence display preserve human oversight in consequential decisions.

REPRODUCIBILITY STATEMENT

The implementation used in this study covers trace normalization, rule calculation, rubric induction, version comparison, fragility routing, custom GRPO rewards, objective-suite construction, resumable inference, and strict scoring. Frozen experimental files fix the 19,026/974 GRPO sampling frame, the 1,580/77 supervised split, the 512-prompt training subset, and the 2680-request objective suite with 2017 relations, all with recorded SHA-256 hashes and single-split assignment per candidate/session. Training uses seed 20260724 with a nested 128-prompt reward probe, four candidates per prompt, and event-complete hashed context views; resampling uses seed 20260713 for diagnostic proxies, 20260730 for date-clustered fragility, and 20260724 for the objective suite. All judges decode greedily. The experiment records contain checkpoints at steps 128 and 256, the environment freeze, hardware and timing logs, raw predictions for all 2680 views per method, and machine-generated tables. Full derivations, per-family transformation metrics, the complete diagnostic tables, and the frozen artifact contract are provided in the appendix.

AI USE STATEMENT

We used generative AI systems as reported experimental components: model-generated teacher responses for structured supervision, two model proxies for the rule audit, and prompted judge baselines. We also used generative AI for code assistance and language editing, including rephrasing and cross-checking this revision against the manuscript and reported artifacts. The authors reviewed the AI-assisted text, code, annotations, outputs, figures, and references and take responsibility for the final content.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2), 2025.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021.

Shuguang Chen, Adil Hafeez, and Salman Paracha. Signals: Trajectory sampling and triage for agentic interactions. *arXiv preprint arXiv:2604.00356*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Hui Huang, Yancheng He, Hongli Zhou, Rui Zhang, Wei Liu, Weixun Wang, Wenbo Su, Bo Zheng, and Jiaheng Liu. Think-J: Learning to think for generative LLM-as-a-judge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026.

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*, 2024.

Shreyas KC. BabelJudge: Measuring LLM-as-a-judge reliability across languages and agent trajectories. *arXiv preprint arXiv:2606.22329*, 2026.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. In *International Conference on Learning Representations*, 2024.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

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

Alexander J. Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3567–3575, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing*, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Leyao Wang, Yanan He, Peng Chen, Asaf Yehudai, Yixin Liu, Rex Ying, Michal Shmueli-Scheuer, and Arman Cohan. Time to REFLECT: Can we trust LLM judges for evidence-based research agents? *arXiv preprint arXiv:2605.19196*, 2026.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in LLM-as-a-judge via reinforcement learning. In *International Conference on Learning Representations*, 2026.

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*, 2023.

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

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

Hao Zhu, Phil Cuvin, Xinkai Yu, Charlotte Ka Yee Yan, Jason Zhang, and Diyi Yang. AutoLibra: Agent metric induction from open-ended human feedback. In *International Conference on Learning Representations*, 2026.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *International Conference on Learning Representations*, 2025.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

TECHNICAL APPENDIX

A FROZEN RULE LIBRARY

The ten deterministic detectors below are the frozen sensor contracts used throughout the paper.

Parent	Signal	Intended observation	Example applicability / risk
Correctness	Outcome success (OSR)	final satisfaction and delivery evidence	weak when no explicit outcome is observed
Correctness	Tool effectiveness	nonfailed, nonempty, nonrepeated calls	semantic usefulness not guaranteed
Correctness	Task evidence (TES)	verified deliveries / delivery claims	task may not require executable verification
Correctness	Task completion (TCR)	completed inferred user requirements	defaults saturate on one-turn sessions
Efficiency	Interaction economy (IES)	normalized turns, tools, and tokens	task difficulty is a confounder
Efficiency	Repeat-free rate	fraction of nonrepeated tool calls	repeated reads may be legitimate
Recovery	Repair rate	resolved / repairable negative events	undefined without a failure episode
Recovery	Recovery rate	successful recovery after a failure	default-perfect values hide noncoverage
Alignment	Dissatisfaction rate (UDR)	negative user turns / user turns	silent abandonment is unobserved
Alignment	Correction burden (UCB)	normalized corrective user effort	unavailable in single-turn interactions

Table 3: Rule signals and their conditional semantics. RAVEL keeps applicability and reliability alongside every value instead of collapsing all signals into an unconditional score.

B SPECIFICATION OF THE NINE DIAGNOSTIC EXPERIMENTS

We run nine diagnostic experiments; E1–E9 reuse existing repository artifacts without new model calls, while the dual-proxy audit and the prompted API-judge baselines are the only components requiring new model calls.

E1: Boundary saturation. For every Production-Unlabeled metric, we compute its mean, median, number of distinct values, and exact mass at zero and one. Boundary mass is interpreted as a measurement-coverage diagnostic, not as agent quality.

E2: Detector-version sensitivity. We match pre- and post-fix exports one-to-one by collection date and root trajectory identifier. We report changed-score and changed-grade rates, mean and 95th-percentile absolute score changes, Spearman and Kendall rank correlations, and overlap of the bottom 1% triage sets. We repeat changed-score summaries independently for each of the 48 dates.

E3: Loop-definition agreement. On Audit-100, the strict detector counts consecutive similar calls, whereas the broad detector also includes oscillation and retry-after-failure patterns. We compare binary detections using agreement, Cohen's κ, and Jaccard similarity, and compare counts using Spearman correlation and maximum observed discrepancy.

E4: Completion-alert applicability. The existing trace-by-trace audit labels all seven completion-rule alerts as confirmed or false. We report alert precision with Wilson 95% intervals before and after the audit's coding-task proxy: at least one Write, Edit, MultiEdit, or Bash call. Because only triggered alerts were reviewed, recall is not estimable.

E5: Applicability and rubric construction. For 2,445 existing AutoLibra judgments, a case-insensitive phrase audit detects explicit N/A expressions only when the emitted score is neutral. This

conservative restriction prevents phrases such as "a system that was not required" from becoming false N/A labels. We compare the original instance mean with the mean after masking explicit N/A entries. We also compare scores from 33 raw criteria and 15 deduplicated canonical criteria, bottom-quartile membership, and five rounds of LADDER coverage. Spearman intervals use 2,000 trajectory-cluster bootstrap resamples so subinstances from the same trace remain together.

E6: Rule–semantic consistency. After removing only explicit N/A entries, canonical criterion scores are averaged within each trace and paired with conceptually corresponding rule signals. We report Spearman correlations, 2,000-resample percentile intervals, and Benjamini–Hochberg q-values over 14 pre-specified pairs. These are proxy-convergence tests between automatic systems, not accuracy against human truth. Resampling in E5–E6 uses seed 20260713.

E7: Version-set partial identification. We treat the matched pre- and post-fix scores as two admissible measurements, without designating either as human truth. Their minimum and maximum define a trajectory-level score interval. We report interval width, whether the letter grade is singleton, bottom-p set retention for $p \in \{1, 5, 10, 25\}\%$, and the fraction of two million seeded random trajectory pairs whose order is certified by non-overlapping intervals. The latter receives a Wilson interval conditional on the sampled pairs.

E8: Temporal-holdout fragility routing. We train an L2-regularized logistic router to predict whether a trajectory's grade changes across detector versions. Its 46 inputs—scores, counts, ratios, quadratic terms, and sufficiently frequent model indicators—come exclusively from the pre-fix export. The first 36 dates (289K trajectories) are training data and the final 12 (113K) are held out. We compare against distance to the nearest grade boundary and report AUROC, average precision, Brier score, and grade-change risk at multiple accepted coverages. Intervals use 1,000 bootstrap resamples of complete test dates.

E9: Applicability-set partial identification. Because neutral zero and N/A share one scalar in the existing judgments, we conduct a sensitivity analysis in which every unresolved zero may either remain in the aggregate denominator or be excluded as inapplicable. This yields exact lower and upper aggregate endpoints for each of 163 subinstances. We compare interval widths and cross-trace order certification before and after removing high-precision explicit N/A statements. This uncertainty set is deliberately conservative; it is not a claim that every neutral entry is N/A.

C Measurement Diagnostics: Full Results

Figures 3 and 4 summarize the measurement diagnostics (Table 4). We additionally report the completed dual-proxy audit, three SFT seeds, and construction of the objective suite. These establish the need for RAVEL and the validity of the experimental instrument; the matched method comparison itself is reported in Tables 1 and 2.

C.1 Completed Training and Frozen Evaluation Instruments

Table 5 separates 35B protocol learning from semantic evaluation. The base model often begins a valid object but truncates before all criteria; SFT makes parsing reliable across seeds. Seed29 and seed43 parse all 77 examples with no truncation, and 74 contain the full strict structure. These gains justify retaining supervised protocol initialization as a stage, but do not establish objective correctness and do not substitute for the separately gated compact SFT.

The objective suite contains 670 original instances and four views per instance, split by candidate/session. Development has 117 successful and 7 failed executions; test has 517 and 29. The imbalance motivates balanced accuracy, AUROC, and strict failure accounting. Of 2,017 relations, 670 test presentation invariance, 670 verification necessity, 670 controlled outcome sensitivity, and seven observed recovery transitions. This instrument is frozen before GRPO evaluation.

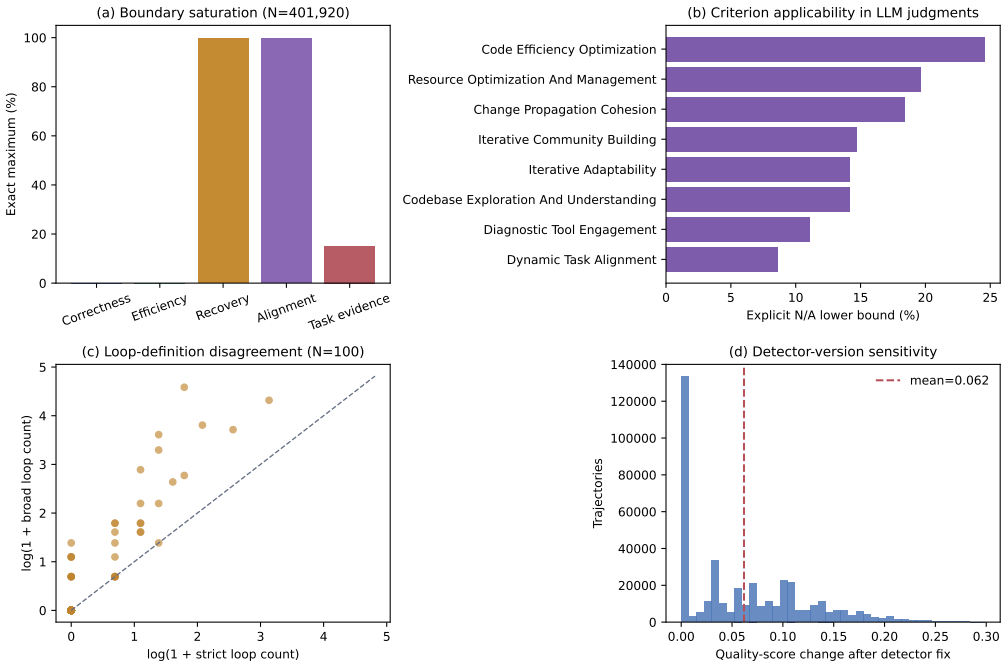

Figure 3: Reproducible pilot audits. (a) Recovery and alignment concentrate at their exact maximum. (b) A high-precision text audit finds criterion-level N/A statements in existing LLM judgments. (c) Strict and broad loop detectors agree on many traces but assign different counts. (d) A detector correction changes the quality score of most production trajectories.

Experiment	Analysis unit	Primary result	Measurement implication
Boundary saturation	402K traces	Recovery and alignment equal 1.0 on 99.74%; TES equals 0 on 61.63%	Default or missing evidence can masquerade as an observed score
Version sensitivity	402K matched pairs	67.1% of scores and 36.9% of grades change; bottom-1% retention is 64.2%	Detector version changes operational ranking
Version-set identification	2.0M sampled pairs	58.0% of orders certified; 24.6% reverse	Some rankings are not identified by available implementations
Fragility routing	113K held-out traces	AUROC 0.743 [0.723, 0.764]; Risk@90 is 33.8%	Provenance features predict measurement instability
Loop definitions	100 traces	25 vs. 33 positives; $\kappa = 0.807$; maximum counts 5 vs. 97	Similar detector names need explicit semantics
Completion applicability	7 audited alerts	Precision 14.3% [2.6%, 51.3%]; gated 100% [20.7%, 100%]	Cross-task rules require preconditions; gated sample is inconclusive
Rubric applicability	2,445 judgments	245 explicit N/A (10.0%); masking yields 95th-percentile score shift 0.304	Neutral and non-applicable judgments must be separated
Applicability sets	12,973 cross-trace pairs	Explicit audit raises certified orderings from 62.4% to 66.5%	Applicability uncertainty should propagate to score intervals
Rubric deduplication	163 subinstances	33 to 15 criteria; rank $\rho = 0.753$ [0.661, 0.816]	Redundant criteria reweight aggregate scores
Rule–semantic consistency	14 proxy pairs	Largest positive $\rho = 0.385$; only one pair has BH $q < 0.05$	Rules and semantic scores provide partial, non-interchangeable evidence
Dual-proxy contract audit	1,800 rule pairs	Core exact agreement 59.6%; 728 disagreements	Proxy disagreement should define uncertainty, not truth
35B SFT protocol learning	3 seeds, 77 dev	Structure rises from 0 to 73–74; two seeds have zero truncation	Structured supervision repairs protocol failures; compact results remain separate
Objective meta-evaluation	670 auto-extracted executions	2,680 views and 2,017 paired contracts	Diagnostic outcome and intervention tests without task-level human scores

Table 4: Summary of completed experiments. Brackets denote 95% intervals: Wilson for alert precision, cluster bootstrap for correlations, and date-cluster bootstrap for fragility routing. Neither detector changes nor dual-proxy agreement is human accuracy.

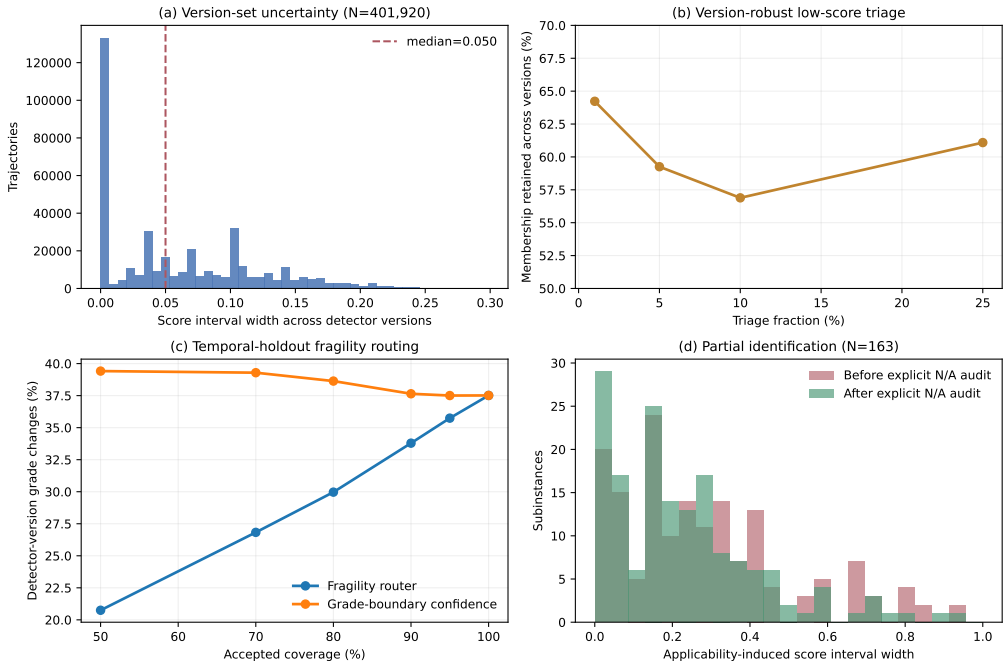

Figure 4: Set-valued and routing diagnostics. (a) Width of the quality-score interval induced by two detector versions. (b) Low-score triage membership retained across versions. (c) Grade-change risk on the final 12 days after accepting trajectories with the lowest predicted fragility; distance to a grade boundary is an ineffective confidence baseline. (d) Score intervals induced by unresolved neutral/N/A states narrow after the explicit N/A audit.

Model	Parse	15 criteria	Structure	Truncated
35B base	52/77	4/77	0/77	61/77
35B SFT seed17	74/77	73/77	73/77	3/77
35B SFT seed29	77/77	74/77	74/77	0/77
35B SFT seed43	77/77	74/77	74/77	0/77

Table 5: Completed 35B development-set protocol study. Structure requires a parseable object and all frozen fields; no row is a semantic-accuracy result or part of the matched compact-student comparison.

C.2 Large-Scale Metric Audit

Table 6 reports descriptive statistics from Production-Unlabeled. The central finding is not that recovery and alignment are solved, but that their current rules rarely observe the events needed to measure them. Exactly 99.74% of traces receive the maximum value for both error recovery and alignment, while 99.38% contain at most one user turn. Task evidence is also sparse: TES equals zero for 61.63% of traces. These values demonstrate why non-applicable metrics must be masked rather than averaged as if observed.

The mean composite quality is 0.7549. Mean quality decreases monotonically from 0.7923 for trajectories with 1–3 tool calls to 0.6060 for trajectories with more than 50. This relationship is descriptive and partly mechanical because tool count contributes to the efficiency score; it must not be interpreted causally.

C.3 Detector-Version Sensitivity

We join the pre- and post-fix metric exports by date and root trajectory identifier (Table 7). The correction changes 270K of 402K quality scores (67.1%), increasing the mean from 0.6931 to 0.7549.

Metric	Mean	At 1.0	At 0.0
Correctness	0.6173	0.00%	0.01%
Efficiency	0.5572	0.00%	0.00%
Error recovery	0.9984	99.74%	0.02%
Alignment	0.9992	99.74%	0.00%
Tool effectiveness	0.6604	31.79%	3.20%
Task evidence	0.2564	15.13%	61.63%

Table 6: Production audit over 402K trajectories. Mass at a boundary can reflect a default or missing evidence rather than genuinely perfect behavior.

Version-sensitivity statistic	Value
Changed quality scores	67.1%
Changed letter grades	36.9%
Pre/post Spearman ρ	0.678
Pre/post Kendall τ	0.510
Bottom-1% retention	64.2%
Bottom-1% Jaccard	0.473

Table 7: Effect of a detector correction on 402K matched trajectories.

More importantly for operational triage, 148K letter grades change (36.9%). Pre/post rankings have Spearman $\rho = 0.678$ and Kendall $\tau = 0.510$; only 64.2% of the pre-fix bottom 1% remain in the post-fix bottom 1% (Jaccard 0.473). The effect is not confined to a single collection window: the changed-score rate ranges from 57.5% to 77.6% across all 48 dates. This natural intervention supports storing detector versions and re-auditing rankings after parser changes.

C.4 Version-Set Identification and Fragility Routing

The two detector implementations define an empirical score interval with mean width 0.0618, median 0.0500, and 95th percentile 0.1765. Only 63.1% of intervals map to a single letter grade. On 2.0M seeded random trajectory pairs, 58.0% of relative orders are certified by non-overlapping intervals (Wilson 95% CI 58.0–58.1%), while 24.6% reverse between versions. Among pairs without certified separation, 58.5% reverse. This is not an accuracy comparison: the interval records what can be concluded without assuming either implementation is authoritative.

Operationally, retaining both implementations forever is expensive, so we ask whether fragile measurements can be recognized from the old export. A 46-feature logistic router trained on the first 36 dates and tested on the final 12 obtains AUROC 0.743 (date-cluster bootstrap 95% CI 0.723–0.764), average precision 0.604 [0.554, 0.672], and Brier score 0.198. The test prevalence of grade change is 37.5%. Accepting the 90% lowest-risk trajectories reduces this diagnostic risk to 33.8% [31.9%, 35.4%]; at 80% and 50% coverage it falls to 30.0% and 20.7%. By contrast, confidence based only on distance to the nearest grade boundary has AUROC 0.460 and becomes less stable as it selects more aggressively (Figure 4c). The router therefore captures provenance-related fragility that a scalar score margin misses, but it does *not* estimate human judgment error.

C.5 Rule Failure Modes in Audit-100

Manual audit exposes three failure modes that motivate Ravel directly.

Cross-task false positives. The initial definition of a "coding trace" was any trace containing a tool message. At least 11 of 19 "no code output" cases were actually question-answering or search tasks. Likewise, a completion-honesty rule flagged seven cases, but six were non-coding tasks. Its audited alert precision is therefore 14.3% (1/7; Wilson 95% CI 2.6–51.3%). A documented coding-task gate based on the presence of Write, Edit, MultiEdit, or Bash retains only the true alert (1/1), but the resulting Wilson interval is 20.7–100%; this tiny sample supports the failure mode, not a general performance claim.

Criterion	Explicit N/A
Code efficiency optimization	24.5%
Resource optimization	19.6%
Change propagation cohesion	18.4%
Codebase exploration	14.1%
Iterative community building	14.7%
All 15 criteria	10.0%

Table 8: Lower bounds from explicit N/A phrases in existing LLM-generated reasons; these are not human applicability labels.

Definition instability. A strict loop detector counts consecutive similar calls, while a broader detector also counts oscillations and retry-after-failure patterns. The strict detector identifies loop segments in 25 of 100 traces; the broader detector identifies 33. Binary agreement is 92.0% (Cohen's $\kappa = 0.807$, Jaccard 0.758), but count correlation does not imply interchangeable magnitudes: Spearman $\rho = 0.884$, while one trajectory receives counts of 5 and 97, a 19.4$\times$ difference. Both definitions are defensible, but they measure different phenomena and should not share an unqualified "loop" label.

Parser and evidence fragility. An early tool-call/result matcher attached only the first result after a multi-call assistant message. Fixing ordered matching changed build-pass rate from 10.7% to 76.8% and self-repair from 25.8% to 50.2%. These are not model improvements; they are measurement corrections. Reliability metadata should therefore include parser coverage and detector version.

C.6 Applicability and Rubric Redundancy

We next audit the existing AutoLibra outputs without treating them as human labels. The canonical file contains 15 criteria evaluated over 163 subinstances from 100 traces, for 2,445 criterion judgments. A high-precision phrase detector, restricted to neutral judgments, finds 245 explicit statements such as "this metric is not applicable" (10.0%). These statements account for 26.8% of all zero scores, showing that the current scalar value conflates applicable-neutral behavior with missing applicability. Table 8 gives the largest criterion-specific lower bounds.

Masking only these explicit N/A entries preserves the broad ordering (Spearman $\rho = 0.947$, trajectory-cluster bootstrap 95% CI 0.890–0.980), yet the 95th-percentile absolute score shift is 0.304 on the $[-1, 1]$ scale. Under the conservative E9 sensitivity set, unresolved zero entries induce a mean aggregate-score interval width of 0.280. Removing explicit N/A statements narrows it to 0.224 and raises the fraction of 12,973 cross-trace pairwise orders certified by non-overlap from 62.4% to 66.5%. Nevertheless, the median admissible rank span remains 59 positions among 163 subinstances. These bounds do not assert that every remaining zero is inapplicable; they quantify how much the current encoding leaves unidentified.

Rubric construction is also consequential: the iterative induction process grows from 7 to 33 raw criteria, reaching estimated aspect coverage of 75.0% in round four before declining to 72.0% in the fifth estimate. Deduplication reduces 33 raw criteria to 15 canonical criteria; raw-versus-canonical instance rankings have $\rho = 0.753$ (cluster-bootstrap 95% CI 0.661–0.816), and only 68.3% of the raw bottom quartile remains in the canonical bottom quartile. Thus applicability and redundancy are not merely presentation details: both can change which cases are triaged.

C.7 Rule–Semantic Proxy Consistency

Finally, we compare conceptually matched rule signals and canonical LLM judgments at the trace level. This is convergent evidence between two automatic systems, not validation against human truth. Technical-execution scores correlate with the code-output rule at $\rho = 0.365$ ($n = 82$, bootstrap 95% CI 0.124–0.575; Benjamini–Hochberg $q = 0.010$). Resilience correlates with self-repair at $\rho = 0.385$ ($n = 47$, CI 0.072–0.642; $q = 0.052$). Other anticipated pairs are weak or mixed: tool selection versus tool effectiveness yields $\rho = 0.172$, while the LLM aggregate versus the existing

satisfaction proxy yields $\rho = -0.118$. Partial convergence supports evidence fusion; inconsistent pairs motivate audited reliability and abstention rather than forced agreement.

C.8 Failure Triage at Scale

Within a 2,956-trajectory low-quality candidate set, tool-loop signals cover 91.0% of traces and are the primary assigned issue in 1,965 cases. Yet manual inspection reveals at least three subtypes: planning paralysis (repeatedly rewriting an unchanged task list), no-progress rereading (successful reads without edits), and legitimate debugging cycles falsely captured by syntactic repetition. This illustrates the division of labor in RAVEL: rules provide high-recall candidates and concrete event spans, while the semantic judge distinguishes intent and progress.

D Additional Diagnostic Tables

E Full Reproducible Diagnostic Results

This appendix reports aggregate outputs from E1–E9 and the frozen objective contracts supporting Tables 1 and 2.

E.1 Boundary Statistics for All Exported Scores

Metric	Mean	Median	Exact zero	Exact one	Distinct values
Composite quality	0.7549	0.7605	0.000%	0.000%	5,052
Correctness	0.6173	0.6274	0.007%	0.000%	4,127
Efficiency	0.5572	0.5709	0.000%	0.000%	8,616
Error recovery	0.9984	1.0000	0.018%	99.743%	3
Alignment	0.9992	1.0000	0.000%	99.743%	7
Outcome success (OSR)	0.5356	0.5000	0.117%	0.003%	8
Tool effectiveness	0.6604	0.6667	3.196%	31.793%	2,346
Task evidence (TES)	0.2564	0.0000	61.630%	15.132%	65
Task completion (TCR)	0.9974	1.0000	0.257%	99.743%	2
Interaction economy (IES)	0.4022	0.4604	5.793%	0.000%	7,888
Repeat-free rate	0.7122	0.7500	0.000%	35.947%	2,547
Repair rate	0.9974	1.0000	0.257%	99.743%	2
Recovery rate	0.9998	1.0000	0.018%	99.982%	2
User dissatisfaction rate	0.0013	0.0000	99.743%	0.000%	3
User correction burden	0.0004	0.0000	99.743%	0.000%	7

Table 9: Full boundary audit for all 15 exported scores on 402K trajectories. Exact boundary mass exposes metrics dominated by defaults or sparse triggering events.

E.2 Detector Changes by Signal

Signal	Changed trajectories	Changed rate	Mean before	Mean after / change
Composite quality	270K	67.1%	0.6931	0.7549 / +0.0618
Correctness	267K	66.4%	0.5119	0.6173 / +0.1054
Repeat-free rate	217K	54.1%	0.5134	0.7122 / +0.1989
Repeated-tool count	217K	54.1%	6.1851	3.8678 / −2.3173
Efficiency	217K	54.1%	0.4578	0.5572 / +0.0994
Tool effectiveness	213K	53.1%	0.4653	0.6604 / +0.1951
Task evidence (TES)	154K	38.4%	0.0000	0.2564 / +0.2564
Outcome success (OSR)	98K	24.4%	0.4901	0.5356 / +0.0455

Table 10: All signals changed by the detector correction. The remaining exported numeric fields are unchanged. Quality-score median change is 0.0500, 95th-percentile absolute change is 0.1765, and maximum absolute change is 0.2988.

E.3 Version-Set and Fragility-Routing Results

Version-set quantity	Value
Mean / median interval width	0.0618 / 0.0500
95th-percentile width	0.1765
Width > 0.10	30.3%
Singleton letter grade	63.1%
Certified sampled pair order	58.0% [58.0%, 58.1%]
Sampled pair order reversed	24.6%
Bottom-10% retention	56.9%

Table 11: Partial identification over the two available detector versions. The 2.0M sampled pairs use seed 20260713; brackets are a Wilson interval conditional on those pairs.

Coverage	Fragility router	Boundary baseline
100%	37.5%	37.5%
95%	35.7%	37.5%
90%	33.8%	37.6%
80%	30.0%	38.6%
70%	26.8%	39.3%
50%	20.7%	39.4%

Table 12: Detector-version grade-change risk on the final 12 dates after retaining the least-fragile trajectories. This is measurement-instability risk, not human-error risk.

E.4 Full Applicability Audit

Criterion	Explicit N/A	N/A rate	N/A / zero	Mean all	Mean after mask
Code efficiency optimization	40	24.5%	33.6%	0.086	0.114
Resource optimization and management	32	19.6%	27.6%	0.031	0.038
Change propagation cohesion	30	18.4%	43.5%	0.368	0.451
Iterative community building	24	14.7%	28.6%	0.190	0.223
Codebase exploration and understanding	23	14.1%	41.1%	0.178	0.207
Iterative adaptability	23	14.1%	42.6%	0.325	0.379
Diagnostic tool engagement	18	11.0%	17.5%	-0.012	-0.014
Dynamic task alignment	14	8.6%	28.6%	0.282	0.309
Resilience in problem solving	13	8.0%	23.6%	0.356	0.387
Problem-solving proficiency	11	6.7%	20.4%	0.313	0.336
Tool-selection appropriateness	7	4.3%	26.9%	0.644	0.673
Traceable decision documentation	5	3.1%	10.0%	-0.227	-0.234
Systematic planning and research	4	2.5%	7.4%	0.141	0.145
Technical-execution proficiency	1	0.6%	9.1%	0.589	0.593
Requirement understanding	0	0.0%	0.0%	0.491	0.491
All criteria	245	10.0%	26.8%	–	–

Table 13: Complete explicit-applicability audit over 163 subinstances per criterion. "N/A / zero" is the fraction of neutral judgments that explicitly state non-applicability. The detector is intentionally a lower-bound heuristic over LLM-generated reasons.

Identification statistic	Before audit	After audit
Mean score-interval width	0.280	0.224
Median width	0.218	0.194
95th-percentile width	0.733	0.598
Certified cross-trace orders	62.4%	66.5%
Median admissible rank span	71	59
95th-percentile rank span	111	104

Table 14: E9 sensitivity bounds before and after removing explicit N/A statements. Remaining zeros are allowed to be applicable-neutral or inapplicable; the table does not claim which state is correct.

E.5 Rubric-Induction Rounds and Rule Audits

Round	Raw criteria	Estimated coverage	Marginal gain
1	7	66.0%	–
2	13	72.0%	+6.0 pp
3	19	74.5%	+2.5 pp
4	25	75.0%	+0.5 pp
5	33	72.0%	−3.0 pp

Table 15: All five LADDER rounds. The evaluated aspect total varies slightly across rounds because these are saved LLM coverage estimates; the decline triggered early stopping and reversion.

Audit quantity	Result
Ungated completion precision	1/7 = 14.3% [2.6%, 51.3%]
Gated completion precision	1/1 = 100% [20.7%, 100%]
Strict loop positives	25/100
Broad loop positives	33/100
Positive-set intersection	25/100
Binary agreement	92.0%
Cohen's κ	0.807
Positive-set Jaccard	0.758
Count Spearman ρ	0.884
Largest observed counts	5 vs. 97

Table 16: Complete aggregate outputs for Audit-100. Completion brackets are Wilson 95% intervals; no completion recall estimate is available.

E.6 All Rule–Semantic Proxy Pairs

Semantic criterion	Rule or proxy	n	Spearman ρ	Bootstrap 95% CI	BH q
Resilience in problem solving	Self-repair	47	0.385	[0.072, 0.642]	0.052
Technical-execution proficiency	Code output	82	0.365	[0.124, 0.575]	0.010
Requirement understanding	1−UDR	100	0.196	[−0.006, 0.401]	0.140
Requirement understanding	1−UCB	100	0.181	[−0.008, 0.377]	0.165
Tool-selection appropriateness	Tool effectiveness	81	0.172	[−0.058, 0.396]	0.252
Technical-execution proficiency	Build pass	56	0.108	[−0.169, 0.394]	0.600
Problem-solving proficiency	Self-repair	47	0.106	[−0.232, 0.423]	0.610
Diagnostic tool engagement	Build pass	55	0.058	[−0.277, 0.403]	0.787
Iterative adaptability	Loop escape	24	0.017	[−0.417, 0.426]	0.938
Iterative adaptability	Self-repair	47	−0.051	[−0.356, 0.283]	0.789
Resource optimization	1−meta overhead	72	−0.103	[−0.331, 0.104]	0.600
LLM aggregate	Satisfaction proxy	100	−0.118	[−0.292, 0.066]	0.427
Dynamic task alignment	1−UDR	91	−0.225	[−0.420, −0.025]	0.112
Dynamic task alignment	1−UCB	91	−0.255	[−0.446, −0.061]	0.069

Table 17: All 14 pre-specified automatic-signal comparisons. Confidence intervals use 2,000 trace-level bootstrap resamples; q-values control the false-discovery rate across all rows. UDR denotes user dissatisfaction rate and UCB denotes user correction burden.

F Frozen Objective Artifact Contract

Layer	Size	Observable contract	Permitted use
Compact GRPO train	512 prompts	Metadata-stratified subset; four candidates; matched budget	Four-way reward ablation at steps 128 and 256
Reward-signal probe	128×4	Structure, finite reward, within-group variance, Naive–Set ranking difference	Train/dev-only go/no-go; never method selection on test
Natural execution	670	Pattern-matched verification command with explicit exit code	Diagnostic AUROC, AUPRC, BAcc, and Execution-Risk; label withheld from prompt
Event-ID equivariance	670 pairs	Score/completion invariant; citations follow ID map	Presentation invariance and evidence equivariance
Verification removal	670 pairs	Removing decisive evidence cannot improve assessment	Evidence necessity and Intervention-Risk
Outcome flip	670 pairs	Assessment follows controlled exit-result direction	Sensitivity only; excluded from natural outcomes
Observed recovery	7 pairs	Recovered execution should improve over failure	Exploratory temporal consistency
Proxy rule audit	180×10	Two-model applicability, fidelity, direction, evidence	Training-side reliability sets; never human validity

Table 18: Frozen compute-bounded training and objective evaluation contracts. Candidate/session-disjoint development data choose thresholds; untouched objective test data report strict results. Model proxies only parameterize training-side uncertainty.

The original and all transformed views of one execution instance remain in the same split. Development contains 124 natural instances and test contains 546; test labels are 517 successes and 29 failures. Every view retains complete task and verification events, while omitted complete events are represented by an auditable hash marker. Invalid model output is a strict error for threshold and relation metrics. The natural execution label, an explicit relation-type field, the expected direction, and proxy decisions are withheld from the judge prompt. However, controlled outcome-flip events contain synthetic marker strings in the model-visible tool output; this can reveal the intervention and limits interpretation of that relation.

F.1 Dual-Proxy and SFT Details

Proxy-audit field	Exact agreement	Cohen's κ
Applicability	81.1%	0.612
Observation fidelity	63.9%	0.492
Quality direction	91.8%	0.795
All core fields exact	59.6%	–
Mean evidence Jaccard	0.471	–
Disagreements	728	–

Table 19: Formal dual-model proxy audit on 180 trajectories and ten rules (1,800 paired decisions). These are cross-model reliability diagnostics, not human-agreement estimates.

The corresponding frozen rule file stores a conservative joint-support point and the hull of joint, GPT, and Claude Beta(1,1) 95% posterior intervals. Point supports range from 0.0168 to 0.6611, so treating every rule as equally reliable is a deliberately strong ablation. The GRPO data store applicability, point, lower, and upper reliability fields for every target.