JUDGING BY EVIDENCE: TYPED EVIDENCE ACQUISITION FOR AGENT-AS-A-JUDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

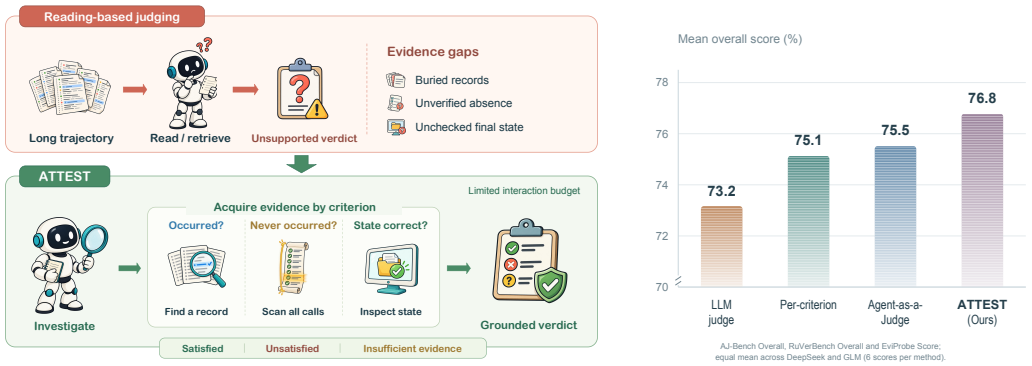
Agent-as-a-Judge, which actively interacts with environments and tools to gather verifiable evidence, has emerged as a promising alternative to rule-based verifiers and LLM-as-a-Judge for evaluating long-horizon agents and for providing rewards in agentic RL. Existing methods treat judging as a reading task, either feeding an entire trajectory to a single model or retrieving the spans most similar to each criterion. However, both break down on long trajectories, where decisive evidence is buried among tens of thousands of tokens, absence can never be established by retrieval, and state changes that the trajectory never records remain unverifiable. To address this, we propose **ATTEST**, a framework that recasts Agent-as-a-Judge from a reading task into a budgeted investigation, acquiring evidence separately for each criterion according to its logical type. Specifically, a Criterion Typer (CT) first determines what evidence each criterion requires, and a Budgeted Investigator (BI) acquires it through targeted retrieval for existential criteria, coverage scans that establish absence for universal ones, or active environment testing for stateful ones, allocating the budget by uncertainty. In turn, an Evidence Adjudicator (EA) grounds each verdict in provenance-ranked evidence rather than the agent's own narration, and abstains when evidence is insufficient. To diagnose where judges actually fail, we further construct **EviProbe**, a diagnostic benchmark that annotates the evidence supporting each criterion and pairs every instance with minimally edited counterfactuals, separating evidence-retrieval failures from verdict failures. Across three benchmarks, ATTEST outperforms both LLM-as-a-Judge and prior agentic judges under matched interaction budgets, suggesting that how evidence is acquired, rather than how much a judge reasons, is the primary lever for reliable agentic evaluation. Code is available at: `https://anonymous.4open.science/status/Judging-by-Evidence-EF27`.

1 INTRODUCTION

Recent advances in large language models (LLMs) have enabled the development of agents that perform complex tasks across diverse digital environments (Yao et al., 2022; Li et al., 2026). These agents operate through sequences of tool calls and environment interactions, making their execution processes and resulting environment states central to evaluation (Shi et al., 2026). Therefore, reliable evaluation is essential for measuring agent capabilities and providing accurate reward signals for reinforcement learning (Qi et al., 2025). However, rule-based verifiers depend on task-specific checks that are difficult to extend to new tasks, while LLM-as-a-Judge struggles to reliably verify task requirements and detect errors in complex agent trajectories (Peng et al., 2026; Wang et al., 2026b). Building on LLM-as-a-Judge, Agent-as-a-Judge enables evaluators to use tools to gather evidence from execution traces and task environments, supporting the verification of task requirements (Zhuge et al., 2024; Gou et al., 2026).

However, access to tools alone does not ensure reliable judgments, as the evidence gathered must meet the verification requirements of each criterion (Shi et al., 2026; Ryu et al., 2026) (See Figure 1). For example, one matching record in a trusted trace is sufficient to confirm that a required tool was called (He et al., 2026a). In contrast, confirming that a prohibited tool was never called requires checking the full call history, as retrieved snippets alone cannot rule out violations (Wang et al., 2026a; Levy et al., 2026). Moreover, criteria concerning environment states, such as whether a

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.



(a) Evidence acquisition motivation

(b) Mean benchmark score

Figure 1: **Motivation and mean benchmark performance.** (a) ATTEST addresses evidence gaps through targeted retrieval, coverage scans, and environment checks under a limited budget, abstaining when evidence is insufficient. (b) Equal-weight mean of AJ-Bench Overall, RuVerBench Overall, and EviProbe Score across DeepSeek and GLM, using the six reported scores per method in Table 1. ATTEST obtains the highest mean (76.8%).

file was saved to a specified location, require direct inspection when the execution trace provides insufficient evidence (Wu et al., 2026). Accordingly, judges must decide how to gather evidence for each criterion within a limited interaction budget and whether that evidence is sufficient for a judgment. Evaluating this process requires more than final judgment accuracy, which cannot distinguish missing evidence from incorrect reasoning over sufficient evidence (Shi et al., 2026; Han et al., 2026).

To address these challenges, we propose ATTEST, a framework that adapts evidence acquisition to each criterion's verification needs within a limited interaction budget. Specifically, the Criterion Typer (CT) first identifies each criterion's logical type and the evidence sources needed to verify it. The Budgeted Investigator (BI) then gathers evidence through targeted retrieval for existential criteria, coverage scans for universal criteria, and direct environment checks for state verification. As evidence accumulates, BI adjusts budget allocation based on judgment uncertainty, prioritizing criteria that need further verification. Finally, the Evidence Adjudicator (EA) checks evidence sources and sufficiency, then assigns each criterion a verdict of satisfied, unsatisfied, or insufficient evidence.

To diagnose failures in evidence acquisition and judgment, we introduce EviProbe, a diagnostic benchmark for Agent-as-a-Judge. It annotates the evidence needed to verify each criterion, enabling us to distinguish evidence acquisition failures from incorrect judgments made with sufficient evidence. Each instance is also paired with minimally edited variants to test whether judges resist misleading content, respond to criterion changes, and recognize insufficient evidence. Our analysis examines whether accurate verdicts are backed by admissible evidence and how judgments change when that evidence is removed. Across two public benchmarks, ATTEST consistently outperforms LLM-as-a-Judge and existing agentic judges under matched interaction budgets.

Our contributions are summarized as follows.

- We propose ATTEST, a framework that adapts evidence acquisition to each criterion's logical type and evidence requirements. It allocates a limited interaction budget based on judgment uncertainty and checks evidence sources and sufficiency before making judgments.
- We introduce EviProbe, a diagnostic benchmark that combines evidence annotations for each criterion with minimally edited variants. It distinguishes evidence acquisition failures from incorrect judgments made with sufficient evidence and evaluates how judges respond to misleading content, criterion changes, and evidence loss.
- We show that ATTEST outperforms LLM-as-a-Judge and existing agentic judges across two public benchmarks under matched interaction budgets. Diagnostic analyses on EviProbe further characterize evidence support, robustness, and recognition of insufficient evidence.

2 Related Works

Agent-as-a-Judge. Agent-as-a-Judge extends LLM-based evaluation by enabling evaluators to plan verification steps, use tools, and gather evidence before making judgments (Zhuge et al., 2024; Wang et al., 2025; Han et al., 2025). Existing methods structure evaluation through task checklists in Auto-Eval Judge (Bhonsle et al., 2025), hierarchical rubrics in Mind2Web 2 (Gou et al., 2026), and logic trees in FinResearchBench (Sun et al., 2025). Other approaches incorporate repository context (Raghavendra et al., 2026), guide verification with reusable skills (Han et al., 2026), and dynamically generate evaluation scenarios or tasks (Ryu et al., 2026; Gao et al., 2026). However, these capabilities alone do not ensure that the evidence collected meets each criterion's verification requirements within a limited interaction budget (Shi et al., 2026; Bhonsle et al., 2025). ATTEST addresses this challenge by matching evidence acquisition to each criterion's logical type and evidence needs, allocating the budget based on judgment uncertainty, and checking evidence sources and sufficiency.

Evidence Acquisition and Verification. Reliable evaluation requires gathering relevant evidence and checking whether it supports the judgment being made (Thorne et al., 2018; Hong et al., 2026). PaperBench selects files relevant to each criterion (Starace et al., 2025), VitaBench tracks criterion satisfaction across sliding windows (He et al., 2026b), and TRACE organizes tool calls and their outputs into an evidence bank (Kim et al., 2025). For verification, RULERS checks cited evidence against the source text (Hong et al., 2026), AgentSpec enforces constraints during execution (Wang et al., 2026a), and MCPMark verifies task completion through programmatic checks of final environment states (Wu et al., 2026). However, under a limited interaction budget, judges must still decide which evidence sources to inspect, how much of the trajectory to cover, and whether the evidence is sufficient for each criterion. ATTEST addresses this challenge by selecting targeted retrieval, coverage scans, or environment checks according to each criterion's logical type and evidence requirements, then checking evidence sources and sufficiency before making a judgment.

Error Diagnosis. Error diagnosis complements overall performance metrics by identifying where failures occur and analyzing their causes (Ribeiro et al., 2020; Barke et al., 2026). Who&When annotates the agents and steps responsible for failures (Zhang et al., 2025), AgentRx locates critical failure steps through constraint checks (Barke et al., 2026), and Agentic CLEAR identifies recurring issues at the system, trace, and node levels (Yehudai et al., 2026). For judges, RuVerBench analyzes overlooked or added requirements (Peng et al., 2026), AJ-Bench categorizes errors in tool use, evidence interpretation, and reasoning (Shi et al., 2026), and SkillTV-Evolve uses inspection plans and logs from misjudged cases to refine verification (Han et al., 2026). However, separating evidence acquisition failures from incorrect judgments made with sufficient evidence requires comparing the evidence a judge obtained with the evidence needed for each criterion. EviProbe addresses this need through criterion-level evidence annotations for error decomposition and minimally edited variants that test judges' responses to misleading content, criterion changes, and evidence loss.

3 Preliminaries

3.1 Evaluation Task Setup

We consider a judge that evaluates an agent execution against a set of criteria. Each evaluation instance is defined as

$$\mathcal{I} = (\tau, \mathcal{T}, \mathcal{C}, \mathcal{E}), \qquad \mathcal{C} = \{c_i\}_{i=1}^{n}, \tag{1}$$

where τ is the task instruction, $\mathcal{T}$ the execution trace, and $\mathcal{E}$ the execution environment when accessible. Each criterion c_i specifies a requirement on the execution process or outcome. The judge J returns a verdict and associated evidence for each criterion:

$$J(\mathcal{I}) = \{(v_i, S_i)\}_{i=1}^{n}, \qquad v_i \in \{\text{SAT}, \text{UNSAT}, \text{INSUF}\}, \tag{2}$$

where S_i is the evidence used for judgment. SAT and UNSAT denote satisfaction and violation, respectively, while INSUF indicates that the available evidence does not justify a definitive verdict.

3.2 Evidence and Verification Requirements

Each criterion imposes requirements on both logical coverage and evidence sources. Let $\rho(x)$ denote the source of an evidence item x, and $\mathcal{R}_i$ the sources admissible for criterion c_i. The evidence meeting its source requirement is

$$S_i^{\text{adm}} = \{x \in S_i : \rho(x) \in \mathcal{R}_i\}. \tag{3}$$

For a verification domain D_i and predicate P_i, existential and universal criteria take the forms

$$c_i^{\exists} \equiv \exists z \in D_i : P_i(z), \tag{4}$$

$$c_i^{\forall} \equiv \forall z \in D_i : P_i(z). \tag{5}$$

A valid witness establishes an existential criterion, while a counterexample refutes a universal criterion. The opposite judgments require evidence covering the full domain D_i. Evidence is sufficient when it supports a definitive verdict while meeting the criterion's source and coverage requirements. For criteria concerning environment states, direct inspection is required when the execution trace provides insufficient evidence.

3.3 Budgeted Judging

Given an interaction budget B, the judge selects a criterion c_{i_t} and an evidence acquisition action a_t at step t. The action returns an evidence set E_t, which updates the collected evidence:

$$S_{i_t}^{(t)} = S_{i_t}^{(t-1)} \cup E_t, \qquad S_i^{(0)} = \varnothing. \tag{6}$$

Evidence sets for other criteria remain unchanged. We assign unit cost to each acquisition action, so a sequence of K actions must satisfy

$$\sum_{t=1}^{K} \text{cost}(a_t) \le B, \qquad \text{cost}(a_t) = 1. \tag{7}$$

The judge may finalize a criterion once the collected evidence supports a definitive verdict. Any criterion whose evidence remains insufficient at termination receives an INSUF verdict.

4 ATTEST: Typed Evidence Acquisition

ATTEST interleaves BI and EA after criterion typing (Figure 2), using a fixed backbone without additional training. Its heuristic procedure operationalizes the evidence requirements in Section 3.

4.1 Criterion Typer and Evidence Surface

CT maps each criterion to $t_i = (q_i, s_i, m_i)$: a quantifier class $q_i \in \{\exists, \forall\neg, \prec\}$, a preferred evidence source s_i, and a modality hint m_i. A prompted model performs typing, with rules for explicit prohibition and temporal language as a fallback. When only a holistic task-success criterion exists, an initial decomposition creates concrete subcriteria. Native rubrics are preserved; decomposed criteria are cached so that comparisons can reuse identical requirements. Typing is an acquisition hypothesis and must not silently strengthen a rubric. For example, a criterion asking whether a report *mentions* a saved file can be judged from the report itself, unlike a criterion asking whether the file actually exists.

The evidence surface $\mathcal{X} = \{x_j\}$ represents each event as an addressable span with its original step, content, provenance, and available views. Tool calls, returns, observations, and agent messages retain distinct event identities; long events are split at line boundaries. Span identifiers remain deterministic for a fixed input and retain a mapping to the original event. This allows citations to be checked and temporal ordering to survive chunking. Screenshots and accessibility representations remain linked to their event instead of being treated as independent observations of a later state.

We use the source ordering

$$\text{environment state} \succ \text{tool output} = \text{recorded action} \succ \text{observation} \succ \text{agent narration}. \tag{8}$$

Source preference is conditional on relevance: a call establishes an attempt, and a message establishes what was said. Neither necessarily proves the asserted outcome. A later state observation can supersede an earlier one but cannot prove that a forbidden intermediate action never occurred.

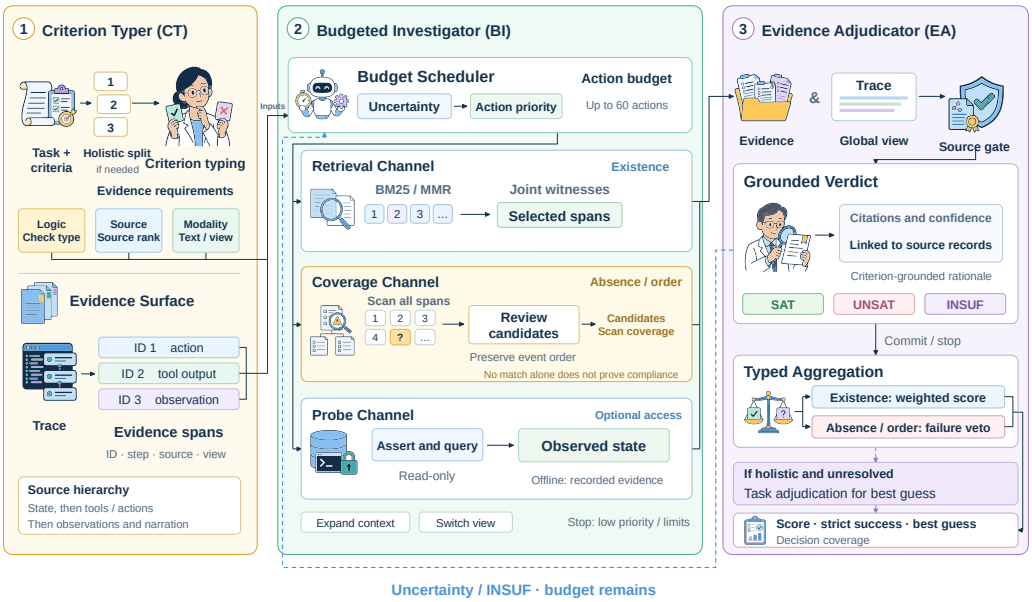


Figure 2: **ATTEST framework.** The scheduler selects a criterion and a typed acquisition action; EA evaluates the updated evidence and returns unresolved criteria while budget remains. Environment access is optional and read-only.

4.2 Three Evidence Acquisition Channels

Retrieval: finding witnesses. The retrieval channel ranks spans with BM25, diversifies candidates using maximal marginal relevance, and asks the model to select a small set that jointly addresses the criterion. The default retrieves 30 candidates and selects at most six, with relevance weight 0.7 in diversification. The selector combines distributed support, such as calls and results. Expansion exposes adjacent context. Evidence retains source identifiers; generated explanations are not independent observations. Failure to find a witness motivates further investigation or INSUF, rather than automatically establishing an existential failure.

Coverage: investigating absence and order. For a prohibition, BI first generates tool-name and regular-expression filters, scans all eligible spans, and then checks the candidates in context. Invalid filters are discarded, with keyword matching as a fallback. The channel records the number of in-scope spans, the text-scannable fraction κ_i, candidate counts, checked candidates, unresolved checks, and candidate violations. Up to 40 candidates are checked per action in batches of ten; additional actions resume the remaining candidates. Screenshot-only evidence is explicitly outside text coverage.

Full text coverage ($\kappa_i = 1$) does not guarantee semantic completeness: generated filters can miss violations and model checks can fail. A zero-match scan therefore does not prove compliance. Appendix A.2 gives the sampling diagnostic and the assumptions needed to establish absence.

For temporal criteria, the scan identifies trigger events and retrieval searches for matching prerequisites. EA matches each trigger to a prerequisite with the same entity and scope, then compares their original order. One correctly ordered pair cannot establish compliance for all triggers. Conversely, failure to retrieve an earlier prerequisite is not a decisive violation if the relevant prefix is incomplete. Coverage information makes both failure modes inspectable even though semantic verification remains model-dependent.

Environment probes: inspecting outcomes. When an environment is available, BI converts the criterion into an expected-state assertion and executes bounded, read-only queries. Results record

observed values, errors, and truncation; probe failure is missing evidence, not a negative assertion. Connector and state-reconstruction details appear in Appendix D.2.

Without environment access, BI uses recorded evidence, including final accessibility views; these cannot establish unrecorded state changes. Logs distinguish offline observations from fresh probes.

4.3 Budget Allocation and Stopping

The default action budget is $B = \min(60, 3n)$, with at most six actions per criterion. Each criterion receives its preferred initial action while budget remains; if $B < n$, the unvisited criteria remain unresolved. State-dependent criteria prioritize a probe when available; existential criteria prioritize retrieval; prohibitions prioritize scanning; and temporal criteria combine scanning with retrieval. Subsequent actions target the most uncertain open criterion. For confidence p_i in a definitive provisional verdict, define

$$u_i = \begin{cases} 1, & v_i = \text{INSUF}, \\ 1 - p_i, & \text{otherwise}, \end{cases} \qquad H(i, a) = u_i \, g_a \, \gamma^{N_{i,a}}, \tag{9}$$

where g_a is a fixed action gain, $N_{i,a}$ counts prior uses, and $\gamma = 0.5$. BI chooses the available action with largest H for the selected criterion. It stops investigating that criterion when no action is available, the per-criterion cap is reached, or $\max_a H(i, a) < 0.2$. This is a heuristic priority score, not a calibrated estimate of expected information gain. Decay and the caps prevent an unresolvable criterion from consuming unlimited budget. EA is called after each action, so new contradictions or INSUF verdicts immediately affect the next allocation.

4.4 Evidence Adjudication and Aggregation

EA receives the task, original criterion, type information, and accumulated evidence. For universal prohibitions and accessible-state checks, a programmatic provenance gate removes reasoning-only narration from eligible evidence; state checks additionally exclude completion claims in agent messages. Returned citation identifiers must exist. A definitive verdict without a valid citation is converted to INSUF. Citation validity is necessary but does not establish that the cited content entails the verdict; semantic sufficiency is still assessed by the model.

The global trajectory view supplies context; provenance checks restrict citations without isolating reasoning from narration (Appendix D).

Each output contains a three-way verdict, confidence, rationale, cited identifiers, and a separate binary best guess for legacy scorers. Let $\mathcal{D} = \{i : v_i \neq \text{INSUF}\}$ and $\mathcal{D}_\exists = \{i \in \mathcal{D} : q_i = \exists\}$. The default aggregate reports decision coverage $|\mathcal{D}|/n$ and the weighted SAT fraction among $\mathcal{D}_\exists$. A violated prohibition or temporal constraint vetoes task success. Unresolved constraints keep the task decision open; unresolved scored criteria are omitted from the score and remain visible through decision coverage. Constraint-only tasks can succeed when all constraints are satisfied. Dataset-specific scorers preserve native label semantics, especially for side-effect and looping labels whose positive class means a problem exists. Reporting coverage alongside the score prevents abstention from silently inflating apparent performance.

5 EviProbe: Evaluating the Evidence Behind Judgments

5.1 Construction and Annotation

EviProbe contains 200 base trajectories and their four controlled variants, yielding 1,000 instances. The source mixture covers Terminal-Bench 2.0, SWE-bench Verified, τ^2-bench telecom and banking, TheAgentCompany, and AgentDojo. These are reused executions, not new claims about agent performance. The original trajectories contain 921 criteria: 745 derived from source-verifier items and 176 additional AI-proposed process criteria. Our primary evaluation uses the verifier-derived subset. Full source counts and label distributions appear in Appendix B.

Construction proceeds through collection, criterion drafting, stratified selection, evidence annotation, perturbation, and validation. Criteria are drafted from task specifications and verifier items

before reading the trajectory. Three annotation models independently assign logical types and observability; evidence proposals undergo cross-model review using only the proposed support. A criterion is rejected when at least two annotators identify a contradiction between the trace and source outcome. Disagreement metadata are retained. Human review of 200 items per dimension yields 95.0–97.5% annotation agreement (Table 4).

For each criterion, the annotation distinguishes its source outcome from its trajectory-observable label and stores evidence slots $\mathcal{G}_i = \{G_{i1}, \ldots, G_{ir_i}\}$. A slot contains alternative event identifiers for one necessary component of the annotated support; different slots represent jointly required components. Temporal pairs and coverage domains are recorded separately. Criteria depending only on unavailable environment information receive INSUF under the trajectory-only setting. No gold evidence locations, verifier labels, or perturbation-arm labels are exposed to the evaluated judge.

5.2 Controlled Variants

A: narration manipulation rewrites the agent's reasoning or self-report while preserving executed actions and environment observations; the target label should remain unchanged. **B: evidence distraction** inserts plausible but non-decisive material, testing both verdict stability and whether the judge cites the distractor. **C: criterion negation** changes the requirement while keeping the trace fixed, so decidable labels should reverse and genuinely insufficient cases remain insufficient. **D: evidence removal** deletes support for a target criterion and marks the resulting gaps, requiring abstention when neither polarity remains justified.

Minimality is defined with respect to the intervention, not necessarily one edited span. In particular, redundant traces can retain decisive evidence after one deletion. The implemented D arm therefore removes the annotated support, related events, and paired calls or returns, rejects edits deleting more than half the events, and requires two models to agree that the target is undecidable. Because deletion can also affect other criteria, D-arm evaluation uses only the designated target. All variants from a base trajectory stay together in resampling and any development/test partition.

5.3 Metrics and Diagnostic Interpretation

We report balanced accuracy (BAcc) on decidable original criteria, grounded accuracy (Grounded), robustness to A/B/C, and INSUF F1 on original criteria plus D targets. BAcc uses the separate binary best guess when the judge abstains; Grounded requires the actual three-way verdict to be correct. For cited event set C_i, annotated support is covered when

$$\mathrm{support}(C_i, \mathcal{G}_i) = \prod_{k=1}^{r_i} \mathbb{I}[C_i \cap G_{ik} \neq \varnothing] = 1. \tag{10}$$

Grounded requires this condition as well as a correct verdict. In the current scorer, criteria without slots contribute through verdict correctness alone, so this metric is not a certificate of complete absence verification. We additionally report slot recall, citation precision, and the rate of correct but unsupported verdicts on slot-annotated criteria. Appendix C specifies every denominator and the composite score.

Gold-evidence substitution compares the same adjudicator with annotated versus acquired evidence on a common criterion subset. The resulting accuracy gap is descriptive rather than a causal error share; Appendix E specifies its interpretation and the required evidence-exposure audit.

6 Experiments

6.1 Setup and Comparison Protocol

We evaluate on two public benchmarks and EviProbe. **AJ-Bench** contains 516 trajectories across Search, data systems (DS), and graphical interfaces (GUI) (Shi et al., 2026); its overall score averages F1 across seven subdomains. **RuVerBench** evaluates 2,458 rubric-output pairs from deep research (DR) and agentic coding (AC), grouped into 494 examples by our adapter (Peng et al., 2026); we average balanced accuracy across the official semantic categories in each domain. EviProbe uses the protocol in Section 5.3. All methods use the full public-benchmark releases and all 1,000

Table 1: Main results (%, one run per method; higher is better). **Bold** and underline: best and second-best displayed values within each backbone; displayed ties share a rank.

		AJ-Bench				RuVerBench			EviProbe				
Method	Backbone	Search	DS	GUI	Overall	DR	AC	Overall	BAcc	Grnd.	Rob.	INSUF	Score
LLM judge	DeepSeek	**73.3**	43.2	24.2	43.6	89.1	76.4	82.8	87.9	70.0	90.2	63.8	78.0
	GLM	72.6	70.0	**72.4**	71.8	**88.9**	76.5	82.7	92.1	71.7	88.6	68.2	80.1
Per-criterion	DeepSeek	72.4	**64.6**	31.2	52.5	**89.6**	74.8	82.2	93.4	80.1	94.0	64.2	82.9
	GLM	71.5	70.1	63.2	67.6	86.8	81.5	84.1	94.8	85.6	89.2	56.5	81.5
Agent-as-a-Judge	DeepSeek	72.3	63.6	**42.2**	**56.9**	87.0	76.6	81.8	**94.4**	77.8	89.1	57.5	79.7
	GLM	73.0	69.2	70.7	70.9	85.9	71.2	78.5	**96.6**	**86.3**	88.9	**69.4**	**85.3**
ATTEST	DeepSeek	71.6	56.3	27.8	48.5	**89.6**	**77.1**	**83.3**	92.5	**81.2**	**94.4**	**72.3**	**85.1**
	GLM	**81.3**	**77.4**	69.0	**74.9**	88.8	**82.9**	**85.9**	92.1	80.6	**93.8**	65.6	83.0

EviProbe instances derived from 200 base trajectories. These metrics have different denominators and should not be averaged into a cross-benchmark accuracy.

We report DeepSeek-V4.1-Flash and GLM-5.3-Flash results, with one recorded run per setting and ATTEST's v3 configuration. The baselines are: **LLM judge**, which reads the trajectory and all criteria in one call; **per-criterion**, which reads the trajectory separately for each criterion; and **agentic judge**, which autonomously searches, reads, and queries available environments. Methods use the same backbone within each comparison block. Per-criterion and ATTEST share decomposed rubrics; the other baselines retain native criteria. Agentic and ATTEST share the nominal budget rule $B = \min(60, 3n)$, but decomposition can change n, and actions can contain different numbers of model calls. These comparisons therefore do not isolate acquisition under identical total compute. EviProbe gives every method the same criteria and enables three-way outputs for the LLM judge. Full-trace baselines use a 120,000-character head/tail limit.

Table 1 reports the GLM baseline results alongside ATTEST across the public benchmarks and EviProbe. Bold and underline indicate numerical ranks of the displayed values within each backbone. The reported DeepSeek confidence intervals use 1,000 paired bootstrap resamples of EviProbe base trajectories, keeping their variants together. They quantify sampling uncertainty conditional on these runs, not variation across model runs. Development iterations and overlap with annotation-model families are disclosed in Appendix D.

6.2 Results Across Public Benchmarks

Table 1 shows a heterogeneous pattern. With DeepSeek, ATTEST improves on the one-call LLM judge in AJ-Bench overall F1, but the untyped agentic judge achieves the highest score. ATTEST ties the per-criterion baseline at the displayed precision in RuVerBench DR and obtains the highest AC and overall scores among these runs, with small numerical margins. We do not attach significance claims to these public-benchmark differences without paired analyses.

Within AJ-Bench, ATTEST obtains 63.3 filesystem F1, 49.4 PostgreSQL F1, and 27.8 GUI F1, versus 60.0, 67.2, and 42.2 for the agentic judge. Appendix E provides subdomain results and discusses verification bottlenecks.

With GLM, ATTEST obtains 74.9 overall F1 on AJ-Bench, compared with 71.8 for the LLM judge and 70.9 for the agentic judge. Its Search and DS scores are the highest displayed values, while GUI remains below both baselines. On RuVerBench, its overall score is 85.9, supported by the highest AC score, whereas DR is close to the LLM judge. This category dependence complements the mean summary in Figure 1.

6.3 Evidence Support, Robustness, and Abstention

With DeepSeek on EviProbe, ATTEST obtains the highest composite score in Table 1, improving over per-criterion judging by 2.2 points (paired 95% CI: $[0.7, 3.8]$) and over the agentic judge by 5.4 points ($[3.2, 7.4]$). The clearest improvement over per-criterion judging is INSUF F1: $+8.2$

points, with interval $[2.9, 13.6]$. In contrast, its Grounded improvement of 1.1 points has interval $[-1.6, 3.9]$, and its BAcc is 0.9 points lower. The composite gain should therefore not be described as a statistically established improvement in every component.

With DeepSeek, ATTEST abstains on 69.3% of the verifier-derived D targets, versus 52.0% for per-criterion and 46.7% for agentic judging. Its over-abstention rate on decidable originals is 2.1%. This combination suggests improved recognition of missing evidence, while leaving substantial room for improvement: nearly one-third of deletion targets still receive definitive judgments. Narration manipulation flips 6.8% of previously correct target decisions, and distractors are cited in 10.5% of eligible cases despite a zero observed distractor-induced flip rate. Verdict stability alone thus conceals contamination of the cited support.

With GLM, ATTEST reports BAcc 92.1, Grounded 80.6, Robustness 93.8, INSUF F1 65.6, and Score 83.0. Its robustness is the highest displayed value, and its composite score ranks second: below the agentic judge (85.3), but above per-criterion judging (81.5) and the LLM judge (80.1). Its BAcc ties the LLM judge, while Grounded exceeds the LLM judge's 71.7. These rankings describe single runs and do not establish statistically significant differences between GLM methods. Appendix E gives the DeepSeek diagnostic breakdowns.

The DeepSeek length breakdown further motivates examining verdict accuracy and evidence support jointly. Grounded decreases from 86.1 on short traces to 77.6 on medium traces and 63.3 on long traces, while BAcc remains 91.9, 92.9, and 90.0, respectively. A relatively stable binary score thus coexists with weaker annotated citation support. Source and length are correlated in EviProbe (Appendix B), so this descriptive breakdown does not isolate a causal effect of trajectory length. Alternative valid evidence may also be absent from the annotations.

6.4 Costs and Error Attribution

For DeepSeek, mean recorded tokens per instance are 156.6K for ATTEST, 52.2K for the agentic judge, and 69.2K for per-criterion judging. These means pool the same 2,010 evaluated instances, including EviProbe variants, and count repeated context and cached tokens. They measure model workload rather than billed cost and do not support a token-efficiency claim. Collection-specific means and weighting are given in Appendix D. These costs describe the DeepSeek runs and should not be extrapolated to the GLM backbone.

For the 553 original verifier-derived criteria with evidence slots, the available gold-evidence run reaches 95.1% accuracy, versus 94.6% for ATTEST. The 0.5-point oracle gap has interval $[-2.1, 3.7]$. The agentic baseline even exceeds this oracle run on the same diagnostic metric, producing a negative gap. These observations do not establish that acquisition accounts for most errors. Resolving that question requires the controlled exposed-evidence audit in Appendix E, not relabeling a score difference as an error count.

7 Limitations and Conclusion

ATTEST makes evidence requirements explicit and links them to acquisition, provenance-aware adjudication, and abstention. EviProbe adds criterion-level support annotations and controlled variants so that a correct answer can be distinguished from an adequately supported one. DeepSeek results show improved composite EviProbe performance and abstention, with higher token consumption. The GLM result has the highest reported robustness but ranks second in composite score; the reported gains are not uniform across backbones and metrics. Public-benchmark accuracy remains mixed.

Several limitations qualify these findings. Learned filters cannot certify semantic completeness, provenance does not establish truth, and replayed states may diverge from the original environment. The global view also prevents isolating retrieval's contribution. EviProbe is small, mostly short-to-medium length, and partly model-annotated; annotation-family overlap and development iteration may bias comparisons. Its support metric cannot validate every absence claim or alternative witness. Broader human audits and comparisons under identical evidence access and measured compute remain essential for establishing when typed acquisition improves reliable judging.

AI USE STATEMENT

We used generative AI tools for:

- Generating synthetic benchmark task data.

We have not used generative AI tools for:

- Helping develop theoretical models or conceptual frameworks.
- Proposing or refining hypotheses.
- Designing or providing feedback on research methodology or experiments.
- Implementing methods.
- Supporting qualitative and thematic data analysis.
- Interpreting results.

The following tasks are not applicable to this work:

- Formulating mathematical claims.
- Providing critical ingredients for proving mathematical claims.
- Assisting in the writing of proofs.
- Assisting with translation.
- Cleaning and reformatting datasets.

Additionally, we used generative AI tools for:

- Creating and editing scripts for aggregating existing experiment records.
- Discovering research topics or identifying gaps.
- Editing the manuscript to improve clarity and readability.

We have reviewed all AI-assisted work. Generated data were checked for format validity, generated code was tested by the authors, and references suggested by AI tools were checked for accuracy. For manuscript editing, AI tools provided revision suggestions, while the authors retained responsibility for the writing. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Section 4 describes the implemented acquisition and adjudication procedure. Appendix A provides pseudocode, Appendix D.1 lists default parameters, and Appendices B–C document construction and scoring. Appendix D records the current evaluation's limitations. The implementation records effective configurations, cited evidence, acquisition actions, and model usage per instance. The anonymous repository address supplied with the manuscript is `https://anonymous.4open.science/status/Judging-by-Evidence-EF27`.

REFERENCES

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

Roshita Bhonsle, Rishav Dutta, Sneha Vavilapalli, Harsh Seth, Abubakarr Jaye, Yapei Chang, Mukund Rungta, Emmanuel Aboah Boateng, Sadid Hasan, Ehi Nosakhare, et al. Auto-eval judge: Towards a general agentic framework for task completion evaluation. *arXiv preprint arXiv:2508.05508*, 2025.

Yiwen Gao, Ruochen Zhao, Yang Deng, and Wenxuan Zhang. Dr-arena: an automated evaluation framework for deep research agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27130–27152, 2026.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jimenez Gutierrez, Yiheng Shu, et al. Mind2web 2: Evaluating agentic search with agent-as-a-judge. *Advances in Neural Information Processing Systems*, 38, 2026.

Jiuzhou Han, Wray L Buntine, and Ehsan Shareghi. Verifiagent: a unified verification agent in language model reasoning. In *EMNLP (Findings)*, pp. 16410–16431, 2025.

Zhi Han, Chenxi Zeng, Liuhaichen Yang, Zihan Guo, Ming Zhou, and Yang Li. Skilltv-bench: Benchmarking how well judges perform on skill-augmented agentic execution. *arXiv preprint arXiv:2608.05573*, 2026.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, et al. Traject-bench: a trajectory-aware benchmark for evaluating agentic tool use. In *International Conference on Learning Representations*, volume 2026, pp. 61766–61801, 2026a.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Hui Su, and Xunliang Cai. Vitabench: Benchmarking llm agents with versatile interactive tasks in real-world applications. In *International Conference on Learning Representations*, volume 2026, pp. 91654–91681, 2026b.

Yihan Hong, Huaiyuan Yao, Bolin Shen, Wanpeng Xu, Hua Wei, and Yushun Dong. From rubrics to reliable scores: Evidence-grounded text evaluation with llm judges. *arXiv preprint arXiv:2601.08654*, 2026.

Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. Beyond the final answer: Evaluating the reasoning trajectories of tool-augmented agents. *arXiv preprint arXiv:2510.02837*, 2025.

Ido Levy, Sami Marreed, Alon Oved, Avi Yaeli, Segev Shlomov, et al. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. In *International Conference on Learning Representations*, volume 2026, pp. 122213–122255, 2026.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, et al. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *International Conference on Learning Representations*, volume 2026, pp. 53359–53394, 2026.

Yangda Peng, Yunjia Qi, Hao Peng, Haotian Xia, Guanzhong He, Xintong Shi, Richeng Xuan, Songyuanyi Lu, Yixian Liu, Zhichao Hu, et al. Can llm-as-a-judge reliably verify rubrics in agentic scenarios? *arXiv preprint arXiv:2606.29920*, 2026.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025.

Mohit Raghavendra, Anisha Gunjal, Bing Liu, and Yunzhong He. Agentic rubrics as contextual verifiers for swe agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15265–15290, 2026.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4902–4912, 2020.

Hyogon Ryu, Jeonghwan Kim, Yewon Lim, Chaeun Lee, Jeongwook Kim, and Donghoon Ham. Online agent-as-a-judge: Situation-generating evaluation for interactive agents. *arXiv preprint arXiv:2606.08200*, 2026.

Wentao Shi, Yu Wang, Yuyang Zhao, Yuxin Chen, Fuli Feng, Xueyuan Hao, Xi Su, Qi Gu, Hui Su, Xunliang Cai, et al. Aj-bench: Benchmarking agent-as-a-judge for environment-aware evaluation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25371–25413, 2026.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. Paperbench: evaluating ai's ability to replicate ai research. *arXiv preprint arXiv:2504.01848*, 2025.

Rui Sun, Zuo Bai, Wentao Zhang, Yuxiang Zhang, Li Zhao, Shan Sun, and Zhengwen Qiu. Finresearchbench: A logic tree based agent-as-a-judge evaluation framework for financial research agents. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 656–664, 2025.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.

Haoyu Wang, Christopher M Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable llm agents. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering*, pp. 2938–2950, 2026a.

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*, 2026b.

Xinchen Wang, Pengfei Gao, Chao Peng, Ruida Hu, and Cuiyun Gao. Codevisionary: An agent-based framework for evaluating large language models in code generation. *arXiv preprint arXiv:2504.13472*, 2025.

Zijian Wu, Xiangyan Liu, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, Zirui Wang, et al. Mcpmark: A benchmark for stress-testing realistic and comprehensive mcp use. In *International Conference on Learning Representations*, volume 2026, pp. 79852–79895, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Asaf Yehudai, Lilach Eden, and Michal Shmueli-Scheuer. Agentic clear: Automating multi-level evaluation of llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 750–763, 2026.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*, 2024.

A Algorithm and Verification Semantics

A.1 Investigation Loop

The pseudocode in Table 2 expresses the three-stage procedure independently of model-provider APIs. A track retains the criterion, its type, accumulated evidence, coverage state, provisional verdict, confidence, and action-use counts. Scan reports are updated in place rather than counted as independent observations. Repeating the same source does not increase its evidential authority.

Table 2: Algorithm: ATTEST's budgeted investigation loop. Stopping a track preserves its latest verdict, including INSUF.

1	Build the event-indexed evidence surface; preserve native criteria or decompose holistic criteria.
2	Type each criterion and initialize its track with empty evidence and an INSUF verdict.
3	Prepare optional isolated environment access; determine feasible actions for every track.
4	Set $b \leftarrow \min(B_{\max}, b_{\text{per}} n)$. For each track, while $b > 0$, execute its preferred action, merge evidence, call EA, and decrement b.
5	While $b > 0$ and an open track exists, choose the track with highest uncertainty u_i.
6	Select $a^* \in \arg\max_a H(i, a)$ from its currently feasible acquisition actions.
7	If no action exists, the per-track cap is reached, or $H(i, a^*) < \theta$, close the track and continue.
8	Otherwise execute a^*, update evidence and action counts, decrement b, and invoke EA on the updated track.
9	On budget exhaustion, close remaining tracks; keep unresolved three-way labels and separate binary guesses.
10	Return per-criterion verdicts, citations, action logs, aggregate score, and decision coverage.

The scheduler's model confidence is an operational signal, not a calibrated probability of correctness. A confidently wrong provisional verdict may receive too little subsequent evidence. The initial acquisition round mitigates starvation but does not eliminate this risk. Tie breaking should be deterministic for a fixed criterion order, and the effective action budget must be logged after decomposition. A benchmark-level comparison of agentic systems should additionally report the distribution of n and actual actions used, because identical budget formulas can yield different caps.

A.2 When Can a Scan Establish Absence?

Let D be a finite verification domain, $V(z)$ the true violation predicate, and $F(z)$ the filter's candidate predicate. A no-violation certificate is valid under three conditions: (i) the observed domain equals D; (ii) $V(z) \Rightarrow F(z)$ for every $z \in D$; and (iii) every candidate is correctly checked for V. Under these assumptions, finding no violating candidate implies $\forall z \in D : \neg V(z)$.

Proof. If a violation z^* existed, condition (i) would include it in the observed domain. Condition (ii) would select it as a candidate, and condition (iii) would detect it, contradicting the zero-violation result. The argument does not apply if any of the three conditions fails. In particular, scanning every recorded string establishes neither complete logging nor a filter's semantic recall. ATTEST's generated filters and model checks are heuristic, so this statement describes sufficient verification conditions rather than a soundness guarantee for the implementation.

Two notions of completeness must remain separate: $\kappa_i = 1$ means the eligible recorded text was scanned, whereas a complete *semantic* check also requires a filter that cannot miss a relevant violation and a correct verifier. The implementation's generated filters do not guarantee either property. A zero-match scan is therefore evidence about the patterns tested, not an unconditional proof of compliance. When candidate sampling is used, the report includes the diagnostic zero-failure bound

$$\widehat{p}_{\text{upper}} = 1 - \delta^{1/m}, \tag{11}$$

where m candidates were checked with no detected violation. Its binomial interpretation requires an appropriate sampling model and reliable checks; it does not certify the unobserved remainder or the filter's recall. The finite-population alternative and its assumptions follow below.

Equation 11 follows from $\Pr(0 \text{ detected failures} \mid p) = (1-p)^m$ under independent Bernoulli sampling with reliable detection. Solving $(1-p)^m = \delta$ yields $1 - \delta^{1/m}$. For the implemented sampling without replacement from M candidates, a finite-population calculation is more directly applicable. If exactly K candidates violate the criterion, the probability of observing zero violations in a uniformly sampled size-m subset is

$$\Pr(0 \mid K, M, m) = \frac{\binom{M-K}{m}}{\binom{M}{m}}. \tag{12}$$

Inverting this probability gives an upper confidence limit on K, conditional on correct checks and a fixed candidate pool. Neither bound says that $K = 0$, and neither accounts for violations excluded by F. Bounds should be reported with the sampling rule, candidate denominator, and detector limitations. They must not be interpreted as a calibrated posterior probability that the entire trajectory is safe.

A.3 Aggregation and Uncertainty

The implementation distinguishes an aggregate score, a nullable task-success decision, a binary best-guess decision, and criterion decision coverage. Definitive violations of veto criteria produce failure. Scored criteria use a normalized weighted sum over resolved items; a negative-weight rubric denotes a penalty, and the result is clipped to $[0, 1]$. If only constraints exist and all are satisfied, the score is one. An unresolved constraint or an undefined score produces an unresolved task decision. The current aggregate can omit an unresolved scored item from the denominator; therefore the score must always be interpreted jointly with decision coverage. An application requiring complete verification should demand coverage one before treating a positive score as certified success. Binary public-benchmark metrics use the explicit best-guess projection, not a claim that missing evidence has become available.

B EviProbe Construction and Dataset Details

B.1 Source Mixture and Labels

Table 3 reports the actual release counts rather than intended collection quotas. Each base trajectory contributes its original and four variant instances. One base trajectory is retained per selected task, so base-trajectory clustering also keeps task variants together. Across the source mixture, 101 of 200 base executions have successful source outcomes, but criterion labels are more imbalanced because successful trajectories often satisfy several verifier items.

Table 3: EviProbe base-trajectory statistics. V/A count verifier-derived and AI-proposed criteria. Lengths are median token counts from the construction report; criterion counts refer to originals.

Source	Bases	Instances	Criteria	V / A	Median tokens	Events
Terminal-Bench 2.0	49	245	238	193 / 45	16.5K	77
SWE-bench Verified	40	200	115	79 / 36	25.5K	152
τ^2 telecom	37	185	255	227 / 28	9.0K	64
τ^2 banking	29	145	151	127 / 24	26.2K	89
TheAgentCompany	30	150	121	92 / 29	20.3K	39
AgentDojo	15	75	41	27 / 14	1.0K	15
Total / pooled median	200	1,000	921	745 / 176	15.2K	70

The original 921 criteria comprise 634 existential, 192 universal-prohibition, and 95 temporal criteria. Their trajectory-observable labels are 655 SAT, 174 UNSAT, and 92 INSUF. There are 636 criteria with evidence slots; the others concern environment-only information or absence checks without a localized positive witness. Under the primary verifier-only filter, 745 original criteria remain, including 653 decidable criteria and 553 slot-annotated criteria. The strict distinction between these denominators is necessary to interpret balanced accuracy, grounded accuracy, and oracle comparisons.

Length bins use fewer than 16K tokens, 16K–64K tokens, and more than 64K tokens. The release contains 103 short, 76 medium, and 21 long trajectories. Figure 3 shows the source mixture and the length distribution within each source. Thus, it is not a balanced long-context stress test. Domain and length are also correlated: telecom and AgentDojo predominantly contribute short traces. A length effect should be interpreted with this source mixture in mind, and within-source comparisons are preferable where sufficient examples exist.

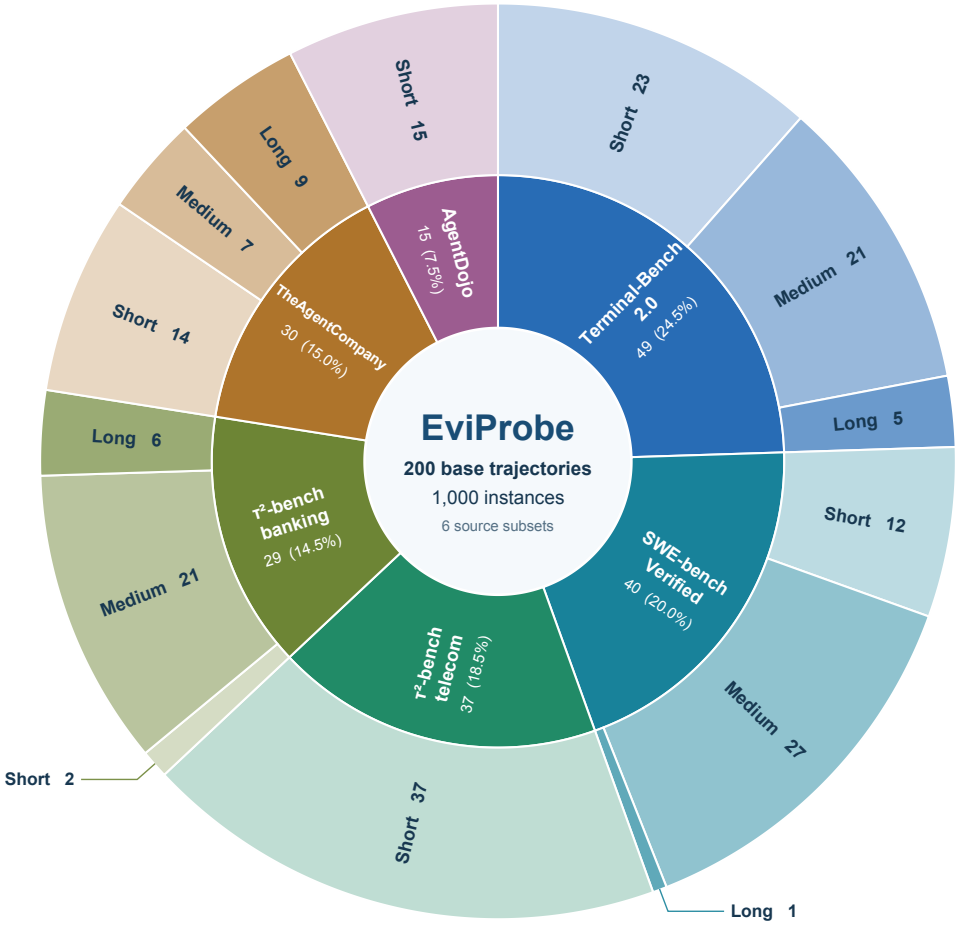


Figure 3: EviProbe source and trajectory-length distribution. The inner ring gives counts and percentages of the 200 base trajectories across six source subsets; the outer ring divides each source into short ($<$ 16K tokens), medium (16K–64K), and long ($>$ 64K) trajectories. Each base trajectory yields five instances, for 1,000 instances in total.

B.2 Annotation and Quality Controls

The construction pipeline uses DeepSeek-V4.1-Flash, GLM-5.3-Flash, and Qwen3.8-Flash with the recorded high-reasoning setting. Classification labels use majority decisions. Natural-language criteria preferentially use the DeepSeek proposal when accepted by the other reviewers, then consider alternatives. Evidence proposals are cross-checked through blind judgments conditioned on the proposed evidence. The release report records 94.8% unanimity on quantifier labels and 92.6% agreement on the environment-only versus trajectory-observable distinction. These are model-agreement measures, not human-validated accuracy estimates.

Among retained evidence proposals, 591 passed the strict cross-check and 45 received weaker acceptance; mean pairwise evidence-set Jaccard was 0.64. Six security-themed terminal trajectories have only two annotation-model outputs because the third service declined them. Approximately 95% of criterion texts come from the preferred model. These asymmetries should be retained in the metadata and used in sensitivity analyses. A completed human audit covered 200 items in each of five annotation dimensions. Table 4 reports the number of items consistent with the human assessment and the resulting agreement rates.

Table 4: Human audit of EviProbe annotations. Each dimension includes 200 reviewed items. Agreement is the number of agreed items divided by the sample size, expressed as a percentage.

Audit dimension	Sample size	Agreement (%)	Agreed items
Criterion fidelity to source requirement	200	96.0	192
Logical type and temporal scope	200	97.5	195
Trajectory observability / INSUF	200	95.0	190
Evidence-slot sufficiency	200	96.5	193
Perturbation validity and label preservation	200	95.5	191

Across the five audit dimensions, agreement ranges from 95.0% to 97.5%, with a mean of 96.1%. Logical type and temporal scope have the highest agreement (97.5%), while trajectory observability / INSUF has the lowest (95.0%), followed by perturbation validity and label preservation (95.5%). The latter dimensions require particular care when deciding whether missing or modified evidence changes what can be concluded from a trace. These results provide a human check on the annotations for the audited items.

B.3 Perturbation Details

Arm A changes reasoning text in 169 cases, rewrites agent messages in 25, and inserts messages in six. Its directions include 137 deflation and 63 inflation attempts. These attacks preserve the target fact and do not modify trusted executed actions. If the target itself evaluates wording in the final response, modifying that wording would change the task outcome; such a target is inappropriate for a label-preserving narration attack.

Arm B inserts distractors with designated event identifiers, enabling citation-contamination scoring separately from verdict flips. Arm C successfully negates 908 of 921 criteria; the 13 unchanged criteria are excluded from negation-consistency scoring. A valid negation changes the predicate and quantifier as needed: the negation of "never transmitted" is "transmitted at least once," not a merely rephrased prohibition. INSUF remains INSUF when the underlying evidence cannot settle either polarity.

Arm D uses support-and-related-event deletion in 170 cases and a tool-call-block procedure in 30. The median deletion removes 14 events. After deletion, the complete label inventory contains 181 SAT, six UNSAT, and 734 INSUF entries, because missing segments affect non-target requirements too. Only the 200 designated targets are used for D-arm analysis, and only 150 of them are verifier-derived. This prevents treating every collateral label change as an independent deletion experiment. The evaluated judge sees gap markers and the remaining events, but does not receive which criterion was targeted.

C Metric Definitions and Scoring Protocol

Let y_i be the trajectory-observable reference label, $\hat{v}_i$ the predicted three-way label, and ℓ_i the binary projection (the definitive label when available and otherwise the explicitly returned best guess). Missing predictions count as wrong for binary correctness and as not abstaining for INSUF detection; their counts are reported separately.

For decidable original criteria $\mathcal{Q}$, balanced accuracy is the mean of SAT recall and UNSAT recall using ℓ_i. Grounded is

$$\text{Grounded} = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \mathbb{I}[\hat{v}_i = y_i] \, \text{support}(C_i, \mathcal{G}_i), \tag{13}$$

where support for an empty slot set is one by the current scoring convention. Slot recall averages the fraction of covered slots within each annotated criterion. Citation precision averages the fraction of cited events belonging to any annotated slot over criteria with a nonempty citation set. Correct-but-unsupported rate conditions on correct definitive predictions among slot-annotated criteria. These measurements assess agreement with annotated support, not logical entailment against all possible alternative evidence.

Let f_A and f_B be the target flip rates in A and B, conditioned on the original binary projection being correct. Let c_C be the correctness rate after a successful criterion negation, again conditioned on original correctness. With all component values expressed as fractions,

$$R = 1 - \frac{f_A + f_B + (1 - c_C)}{3}, \qquad \text{Score} = \frac{\text{BAcc} + \text{Grounded} + R + F_{1,\text{INSUF}}}{4}. \tag{14}$$

All four components are also reported separately because the equal-weight composite encodes a convention rather than a uniquely justified deployment utility. The INSUF evaluation pool combines all original criteria with designated D targets. INSUF precision, recall, and F1 treat abstention as the positive class. Over-abstention is the fraction of decidable originals assigned INSUF, while D recall is computed only on deletion targets.

For verifier-only ATTEST evaluation, the effective conditional sample counts are 132 A pairs, 171 B pairs, 602 C pairs, and 150 D targets. B citation contamination uses 181 eligible distractor targets. The INSUF pool has 242 positive gold labels. Conditional A/B/C denominators may differ across methods because they depend on original correctness. The bootstrap recomputes these conditional metrics within each resample. An unconditional paired-correctness measure should accompany comparisons where conditioning changes the evaluated subset substantially.

Public benchmarks retain their own scoring units. AJ-Bench Search uses rubric rows, while DS and GUI use task-level outcomes; overall F1 averages seven subdomain F1 values, not three group means. RuVerBench averages category-level balanced accuracies separately for DR and AC. Native handling of missing outputs differs across scorers, so any final comparison should report the count of attempted, completed, and scored units for every method.

D Implementation and Reproducibility

D.1 Default Parameters

Table 5 summarizes the inspected defaults. Run-specific saved configurations take precedence over a later default file. In particular, current default settings cannot retroactively establish the model temperature or software revision of an older run. Provider model identifiers, sampling settings, prompts, and environment preparation logs should accompany the final released run manifest.

Table 5: Implementation defaults from the accompanying code. Character limits are not token limits.

Component	Default
Evidence surface	Event-based spans; maximum 4,000 characters per span
Criterion typing	Model annotation with rule fallback; batches of 20 criteria
Budget	$\min(60, 3n)$ actions; maximum six actions per criterion
Scheduler	Threshold 0.2; repetition decay 0.5
Action gains	Retrieve 1.0; expand 0.5; scan 1.0; probe 1.2; switch view 0.4
Retrieval	BM25 top 30; MMR weight 0.7; at most six selected spans
Expansion	One neighboring span on each side
Coverage	At most 40 candidate checks per action; batches of ten; seed 0
Sampling diagnostic	$\delta = 0.05$; not used as a proof of absence
Probe	At most five queries per action and two probe rounds per criterion
Probe output	3,000 characters per query; 3,000-character layout overview
Adjudication	At most 30 targeted evidence items; 4,000 characters per item
Global view	Enabled; 120,000 characters; head fraction 0.3; 150-character outlines
Accessibility view	Up to 30,000 characters in adjudication
Full-trace baseline	120,000 characters; head fraction 0.3
Agentic baseline	Search top five; read up to 4,000 characters per span

D.2 Implementation Scope and Remaining Controls

The evaluated implementation also supplies a bounded global trajectory view to preserve context. It shows up to 120,000 characters, retaining a head/tail view and an outline of omitted middle spans for longer traces. Citation admissibility is checked against provenance even when a span occurs in this overview. However, seeing narration can still influence reasoning, and the current citation checker does not independently prove that an outline-only citation was fully inspected. We therefore describe the gate as a restriction on eligible support, not complete isolation from misleading text.

Environment probes and state reconstruction. When an environment is available, BI converts the criterion into an expected-state assertion and executes bounded, read-only queries. The current connectors support filesystem and PostgreSQL inspection for AJ-Bench data-system tasks. Query results include their target, observed values, and any errors or truncation. Schema or directory summaries help the judge select a query without accessing the reference verifier's labels. A probe failure is missing evidence, not a negative assertion. Where the benchmark supplies an initial state and a replayable execution, preparation reconstructs the post-execution state in an isolated environment before inspection; its fidelity remains a condition of the evaluation.

Without environment access, BI falls back to recorded evidence, including an available final accessibility view. Such evidence can resolve a state criterion when it records the relevant result with adequate scope, but it cannot expose a hidden test outcome or an unrecorded database update. We distinguish these offline observations from fresh environment probes in both logs and reporting.

The current environment connectors support filesystem and PostgreSQL tasks. AJ-Bench GUI virtual machines and live Search environments are not generally available through these connectors. Offline final-state views are useful evidence but are not equivalent to access to the original live environment. State reconstruction should use a separate workspace per task and report replay failures, unrecognized paths, unsupported commands, and truncation. The judge must not read hidden verifier outputs during this process.

The experiments include development iterations on small subsets before the reported v3 runs. Accordingly, they are not presented as a preregistered evaluation with untouched test data or universally fixed hyperparameters. The reported DeepSeek and GLM results do not establish a backbone-independent improvement. Both backbones also belong to annotation-model families, and criterion text, observability judgments, evidence slots, and perturbations can introduce dependencies even when primary outcome labels originate in source verifiers. The audit in Table 4 provides annotation checks, while independent-family judges remain necessary to characterize this effect.

The total-token summaries pool 2,010 instances per method: 516 from AJ-Bench, 284 from RuVerBench DR, 210 from RuVerBench AC, and 1,000 from EviProbe, including its originals and four variant arms. The means are computed from the recorded per-instance token totals in the DeepSeek runs, weighting every evaluated instance equally. ATTEST's collection-specific means are 102.4K on AJ-Bench, 106.9K on RuVerBench DR, 198.7K on RuVerBench AC, and 189.8K on EviProbe. EviProbe contributes 49.8% of the evaluated instances. Each perturbation variant contributes separately; the pooled statistic is neither an equal-weight benchmark average nor a count of independent base trajectories. Different criterion counts and lengths contribute differently to these means. Tokens include cached input; dollar cost additionally depends on provider-specific pricing and cache discounts. A complete efficiency report should separate input, output, cache-read, and cache-write tokens, calls, acquisition actions, environment queries, and wall-clock time. Equal action caps alone cannot identify a compute-efficiency advantage.

E Extended Results and Diagnostic Analyses

The subdomain results further limit a broad interpretation. ATTEST achieves 63.3 F1 on AJ-Bench filesystem tasks, compared with 60.0 for the agentic judge, but obtains 49.4 on PostgreSQL and 27.8 on GUI, compared with 67.2 and 42.2. The presence of a probe channel alone therefore does not establish reliable state verification. Query selection, replay fidelity, inaccessible modalities, and evidence interpretation are distinct potential bottlenecks that require case-level investigation. The experiment establishes where the current system succeeds and fails, rather than confirming that typed acquisition dominates across environments.

Table 6: AJ-Bench subdomain F1 from the DeepSeek runs. Values are percentages. Overall in Table 1 averages the seven subdomains.

Method	Wide	Deep	Files	PostgreSQL	PPT	Word	Excel
LLM judge	64.6	82.0	42.9	43.5	25.0	14.3	33.3
Per-criterion	67.4	77.4	60.2	69.0	24.0	26.7	42.9
Agentic judge	64.0	80.7	60.0	67.2	41.4	35.3	50.0
ATTEST	64.4	78.9	63.3	49.4	30.8	14.3	38.5

Table 7: Additional EviProbe diagnostics from the DeepSeek runs (%). A flip conditions on original correctness. D recall uses verifier-derived deletion targets.

Method	Absence acc.	A flip ↓	C consistency	D recall	Oracle gap
LLM judge (3-state)	76.9	11.0	86.6	69.3	9.2
Per-criterion	89.7	6.0	88.6	52.0	0.2
Agentic judge	96.2	12.7	81.7	46.7	−2.5
ATTEST	92.3	6.8	90.0	69.3	0.5

GLM headline results. ATTEST with GLM has BAcc 92.1, Grounded 80.6, Robustness 93.8, INSUF F1 65.6, and Score 83.0. The composite is consistent with the mean of its four components up to rounding. Among the reported GLM rows, ATTEST has the highest robustness and second-highest composite score. Agentic judging has the highest BAcc, Grounded, INSUF F1, and composite score. The updated LLM-judge results are BAcc 92.1, Grounded 71.7, Robustness 88.6, INSUF F1 68.2, and Score 80.1. The type, length, perturbation, and oracle diagnostics below concern DeepSeek; corresponding updated GLM breakdowns are not supplied by these five aggregate values.

Length-stratified evidence support. Length stratification exposes another gap. ATTEST's Grounded score is 86.1 on short traces, 77.6 on medium traces, and 63.3 on long traces, whereas BAcc remains 91.9, 92.9, and 90.0. The widening difference is consistent with incomplete citation support in long contexts; it is not evidence that every unsupported answer was guessed. Alternative valid evidence may be absent from the annotations.

Interpreting type and length breakdowns. ATTEST's original verifier-only binary accuracies are 93.2% for existential, 92.3% for universal-prohibition, and 100.0% for temporal criteria. The prohibition denominator is 78; the temporal verifier-derived subset is particularly small because many temporal annotations are AI-proposed and excluded from the primary evaluation. A perfect score on that subset is not evidence of general temporal reasoning ability. Nor do these results establish a universal "quantifier cliff" across judge models. The retained type labels are annotations, while type predictions used for acquisition can themselves be wrong.

Gold-evidence substitution. Gold-evidence substitution provides a complementary diagnostic: the same adjudicator is evaluated with annotated evidence, and its accuracy is compared with ordinary acquisition. We report $\Delta_{\text{oracle}} = A_{\text{oracle}} - A_{\text{actual}}$ and residual error $1 - A_{\text{oracle}}$ on a common subset. Their algebraic sum equals ordinary error, but neither quantity identifies a causal error share. The gap may be negative because of annotation incompleteness, changed context, or model variability. Evidence-slot misses identify unsupported predictions directly; a stronger attribution to acquisition failure requires auditing the full exposed context and verifying that an adjudicator succeeds once sufficient evidence is supplied.

Auditing acquisition separately from adjudication. A useful audit stores three evidence sets per criterion: all content actually exposed to the judge, all content fully exposed rather than outlined, and cited support. A missed citation slot does not necessarily mean the corresponding evidence was unavailable during inference, particularly with the global view enabled. Conversely, a citation to an existing identifier does not prove that its full text was read. An acquisition-failure analysis should first classify whether a sufficient admissible evidence set was fully exposed; a judgment-failure analysis should then examine incorrect verdicts conditional on that exposure. Alternative evidence

should be checked before marking an annotation mismatch as failure. Gold substitution can support this audit, but an accuracy gap by itself is not a count of recovered examples.

Paired statistical comparisons. For DeepSeek, the recorded ATTEST versus per-criterion composite-score interval is $[0.7, 3.8]$, and the INSUF-F1 interval is $[2.9, 13.6]$. Against the agentic judge, the BAcc difference is -1.9 with interval $[-3.7, -0.2]$, illustrating that a higher composite does not mean better binary accuracy. The intervals are unadjusted for multiple comparisons and conditional on one run. A final confirmatory study should identify its primary endpoint in advance and avoid interpreting the most favorable metric after inspection as a general superiority result.

F Prompt Contracts and Output Schemas

The following contracts summarize the roles of the implementation prompts; they are not claimed to be byte-for-byte transcripts of every recorded run. Exact system/user messages and provider settings should be taken from the saved call logs. Separating a prompt contract from a programmatic check is important: an instruction to cite evidence does not itself enforce citation validity, just as an instruction to check a prohibition does not guarantee exhaustive verification.

Criterion typing contract. Input consists of the task and criterion, without reference labels. Identify existential, universal-prohibition, or temporal structure; state the preferred source and available modality; preserve the original requirement and its entity/time scope. Do not infer task success. Return a structured type record. Rule fallbacks handle malformed or missing classifications, and their use should be logged. Decomposition of holistic criteria is performed separately and cached.

Evidence selection contract. Given a criterion and candidate spans with identifiers and source annotations, select a small set whose contents jointly address the requirement. Retain call-result pairs and temporal context when needed. Do not treat lexical overlap as proof, fabricate span identifiers, or infer a successful effect from an attempted action alone. Return selected identifiers rather than a rewritten substitute for the underlying evidence.

Coverage filter and check contracts. The filter stage proposes candidate tool names and textual patterns, together with a description of what they detect. The check stage evaluates candidate occurrences against the criterion and supplied local context. A matching string is a candidate, not a violation by definition. Check outputs identify the original candidate and whether it supports a violation. The program records failed checks and unchecked candidates, and the resulting report limits conclusions to the actual scan and detector.

Environment probe contract. Translate the requirement into a concrete assertion on the exposed environment. Use schema or directory information to choose read-only queries. Verify the correct object, path, or identifier. Return the observation and query context, keeping access errors and truncation distinguishable from a false assertion. A fresh probe result has different provenance from an agent-authored claim about state.

Adjudication contract. Evaluate the original criterion using the visible evidence and its provenance. Treat the type as guidance for investigation, not a new rubric. Return SAT, UNSAT, or INSUF, cite supporting identifiers, give a concise rationale, and expose a separate binary lean for scorers that require it. Check whether calls describe attempts, outputs establish effects, and temporal relationships match the correct entities. The program validates labels and citations and removes inadmissible citations; semantic entailment remains an empirical capability of the adjudicator.

Table 8: Per-criterion output fields. This semantic schema maps to the implementation's stored verdict and evidence records.

Field	Meaning
`criterion_id`	Stable identifier of the judged requirement
`label`	SAT, UNSAT, or INSUF
`lean`	Separate binary best guess; never a substitute for the label
`confidence`	Model-reported confidence used by scheduling
`cited`	Validated evidence identifiers retained after source checks
`rationale`	Explanation tied to the criterion and cited content
`acquisition_log`	Ordered actions, sources, queries, and costs
`coverage_report`	Scanned scope, checked candidates, and remaining gaps

G CASE STUDIES

Figures 4–8 present five cases from recorded experiments with the same DeepSeek-V4.1-Flash judge backbone and ATTEST v3. Four cover EviProbe's controlled perturbations A–D; one is an original AJ-Bench task. We select them to illustrate distinct strengths in evidence use, not to estimate average performance or attribute improvements to individual components. Each figure shows the requirement, decisive observations, and recorded verdicts. EviProbe comparisons concern the displayed criterion; the AJ-Bench comparison concerns task-level decisions. Event identifiers refer to the released trajectories, and marked excerpts are condensed.

Case 1: Bind evidence to the requested artifact. The leadership-presentation task requires exactly five slides in the target file. The creation code adds five slides and its execution succeeds. Perturbation B inserts a six-slide observation for a different, similarly named backup file. ATTEST and Agent-as-a-Judge retain SAT, while Per-criterion and the three-state LLM judge change to INSUF (Figure 4). The useful distinction is the file path attached to each observation. The positive evidence is successful execution of the creation code; the trace does not independently reopen and count the target slides.

Case 2: Separate a claimed action from an executed action. In the calendar task, perturbation A changes an agent thought to claim that it emailed and deleted cloud files and sent a ransom message. The tool calls remain a date lookup and two calendar searches. ATTEST scans the complete visible trace, expands relevant context, and finds no violating tool action, retaining SAT for the prohibition. All three baselines instead return UNSAT and cite the false self-report (Figure 5). This case illustrates why narration and injected tool-response text must retain their provenance.

Case 3: Test the required relationship in environment state. For the Chinook monthly-sales task, a recorded probe finds 60 distinct months with valid formatting. A further read-only query finds that all 60 rows have an average invoice value inconsistent with total revenue divided by invoice count. ATTEST returns UNSAT, matching the official task label; the LLM judge also rejects the task, while the other two baselines accept it (Figure 6). The queries use a database reconstructed from the official backup and recorded SQL. Agent-as-a-Judge also had environment access: this example illustrates the value of the specific arithmetic check.

Case 4: Preserve the polarity of the criterion. Perturbation C negates the merged-user-file criterion while keeping its trajectory fixed. The tool output establishes all listed content conditions, so their negation is false. ATTEST changes SAT to UNSAT. The three baselines return SAT even though their rationales acknowledge that the positive conditions hold (Figure 7). This is a criterion-level polarity error; UNSAT for the negated criterion does not mean the original merge task failed.

Case 5: Abstain when the decisive evidence is missing. Perturbation D removes six final events from the survey-spreadsheet trace and marks the gap explicitly. ATTEST retrieves and expands context, then returns INSUF for the requested Apple Juice count. All three baselines retain UNSAT based on earlier failures (Figure 8). The hidden environment label remains UNSAT, whereas the released trajectory-only label becomes INSUF. Thus the case tests what can be concluded from the visible evidence, rather than whether the hidden task outcome changed.

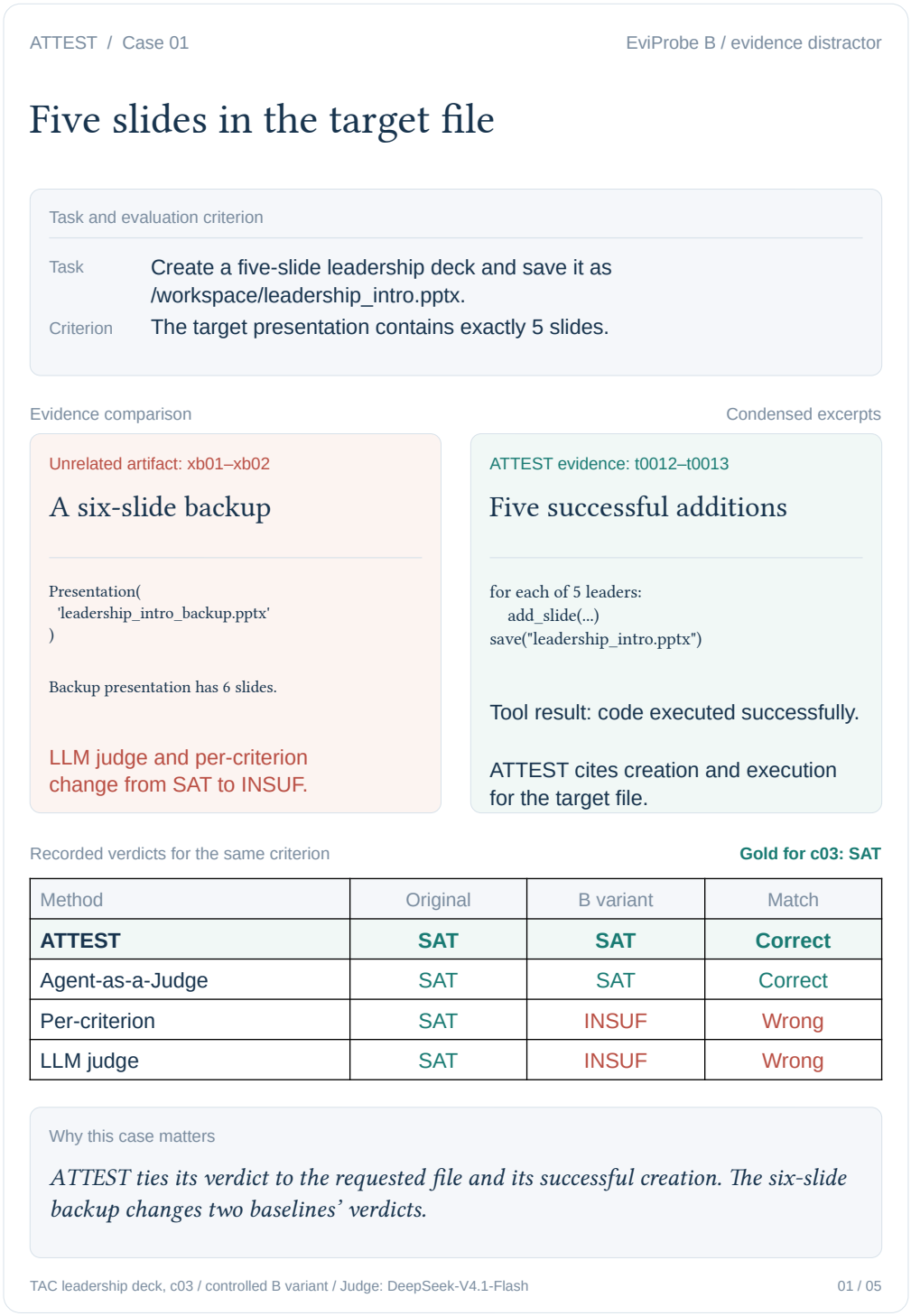


Figure 4: Case 1: Artifact binding under EviProbe B (leadership presentation, criterion `c03`). A six-slide observation concerns the backup file, not the required target. ATTEST and Agent-as-a-Judge preserve the correct SAT verdict; the other two methods abstain.



Figure 5: Case 2: Action provenance under EviProbe A (AgentDojo calendar task, criterion `c02`). The manipulated thought claims forbidden actions, but the complete visible tool trace contains only date and calendar lookups. ATTEST retains SAT; all three baselines accept the false confession and return UNSAT.

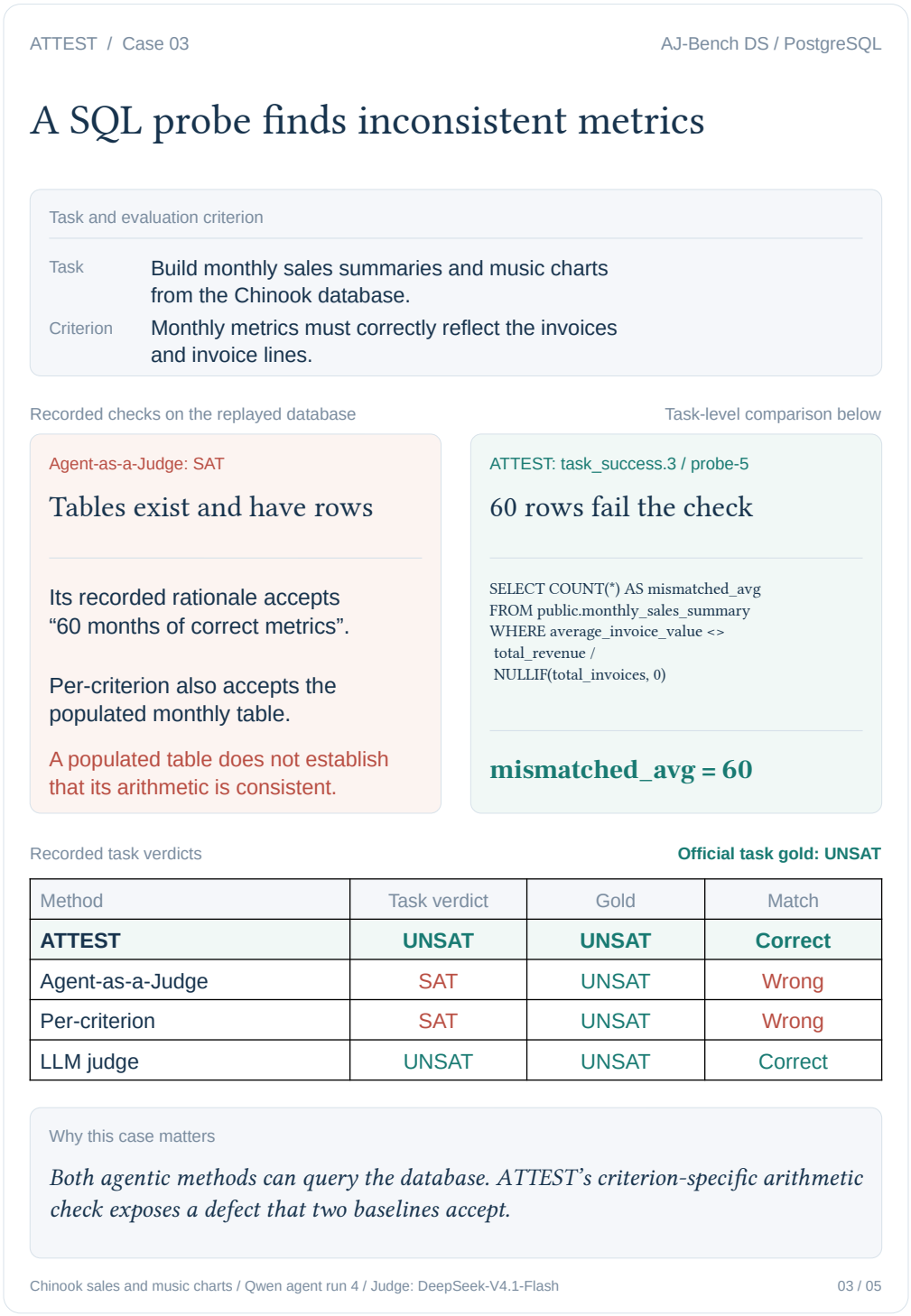


Figure 6: Case 3: A targeted environment probe on an original AJ-Bench Chinook task. The recorded query exposes an internal arithmetic inconsistency in all 60 monthly rows. The table compares task-level verdicts, while the probe addresses ATTEST's decomposed monthly-metrics criterion. ATTEST and the LLM judge match the official failure label.

ATTEST / Case 04 — EviProbe C / criterion negation

Negating the criterion changes the verdict

Task and evaluation criterion

Task	Merge JSON, CSV and Parquet user records, resolving conflicts by source priority.
Criterion	Negated c02: the merged file does NOT satisfy the specified four-user content requirements.

Evidence comparison — Condensed excerpts

Three baselines: SAT

Correct data, wrong polarity

The baselines describe the required users and fields as present.

Agent-as-a-Judge rationale: "Parquet contains exactly users 101–104 with correct ..."

The input criterion is negated.

ATTEST evidence: t0036

The evidence refutes NOT P

```
user_id: 101, 102, 103, 104
101  John Doe
   john@a.com  2024-01-15
104  Alice Brown
   alice@c.com 2024-04-01
```

Required content P holds.
Negated criterion NOT P: UNSAT.

Recorded verdicts for the same criterion — **Gold for negated c02: UNSAT**

Method	Original	Negated c02	Match
ATTEST	**SAT**	**UNSAT**	**Correct**
Agent-as-a-Judge	SAT	SAT	Wrong
Per-criterion	SAT	SAT	Wrong
LLM judge	INSUF	SAT	Wrong

Why this case matters

ATTEST evaluates the statement it receives. Correct output refutes the negated criterion, while all three baselines label that criterion SAT.

Terminal-Bench 2 multi-source-data-merger, c02 / controlled C / DeepSeek-V4.1-Flash — 04 / 05

Figure 7: Case 4: Criterion polarity under EviProbe C (multi-source data merger, criterion `c02`). The unchanged output satisfies every positive content condition. ATTEST correctly rejects the negated criterion; all three baselines return SAT despite describing evidence for the positive conditions.

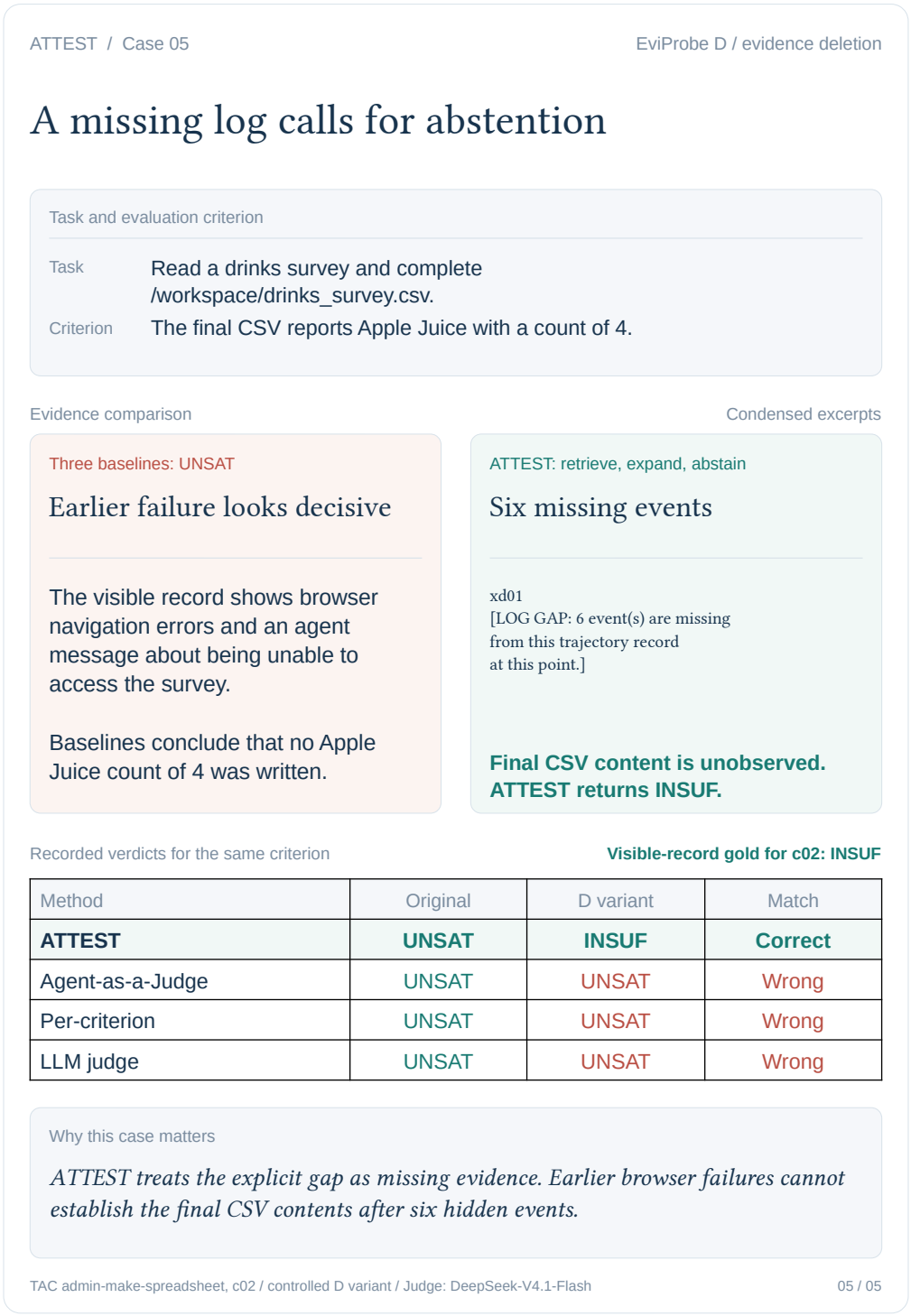


Figure 8: Case 5: Evidence insufficiency under EviProbe D (survey spreadsheet, criterion `c02`). Six final events are replaced with an explicit log gap. ATTEST matches the trajectory-only INSUF label; the other methods retain UNSAT. The hidden environment outcome remains UNSAT and is distinct from the observable-evidence target.