

# Judges Follow the Reference: The Derivation Gap in Verifying Tool-Call Arguments

Linhai Ma<sup>1,2</sup>, Yan Wang<sup>1,2,\*</sup>, Lingfei Qian<sup>1,2,\*</sup>,  
Yupeng Cao<sup>1,2</sup>, Víctor Gutiérrez-Basulto<sup>3</sup>

<sup>1</sup>The Fin AI, USA, <sup>2</sup>Yale University, USA, <sup>3</sup>Cardiff University, UK

Correspondence: {wy2266336,lfqian94}@gmail.com

## Abstract

Tool-call judges evaluate argument values proposed by LLM agents. We find that they accept a correct value less often when their input holds no reference for it, that is, no text that says what the value should be. This happens when the value is a default value from the function schema that the user did not mention, when it has to be inferred, or when the text it comes from was cut from the judge input. We call this drop in acceptance the *derivation gap*. It appears across all seven judges we evaluate, including System One models CLM and Jev. The gap is not only a truncation artifact: correct default values have the largest gap on a benchmark with no truncation. A controlled source-relocation test shows that judges follow the reference in their input: they favor a visible incorrect value over a correct value whose source is hidden. We introduce DEREf (Derived Reference): a separate model reads the full history without seeing the candidate and writes a reference for the judge. Across three benchmarks, DEREf improves overall argument AUROC for all judges tested. On the controlled test, it raises AUROC from chance or below to between 98.40 and 99.59. The code is available at [https://github.com/SarielMa/DeRef\\_release](https://github.com/SarielMa/DeRef_release)

## 1 Introduction

LLM agents act through tool calls. Wrong argument values are among the most common errors in the tool calls of LLM agents, as in FC-RewardBench (FCRB) (Agarwal et al., 2025). Agent pipelines therefore use judges to check tool calls, as reward models (Agarwal et al., 2025; Li et al., 2026b; Lin et al., 2026) and guards before execution (She et al., 2026). Many checks operate at the argument level, from the BFCL benchmark to ProvenanceGuard and ToolPRM (Patil et al., 2025; She et al., 2026; Lin et al., 2026).

An argument judge sees the context and a proposed value, the *candidate*, and decides whether

it is right. But a correct value can reach a call in several ways. It can be copied from visible text, taken from an earlier tool return, filled in from a default that the function schema specifies when the user does not mention the argument, reformatted, or inferred. We call each way a *derivation path*. Figure 1 shows one such path: a correct vehicle ID taken from an earlier tool return. The source of the ID is not visible to the judge, and all four judges reject the value. Correct arguments that the input does not show directly, like this ID, are accepted much less often than those it does. We call this drop in acceptance rate the *derivation gap*.

The gap appears in all seven judges we study. Two are System One judges, scoring models that return a probability rather than text (Kwok et al., 2026; Almeida, 2026). Five are LLM judges from the Qwen, Llama, and Mistral families (Yang et al., 2025; Qwen Team, 2026; Grattafiori et al., 2024; Jiang et al., 2023). Scaling the judge from 4B to 27B parameters does not remove the gap. The gap is also not only a matter of missing information: on FCRB, where nothing is cut and each default is shown, correct defaults have the largest gap.

Controlled tests point to one cause: a judge tends to accept a value that is stated in its input, and to reject one that is not. It does not reliably work out the correct value on its own. When we hide the source of a correct value and leave a wrong value visible, AUROC falls to chance or below for every judge tested. Giving the judge the full history narrows the gap on values from earlier steps. It does not help on values that match no span of the history. But when the input states the expected value and the reason for it, judges accept values they rejected before.

This points to a simple lever. If judges rely on a reference in their input, we can supply one. DEREf (Derived Reference) asks a model to read the full history without seeing the candidate. The model writes down the expected value of each argument

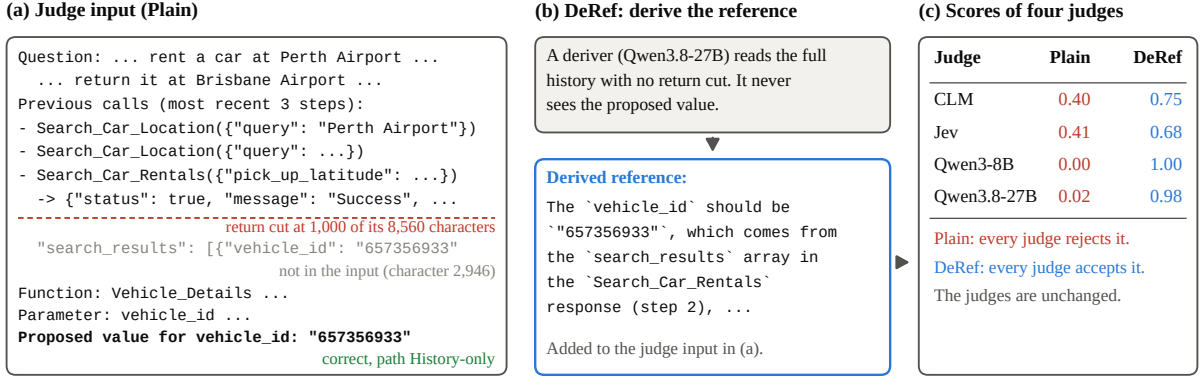


Figure 1: A correct argument on the HISTORY-ONLY path, from CFB. (a) The judge input keeps the first 1,000 characters of the search result, and the correct vehicle ID appears later. (b) DEREF asks a derivator to read the full history without the candidate and to write the expected value and its source. The reference goes into the same judge input. (c) Scores of the four judges, rounded to two decimals. With Plain input, every judge scores the value below 0.5 and below its own acceptance threshold  $\tau$  (Section 3.3); for Qwen3-8B, whose  $\tau$  on CFB is 0.002, the unrounded score is below that. With DEREF, every judge scores it above both.

and where it comes from (Figure 1b). The judge then scores the candidate with this reference in its input. In Figure 1c, all four judges then accept the vehicle ID. Unlike judges that write their own answer first, DEREF uses a separate model that reads the full history.

**Our contributions are as follows:** (1) We identify a systematic derivation gap across seven argument judges, from System One models to a 27B LLM. On ComplexFuncBench (CFB) and FinTrace it lies between 22.67 and 36.13 points for every judge, and on FCRB, with no truncation, correct defaults have a gap between 14.40 and 50.51 points. (2) Through controlled source relocation and reference substitution, we show that judgments track explicit reference values in the input, even when these favor an incorrect candidate over the correct one. (3) We introduce DEREF, which derives a shared reference from the full history without seeing the candidate. Without retraining or modifying the judges, it improves overall argument AUROC for every judge tested, on all three benchmarks. Averaged over tested judges, it is significantly above the strongest prior method, RevisEval, on every benchmark. On the controlled test with a hidden source, it lifts every judge from chance or below to between 98.40 and 99.59.

## 2 Related Work

**Judges for tool-calling.** Benchmarks check calls against gold answers, per argument (Patil et al., 2025) or per call (Zhong et al., 2025); without gold answers, learned judges score whole calls (Agarwal

et al., 2025; Li et al., 2026b) or each argument as a step (Lin et al., 2026), and benchmarks of such judges find that they tend to label steps as correct and agree less with a reference as tasks get harder (Li et al., 2026a; Fan et al., 2026b; Verma et al., 2026). Guards check calls before they run: ProvenanceGuard finds evidence for each argument with the value in hand (She et al., 2026), PACT and Agent-Sentry track where each value came from (Fan et al., 2026a; Sequeira et al., 2026), and TwinCheck repairs a flagged call from the trace (Dai and Huang, 2026). Jev trails GPT-6 where the verdict has to be derived rather than read off the text (Li et al., 2026c), and factuality metrics drop on summaries that need deeper reasoning (Ramprasad and Wallace, 2025). To our knowledge, none of these studies splits judge errors by derivation path.

**Reference-guided judging.** Judges grade better with a reference, and several have the judge write its own answer before seeing the candidate (Zheng et al., 2023; Zeng et al., 2024; Li et al., 2026d; Zhou, 2026; Andrade et al., 2026; Li et al., 2026e; Dhuliawala et al., 2024; Raghavendra et al., 2026), or revise the candidate into one (Zhang et al., 2025a); without a reference, judges agree less with experts and credit wrong answers (Krumdick et al., 2025; Chalamalasetti and Vajjala, 2026), and with a wrong reference they reject answers that match it (Lee et al., 2026). These judges write the reference from the question and their own knowledge, which is often not enough for an argument judge, whose expected value can sit in a tool return cut from its input, and a reference the judge writes from

its own input, as in LLMBar, stays below DEREf (Section 5.2).

**Correct content without support.** A value without support in the context is not always wrong: unsupported entities in summaries are often factual (Cao et al., 2022), and RAG models often answer correctly when the context is insufficient (Joren et al., 2025). Tool-call arguments make this common, since a correct default, format change, or inferred value has no matching span in the dialogue, and we measure how often judges treat such values as errors. A source in the context can also go unused (Modarressi et al., 2025; Kate et al., 2025). App. I discusses each line of work in more detail.

### 3 The Derivation Gap

This section identifies the derivation gap and investigates its causes. Section 3.1 defines the judging task and five derivation paths based on how argument values relate to the history and the function schema. Section 3.2 describes the benchmarks, judges, and judge input. Section 3.3 measures differences in acceptance rates for correct values across paths. Finally, Section 3.4 uses controlled tests to examine the roles of missing information and explicit reference values, motivating the introduction of DEREf in Section 4.

#### 3.1 Argument-Level Judging and Derivation Paths

**Task.** An agent acts in a state with history  $H$ , which contains the user turns, earlier tool calls, and their full returns. The agent proposes a call to a function  $f$  with  $k$  arguments, where argument  $i$  has a name  $a_i$  and a value  $v_i$ . An argument judge checks one value at a time using a context  $X$  built from  $H$  and the schema of  $f$ .  $X$  does not need to contain all of  $H$  (Section 3.2), so the text that determines a correct value can be present in  $H$  but absent from  $X$ . For each argument, the judge returns a score  $s(v_i | a_i, X) \in [0, 1]$ , representing its estimated probability that  $v_i$  is correct.

**Derivation paths.** A correct value is not always a copy of text that the judge can see. The derivation of a value  $v$  can be described by a *source*  $e$  and a *transformation*  $t$ . The source is a span of  $H$  or a field of the function schema, such as a default. The transformation copies the source, changes its format, or infers a new value from it, so that  $t(e) = v$ . Each candidate value, whether right or wrong, has a *derivation path*, determined by text-matching

rules (Appendix B). The paths capture how directly a value can be checked against the dialogue shown to the judge. There are five path types, and a value takes the first one that matches:

- **VISIBLE:**  $v$  appears in the dialogue part of  $X$ , that is, the user turns, shown calls, and retained portions of their returns.
- **HISTORY-ONLY:**  $v$  appears in  $H$  but not in the dialogue part of  $X$ . The matching text was cut or dropped, as for the vehicle ID in Figure 1.
- **FORMAT:**  $v$  matches a span of  $H$  after a format-normalization step, such as converting two date expressions to the same form.
- **DEFAULT:**  $v$  matches a schema default or a “don’t care” value explicitly specified in the parameter description.
- **NO-SOURCE:** no match is found under the preceding rules.

Correctness is determined by the benchmark’s gold call, not by the assigned path.

#### 3.2 Benchmarks and Judges

**Benchmarks.** We use three tool-calling benchmarks, each with a gold call at every step (Appendix A). FC-RewardBench (FCRB) gives 630 single-turn requests, each with a correct call and a wrong call made by a real model (Agarwal et al., 2025). ComplexFuncBench (CFB) has multi-step tasks with long tool returns (Zhong et al., 2025). Its wrong calls come from the trajectories of 30 models (AgentSuite, 2026), which gives 347 correct and 644 wrong calls. FinTrace gives 2,334 golden calls for financial queries (Cao et al., 2026) and 2,334 wrong calls, each made from a golden call by changing one argument by rule.

**Judges.** We study seven judges (Appendix F). CLM (Kwok et al., 2026) and Jev (Almeida, 2026) are System One judges, scoring models that return a probability for a question about their input. The other five are LLM judges: Qwen3 at 4B and 8B (Yang et al., 2025), Qwen3.8-27B (Qwen Team, 2026), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). Their score is the probability of “Yes” at the first output token, with thinking turned off. Section 3.3 reports all seven judges. The controlled tests of Section 3.4 and the main experiments of Section 5 use four of them: CLM, Jev, Qwen3-8B, and Qwen3.8-27B.

**Judge input.** All judges see the same context  $X$ , which we call the *Plain input*. We choose a truncated input on purpose: it keeps the last few steps and cuts each return to 1,000 characters (Appendix A), which is the input a judge sees in practice. Deployed agents stack cut context the same way: the OpenAI Agents SDK provides a trimmer that cuts tool outputs above 500 characters outside the last two turns, and Anthropic’s guidance for long-running agents clears old tool results and compacts earlier turns (OpenAI, 2026; Rajasekaran et al., 2025). FCRB requests have no earlier calls, so nothing is cut there.

**Statistics.** All confidence intervals (CIs) are 95% bootstrap intervals over 1,000 resamples of queries. A difference is called significant when its CI excludes zero (Appendix C).

### 3.3 Measuring the Gap

**Metric.** We compare how often a judge accepts correct values on different derivation paths. A judge accepts a value when its score is above a threshold. One fixed threshold for all judges is not fair, because judges accept values at very different score levels. On CFB, for example, Qwen3-8B accepts 80% of correct arguments at a score of 0.002, and CLM at a score of 0.631. For each judge, dataset, and input, we therefore set a threshold  $\tau$  at which the judge accepts 80% of correct arguments. Let  $A_\tau(p)$  be the share of correct arguments on path  $p$  that the judge accepts at  $\tau$ . The *derivation gap* of path  $p$  is

$$\text{Gap}_\tau(p) = A_\tau(\text{VISIBLE}) - A_\tau(p). \quad (1)$$

A positive gap means that correct values on  $p$  are accepted less often than correct VISIBLE values. We say that values on the VISIBLE path are *in view* and that values on the other four paths are *out of view*. We also report the gap of the out-of-view values pooled together.

The gap compares acceptance rates and not AUROC within a path. This is because AUROC within a path can stay the same when a judge scores all values on that path lower, right and wrong alike. A deployed judge applies one threshold to every argument, so such a shift means more false rejections on that path. Appendix C gives the rule for ties at  $\tau$ . It also repeats the analysis with  $\tau$  set to accept 70% or 90%, where every CFB and FinTrace gap tested stays significant.

**Results.** Table 1 gives the gap for all seven judges. For each benchmark, it reports the pooled out-of-view gap and the gap on the main out-of-view path of that benchmark: DEFAULT on FCRB, HISTORY-ONLY on CFB, and NO-SOURCE on FinTrace. On CFB and FinTrace, the out-of-view gap is significant for every judge. On FCRB it is small, and defaults carry the largest gap. Scale does not remove the gap, since Qwen3.8-27B, the largest judge we run, has a significant out-of-view gap on all three benchmarks, between 12.29 and 29.76 points.

**Base rates.** A gap could also arise from the base rate of a path, the proportion of its values that are correct: if the base rate of a path is lower, a judge would be right to trust its values less. On FCRB, 77% of default values are correct, against 75% of values in view. On CFB, 83% of history-only values are correct, against 82% in view. So base rates do not explain the gap in either case. On FinTrace, only 45% of no-source values are correct, against 97.5% in view, so this low base rate may account for part of the gap.

**Truncation is not the whole story.** On CFB, truncation is an obvious cause of the history-only gap, because it removes the text that these values match. But it cannot be the only cause. On FCRB nothing is cut, the default is written in the schema the judge reads, and correct defaults still carry the largest gap of any path. A judge might reasonably hesitate over a default that the user never mentioned. Yet base rates do not explain this gap. This suggests a second explanation besides missing information: the input has no explicit reference, that is, no text that states what the value should be. Section 3.4 examines both explanations with controlled tests.

### 3.4 Judges Follow the Reference

Two explanations remain for a rejected correct value. The judge may lack the information. Or it may have the information but no explicit reference. The tests below examine these two explanations. They use the four judges named in Section 3.2.

**Test 1: judges prefer visible distractors when sources are hidden.** If a judge merely lacked information, hiding the source should push its score toward uncertainty. Instead it follows the wrong value it can see. We build FCRB-Relocate from 257 FCRB states. We move the value of one argument from the user request into the return of an added earlier call. The request now points to the right entry by one of its attributes. The return

| Judge        | FCRB                   |                         | CFB                     |                         | FinTrace                |                         |
|--------------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|              | Out of view            | Default                 | Out of view             | History-only            | Out of view             | No-source               |
| CLM          | 8.15<br>[3.92, 12.07]  | 31.93<br>[24.36, 41.20] | 30.76<br>[27.95, 34.02] | 66.63<br>[59.42, 73.00] | 25.95<br>[23.46, 28.22] | 28.28<br>[25.34, 30.81] |
| Jev          | 8.11<br>[4.44, 11.74]  | 20.26<br>[12.68, 27.32] | 22.67<br>[19.08, 27.71] | 59.39<br>[54.22, 68.00] | 27.78<br>[25.20, 29.87] | 27.47<br>[24.54, 30.25] |
| Qwen3-4B     | 4.37<br>[0.37, 8.43]   | 21.26<br>[13.91, 29.63] | 34.71<br>[32.87, 36.78] | 76.81<br>[71.30, 81.93] | 23.30<br>[20.91, 25.94] | 23.29<br>[20.73, 26.35] |
| Qwen3-8B     | 0.63<br>[-3.78, 4.35]  | 14.40<br>[4.69, 22.14]  | 33.30<br>[31.13, 35.65] | 74.87<br>[68.09, 80.47] | 24.04<br>[22.13, 26.22] | 25.72<br>[23.59, 28.31] |
| Qwen3.8-27B  | 12.29<br>[8.19, 15.81] | 36.03<br>[27.56, 44.48] | 25.99<br>[21.10, 30.72] | 57.08<br>[50.28, 63.87] | 29.76<br>[27.44, 32.10] | 31.68<br>[28.98, 34.22] |
| Llama-3.1-8B | 8.31<br>[4.54, 12.33]  | 49.00<br>[40.85, 56.87] | 23.95<br>[19.54, 28.02] | 55.97<br>[50.37, 62.85] | 23.31<br>[20.53, 25.64] | 23.54<br>[20.25, 26.28] |
| Mistral-7B   | 12.25<br>[8.33, 16.03] | 50.51<br>[42.23, 58.31] | 36.13<br>[33.67, 38.43] | 80.54<br>[73.82, 86.99] | 30.68<br>[28.61, 32.85] | 32.47<br>[30.29, 34.82] |

Table 1: Derivation gap  $\text{Gap}_\tau$  (in points) with Plain input, with its 95% CI below. Out of view pools the four paths other than VISIBLE. The second column of each dataset is its main path.

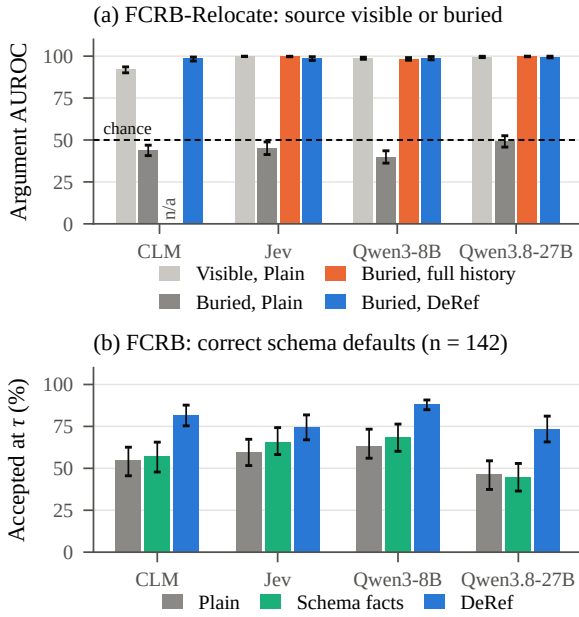


Figure 2: (a) Argument AUROC on FCRB-Relocate, with the source of the correct value in view (visible) or past the cut (buried). CLM is run at its default input length of 2,048 tokens and is not given the full history. (b) Acceptance at  $\tau$  of the 142 correct FCRB defaults whose DEREf reference says the default applies. Schema facts restates the default and lists related words in the request. Error bars show 95% CIs.

lists eight entries, the correct one and seven distractors whose values come from other FCRB samples. Each state has two versions that differ only in the order of the entries. The correct entry comes second, within the first 1,000 characters that the judge sees, or last, past that cut. The first distractor stays in view in both versions. This mirrors how history-only values arise in CFB (Appendix A). Burying the source drops argument AUROC from between 91.89 and 99.88 to between 39.80 and 49.03 for all

four judges (Figure 2a). Against the distractor in view alone, AUROC is between 26.71 and 39.88, so all four judges rank the wrong value they can see above the correct value that was cut. Given the full history, Jev, Qwen3-8B, and Qwen3.8-27B recover, to between 98.39 and 99.86.

**Test 2: restoring the information closes the gap only where a source was cut.** If truncation were the whole cause, the uncut history should close the gap on every path. We give Jev, Qwen3-8B, and Qwen3.8-27B the full history, with returns uncut and no window, up to each judge’s input limit (Appendix A). On CFB, this narrows the gap on history-only values by between 37.50 and 62.84 points, and for Qwen3.8-27B the remaining gap is no longer significant (Table 17 in Appendix H). Argument AUROC on these values rises by between 11.45 and 25.95 points. But on no-source values it raises AUROC for none of the three judges, and it lowers it for Jev by 2.83 points and for Qwen3-8B by 13.83 points (Table 18 in Appendix H). Putting back only the one cut step that holds the source has the same effect on FinTrace: it raises the scores of correct history-only values for all four judges, against a placebo step of the same kind (Appendix A). A reference also helps on top of the uncut history (Section 5.3).

**Test 3: the information can be in view and still not count.** In FCRB the default is written in the parameter description that the judge sees, and nothing is cut, yet judges accept correct defaults less often than correct values in view (Table 1). On the 142 of the 185 correct defaults whose DEREf reference (Section 4) says that the default applies, a line that restates the schema default and lists which related words appear in the request changes acceptance by

at most 6.18 points. The reference itself, which writes out this derivation, raises it to between 73.02 and 88.06 (Figure 2b). This subset is chosen by the reference. Table 17 in Appendix H reports all 185, with the same picture.

**A matching reference lifts wrong values too.** The three tests say that judges need a reference. The last test shows that judges follow a reference even when it supports a wrong value. On CFB, we add snippets from the full history around each place where the candidate value occurs, built by the snippet rule of Appendix G. Such snippets contain the value by construction, so they agree with any candidate. On the 59 wrong history-only arguments, they raise the mean score for all four judges, by between 14.28 and 39.39 points. At the 0.5 decision threshold, CLM, Jev, and Qwen3-8B then accept between 28 and 49 of the 59, while Qwen3.8-27B accepts 2. On FinTrace-HC, which pairs for the same argument a wrong value that appeared earlier in the history (H) with one that did not (C), CLM, Jev, and Qwen3-8B score H higher than C (Appendices A and G).

These tests show one pattern. A judge scores a value higher when a reference in its input matches it, whether the value is right or wrong. It does not reliably rebuild the derivation on its own, even from the full history.

## 4 Deref: Deriving the Reference

DEREF gives the judge a reference that is ready to use. It has two steps (Figure 1b). A deriver model reads the full history  $H$  and the schema, with returns uncut and no window (Appendix F). The deriver sees the function called at the step under judgment and its schema, but no candidate value. For each argument it writes the expected value and its source. If the history does not fix the value, it says so, and that statement becomes the reference. Our deriver reads everything, so it can name values whose source was cut from the judge input (Section 5.3). Because the deriver never sees the candidate, the reference cannot echo the value under test. The deriver writes one reference per argument of a state, and all candidates at that state share it. We use Qwen3.8-27B (Qwen Team, 2026) as the deriver for every judge, with thinking on (Appendix G gives the prompt and Appendix F the settings). We insert the reference into the judge input, after the history and before the function description. The judge itself is unchanged and scores

each argument as before. In all main experiments, every judge keeps its usual truncated input and reads the reference in addition (Appendix A).

## 5 Experiments

We ask whether DEREF improves every judge and how it compares with prior ways to give the judge evidence (RQ1), where the gains come from (RQ2), and whether the reference drives the judgment (RQ3).

### 5.1 Setup

**Data and judges.** We use the three benchmarks of Section 3.2 and the two controlled sets (Section 3.4 and Appendix A). Our main experiments run DEREF with four of the seven judges: CLM, Jev, Qwen3-8B, and Qwen3.8-27B. Qwen3-8B is the frozen backbone of CLM, so the two share one base model. Qwen3.8-27B is the largest open LLM we run.

**Inputs.** Plain is the truncated input of Section 3.2. We compare DEREF with three prior methods (Appendix G). ProvenanceGuard evidence gives the judge snippets of the history that a model quotes as the source of the proposed value (She et al., 2026). RevisEval gives the judge a revised version of the proposed call (Zhang et al., 2025a). Both use Qwen3.8-27B with the same settings as our deriver, and both see the proposed value. Self-generated reference has the judge model write the expected value from its own truncated input before it judges (Zeng et al., 2024). It needs a judge that writes text, so it runs with Qwen3-8B and Qwen3.8-27B. Full history has no window and no cut, up to about 32K tokens (FCRB has no cut). We run it with Jev and the two LLM judges. We report argument AUROC and  $\text{Gap}_\tau$  with  $\tau$  set again for each input. Appendix E gives call-level AUROC.

### 5.2 Main Results

**DEREF improves every judge.** Table 2 answers RQ1. It covers four judges and three benchmarks, that is, 12 pairs of judge and benchmark. DEREF raises argument AUROC over Plain in all 12 pairs, by between 1.44 and 22.14 points, and every gain is significant (Table 16 in Appendix H). The Llama and Mistral judges gain between 8.78 and 20.84 points, also significantly (Table 14 in Appendix G). The gains carry over to call-level AUROC, where all 12 are again significant (Appendix E).

| Judge and evidence                           | FCRB                   | CFB                    | FinTrace               |
|----------------------------------------------|------------------------|------------------------|------------------------|
| <i>CLM</i>                                   |                        |                        |                        |
| Plain                                        | 72.51                  | 63.41                  | 83.36                  |
| ProvenanceGuard evidence (She et al., 2026)  | 75.17                  | 69.62                  | 84.75                  |
| RevisEval (Zhang et al., 2025a)              | 83.26                  | 81.78                  | 86.07                  |
| DEREF (ours)                                 | <b>89.59</b>           | <b>85.55</b>           | <b>89.10</b>           |
| $\Delta$ vs. RevisEval                       | +6.33<br>[4.65, 7.98]  | +3.77<br>[1.72, 5.90]  | +3.03<br>[1.42, 4.79]  |
| <i>Jev</i>                                   |                        |                        |                        |
| Plain                                        | 90.83                  | 75.01                  | 89.11                  |
| Full history                                 | —                      | 79.82                  | 90.08                  |
| ProvenanceGuard evidence                     | 91.00                  | 80.74                  | 89.57                  |
| RevisEval                                    | 92.23                  | 87.15                  | 91.77                  |
| DEREF (ours)                                 | <b>93.30</b>           | <b>88.73</b>           | <b>93.10</b>           |
| $\Delta$ vs. RevisEval                       | +1.07<br>[−0.05, 2.13] | +1.57<br>[0.31, 2.98]  | +1.33<br>[0.77, 1.88]  |
| <i>Qwen3-8B</i>                              |                        |                        |                        |
| Plain                                        | 78.79                  | 66.79                  | 85.15                  |
| Full history                                 | —                      | 66.92                  | 84.74                  |
| Self-generated reference (Zeng et al., 2024) | 81.80                  | 68.28                  | 87.95                  |
| ProvenanceGuard evidence                     | 77.71                  | 64.80                  | 85.21                  |
| RevisEval                                    | 87.06                  | 81.75                  | 90.52                  |
| DEREF (ours)                                 | <b>89.48</b>           | <b>83.13</b>           | <b>92.28</b>           |
| $\Delta$ vs. RevisEval                       | +2.42<br>[0.89, 4.01]  | +1.38<br>[−0.64, 3.26] | +1.75<br>[0.97, 2.53]  |
| <i>Qwen3.8-27B</i>                           |                        |                        |                        |
| Plain                                        | 91.95                  | 78.92                  | 89.31                  |
| Full history                                 | —                      | 85.45                  | 89.77                  |
| Self-generated reference                     | 92.52                  | 81.83                  | 90.05                  |
| ProvenanceGuard evidence                     | 91.94                  | 88.08                  | 90.09                  |
| RevisEval                                    | <b>93.70</b>           | 93.54                  | <b>92.53</b>           |
| DEREF (ours)                                 | 93.39                  | <b>95.09</b>           | 92.25                  |
| $\Delta$ vs. RevisEval                       | −0.32<br>[−1.48, 0.80] | +1.55<br>[0.25, 2.83]  | −0.27<br>[−0.93, 0.38] |

Table 2: Argument AUROC (in points) of four judges with each kind of evidence (Section 5.1). Bold marks the best input for each judge and benchmark.  $\Delta$  is DEREF minus RevisEval, with its 95% CI below. FCRB has no cut, so Full history equals Plain there. Table 16 adds DEREF-v and the gain over Plain.

**Comparison with prior methods.** RevisEval is the strongest prior method in all 12 pairs. DEREF is above it in 10 pairs and significantly above it in 8. Averaged over the four judges, DEREF is significantly above RevisEval on every benchmark (Appendix H). In the other two pairs, Qwen3.8-27B on FCRB and FinTrace, RevisEval is higher by 0.32 and 0.27 points, and neither is significant. DEREF is significantly above ProvenanceGuard evidence in all 12 pairs. Self-generated reference gains at most 3.01 points over Plain and stays below DEREF in all six pairs where it runs.

**More context and more reasoning.** Giving the judge the full history instead of a reference is significantly below DEREF (Table 2). Averaged over the three judges that are given it, the difference is 11.58 points on CFB and 4.35 on FinTrace (Appendix H). More reasoning by the judge does not make up for the reference either. Qwen3.8-27B judging directly with thinking on stays below the same model reading the reference with thinking

| Variant                    | AUROC | $\Delta$               |
|----------------------------|-------|------------------------|
| DEREF                      | 95.09 | —                      |
| w/o full history           | 85.46 | +9.63<br>[7.53, 11.86] |
| Qwen3-8B as deriver        | 88.56 | +6.53<br>[4.70, 8.39]  |
| Deriver thinking off       | 88.18 | +6.91<br>[5.42, 8.47]  |
| Parsed expected value only | 90.90 | +4.19<br>[3.15, 5.43]  |

Table 3: Ablation of DEREF on CFB with Qwen3.8-27B (argument AUROC in points). Each variant changes one part of DEREF. The judge reads the truncated input with thinking off throughout.  $\Delta$  is DEREF minus the variant, with its 95% paired bootstrap CI.

| Judge thinking          | Reference             |                         | $\Delta_{\text{ref}}$   |
|-------------------------|-----------------------|-------------------------|-------------------------|
|                         | Without               | With                    |                         |
| Off                     | 85.45                 | 96.01                   | +10.56<br>[8.82, 12.26] |
| On                      | 90.27                 | 94.74                   | +4.47<br>[2.74, 6.33]   |
| $\Delta_{\text{think}}$ | +4.82<br>[3.18, 6.46] | −1.26<br>[−1.86, −0.79] | —                       |

Table 4: Reference availability and judge thinking on CFB with Qwen3.8-27B (argument AUROC in points). All four settings give the judge the full history. The two settings with a reference reuse the same references, written without the candidate and with deriver thinking on.  $\Delta_{\text{ref}}$  is With minus Without within each row, and  $\Delta_{\text{think}}$  is On minus Off within each column, with 95% paired bootstrap CIs.

off on all three benchmarks, by between 2.82 and 10.92 points (Appendix G).

**Controlled sets.** On the buried version of FCRB-Relocate, where all four judges fell to chance or below (Section 3.4), DEREF brings them back to between 98.40 and 99.59 (Figure 2a). This includes CLM, which is not given the full history. On the visible version and on FinTrace-HC, where the judges were already accurate, DEREF makes no judge significantly worse (Table 6 in Appendix A).

### 5.3 Ablation: Where the Gain Comes From

Table 3 removes one part of DEREF at a time. We show CFB with Qwen3.8-27B because CFB has the largest gap on a single path and includes sources that were cut from the judge input. Every drop is significant, and the largest comes from writing the reference from the judge’s cut input instead of the full history.

**Why keep the judge.** Keeping only the parsed expected value costs little where the reference states a value (1.39 points) and much where the parser finds none (22.37 points), so the text beyond the

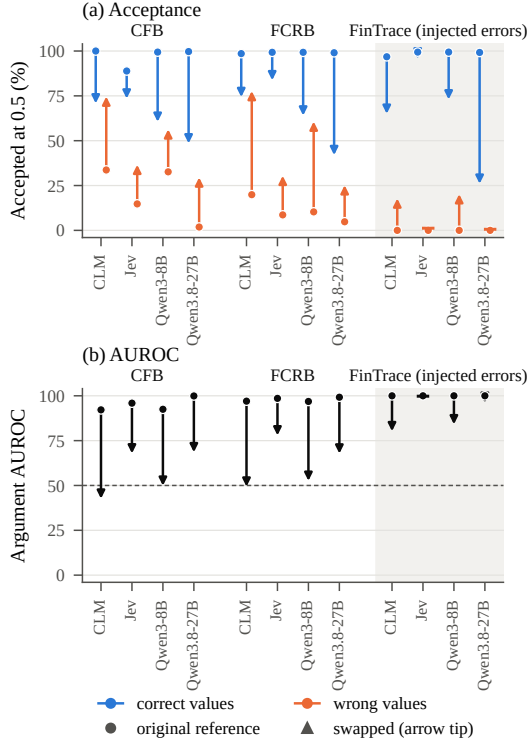


Figure 3: Changing one value in the reference. Dots show the original reference, which names the correct value. Arrow heads show it with a wrong value from the same state.

stated value helps mainly where it still carries the value (Appendix D). This is also why we keep the judge instead of matching the candidate against the parsed value: where the parser finds no value, a string match has nothing to compare, and the judge still uses the rest of the reference.

**Reference or more thinking.** Table 4 crosses the reference with judge thinking, with the full history in all four settings. The best setting is the reference with thinking off, at 96.01. So the gain comes from the reference, not from more reasoning by the judge.

**Why hide the candidate.** A deriver that sees the candidate, DEREf-v, is above DEREf on CFB and below it on FCRB and FinTrace, and it names the wrong value under test as the expected value up to six times as often (Appendix G). We therefore keep the deriver that does not see the value.

#### 5.4 Changing the Reference Changes the Judgment

We now test whether judges follow the DEREf reference itself. We take up to 500 arguments per benchmark whose reference names the correct value (332 on CFB, 417 on FCRB, and 500 on Fin-

Trace). In the reference text, we replace that value with a wrong value for the same argument from the same state and keep all other text. On CFB and FCRB, where the wrong values come from real models, all four judges follow the change (Figure 3). At the 0.5 threshold, acceptance of correct values drops by between 15.36 and 57.07 points, and acceptance of wrong values rises by between 20.38 and 57.79 points. Argument AUROC falls from between 92.16 and 99.91 to between 42.63 and 77.92. All 24 changes are significant. The swap also overturns judgments that agreed with the labels under Plain. For correct values that a judge accepted with Plain input, acceptance drops by between 10.85 and 53.62 points. For values labeled wrong that it rejected, acceptance rises by between 11.00 and 50.74 points. FinTrace behaves differently, because we made its wrong values by rule, and under Plain the judges accept at most 11.44% of them at 0.5. After the swap, their acceptance rises by at most 20.20 points, and AUROC stays between 80.26 and 99.77. Still, CLM, Qwen3-8B, and Qwen3.8-27B lower their acceptance of correct values by between 26.40 and 73.20 points, and Jev by 3.00 points. Since the reference is the only text we change, the change in judgment comes from the reference (examples in Appendix J).

**Summary.** RQ1: DEREf improves argument AUROC for every tested judge on every benchmark and significantly outperforms RevisEval when averaged over judges. RQ2: deriving a reference from the full history provides gains that more context or judge thinking alone does not achieve. RQ3: swapping the reference to name a wrong value can reverse judgments on CFB and FCRB.

## 6 Conclusion

Argument judges tend to score the candidate against explicit reference values in their input. When a correct value matches only a return that was cut, or is a default with nothing saying that it applies, the input holds no such reference, and judges of every type and size we test accept it less often than a value shown in the input. On the two multi-step benchmarks, the same holds for values that match no span of the history. DEREf derives the reference from the full history before the candidate is seen. Providing this reference improves overall argument AUROC for all six judges we test on three benchmarks, with no change to the judges, and substantially reduces the gap on cut returns and

on defaults.

## Limitations

Each argument has one gold value in our data, so a reference that names it can lead judges to reject another value that is also correct (Appendix A). On CFB and FinTrace, the wrong labels come from our own construction, and an audit of 100 items per dataset by two annotators could not confirm 3% and 4% of them. Appendix A runs a worst-case analysis of this label uncertainty, and every gain of DEREf over Plain remains significant. FCRB labels follow the official BFCL checker and are not independently audited.

On FinTrace, DEREf does not shrink the gap on no-source values for three of four judges (Table 17). The low base rate of this path may account for part of that gap (Section 3.3).

Main results use one deriver model, Qwen3.8-27B, held fixed across judges and input conditions. In the ablation (Table 3), a weaker deriver, Qwen3-8B, still improves Qwen3.8-27B’s argument AU-ROC over Plain on CFB. Broader evaluation of derivers from other model families is left to future work.

Our judges include dedicated scoring models (CLM and Jev) and open LLMs of up to 27B parameters, which show the pattern across judge types and sizes. Evaluation on general-purpose frontier LLMs as judges is left to future work.

## Ethical Considerations

We use three public benchmarks as released, together with public trajectories of agent runs on ComplexFuncBench. Their requests ask for function calls in tasks such as travel booking and financial queries, and their tool returns hold data such as hotel listings and company financials. We do not collect new data from people, and we do not try to identify anyone. Names in the requests come from the benchmarks, and we add no personal information. Two authors checked samples of labels and paths by hand, and we used no crowd workers. Jev, which we reach through an outside API, receives only benchmark text and model outputs.

Argument judges can gate tool calls that have real effects, such as a booking or a trade. Our results show that judges follow the reference in their input, and the swap test shows that a wrong reference overturns many judgments. So a deriver that errs, or a tool return written to mislead it, could

make a judge accept a wrong argument. We therefore see DEREf as support for review, not a replacement for it. Calls with high stakes should still go to a person or to a check that does not rely on the judge. We recommend keeping each derived reference with its scores, so that a reviewer can see why a value was accepted.

The benchmarks are task-oriented, and their texts are function requests, schemas, and tool returns. We did not see offensive content in the items we read for our label and path checks. We did not screen every item, and we use the texts as released.

## Artifact Use and Licensing

FC-RewardBench and ComplexFuncBench are released under Apache-2.0, and FinTrace under MIT. CLM, Qwen3, Qwen3.8-27B, and Mistral-7B-Instruct-v0.3 are released under Apache-2.0, and vLLM under Apache-2.0. We use Llama-3.1-8B-Instruct under the Llama 3.1 Community License. We access Jev through OpenRouter under its terms of service. After review, we will release our code, the path labels, and the two controlled sets.

## Use of AI Assistants

The authors used AI assistants, including ChatGPT, to support language polishing, brainstorming, and LaTeX editing. All technical claims, experimental results, analysis, and final writing decisions were reviewed and verified by the authors.

## References

- Mayank Agarwal, Ibrahim Abdelaziz, Kinjal Basu, Merve Unuvar, Luis A. Lastras, Yara Rizk, and Pavan Kapanipathi. 2025. [ToolRM: Outcome reward models for tool-calling large language models](#). *Preprint*, arXiv:2509.11963.
- AgentSuite. 2026. [complex-func-bench agent trajectories](#). Hugging Face dataset. <https://huggingface.co/datasets/AgentSuite/complex-func-bench-trajectories>. Accessed 2026-10-04.
- Diogo Almeida. 2026. [Introducing system one models & Jev](#). TypeSafe blog. <https://typesafe.ai/blog/introducing-system-one-models-and-jev>.
- Moises Andrade, Joonhyuk Cha, Brandon Ho, Vriksha Srihari, Karmesh Yadav, and Zsolt Kira. 2026. [Let’s think in two steps: Mitigating agreement bias](#)

- in MLLMs with self-grounded verification. In *International Conference on Learning Representations (ICLR)*, pages 36919–36972.
- Meng Cao, Yue Dong, and Jackie Cheung. 2022. **Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.
- Yupeng Cao, Haohang Li, Weijin Liu, Wenbo Cao, Anke Xu, Lingfei Qian, Xueqing Peng, Minxue Tang, Zhiyuan Yao, Jimin Huang, K. P. Subbalakshmi, Zining Zhu, Jordan W. Suchow, and Yangyang Yu. 2026. **FinTrace: Holistic trajectory-level evaluation of LLM tool calling for long-horizon financial tasks**. *Preprint*, arXiv:2604.10015.
- Kranti Chalamalasetti and Sowmya Vajjala. 2026. **LLM judges can be too generous when there is no reference answer**. *Preprint*, arXiv:2607.12885.
- Jiaxuan Dai and Tianyi Huang. 2026. **TwinCheck: Evidence-grounded negative-twin verification for stateful tool agents**. *Preprint*, arXiv:2609.26911.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. **Chain-of-verification reduces hallucination in large language models**. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.
- Linfeng Fan, Ziwei Li, Yuan Tian, Yichen Wang, Rongsheng Li, and Xiong Wang. 2026a. **The granularity mismatch in agent security: Argument-level provance solves enforcement and isolates the LLM reasoning bottleneck**. *Preprint*, arXiv:2605.11039.
- Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. 2026b. **AgentProcessBench: Diagnosing step-level process quality in tool-using agents**. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD 2026)*, Jeju Island, Republic of Korea. ACM.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. **The Llama 3 herd of models**. *Preprint*, arXiv:2407.21783.
- Anisha Gunjal and Greg Durrett. 2024. **Molecular facts: Desiderata for decontextualization in LLM fact verification**. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.
- Qisheng Hu, Quanyu Long, and Wenya Wang. 2025. **Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance?** In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6313–6336, Albuquerque, New Mexico. Association for Computational Linguistics.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. **Mistral 7B**. *Preprint*, arXiv:2310.06825.
- Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2025. **Sufficient context: A new lens on retrieval augmented generation systems**. In *International Conference on Learning Representations (ICLR)*, pages 20310–20334.
- Kiran Kate, Tejaswini Pedapati, Kinjal Basu, Yara Rizk, Vijil Chenthamarakshan, Subhajit Chaudhury, Mayank Agarwal, and Ibrahim Abdelaziz. 2025. **LongFuncEval: Measuring the effectiveness of long context models for function calling**. *Preprint*, arXiv:2505.10570.
- Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. **No free labels: Limitations of LLM-as-a-judge without human grounding**. *Preprint*, arXiv:2503.05061.
- Jacky Kwok, Hangoo Kang, Tarun Suresh, Jon Saad-Falcon, Marco Pavone, Christopher R  , and Azalia Mirhoseini. 2026. **Contrastive language models: A system one model for fast and generalizable decision-making**. Notion blog. Blog: <https://contrastive-lm.notion.site>; code: <https://github.com/Contrastive-LM/CLM>; weights: <https://huggingface.co/Contrastive-LM/CLM-v0.1-8B>.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. **Efficient memory management for large language model serving with PagedAttention**. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP ’23*, pages 611–626, New York, NY, USA. ACM.
- Dongryeol Lee, Yerin Hwang, Taegwan Kang, Minwoo Lee, Younhyung Chae, and Kyomin Jung. 2026. **Judging against the reference: Uncovering knowledge-driven failures in LLM-judges on QA evaluation**. *Preprint*, arXiv:2601.07506.
- Dawei Li, Yuguang Yao, Zhen Tan, Huan Liu, and Ruocheng Guo. 2026a. **ToolPRMBench: Evaluating and advancing process reward models for tool-using agents**. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 12378–12391, San Diego, California, United States. Association for Computational Linguistics.

- Renhao Li, Jianhong Tu, Yang Su, Yantao Liu, Fei Huang, Hamid Alinejad-Rokny, Derek F. Wong, Junyang Lin, and Min Yang. 2026b. **ToolRM: Towards agentic tool-use reward modeling**. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 8613–8640, San Diego, California, United States. Association for Computational Linguistics.
- Yubo Li, Yidi Miao, Ramayya Krishnan, and Rema Padman. 2026c. **JEV-as-a-judge: Accept when confident, escalate when unsure**. *Preprint*, arXiv:2609.26550.
- Zeyan Li, Jing Peng, and Jianfeng Xu. 2026d. **Backbone-adaptive evidence routing for robust pairwise LLM judging**. *Preprint*, arXiv:2609.30751.
- Zhuo Li, Yupeng Zhang, Pengyu Cheng, Jiajun Song, Mengyu Zhou, Hao Li, Shujie Hu, Yu Qin, Erchao Zhao, Xiaoxi Jiang, and Guanjuan Jiang. 2026e. **MARCH: Multi-agent reinforced check for hallucination**. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 39389–39415, San Diego, California, United States. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. **Let’s verify step by step**. In *International Conference on Learning Representations (ICLR)*, pages 39578–39601.
- Jianghao Lin, Yuanyuan Shi, Xin Peng, Renjie Ding, Hairui Wang, Yuxuan Peng, Bizhe Bai, Weixi Song, Fengshuo Bai, Huacan Chai, Weinan Zhang, Fei Huang, and Ying Wen. 2026. **ToolPRM: Fine-grained inference scaling of structured outputs for function calling**. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18792–18804, San Diego, California, United States. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xixi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. **FactScore: Fine-grained atomic evaluation of factual precision in long form text generation**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. **NoLiMa: Long-context evaluation beyond literal matching**. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44554–44570. PMLR.
- OpenAI. 2026. **Tool output trimmer**. OpenAI Agents SDK documentation. [https://openai.github.io/openai-agents-python/ref/extensions/tool\\_output\\_trimmer/](https://openai.github.io/openai-agents-python/ref/extensions/tool_output_trimmer/). Accessed 2026-10-07.
- Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. **The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models**. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48371–48392. PMLR.
- Qwen Team. 2026. **Qwen3.8-27B**. Hugging Face model card. <https://huggingface.co/Qwen/Qwen3.8-27B>. Accessed 2026-10-04.
- Mohit Raghavendra, Anisha Gunjal, Bing Liu, and Yunzhong He. 2026. **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)*, pages 15265–15290, San Diego, California, United States. Association for Computational Linguistics.
- Prithvi Rajasekaran, Ethan Dixon, Carly Ryan, and Jeremy Hadfield. 2025. **Effective context engineering for AI agents**. Anthropic engineering blog. <https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>. Accessed 2026-10-07.
- Sanjana Ramprasad and Byron C. Wallace. 2025. **Do automatic factuality metrics measure factuality? A critical evaluation**. In *Advances in Neural Information Processing Systems*, pages 145976–146001.
- Rohan Sequeira, Stavros Damianakis, Umar Iqbal, and Konstantinos Psounis. 2026. **Agent-sentry: Bounding LLM agents via execution provenance**. *Preprint*, arXiv:2603.22868.
- Yining She, Yiliang Liang, and Eunsuk Kang. 2026. **Safeguarding LLM agents from misalignment through provenance analysis**. In *Proceedings of the 41st IEEE/ACM International Conference on Automated Software Engineering (ASE ’26)*, Munich, Germany. ACM.
- Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. **Evaluating the factual consistency of large language models through news summarization**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada. Association for Computational Linguistics.
- Abhigya Verma, Amit Kumar Saha, Segaranasan Subramanian, and Sai Harshitha Aluru. 2026. **Agent-JudgeBench: A multi-difficulty benchmark for evaluating LLM judges on agentic tool-calling**. *Preprint*, arXiv:2608.26623.
- Miriam Wanner, Benjamin Van Durme, and Mark Dredze. 2025. **DnDScore: Decontextualization and decomposition for factuality verification in long-form**

- [text generation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23609–23626, Suzhou, China. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. [Evaluating large language models at evaluating instruction following](#). In *International Conference on Learning Representations (ICLR)*, pages 40193–40219.
- Qiyuan Zhang, Yufei Wang, Tiezheng Yu, Yuxin Jiang, Chuhan Wu, Liangyou Li, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025a. [RevisEval: Improving LLM-as-a-judge via response-adapted references](#). In *International Conference on Learning Representations (ICLR)*, pages 99866–99893.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. [The lessons of developing process reward models in mathematical reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, Vienna, Austria. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-bench and Chatbot Arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Datasets and Benchmarks Track.
- Lucen Zhong, Zhengxiao Du, Xiaohan Zhang, Haiyi Hu, and Jie Tang. 2025. [ComplexFuncBench: Exploring multi-step and constrained function calling under long-context scenario](#). *Preprint*, arXiv:2501.10132.
- Chenyu Zhou. 2026. [More convincing, not more correct: Self-play reward hacking of reference-free LLM judges](#). *Preprint*, arXiv:2607.05904.

## A Data and Judge Inputs

**Choice of benchmarks.** We need three things: multi-step trajectories with real tool returns, so that sources can be cut, a gold value for each argument, and wrong calls made by real models. TR-Bench\_BFCL uses the same BFCL tasks as FCRB in its single-turn part and compares whole turns of calls in its multi-turn part (Li et al., 2026b). The data of ProvenanceGuard label calls for safety or judge tasks by their final outcome, so they give no gold value per argument (She et al., 2026).

**FCRB.** From the 1,500 single-turn requests of FC-RewardBench we drop the 15 with unparseable wrong outputs. We keep the 650 whose wrong call fails the value check of BFCL (error type `value_error`). We then remove 20 whose only recorded message is a system prompt. We check each argument of the wrong call on its own with the BFCL checker. It is labeled wrong when the checker rejects its value and carries no label otherwise. This leaves 630 requests, each with one correct and one wrong call, and 1,719 correct and 769 wrong argument values. FCRB has no earlier calls, so the judge sees the request only.

**CFB.** We use the released trajectories of 30 models on the 1,000 CFB tasks (revision `fcc906da`). For each trajectory we find the first step where the model call differs from the gold call. We keep the divergences in which the model calls the gold function with different argument values, since only these give an argument to label; a call to another function or a missing call is left out. The gold history up to that step is the state. The gold call is the correct call and the model call is the wrong call. Models that make the same wrong call at a state are counted once, so a state can carry several wrong calls. Parallel calls are matched to gold calls by position. We drop 30 wrong argument values whose parallel calls to the same function could not be matched in order. This leaves 347 correct and 644 wrong calls, with 1,639 correct and 999 wrong argument values.

**FinTrace.** FinTrace gives 800 test queries with golden trajectories (Cao et al., 2026). It labels whole trajectories but no wrong calls at the argument level. We make wrong calls from golden calls by rule, with a fixed seed (20260930). Each wrong call keeps the endpoint and changes one argument. A ticker becomes a ticker used in another query, a date or year moves by one to three years (a from

and to pair moves together), or the reporting period switches between annual and quarter. We keep calls whose function descriptions come from the documentation of the data provider, and we drop 10 calls whose request has an unfilled template. This leaves 2,334 correct and 2,334 wrong calls, with 7,428 correct and 2,796 wrong argument values.

**Label checks and sensitivity analysis.** Two annotators independently checked 100 random arguments labeled wrong from each of CFB and FinTrace, without seeing any judge score or derived reference. They agreed on 94% of the CFB items and 88% of the FinTrace items, and settled the rest by discussion. Of 100 CFB arguments, 97 were confirmed wrong, 2 could also be correct, and 1 was unclear. Of 100 FinTrace arguments, 96 were confirmed wrong, 2 could also be correct, and 2 were unclear. We checked CFB and FinTrace because their argument labels come from our construction procedures. FCRB labels follow the official BFCL checker and were kept unchanged. To bound the effect of these labels, we run a worst-case deletion (Table 5). For each judge and dataset, we remove the share of wrong values that the audit could not confirm, 3% on CFB and 4% on FinTrace, choosing the values whose removal lowers the gain of DEREF over Plain the most. This targets the way such noise favors DEREF: a reference that names the gold value leads the judge to reject a value that is labeled wrong but also correct, which counts as a correct rejection. On both datasets, every judge keeps a significant gain, between 11.76 and 20.19 points on CFB and between 0.89 and 6.42 points on FinTrace. We do not apply this test to the comparison with RevisEval, since its revised call also names the gold value, so the two methods share this effect and a deletion chosen against one of them would not model label noise. The derivation gap and  $\tau$  use only correct arguments, so these negative-label ambiguities do not affect Table 1.

**Judge input.** Every judge reads the same text (Figure 4). The state gives the request and the most recent earlier calls with their returns. CFB shows the last 3 steps and FinTrace the last 5 calls of earlier turns. Each return is cut to its first 1,000 characters. If the longest CLM input of a state is still above 2,048 tokens, we drop the returns of the earliest shown step, then whole early steps, and last the middle of the request, keeping at least 40 tokens of it. A state that is still too long is left out. Correct and wrong calls at a state share the state

Question: Generate a digital image with captured 'a man wearing a red dress'.  
Previous calls (most recent 5):  
none  
Function: generate\_human\_image -- Generates a digital image of a human subject based on the provided text prompt. This function is tailored to create images representing different human demographics such as girls, boys, women, men, and children.  
Parameter: image\_quality -- The desired quality level of the generated image.;  
type: string; enum: ["low", "medium", "high"]; default: "high"  
Proposed value for image\_quality: "high"

Figure 4: A Plain judge input from FCRB, a correct value on the DEFAULT path. Long lines are wrapped here for space. The judge question is “Is this the correct value for this parameter, given the question?” DEREf inserts its block after the state, before the Function line.

| Judge        | CFB   |                         | FinTrace |                      |
|--------------|-------|-------------------------|----------|----------------------|
|              | Full  | Worst case              | Full     | Worst case           |
| CLM          | 22.14 | 20.19<br>[18.13, 22.42] | 5.74     | 1.87<br>[0.64, 2.90] |
| Jev          | 13.72 | 11.76<br>[9.79, 13.71]  | 3.99     | 0.89<br>[0.23, 1.30] |
| Qwen3-8B     | 16.34 | 14.05<br>[12.82, 16.28] | 7.13     | 3.37<br>[2.46, 4.42] |
| Qwen3.8-27B  | 16.17 | 13.84<br>[11.72, 16.01] | 2.94     | 1.15<br>[0.42, 2.13] |
| Llama-3.1-8B | 16.98 | 15.37<br>[13.80, 16.47] | 8.78     | 5.22<br>[4.29, 6.45] |
| Mistral-7B   | 20.84 | 19.70<br>[17.49, 21.99] | 9.16     | 6.42<br>[5.40, 7.37] |

Table 5: Worst-case deletion of unconfirmed wrong labels. Full is the gain of DEREf over Plain in argument AUROC (points) on all wrong values. Worst case removes, for each judge, the 3% of CFB wrong values and 4% of FinTrace wrong values whose removal lowers that gain the most, and reports the gain that remains, with its 95% CI below. FCRB labels follow the BFCL checker and are not audited.

text. After the state come the function line, the parameter line, and the proposed value. On FCRB the parameter line adds the type, the enum, and the default of the schema. On CFB it adds the enum, and on FinTrace it adds nothing. The question to the judge is “Is this the correct value for this parameter, given the question?” CLM is CLM-v0.1-8B. We call Jev through OpenRouter with the version id typesafe/jev-1.13-20260917, and we check that every reply returns this id. When we ask Jev again on 500 random inputs, 497 answers fall on the same side of 0.5 as before, and scores differ by at most 0.17.

**Reference block and extra truncation.** The reference goes after the state under the header “Expected value (written before seeing the proposed value):”. If the state with this block exceeds 2,048 CLM tokens, the state takes further steps of the same ladder from its own level. This removes more

early history for 131 of the 2,638 CFB arguments (39 states) and 786 of the 10,224 FinTrace arguments (194 states). All four judges read the same DEREf input.

**Full history.** The full-history input shows every earlier call and return, with no window and no cut. On FinTrace, 248 of the 10,224 arguments have an input longer than 32,768 tokens for a Qwen judge. For these we drop the oldest returns, and then the oldest steps if needed, until the input fits. No CFB input needs this. DEREf + full adds the reference to this input and uses the same rule. Jev accepts less text (Table 11). For 104 FinTrace arguments with full history and 102 with DEREf + full, its input drops one to three more of the oldest returns by the same rule. Every other Jev input equals the Qwen input. Without these arguments, no difference that involves the full history changes sign, and AUROC differences move by at most 0.46 points.

**FCRB-Relocate.** This set mirrors how history-only values arise in CFB: for 263 of its 279 correct history-only values (94.3%), the source return is in the judge input but cut before the value. We start from FCRB correct values that appear in the request. We move one such value into the return of an added earlier call, `find_saved_items`, a name that no FCRB function uses. The request is rewritten to point to the right entry by an attribute that only this entry has. The return lists eight entries. One holds the correct value. Seven distractors hold values of the same kind from other FCRB samples, taken first from the same parameter name, then from the same function, then from any parameter. In the visible version the correct entry comes second. In the buried version it comes last. When the correct entry would still start before character 1,000, every entry in both versions gets the same filler note, repeated until the correct entry of the buried version starts after that point, so the two

versions hold the same eight entries in a different order. In the buried returns, the correct value starts between characters 1,009 and 1,201. The first distractor (H) is in view in both versions and starts by character 33. The wrong values are H for every state and, where one exists, a value that appears in no text of the state (C). This gives 257 correct values and 428 wrong values (257 H and 171 C) per version. We drop states in which a distractor equals the correct value after lower-casing and removing all characters other than letters and digits. Code asserts every placement above. On the buried version, AUROC of the correct value against H alone is 34.21 for CLM, 28.23 for Jev, 26.71 for Qwen3-8B, and 39.88 for Qwen3.8-27B. With the full history, the buried version reaches 99.81 [99.65, 99.93] for Jev, 98.39 [97.54, 99.12] for Qwen3-8B, and 99.86 [99.65, 99.99] for Qwen3.8-27B, as high as DEREf (Table 6). So this set shows that judges fail when the source is out of view, not that DEREf beats the full history. In a check of 19 states, the rewritten request and the full return fix the correct value in all 19, and H is wrong in all 19. Two of the 19 rewrites have wording problems that do not change the label.

**FinTrace-HC.** We start from the 1,099 FinTrace wrong calls whose original values all appear in the request. For the changed argument, H takes another value that the same key holds in an earlier call or return of the trajectory. H must have the same JSON type as the correct value, must differ from it, and must not appear in the request. Synonyms of the reporting period do not count as different. H keeps the rest of the call and the state. C is the rule-made wrong call of the same state, kept when all its changed values are on the NO-SOURCE path. We drop H values that the golden trajectory uses in a later call (35 of 172), and tickers of the same company as the correct ticker (15). This leaves 122 H values, 72 history-only and 50 visible. Table 6 compares the 122 correct values with them. The paired H minus C test of Section 3.4 uses the 106 of these states that also have a C (51 queries). In a check of 20 samples, all 20 changed values are wrong at their step.

**Revealing the source.** The test uses FinTrace correct history-only values whose first occurrence was lost from the window. We exclude booleans, empty strings, and numbers of at most two characters. We also exclude other numbers and strings of at most three characters, which can match by

| Judge                                             | Plain | DEREF | $\Delta$                 |
|---------------------------------------------------|-------|-------|--------------------------|
| <i>Relocate, buried</i> (257 correct, 428 wrong)  |       |       |                          |
| CLM                                               | 43.73 | 98.40 | +54.67<br>[51.22, 58.04] |
| Jev                                               | 45.00 | 98.70 | +53.70<br>[49.74, 57.47] |
| Qwen3-8B                                          | 39.80 | 98.87 | +59.07<br>[55.35, 62.76] |
| Qwen3.8-27B                                       | 49.03 | 99.59 | +50.56<br>[46.98, 53.92] |
| <i>Relocate, visible</i> (257 correct, 428 wrong) |       |       |                          |
| CLM                                               | 91.89 | 99.42 | +7.52<br>[5.64, 9.53]    |
| Jev                                               | 99.88 | 99.99 | +0.11<br>[0.03, 0.21]    |
| Qwen3-8B                                          | 98.82 | 99.68 | +0.87<br>[0.17, 1.64]    |
| Qwen3.8-27B                                       | 99.58 | 99.95 | +0.37<br>[0.06, 0.81]    |
| <i>FinTrace-HC</i> (122 correct, 122 wrong)       |       |       |                          |
| CLM                                               | 91.98 | 96.45 | +4.46<br>[1.49, 7.76]    |
| Jev                                               | 98.90 | 99.32 | +0.42<br>[-0.38, 1.25]   |
| Qwen3-8B                                          | 95.99 | 97.95 | +1.97<br>[0.15, 4.28]    |
| Qwen3.8-27B                                       | 98.56 | 99.26 | +0.70<br>[-0.04, 1.55]   |

Table 6: Argument AUROC (in points) on the two controlled sets.  $\Delta$  is DEREf minus Plain, with its 95% CI below. No difference is significantly negative.

chance. We keep the 167 values whose source step lies outside the window (130 states, 93 queries). The reveal version inserts that step at its place in time. A long return shows 1,000 characters centered on the value. The placebo inserts another step of the same kind from outside the window of the same trajectory. It does not contain the value and has the closest displayed length. Both versions use the same header. The revealed step is a return for 80 values, an earlier call of the agent for 82, and assistant text for 5. Against the placebo, the source raises the mean score of correct values by between 6.30 and 14.29 points for all four judges, while wrong values in the same states move by at most 3.49 points. On revealed returns alone, the source raises correct scores significantly for all four judges.

**Schema facts.** For Figure 2b we add one line after the proposed value on every FCRB argument whose schema has a default (1,037 items). The line restates the default. It then quotes up to 40 characters around each check word found in the request, or lists the check words when none is found. The check words are the parameter name, the name with spaces for underscores, the default value, and

the enum values.

## B Derivation Path Census

We assign paths in this order, and the first match wins.

**VISIBLE:** the value is found in the content of the Plain state, which is the request, the shown calls, and the shown returns as cut. Template text, the function and parameter lines, and the proposed value do not count. **HISTORY-ONLY:** the value is found in the full history, which is the request and every earlier call and return, uncut. **FORMAT:** the value matches the request or a piece of the full history after one change. The change can be lower-casing and collapsing whitespace, or reading both sides as the same calendar date, the same number, or the same boolean. **DEFAULT:** the value equals the schema default, or it is a “don’t care” value that the parameter description names. **NOSOURCE:** none of the above. A value is found when its text is a case-sensitive substring of the search text. A number must not touch a letter, a digit, or a decimal point, so 10 does not match 2010 or 10.5. A string shorter than four characters must not touch a letter or a digit. A list is found when every element is found. The path depends only on the value under test, so wrong values have paths too. Only FCRB schemas have defaults, so the DEFAULT path occurs only on FCRB.

## C Metrics and Statistics

**Threshold and ties.** For each judge, dataset, and input,  $\tau$  is the score at which the judge accepts 80% of correct arguments. Scores above  $\tau$  count as accepted. Scores equal to  $\tau$  count with a fraction  $q$  chosen so that exactly 80% are accepted. This matters for judges whose scores pile up at one value. The choice of 80% does not drive Table 1. Table 7 repeats its six columns with  $\tau$  set to accept 70% and 90% of correct arguments, for the five judges run at all three rates. On CFB and FinTrace, all 20 cells keep a CI above zero at every rate. On FCRB, where the gap is small, three of the 30 cells lose significance at 70% or 90%.

**Bootstrap.** We resample queries 1,000 times with seed 20260930. For argument AUROC we draw queries with replacement. For call AUROC and  $\text{Gap}_\tau$  we give the queries multinomial weights, which has the same distribution but a different random sequence. For  $\text{Gap}_\tau$ ,  $\tau$  and  $q$  are set again in each bootstrap sample. A 95% CI runs from the

2.5th to the 97.5th percentile over these samples. A paired difference computes both sides on the same samples.

## D Reading the Reference

Some analyses need the value that a reference states. They are the ablation that keeps only the expected value, the swap test, and the shares of references that name the correct value (Section 5.3). A rule-based parser reads this value. It first removes Markdown emphasis. It then matches a fixed set of sentence patterns, such as “should be” followed by a value, and takes the value as a substring of the reference. If no pattern matches, the reference counts as not fixing the value. It also counts as not fixing the value when the same sentence gives a second candidate value. This last rule changes 1, 22, and 51 argument slots on FCRB, CFB, and FinTrace. Parallel calls to one function at a step share a slot, and the reference then lists the values it expects, so these slots fall under this rule.

**Checks.** We check the parser on samples drawn with seed 20260930, and no argument is drawn twice. For each benchmark, a check reads 20 correct and 10 wrong arguments for which the parser finds a value, and asks whether that value is the conclusion of the reference. We fixed the parser after each failed round. In the fourth round, it took the right value for 30 of 30 arguments on every benchmark, so we accepted it. The parser can also miss a stated value. So the first two rounds also read 20 correct arguments per benchmark for which the parser found no value although the reference text contains the correct value. In the second round, the reference does state a value for some of them, so the parser misses a small share of stated values. The last two rounds changed only one fallback pattern and added the rule for a second candidate value. A miss counts a stated value as not fixed, so the shares of references that name the correct value can only be too low.

**Parsed expected value only.** This ablation replaces the text of each reference block by one sentence. When the parser finds a value, the sentence is “Expected value for {name}: {value}.”, with the value copied from the reference. Otherwise it is “The context does not determine the value of {name}.” Everything else is the same as in Deref, including the header of the block and its place in the input. Table 8 splits the result by whether the

| Dataset              | Judge       | Out of view             |                         |                         | Main path               |                         |                         |
|----------------------|-------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|                      |             | 70%                     | 80%                     | 90%                     | 70%                     | 80%                     | 90%                     |
| FCRB (Default)       | CLM         | 5.85<br>[1.30, 10.89]   | 8.15<br>[3.92, 12.07]   | 6.20<br>[3.58, 9.51]    | 32.60<br>[24.82, 41.14] | 31.93<br>[24.36, 41.20] | 26.29<br>[19.87, 34.75] |
|                      | Jev         | 11.82<br>[7.35, 15.90]  | 8.11<br>[4.44, 11.74]   | 2.22<br>[−0.75, 4.77]   | 32.99<br>[24.41, 41.88] | 20.26<br>[12.68, 27.32] | 9.41<br>[3.16, 14.70]   |
|                      | Qwen3-4B    | 3.77<br>[−0.43, 7.78]   | 4.37<br>[0.37, 8.43]    | 3.72<br>[0.75, 6.50]    | 25.82<br>[18.43, 34.44] | 21.26<br>[13.91, 29.63] | 15.64<br>[8.63, 22.33]  |
|                      | Qwen3-8B    | 1.15<br>[−2.98, 4.88]   | 0.63<br>[−3.78, 4.35]   | −1.00<br>[−3.85, 1.77]  | 11.97<br>[3.59, 19.85]  | 14.40<br>[4.69, 22.14]  | 3.52<br>[−1.60, 9.31]   |
|                      | Qwen3.8-27B | 16.93<br>[11.92, 21.19] | 12.29<br>[8.19, 15.81]  | 6.07<br>[3.26, 8.78]    | 49.55<br>[40.32, 56.73] | 36.03<br>[27.56, 44.48] | 16.88<br>[10.16, 23.69] |
|                      |             |                         |                         |                         |                         |                         |                         |
| CFB (History only)   | CLM         | 41.68<br>[38.43, 45.44] | 30.76<br>[27.95, 34.02] | 17.15<br>[15.33, 18.97] | 77.22<br>[70.77, 82.83] | 66.63<br>[59.42, 73.00] | 38.19<br>[32.88, 44.93] |
|                      | Jev         | 28.50<br>[22.35, 33.74] | 22.67<br>[19.08, 27.71] | 16.56<br>[13.77, 18.55] | 74.59<br>[66.98, 79.65] | 59.39<br>[54.22, 68.00] | 42.09<br>[35.21, 49.18] |
|                      | Qwen3-4B    | 47.34<br>[44.29, 50.65] | 34.71<br>[32.87, 36.78] | 18.86<br>[17.73, 20.08] | 86.84<br>[82.10, 91.14] | 76.81<br>[71.30, 81.93] | 50.29<br>[44.37, 57.15] |
|                      | Qwen3-8B    | 43.72<br>[40.11, 47.33] | 33.30<br>[31.13, 35.65] | 18.96<br>[17.95, 20.33] | 83.63<br>[78.15, 88.16] | 74.87<br>[68.09, 80.47] | 46.56<br>[40.84, 53.62] |
|                      | Qwen3.8-27B | 34.79<br>[28.64, 40.36] | 25.99<br>[21.10, 30.72] | 14.47<br>[11.85, 17.66] | 76.27<br>[69.90, 82.04] | 57.08<br>[50.28, 63.87] | 28.22<br>[23.28, 36.22] |
|                      |             |                         |                         |                         |                         |                         |                         |
| FinTrace (No source) | CLM         | 32.99<br>[30.19, 36.14] | 25.95<br>[23.46, 28.22] | 15.19<br>[13.79, 16.78] | 34.67<br>[31.45, 38.15] | 28.28<br>[25.34, 30.81] | 17.09<br>[15.48, 18.93] |
|                      | Jev         | 36.44<br>[33.57, 39.11] | 27.78<br>[25.20, 29.87] | 15.79<br>[14.20, 17.37] | 36.06<br>[32.88, 39.00] | 27.47<br>[24.54, 30.25] | 16.53<br>[14.56, 18.45] |
|                      | Qwen3-4B    | 30.54<br>[27.77, 33.99] | 23.30<br>[20.91, 25.94] | 13.52<br>[11.86, 15.30] | 30.44<br>[27.39, 34.24] | 23.29<br>[20.73, 26.35] | 13.79<br>[11.80, 15.84] |
|                      | Qwen3-8B    | 31.48<br>[28.91, 34.47] | 24.04<br>[22.13, 26.22] | 14.42<br>[12.97, 15.81] | 32.93<br>[30.18, 36.21] | 25.72<br>[23.59, 28.31] | 15.80<br>[14.02, 17.53] |
|                      | Qwen3.8-27B | 40.02<br>[37.07, 42.60] | 29.76<br>[27.44, 32.10] | 16.47<br>[14.98, 17.83] | 42.28<br>[39.03, 45.30] | 31.68<br>[28.98, 34.22] | 17.65<br>[15.97, 19.38] |
|                      |             |                         |                         |                         |                         |                         |                         |

Table 7: Derivation gap  $\text{Gap}_\tau$  (in points) when  $\tau$  is set to accept 70%, 80%, or 90% of correct arguments, with Plain input. The 80% columns repeat Table 1. Out of view pools the four paths other than VISIBLE; the main path of each dataset is in parentheses. Llama-3.1-8B and Mistral-7B were run at 80% only, as additional rows of Table 1, so the check covers the five judges run at all three rates. 95% CIs come from the same bootstrap, with  $\tau$  and  $q$  set again in every resample.

|          | Value     | Correct / wrong | DEREF | EV only | $\Delta$                 |
|----------|-----------|-----------------|-------|---------|--------------------------|
| FCRB     | Stated    | 1,577 / 620     | 94.69 | 94.93   | −0.24<br>[−0.89, 0.40]   |
|          | Not found | 142 / 149       | 72.37 | 73.09   | −0.72<br>[−4.24, 3.13]   |
| CFB      | Stated    | 1,290 / 523     | 97.78 | 96.39   | +1.39<br>[0.85, 2.05]    |
|          | Not found | 348 / 470       | 82.78 | 60.41   | +22.37<br>[16.99, 27.46] |
| FinTrace | Stated    | 3,384 / 1,763   | 95.99 | 97.11   | −1.13<br>[−1.46, −0.75]  |
|          | Not found | 4,044 / 1,033   | 82.50 | 78.83   | +3.67<br>[2.26, 5.33]    |

Table 8: Keeping only the parsed expected value (EV only), split by whether the parser finds a value stated in the reference of the argument slot. Argument AUROC (in points) with Qwen3.8-27B.  $\Delta$  is DEREF minus EV only, with its 95% CI below. On CFB, the seven arguments of the one deriver prompt without an answer are left out.

parser finds a value for the argument slot. Misses of the parser fall only into the second group, so the first group does not depend on them. Where a value is stated, the rest of the text adds 1.39 points on CFB, makes no significant difference on FCRB,

and costs 1.13 points on FinTrace. Where no value is found, it adds 22.37 points on CFB and 3.67 points on FinTrace. There the reference text still contains the correct value for 77.01% and 45.03% of the correct arguments.

| Judge       | FCRB  |       |                          | CFB   |       |                          | FinTrace |       |                       |
|-------------|-------|-------|--------------------------|-------|-------|--------------------------|----------|-------|-----------------------|
|             | Plain | DEREF | $\Delta$                 | Plain | DEREF | $\Delta$                 | Plain    | DEREF | $\Delta$              |
| CLM         | 66.19 | 85.84 | +19.66<br>[17.10, 22.27] | 52.56 | 73.47 | +20.91<br>[17.38, 24.76] | 80.34    | 81.55 | +1.21<br>[0.00, 2.37] |
| Jev         | 83.46 | 88.12 | +4.67<br>[3.15, 6.11]    | 56.56 | 78.67 | +22.11<br>[18.18, 25.96] | 87.50    | 90.06 | +2.56<br>[1.88, 3.25] |
| Qwen3-8B    | 73.98 | 86.07 | +12.09<br>[9.80, 14.19]  | 55.47 | 67.46 | +11.99<br>[8.62, 15.51]  | 82.19    | 87.31 | +5.12<br>[4.08, 6.21] |
| Qwen3.8-27B | 85.43 | 88.72 | +3.29<br>[1.74, 4.83]    | 68.78 | 85.98 | +17.20<br>[13.84, 20.55] | 86.69    | 88.08 | +1.39<br>[0.63, 2.20] |

Table 9: Call AUROC (in points) with Plain input and with DEREF.  $\Delta$  is DEREF minus Plain, with its 95% CI below. Only the argument scores change with the input. For CLM on FinTrace the lower bound is 0.0005 points.

## E Call-Level Results

Guards accept or reject a whole call (She et al., 2026), so we also score each call from its argument scores. Process reward models combine step scores by a product or a minimum (Lightman et al., 2024; Zhang et al., 2025b), and the call score takes the minimum.

**Call score.** For a call with argument scores  $s_i$  and whole-call score  $w$ , the call score is

$$s_{\text{call}} = \min\left(w, \min_i s_i\right), \quad (2)$$

The whole-call score always comes from the Plain input. Only the argument scores change with the input. Most arguments of a wrong call are not the changed one and carry no label. All but 134 of them have the same judge input, word for word, as an argument of the correct call at the same state, and they take that argument’s score under every input. The other 134, all on FCRB, get their own score under every input.

**Decomposed verification.** Fact checkers often split a text into atomic claims and verify each one (Min et al., 2023). Atomic claims can lose the context needed to check them (Gunjal and Durrett, 2024; Wanner et al., 2025), and splitting helps weaker verifiers and can hurt stronger ones (Hu et al., 2025). Process reward models also combine step scores, and the best rule depends on how the step labels were made (Lightman et al., 2024; Zhang et al., 2025b). An argument judge checks a call in the same way, one argument at a time.

**The gap costs at the call level.** Taking the minimum assumes that the weakest argument of a call is a wrong one. A correct value out of view often breaks this assumption (Table 10). Compared with the whole-call score alone, the minimum raises call

AUROC on FCRB, the dataset with the smallest gap, for every judge by between 8.15 and 16.21 points. On CFB, which has the largest gap on a single path, it adds nothing for any judge (from  $-3.04$  to 1.22 points, none significant).

|                     | FCRB                    | CFB                     | FinTrace                |
|---------------------|-------------------------|-------------------------|-------------------------|
| Mixed correct calls | 325                     | 197                     | 1,107                   |
| Chance              | 50.45                   | 48.77                   | 48.73                   |
| CLM                 | 62.77<br>[7.33, 17.62]  | 93.40<br>[40.97, 48.01] | 75.07<br>[23.28, 29.28] |
| Jev                 | 62.72<br>[7.49, 16.93]  | 85.25<br>[32.34, 40.93] | 84.19<br>[32.85, 38.15] |
| Qwen3-4B            | 56.02<br>[2.02, 9.23]   | 91.53<br>[38.63, 46.33] | 83.59<br>[32.23, 37.36] |
| Qwen3-8B            | 48.96<br>[−5.02, 2.11]  | 90.36<br>[37.99, 45.27] | 82.11<br>[30.98, 35.66] |
| Qwen3.8-27B         | 68.31<br>[12.89, 22.59] | 82.49<br>[28.37, 39.00] | 85.68<br>[34.68, 39.40] |
| Llama-3.1-8B        | 63.33<br>[8.38, 17.17]  | 70.05<br>[14.98, 27.73] | 79.27<br>[27.52, 33.15] |
| Mistral-7B          | 59.89<br>[5.57, 13.40]  | 94.62<br>[42.31, 49.05] | 82.01<br>[30.73, 35.89] |

Table 10: Share (%) of correct calls whose lowest-scored argument is out of view, among correct calls with arguments both in and out of view, under Plain input. Chance is the share of out-of-view arguments in these calls. Below each share is the 95% CI of its difference from chance, in points. No share is significantly below chance.

**The lowest argument of a correct call.** Table 10 shows which argument takes the minimum in correct calls with arguments both in and out of view. For 20 of the 21 pairs of judge and benchmark, the lowest-scored argument is out of view more often than chance, and for none less often. The one exception is Qwen3-8B on FCRB, where about half of these calls tie for the lowest score. On CFB, DEREF lowers this share for all four judges, by between 10.28 and 24.37 percentage points, but it stays above chance.

**DEREF at the call level.** Table 9 gives call AUROC with Plain and with DEREf. DEREf raises call AUROC for all four judges on all three benchmarks, by between 1.21 and 22.11 points, and all 12 gains are significant. On CFB, where taking the minimum adds nothing under Plain, DEREf raises call AUROC by between 11.99 and 22.11 points.

## F Implementation Details and Cost

**Models and settings.** Table 11 lists the models and settings. We train no model and search no hyperparameters.

**Cost.** Table 12 gives the cost of DEREf. The deriver writes one reference per argument slot of a state, and all candidates for that slot share it. A slot is a pair of state and parameter name, so parallel calls to one function at a step share a slot, and a parameter that only the wrong call fills gets a slot with no correct value. One deriver prompt serves 1.33 candidates on FCRB, 1.83 on CFB, and 1.38 on FinTrace on average. The judge reads the reference inside its usual input. The longest DEREf input has 408 tokens on FCRB and 2,048 on CFB and FinTrace, as CLM counts them against its 2,048-token setting. A judge that reads the full history reads a long input for every candidate, with 3,317 tokens on average on CFB and 6,761 on FinTrace. ProvenanceGuard evidence also runs Qwen3.8-27B, but it needs one prompt per candidate value, 15,141 in all after removing prompts with identical text, against 10,741 deriver prompts.

## G Further Results

**A wrong value from the history.** FinTrace-HC pairs two wrong values for the same argument in the same state, one that appeared earlier in the history (H) and one that did not (C). On the 106 pairs in which both values exist, the mean score of H minus C is 7.79 [5.08, 10.39] for CLM, 5.06 [3.08, 7.36] for Jev, 3.79 [0.88, 7.53] for Qwen3-8B, and 0.72 [−0.08, 2.01] for Qwen3.8-27B, with query-clustered bootstrap CIs; the first three are significant.

**The reference on top of the full history.** On CFB, DEREf with the full history reaches 84.97 for Qwen3-8B and 96.01 for Qwen3.8-27B. It beats the full history alone by 18.04 and 10.56 points, and DEREf on the truncated input by 1.84 and 0.92 points. Its largest gain over the full history is on no-source values, 32.87 and 18.87 points. On

FinTrace, it reaches 91.94 and 92.52, and it beats the full history alone by 7.20 and 2.75 points. For Jev, it beats the full history alone by 9.35 points on CFB and 3.18 on FinTrace.

**A deriver that sees the value.** DEREf-v tests the choice of a deriver that does not see the candidate. Its deriver gets the DEREf prompt with the candidate value added and the opening words “Before seeing any proposed value:” removed. Everything else is the same, including the header of the reference block, which still says that the reference was written before seeing the proposed value. We keep this header so that the judge input changes only through the reference itself. Its references differ in length from those of DEREf, so 152 CFB inputs and 212 FinTrace inputs keep a different amount of early history within 2,048 tokens.

Averaged over the four judges, DEREf-v is above DEREf on CFB by 2.68 points, and below it on FCRB and FinTrace by 2.57 and 0.84 points (Table 16). It also needs one derivation per candidate value, 15,275 prompts after removing prompts with identical text, against 10,741, since candidates cannot share a reference that names one of them. DEREf-v names the wrong value under test as the expected value for 6.61% of wrong CFB arguments and 10.92% of wrong FCRB arguments, against 1.10% and 8.19% for DEREf. RevisEval keeps the wrong value under test in its revised call for 17.02% and 21.46%, and for 12.66% of wrong FinTrace arguments, against 1.00% for DEREf-v and 0.43% for DEREf. At 0.5, Qwen3.8-27B accepts 18.42% of wrong CFB values with DEREf-v, against 8.61% with DEREf, and 21.20% of wrong FCRB values, against 13.00%. Over the three benchmarks, DEREf-v writes the wrong value under test as the expected value for 178 wrong arguments. In a sample of 10 of them, 8 wrong values are close to the correct one, such as a long base64 string whose decoded text differs in one or two characters, a longitude with fewer digits, or a date one day off.

**Deriver seed.** We sample the deriver again with two more seeds and score CLM, Qwen3-8B, and Qwen3.8-27B with each new reference. DEREf is significantly above Plain for all three seeds, all three judges, and all three benchmarks, 27 cases in all. Against the original seed, none of the 18 differences of the two new seeds in DEREf AUROC is significant, and the largest is 0.80 points.

| Item               | Setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CLM                | CLM-v0.1-8B (Kwok et al., 2026), one checkpoint file (SHA-256 b2b4a8c9...). We serve it with vLLM 0.30.0 in bf16. The score is the probability it gives to the answer “true” against “false”. We run it with the released serving settings, which truncate inputs to 2,048 tokens.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Jev                | typesafe/jev-1.13-20260917 (Almeida, 2026), called through OpenRouter. In a length probe, it read inputs of up to 32,831 tokens by its own count and refused longer ones. A request holds only the state, the model id, and one question. We use the returned probability and set no sampling parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Qwen judges        | Qwen3-4B and 8B (Yang et al., 2025), and Qwen3.8-27B (Qwen Team, 2026), with 4.02B, 8.19B, and 27.78B parameters (26.90B in the text part of Qwen3.8-27B). We use one revision of each (1cfa9a7, b968826, and 1d4bf0f). They run in bf16 with transformers 5.18.0 and left padding. Thinking is off. The score is the softmax over the logits of “Yes” and “No” at the first output position. Where a judge runs with thinking on (Tables 4 and 15), it first generates its thinking, up to 8,192 tokens, after which we append the closing tag if it has not been written, and we then read the same two logits at the first position of the answer, without parsing the answer text. The two tokens hold less than half of the probability mass for at most 1.92% of inputs on any benchmark. Truncated inputs run in batches of 32. Full-history inputs run in batches sorted by length, at most 32 inputs and at most 65,536 padded tokens for Qwen3-8B and 32,768 for Qwen3.8-27B. |
| Other LLM judges   | Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), revisions 0e9e39f and c170c70. We read them as the Qwen judges, in bf16 with left padding, with the same system prompt and question. Neither tokenizer has a padding token, so we pad with the end-of-sequence token.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Generation         | Qwen3.8-27B with vLLM 0.30.0 in bf16, with thinking on at medium reasoning effort. Deriver prompts hold up to 32,768 tokens, and the oldest returns are dropped beyond that, which affects 190 of the 7,428 FinTrace deriver prompts. The deriver samples once per prompt with temperature 1.0, top-p 0.95, and top-k 20, the values recommended in the generation config of the model, and with seed 20260930. It writes at most 8,192 new tokens, and the reference is the text after the thinking. If no text follows the thinking, the judge gets the Plain input. This happens for one CFB prompt, which serves seven arguments, and no deriver prompt reaches the limit. ProvenanceGuard evidence and RevisEval use the same model and decoding settings. In the ablation, Qwen3-8B as deriver uses the values recommended for it, with temperature 0.6, and the deriver without thinking uses greedy decoding and at most 1,024 new tokens.                                      |
| Hardware, software | Every job runs on one NVIDIA B200 GPU. Models run with Python 3.11.16, PyTorch 2.13.0 with CUDA 13.0, vLLM 0.30.0 (Kwon et al., 2023), and transformers 5.18.0. Analysis uses Python 3.12.11, NumPy 2.2.6, pandas 2.3.2, and SciPy 1.16.1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Statistics         | AUROC uses our own rank code, which counts ties as one half. CIs come from the bootstrap in Appendix C.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Randomness         | Seed 20260930 fixes every random step: the sampling of the deriver and the prior methods, the FinTrace wrong calls, the two controlled sets, the swap arguments, the case studies, and the bootstrap. Qwen scores shift a little with batch composition. In our checks, one score moved by at most 0.185 and argument AUROC by at most 0.89 points. A rerun of CLM on all CFB arguments leaves 98.2% of scores unchanged and AUROC the same.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Compute            | The results in this paper took 44 jobs and 41.98 GPU hours, with model loading: 6.31 for generation, 27.66 for judges on truncated inputs, and 8.01 for judges on the full history. Runs of variants that we do not report took another 15.40, and pilot runs, smoke tests, exploration, and failed jobs took 12.75. Jev took 325,637 billed calls for \$23.01, and 47,514 more for variants that we do not report.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Table 11: Models and settings.

| Benchmark | Deriver (Qwen3.8-27B) |                     |                |               |       | Judge input (Qwen3-8B tokens) |               |                |
|-----------|-----------------------|---------------------|----------------|---------------|-------|-------------------------------|---------------|----------------|
|           | Prompts               | Candidates / prompt | Input tokens   | Output tokens | GPU s | Plain                         | DEREF         | Full           |
| FCRB      | 1,871                 | 1.33                | 195 (365)      | 217 (968)     | 12    | 165 (352)                     | 236 (444)     | 165 (352)      |
| CFB       | 1,442                 | 1.83                | 2,892 (29,327) | 513 (4,763)   | 138   | 841 (2,031)                   | 1,044 (2,084) | 3,317 (29,673) |
| FinTrace  | 7,428                 | 1.38                | 6,853 (32,651) | 557 (6,351)   | 1,684 | 1,093 (2,084)                 | 1,169 (2,084) | 6,761 (32,646) |

Table 12: Cost of DEREF. Token counts give the mean and, in parentheses, the maximum. Deriver tokens use the tokenizer and chat template of Qwen3.8-27B, and judge tokens those of Qwen3-8B, so they include the template text. Output tokens include the thinking. GPU seconds cover generation without model loading. The run also held other prompts, so we split its time by prompt tokens.

| Per input, batch size 1                                      | FCRB | CFB   | FinTrace |
|--------------------------------------------------------------|------|-------|----------|
| CLM, DEREf input (ms)                                        | 8.7  | 8.0   | 8.0      |
| Qwen3.8B, DEREf input (ms)                                   | 34.7 | 66.9  | 57.6     |
| Qwen3.8-27B, DEREf input (ms)                                | 52.9 | 120.4 | 93.8     |
| Qwen3.8-27B, direct judge with thinking, truncated input (s) | 2.00 | 7.46  | 4.52     |
| Qwen3.8-27B, direct judge with thinking, full history (s)    | –    | 11.20 | 5.53     |
| Deriver, one reference prompt (s)                            | 2.44 | 7.27  | 6.34     |

Table 13: Wall-clock time per input on one GPU in bf16, mean over 200 inputs drawn at random from each benchmark. The deriver row is the time to write one reference, which does not depend on the candidate and is shared by every candidate of that argument at that state.

**Judges outside the Qwen family.** Table 1 includes Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3, and every one of their 12 gaps there is significant. We also run DEREf with them, with the same references as for the other judges (Table 14). DEREf raises their argument AUROC on all three benchmarks, by between 8.78 and 20.84 points, and all six gains are significant.

**Deriver prompt.** When a step calls the same function more than once, the deriver names each value it expects, such as the two hotel IDs that a request for the two cheapest hotels asks for. The deriver gets the system prompt “You are a careful evaluator of tool calls made by an AI assistant.” Its user message gives the request, every earlier call and return uncut under the header “History (all earlier calls and their returns, oldest first):”, and the function and parameter lines of the call under judgment. It ends with this instruction: “Before seeing any proposed value: what should the value of parameter {name} be in this call, and where in the context does it come from? Quote the source text if there is one, or say that the context does not determine it. Answer in at most 3 sentences.” The message contains no value of the call under judgment and no other candidate at that step.

**Snippet rule.** For a candidate value, we search the full history for every place where it occurs and cut a snippet of 150 characters on each side of each occurrence; nearby windows are merged, a merged snippet is cut to 450 characters, and at most three snippets are kept, most recent first. The test of Section 3.4 adds these snippets to the judge input on CFB.

**ProvenanceGuard evidence.** ProvenanceGuard checks a proposed call in stages, and its provenance stage asks an LLM to find, for each derivable parameter value, the evidence in the user query or earlier tool outputs that supports it (She et al., 2026).

ProvenanceGuard then labels the call misaligned when no evidence is found. We keep the provenance stage and hand its evidence to the judge, so that the judge returns a score. Qwen3.8-27B sees the proposed value and quotes up to two passages that the value should come from. It runs with thinking on, and we take the quotes from the text after the thinking. A quote counts when it matches a piece of the full history, exactly or by a long common substring. On FinTrace and FCRB, the snippets around the quotes follow the snippet rule above. On CFB, it uses windows of 100 characters on each side without a length cap. An argument without a snippet keeps its Plain score.

**Revised calls.** RevisEval revises the proposed call from the full history, one revised call per proposed call, and the judge reads the revised call (Zhang et al., 2025a).

**The deriver as the judge.** A natural alternative is to let the deriver judge on its own. Table 15 lets Qwen3.8-27B judge directly, with thinking on and the same thinking budget as the deriver, over the truncated input and over the full history. On every benchmark it stays below the same model reading the DEREf reference with thinking off, by 3.60 points on FCRB, 10.92 on CFB, and 3.58 on FinTrace with the truncated input, and by 5.74 and 2.82 with the full history. Thinking does not close the gap either: with thinking on, the out-of-view gap of the direct judge is 12.96 on FCRB, 28.98 on CFB, and 23.17 on FinTrace. Table 16 gives the scores of the other three judges reading the reference, which we compare with this direct judge. Against the direct judge on the truncated input, Jev reading the reference is higher on all three benchmarks, by 3.5 to 4.6 points, CLM is within 1.4 points on all three, and Qwen3-8B is within 1.1 points on FCRB and CFB and 3.6 points higher on FinTrace. Against the direct judge on

| Judge        | FCRB  |       |                          | CFB   |       |                          | FinTrace |       |                        |
|--------------|-------|-------|--------------------------|-------|-------|--------------------------|----------|-------|------------------------|
|              | Plain | DEREF | $\Delta$                 | Plain | DEREF | $\Delta$                 | Plain    | DEREF | $\Delta$               |
| Llama-3.1-8B | 74.05 | 89.36 | +15.31<br>[13.48, 17.02] | 61.31 | 78.29 | +16.98<br>[14.02, 19.97] | 82.29    | 91.07 | +8.78<br>[7.63, 10.12] |
| Mistral-7B   | 73.92 | 88.90 | +14.97<br>[13.11, 17.12] | 66.70 | 87.55 | +20.84<br>[18.50, 23.54] | 81.39    | 90.55 | +9.16<br>[7.91, 10.40] |

Table 14: Argument AUROC (in points) of the two judges outside the Qwen family, with Plain input and with DEREF.  $\Delta$  is DEREF minus Plain, with its 95% CI below.

the full history, Jev is 1.5 points lower on CFB and 3.4 higher on FinTrace, CLM is 4.7 and 0.6 points lower, and Qwen3-8B is 7.1 points lower on CFB and 2.6 higher on FinTrace.

**Self-generated reference.** Following LLMBar (Zeng et al., 2024), the judge model first writes the expected value from its own truncated input, with thinking off, greedy decoding, and at most 384 new tokens. It then reads that reference as it would read DEREF.

## H Full Comparison Grid

Table 16 gives argument AUROC for each pair of judge and benchmark with every input, extending Table 2. Averaged over the four judges, DEREF minus Plain is 7.92 [6.72, 9.07] on FCRB, 17.09 [15.47, 18.85] on CFB, and 4.95 [4.00, 6.02] on FinTrace; minus ProvenanceGuard evidence it is 7.48 [6.28, 8.63], 12.31 [10.88, 13.81], and 4.28 [3.35, 5.27]; and minus RevisEval it is 2.38 [1.29, 3.48], 2.07 [0.71, 3.47], and 1.46 [0.67, 2.27]. Averaged over the three judges that read the full history, DEREF minus the full history is 11.58 [10.22, 12.97] on CFB and 4.35 [3.55, 5.22] on FinTrace. ProvenanceGuard evidence and RevisEval are the prior methods of Section 5.1. We run both with all four judges, and the full history with every judge but CLM. In Table 16, RevisEval has the higher AUROC of the two prior methods for every judge and benchmark, so  $\Delta$  vs. RevisEval is also the difference against the best prior method. On CFB, DEREF minus the full history is 8.91 [7.27, 10.62] for Jev, 16.20 [14.03, 18.30] for Qwen3-8B, and 9.64 [7.81, 11.38] for Qwen3.8-27B. The self-generated reference minus Plain is 3.01 [1.59, 4.52], 1.49 [−0.06, 3.20], and 2.80 [2.11, 3.57] for Qwen3-8B on FCRB, CFB, and FinTrace, and 0.57 [−0.05, 1.18], 2.92 [1.76, 4.11], and 0.74 [0.15, 1.30] for Qwen3.8-27B. DEREF minus the self-generated reference is 7.68 [5.92, 9.40], 14.85 [11.91, 17.62], and 4.33 [3.52, 5.14] for Qwen3-8B,

and 0.87 [−0.08, 1.84], 13.26 [10.99, 15.53], and 2.20 [1.24, 3.25] for Qwen3.8-27B. DEREF minus ProvenanceGuard evidence on CFB is 15.93 [14.15, 18.02] for CLM, 7.99 [6.16, 9.76] for Jev, 18.33 [16.15, 20.29] for Qwen3-8B, and 7.01 [5.09, 8.88] for Qwen3.8-27B. Every CI of DEREF against ProvenanceGuard evidence, on all three benchmarks, lies above zero. Table 17 gives the derivation gap on the main path of each benchmark, and Table 18 gives each kind of evidence on the two main paths of CFB, extending the ablation of Section 5.3.

| Qwen3.8-27B as                             | FCRB                    | CFB                       | FinTrace                |
|--------------------------------------------|-------------------------|---------------------------|-------------------------|
| Direct judge, thinking on, truncated input | 89.79                   | 84.17                     | 88.67                   |
| Direct judge, thinking on, full history    | –                       | 90.27                     | 89.70                   |
| Judge of the DEREf reference, thinking off | 93.39                   | 95.09                     | 92.25                   |
| with the full history                      | –                       | 96.01                     | 92.52                   |
| Direct – DEREf, truncated input            | –3.60<br>[–4.97, –2.35] | –10.92<br>[–13.12, –8.76] | –3.58<br>[–4.37, –2.83] |
| Direct – DEREf, full history               | –                       | –5.74<br>[–7.40, –4.26]   | –2.82<br>[–3.57, –2.09] |

Table 15: Argument AUROC (in points) of Qwen3.8-27B judging directly with thinking on, against the same model reading the DEREf reference with thinking off. FCRB has no cut, so the full history equals the truncated input there.  $\Delta$  rows give the paired difference with its 95% CI below.

| Judge and evidence                           | FCRB                     | CFB                      | FinTrace               |
|----------------------------------------------|--------------------------|--------------------------|------------------------|
| <i>CLM</i>                                   |                          |                          |                        |
| Plain                                        | 72.51                    | 63.41                    | 83.36                  |
| ProvenanceGuard evidence (She et al., 2026)  | 75.17                    | 69.62                    | 84.75                  |
| RevisEval (Zhang et al., 2025a)              | 83.26                    | 81.78                    | 86.07                  |
| DEREF (ours)                                 | 89.59                    | 85.55                    | 89.10                  |
| DEREF-v (ours)                               | 86.18                    | 88.93                    | 85.88                  |
| $\Delta$ vs. Plain                           | +17.08<br>[15.06, 19.22] | +22.14<br>[19.75, 24.74] | +5.74<br>[4.08, 7.54]  |
| $\Delta$ vs. RevisEval                       | +6.33<br>[4.65, 7.98]    | +3.77<br>[1.72, 5.90]    | +3.03<br>[1.42, 4.79]  |
| <i>Jev</i>                                   |                          |                          |                        |
| Plain                                        | 90.83                    | 75.01                    | 89.11                  |
| Full history                                 | –                        | 79.82                    | 90.08                  |
| ProvenanceGuard evidence                     | 91.00                    | 80.74                    | 89.57                  |
| RevisEval                                    | 92.23                    | 87.15                    | 91.77                  |
| DEREF (ours)                                 | 93.30                    | 88.73                    | 93.10                  |
| DEREF-v (ours)                               | 91.81                    | 92.23                    | 93.54                  |
| $\Delta$ vs. Plain                           | +2.47<br>[1.55, 3.39]    | +13.72<br>[11.56, 15.79] | +3.99<br>[3.23, 4.85]  |
| $\Delta$ vs. RevisEval                       | +1.07<br>[−0.05, 2.13]   | +1.57<br>[0.31, 2.98]    | +1.33<br>[0.77, 1.88]  |
| <i>Qwen3-8B</i>                              |                          |                          |                        |
| Plain                                        | 78.79                    | 66.79                    | 85.15                  |
| Full history                                 | –                        | 66.92                    | 84.74                  |
| Self-generated reference (Zeng et al., 2024) | 81.80                    | 68.28                    | 87.95                  |
| ProvenanceGuard evidence                     | 77.71                    | 64.80                    | 85.21                  |
| RevisEval                                    | 87.06                    | 81.75                    | 90.52                  |
| DEREF (ours)                                 | 89.48                    | 83.13                    | 92.28                  |
| DEREF-v (ours)                               | 85.99                    | 87.89                    | 91.72                  |
| $\Delta$ vs. Plain                           | +10.69<br>[8.85, 12.55]  | +16.34<br>[13.83, 18.69] | +7.13<br>[6.00, 8.29]  |
| $\Delta$ vs. RevisEval                       | +2.42<br>[0.89, 4.01]    | +1.38<br>[−0.64, 3.26]   | +1.75<br>[0.97, 2.53]  |
| <i>Qwen3.8-27B</i>                           |                          |                          |                        |
| Plain                                        | 91.95                    | 78.92                    | 89.31                  |
| Full history                                 | –                        | 85.45                    | 89.77                  |
| Self-generated reference                     | 92.52                    | 81.83                    | 90.05                  |
| ProvenanceGuard evidence                     | 91.94                    | 88.08                    | 90.09                  |
| RevisEval                                    | 93.70                    | 93.54                    | 92.53                  |
| DEREF (ours)                                 | 93.39                    | 95.09                    | 92.25                  |
| DEREF-v (ours)                               | 91.50                    | 94.18                    | 92.22                  |
| $\Delta$ vs. Plain                           | +1.44<br>[0.37, 2.51]    | +16.17<br>[13.89, 18.31] | +2.94<br>[2.04, 3.88]  |
| $\Delta$ vs. RevisEval                       | −0.32<br>[−1.48, 0.80]   | +1.55<br>[0.25, 2.83]    | −0.27<br>[−0.93, 0.38] |

Table 16: Argument AUROC (in points) of four judges with each kind of evidence. ProvenanceGuard evidence and RevisEval are the prior methods (Section 5.1). DEREF-v lets the deriver see the candidate value (Section 5.3). Full history gives Jev and the LLM judges the uncut history (Appendix A); FCRB has no cut, so there it equals Plain. Self-generated reference has the judge write its own reference from its own input (Section 5.1).  $\Delta$  rows give the difference of DEREF against Plain and against RevisEval, the stronger prior method in every cell, with its 95% CI below. Table 2 gives the main inputs, and Table 9 gives call AUROC.

| Judge                                      | Plain | Full history            | DEREF | $\Delta$                   |
|--------------------------------------------|-------|-------------------------|-------|----------------------------|
| <i>FCRB, DEFAULT (185 correct)</i>         |       |                         |       |                            |
| CLM                                        | 31.93 | –                       | 3.80  | −28.13<br>[−36.67, −20.70] |
| Jev                                        | 20.26 | –                       | 10.28 | −9.98<br>[−15.63, −4.20]   |
| Qwen3-8B                                   | 14.40 | –                       | −3.93 | −18.33<br>[−25.50, −9.25]  |
| Qwen3.8-27B                                | 36.03 | –                       | 12.12 | −23.91<br>[−32.30, −15.69] |
| <i>CFB, HISTORY-ONLY (279 correct)</i>     |       |                         |       |                            |
| CLM                                        | 66.63 | –                       | 28.26 | −38.37<br>[−47.49, −30.14] |
| Jev                                        | 59.39 | 21.89<br>[13.69, 29.20] | 38.35 | −21.04<br>[−30.54, −13.38] |
| Qwen3-8B                                   | 74.87 | 12.03<br>[6.66, 18.78]  | 31.37 | −43.50<br>[−55.21, −33.14] |
| Qwen3.8-27B                                | 57.08 | 3.36<br>[−2.76, 9.20]   | 26.21 | −30.87<br>[−40.51, −21.81] |
| <i>FinTrace, NO-SOURCE (2,150 correct)</i> |       |                         |       |                            |
| CLM                                        | 28.28 | –                       | 30.68 | +2.41<br>[−0.43, 5.55]     |
| Jev                                        | 27.47 | 29.09<br>[26.32, 31.64] | 27.68 | +0.21<br>[−1.91, 2.18]     |
| Qwen3-8B                                   | 25.72 | 29.04<br>[26.70, 31.54] | 26.41 | +0.69<br>[−2.18, 3.12]     |
| Qwen3.8-27B                                | 31.68 | 34.23<br>[31.56, 36.81] | 29.13 | −2.55<br>[−4.86, −0.12]    |

Table 17: Derivation gap  $\text{Gap}_\tau$  (in points) on the main path of each benchmark, with  $\tau$  set for each input. Full history gives Jev and the LLM judges the uncut history (Appendix A), with its 95% CI below; CLM cannot read it, and FCRB has no cut.  $\Delta$  is DEREF minus Plain, with its 95% CI below. On FCRB, the acceptance at  $\tau$  of the 185 correct defaults is 52.43, 64.08, 65.95, and 50.54 with Plain input and 78.92, 76.03, 86.20, and 73.35 with DEREF, for the four judges in table order; for the 793 correct values in view it is 84.36, 84.34, 80.35, and 86.57 with Plain input and 82.72, 86.31, 82.27, and 85.47 with DEREF.

| Input                                            | CLM                      | Jev                      | Qwen3-8B                  | Qwen3.8-27B              |
|--------------------------------------------------|--------------------------|--------------------------|---------------------------|--------------------------|
| <i>CFB, HISTORY-ONLY (279 correct, 59 wrong)</i> |                          |                          |                           |                          |
| Plain (AUROC)                                    | 64.33                    | 67.78                    | 49.18                     | 77.56                    |
| ProvenanceGuard evidence – Plain                 | +5.01<br>[−4.04, 13.34]  | +7.81<br>[0.40, 16.07]   | +17.64<br>[7.83, 26.98]   | +4.25<br>[−2.44, 11.69]  |
| RevisEval – Plain                                | +23.93<br>[16.90, 30.81] | +26.18<br>[17.28, 36.16] | +37.45<br>[27.23, 47.61]  | +18.33<br>[10.81, 25.64] |
| Full history – Plain                             | –                        | +12.80<br>[2.93, 23.17]  | +25.95<br>[16.29, 34.90]  | +11.45<br>[3.90, 19.93]  |
| DEREF – Plain                                    | +27.16<br>[17.63, 36.74] | +21.83<br>[13.32, 30.54] | +43.69<br>[34.07, 53.29]  | +16.87<br>[9.93, 23.04]  |
| <i>CFB, NO-SOURCE (462 correct, 747 wrong)</i>   |                          |                          |                           |                          |
| Plain (AUROC)                                    | 58.13                    | 77.14                    | 61.40                     | 75.46                    |
| ProvenanceGuard evidence – Plain                 | +5.49<br>[1.32, 9.83]    | +2.23<br>[0.51, 4.04]    | −13.18<br>[−16.95, −9.43] | +5.36<br>[3.93, 6.97]    |
| RevisEval – Plain                                | +19.87<br>[15.49, 24.39] | +13.23<br>[10.55, 16.08] | +18.75<br>[15.10, 22.67]  | +16.93<br>[13.50, 20.62] |
| Full history – Plain                             | –                        | −2.83<br>[−4.51, −1.28]  | −13.83<br>[−18.90, −9.00] | −0.28<br>[−2.56, 2.05]   |
| DEREF – Plain                                    | +25.04<br>[20.15, 29.88] | +11.10<br>[8.38, 13.84]  | +18.82<br>[14.49, 23.44]  | +17.80<br>[14.00, 21.82] |

Table 18: Argument AUROC (in points) on the two main paths of CFB. The first row of each block gives AUROC with Plain input, and the other rows give differences with their 95% CIs below. The wrong values of a path are those whose own value lies on that path. An argument with no item for an input keeps its Plain score. – marks inputs that we do not run with that judge.

## I Extended Related Work

This appendix expands the related work of Section 2.

**Judges for tool-calling.** Tool-call benchmarks check calls against gold answers. BFCL matches each argument value against a set of allowed answers (Patil et al., 2025), and ComplexFuncBench judges a whole call with rules, response matching, and an LLM (Zhong et al., 2025). Without gold answers, learned judges score or compare whole calls (Agarwal et al., 2025; Li et al., 2026b) or score each argument as a step (Lin et al., 2026). ToolPRMBench and AgentProcessBench evaluate step-level judges for tool-using agents, and the latter finds that judges tend to label steps as correct (Li et al., 2026a; Fan et al., 2026b). AgentJudgeBench finds that LLM judges of tool calls agree less with a programmatic reference as tasks get harder (Verma et al., 2026). Guards check calls before they run. ProvenanceGuard has an LLM find evidence for each derivable argument with the value in hand, and later, without the proposed action, check whether more than one action fits the context (She et al., 2026). PACT tracks where each argument value came from (Fan et al., 2026a). In Agent-Sentry, the agent reports whether each value is copied verbatim, derived, or generated (Sequeira et al., 2026). When rules flag a suspected failure in a proposed call, TwinCheck repairs it or writes a constrained alternative from the trace and the flagged evidence, and a verifier compares the two (Dai and Huang, 2026). Jev trails GPT-6 where the verdict has to be derived rather than read off the text (Li et al., 2026c), and factuality metrics drop on summaries that need deeper reasoning to check (Ramprasad and Wallace, 2025). To our knowledge, none of these studies splits judge errors by derivation path.

**Reference-guided judging.** Judges grade better with a reference answer. In MT-Bench, the judge first answers a math question without seeing the answers under review, then grades against its own answer, which reduces its errors (Zheng et al., 2023). LLMBAR and the candidate-blind head of BAER also have the judge write its own output first (Zeng et al., 2024; Li et al., 2026d), and RevisEval builds the reference by revising the candidate (Zhang et al., 2025a). Without a reference, judges agree less with experts (Krumdick et al., 2025) and give credit to wrong answers (Chalamalasetti and Vajjala, 2026). On math and code, judges without a

reference reward wrong answers that look convincing, and a judge that commits to its own solution before grading accepts far fewer of them (Zhou, 2026). SGV finds that verifiers of agent trajectories tend to agree with the candidate, and it first writes priors without the candidate (Andrade et al., 2026). MARCH and factored Chain-of-Verification also check claims without showing the checker the answer under test (Li et al., 2026e; Dhuliawala et al., 2024). For code agents, an agent explores the repository and writes a rubric before scoring any patch (Raghavendra et al., 2026). When a QA reference is swapped for a wrong one, judges often reject answers that match it, which the authors trace to the judges’ own knowledge (Lee et al., 2026). These judges write the reference from the question and their own knowledge, which is often not enough for an argument judge, whose expected value can sit in a tool return cut from its input. A reference that the judge writes from its own input, as in LLM-Bar, improves on Plain input by at most 3.01 points but stays below DEREf (Section 5.2); writing the DEREf reference from the cut input instead of the full history alone costs 9.63 points on CFB (Table 3), and on CFB and FCRB our judges follow a swapped reference (Section 5.4).

**Correct content without support.** A value without support in the context is not always wrong. Many entities that summarizers add without support are still factual (Cao et al., 2022). RAG models also often answer correctly when the context is insufficient (Joren et al., 2025). Models also score an inconsistent summary higher when its text is copied from the source (Tam et al., 2023). Tool-call arguments make this common, since a correct default, format change, or inferred value has no matching span in the dialogue. We do not treat such values as errors, and we measure how often judges do. A source in the context can also go unused. Models miss facts that share few words with the question (Modarressi et al., 2025), and they do worse as tool returns grow longer (Kate et al., 2025).

## J Qualitative Case Studies

Figures 5 and 6 show two arguments, one for each of two findings. Figure 1 already shows a correct value whose source was cut. For each finding, we wrote a condition before reading any score and drew one argument at random from those that meet it, with seed 20260930. A script copies all text from the judge inputs and the deriver outputs. The

### Judge input (Plain), excerpt

Question: Could you provide me with the details of Adriel's internship experiences, including the name of the organizations and the dates of the internships?  
 Function: get\_adriel\_detail\_experience\_and\_education -- Retrieves detailed information about Adriel's professional experiences and educational background, including the type of experience or education, name of the organization or institution, and relevant dates.  
 Parameter: experience\_or\_education\_name -- The name of the organization where the experience was gained or the institution where the education was obtained.; type: string; default: "Not specified"  
 Proposed value for experience\_or\_education\_name: "Not specified"

### Line added for schema facts

Schema default for experience\_or\_education\_name: "Not specified". The question does not contain any of these words: "experience\_or\_education\_name", "experience or education name", "Not specified".

### DEREF reference

The question asks broadly for all of Adriel's internship experiences without naming any specific organization, so the context does not determine a particular value for `experience\_or\_education\_name`. The parameter should take its default value of "Not specified" to retrieve all matching internship details.

| Judge       | Plain           | Schema facts    | DEREF           |
|-------------|-----------------|-----------------|-----------------|
| CLM         | 0.3718 / 0.6567 | 0.5092 / 0.6640 | 0.8924 / 0.7014 |
| Jev         | 0.7300 / 0.8600 | 0.7300 / 0.8600 | 0.9600 / 0.9300 |
| Qwen3-8B    | 0.0000 / 1.0000 | 0.0032 / 1.0000 | 1.0000 / 1.0000 |
| Qwen3.8-27B | 0.1824 / 0.6514 | 0.1824 / 0.4688 | 0.9903 / 0.9627 |

Figure 5: A correct default on FCRB (live\_multiple\_171-71-0). Each cell gives the judge's score and its threshold  $\tau$  for that input (score /  $\tau$ ). With DEREF, the score of Qwen3-8B equals its  $\tau$ .

excerpts show the lines relevant to the argument, and as in Figure 4 we write the long dash after a function or parameter name as --. Scores are each judge's probability that the value is correct.

**A default.** In Figure 5, the request asks for the details of Adriel's internships and names no organization. The correct value of experience\_or\_education\_name is the schema default, "Not specified", which the parameter line states. With Plain input and with the schema-facts line, all four judges score it below their thresholds  $\tau$ . The DEREF reference says that the request names no organization, so the parameter should take its default. With it, CLM, Jev, and Qwen3.8-27B score the value above their thresholds, and Qwen3-8B scores it at its threshold of 1.0000, which counts as accepted only with the tie fraction  $q$  (Appendix C). The condition is this pattern for all four judges. One of the 142 arguments of Figure 2b meets it.

**A swapped reference.** In Figure 6, the request asks for an alarm at 5 in the afternoon. The correct value of new\_alarm\_time is 17:00, and the wrong FCRB call, made by a real model, uses 15:00. The original reference names 17:00, and the swapped reference changes only that value. With the orig-

inal reference, all four judges score the correct value at or above 0.5 and the wrong value below 0.5. With the swapped reference, all four judges reverse both judgments. Under Plain input, CLM and Qwen3-8B score both values at or above 0.5, Jev scores only the correct value above 0.5, and Qwen3.8-27B scores only the wrong value above 0.5. The condition is that all four judges reverse both judgments at 0.5. Of the 749 CFB and FCRB arguments in the swap test, 14 meet it. We exclude the two pairs whose values differ only by a power of ten, 5.0 against 0.05 and 16 against 1600, since the schema leaves their unit open, and draw from the remaining 12.

### Judge input for the correct value (Plain), excerpt

Question: Could you set a new alarm for 5 in the afternoon labeled 'Grocery run'?  
Previous calls (most recent 5):  
none  
Function: Alarm\_1\_AddAlarm -- This function sets a new alarm with a specified time and an optional name.  
Parameter: new\_alarm\_time -- The time to set for the new alarm, in 24-hour format (HH:MM).; type: string  
Proposed value for new\_alarm\_time: "17:00"

### Original Deref reference

The value should be **"17:00"**, since the parameter requires 24-hour HH:MM format. This is derived from the user's request text: "set a new alarm for **5** in the afternoon**"**."

### Swapped reference

The value should be **"15:00"**, since the parameter requires 24-hour HH:MM format. This is derived from the user's request text: "set a new alarm for **5** in the afternoon**"**."

| Judge       | Plain   |        | Original reference |        | Swapped reference |        |
|-------------|---------|--------|--------------------|--------|-------------------|--------|
|             | Correct | Wrong  | Correct            | Wrong  | Correct           | Wrong  |
| CLM         | 0.7870  | 0.7632 | 0.8059             | 0.4389 | 0.3369            | 0.8281 |
| Jev         | 0.9800  | 0.0600 | 0.9800             | 0.0100 | 0.4100            | 0.9700 |
| Qwen3-8B    | 1.0000  | 1.0000 | 1.0000             | 0.0000 | 0.0000            | 1.0000 |
| Qwen3.8-27B | 0.3486  | 0.9579 | 0.9914             | 0.0003 | 0.0005            | 0.9890 |

Figure 6: A swapped reference on FCRB (live\_multiple\_809-176-0). The correct value is 17:00 and the wrong value is 15:00. Cells give scores.