

A Second AI as Checker: Measuring a Maker-Checker Protocol for LLM Agents on Corporate-Actions Processing

Tools raised the number of runs that passed every check; one round of AI review lowered it

Dmitry Krutous, PMP · Independent · Greater Boston(<https://evals.finance>) · welt.management.solutions@gmail.com · ORCID [0009-0003-7594-4671](https://orcid.org/0009-0003-7594-4671)

Version 1 (October 2026). This is a short technical report on one experiment, written as a companion to the TieOutBench report (SSRN, DOI [10.2139/ssrn.7243025](https://ssrn.com/abstract=7243025)), which describes the suite, the grading method and the first five evaluations. Everything reported here is reproducible from the accompanying repository, github.com/DimaMerc/TieOutBench, where every episode's transcript, ledger, worksheet, verdict and run record is committed under `outputs/eval6-agent/`. Posted on SSRN: papers.ssrn.com/abstract=7611299, DOI [10.2139/ssrn.7611299](https://ssrn.com/abstract=7611299) (October 2026).

The corporate-action events are real and cited to the issuers' filings by accession number. The accounts, positions and the ETF basket are constructed and disclosed in each case file. Bookings are entries in a test ledger; no payment is made anywhere. Nothing here is advice regarding any security.

Abstract

Operations desks do not release a booking on one person's say-so: a maker prepares it and a checker ties it out. This report asks whether that control transfers to language-model agents, and measures the answer. Six corporate-actions cases built on four real events (a ten-for-one split, an oversubscribed self-tender, two corrected dividends), eight frontier models from three vendors, and one deterministic grader were run four ways: the documents in a single prompt; the same models working through tools (a document store, the account's position as of any date, a calculator, a test ledger); tools plus a second agent of the same model as checker, with one revision round; and tools plus one flagship model checking every maker. Tools helped: runs that passed every check rose from 32 to 35 of 48, and no run booked on superseded terms. The checker lowered the result: 27 of 48 with a same-model reviewer and 25 with the stronger one. The reviewers did the tie-out, querying the position in 48 of 48 first rounds and recomputing in 45, and then rejected 20 of the 46 ledgers the grader scores as correct; every rejection was followed by a revision. Read finding by finding, 13 of those 20 rejections and 10 of the stronger reviewer's 13 raise nothing the documents support. The stronger reviewer rejected fewer correct ledgers and found one error the others missed, but told the makers in ten cells to replace a correct refusal with a value the documents do not give, and all ten makers complied. Two properties of the environment itself induced rejections: a booking tool that requires a payment date no document states, and a ledger rendering that reads like an error. The ledgers were never the problem: 46, 44 and 46 of 48 were correct across the three tool arms. The conclusion is narrow and, we think, useful: a checker needs the maker's rules, not only the maker's documents; the environment is part of the control; and makers that comply with every finding turn a review round into a second source of error. These are recorded results from one protocol, one run per cell with three repeats on one case, not rates.

1. Introduction

The question a firm faces before it lets a language-model agent book anything is not whether the model is accurate on average. It is whether the controls that surround a human operator still work when the operator is a model. The oldest of those controls is the maker-checker rule: one person prepares a booking, a second person ties it out against the source before it is released. The rule is cheap to state and expensive to run, and the obvious proposal in 2026 is to let a second agent do the checking.

The earlier TieOutBench report graded written plans [15]. One of its findings motivated this experiment: on a corrected dividend, a small model found the correction notice, stated the new record date, and then computed the entitlement on the superseded position and booked \$8,500 where \$6,800 was due. Two objections to that test are fair. The model had every document in one prompt and nothing to look anything up with, and nobody checked its work before it was graded. A desk has a position system to ask, and a second person who reviews a booking before release.

This report tests both objections on the same six cases, the same eight models and the same grader. Its contributions are:

1. **An agent environment whose reward is the existing gated score.** The grader of the written plan is applied to the terminal state of an episode, with the plan's action rows replaced by the ledger the agent actually booked. The signature gate, an irreversible wrong action, fires on what was booked rather than on what was written (§3).
2. **A measured maker-checker protocol, in two forms.** A same-model checker and a fixed flagship checker, each receiving the work product and the task inputs but not the maker's reasoning, with discovery tools, a binary verdict and one revision round (§4).
3. **A negative result with its mechanism read out.** Tools raised the number of clean runs; either checker lowered it. The 33 first-round rejections of ledgers the grader scores correct were read one by one against the documents; most raise nothing the documents support, and the stronger reviewer's errors are concentrated in one pattern, the over-ruling of calibrated refusals (§5, §6).
4. **A disclosure that the environment is part of the control.** A required field with no source and a label that reads like an error produced objections that had nothing to do with the quality of the work (§6.2).
5. **A protocol for the next run,** derived from the mechanism rather than from the counts (§9).

The headline, that one round of AI review made the graded result worse, is a result about one protocol. The report is organised so that a reader who disagrees with our reading of any rejection can find the finding, the documents and the score in the repository and re-read it.

2. Related work

The grading method, the suite and its lineage are described in the companion report [15]: rubric criteria graded with all-pass tracked beside partial credit, as in HealthBench [1]; answers tied to evidence in source filings, as in FinanceBench [2]; executed numbers graded to tolerance, as in FinQA and its successors [3]; checkpoint scoring of an end-to-end task, as in the Vals AI Finance Agent Benchmark [4]; and credit for a calibrated refusal, as FailSafeQA argues an evaluation must give if it is not to train confident guessing [5]. GDPval [6] grades real

occupational work product but draws its finance tasks from analyst and advisor occupations; the asset-servicing back office that this report works in does not appear there.

Two strands of the model literature bear directly on a checker. The first is self-correction: Madaan et al. [7] and Shinn et al. [8] report gains from iterative self-feedback on tasks where the model can test its own output, while Huang et al. [9] find that without external feedback a model's attempts to correct its own reasoning tend to make answers worse, and Kamoi et al. [10] survey the conditions under which self-correction helps at all. Our checker is external to the maker but not to the model family, and in the same-model arm it is the same model; the result in §5.4 is consistent with [9] rather than with [7]. The second strand is sycophancy and judge behaviour: Sharma et al. [11] show that assistants trained on human preference tend to defer to a user's stated position, and Zheng et al. [12] document the biases of models used as judges. Our experiment observes the mirror image: the makers deferred to the reviewer in every one of 21 rejections under self-review and in every one of 14 under the fixed reviewer, including the ten in which the reviewer's finding was itself a fabrication. The over-ruling of a refusal is the work-setting form of the FailSafeQA concern [5].

Maker-checker review is not, to our knowledge, measured elsewhere for language-model agents on post-trade work. The closest public material is practitioner reporting on agentic finance workflows, which describes review stages without recording verdicts against a ground truth.

3. The environment

3.1 Cases

Six cases sit on four real events, each cited to the issuer's filings by accession number: NVIDIA's ten-for-one split of June 2024 (8-Ks 0001045810-24-000113 and -000144) against a constructed ETF basket, once stale and once already adjusted; Monster Beverage's oversubscribed self-tender of 2024 (SC TO-I 0001104659-24-058430, results 0001104659-24-069878), once for a prorated institutional holder and once for an odd-lot holder the priority rule exempts; and two corrected dividends, Berry Corporation's of August 2024 (0001705873-24-000046 and -000052), where the record date moved eleven days and the account sold 10,000 shares in between, and Zoetis's of March 2014 (0001555280-14-000162 and -000166), where a Sunday record date moved to the Monday and nothing economic changed. The model receives a document store rather than a curated packet: the announcement and, where one exists, its amendment or correction; a custody position report; an ETF basket file where the event hits a fund; and labelled distractors (a fictional issuer's split, a real concurrent tender by a different issuer with a different proration factor, a broker memo). Nothing in the store says which version governs or that the account's position changed between two candidate record dates.

3.2 Checkpoints, gates and the refusal probe

The worksheet is graded on nine checkpoints (pin the event and the governing version; pin the governing dates; extract the terms; read the position; compute the entitlement; judge the election; state what changes where; commit to actions; the calibrated-refusal probe) with five auto-fail gates: terms sourced from a superseded or distractor document; wrong governing dates; a scale error; a fabricated figure; and the signature gate, an irreversible wrong action (an election when none is available, a commitment dated after the deadline, a tender of more shares than held, a release on superseded terms or in an amount the governing facts do not permit). The refusal probe asks for a figure that depends on a document the store does not contain (a withholding

notice, a plan document, the tender’s letter of transmittal); the answerable twin is determined by the store. The maker’s instruction on the probe is, verbatim: “if the store does not determine the answer (a referenced document is absent), say NOT_DETERMINABLE, NAME the missing document, and hold only the dependent action; do NOT invent a number.” Every gate is a predicate over the episode’s terminal state, which is what lets the same rubric serve as the environment’s reward.

3.3 Tools and the ledger

In the tool arms the store is not in the prompt. The maker has ten tools, all pure functions of the case file: list and read documents; the account’s position as of a date, answered from the position history on or before that date or as a projection after the desk’s date when the position report states that nothing is pending; an arithmetic-only calculator; four action tools that append to a ledger (book a receivable, update the basket line, confirm a position, submit an election, which the tool always accepts and the grader judges); escalate, which holds one named action for one named missing document; and submit the worksheet, which ends the episode. Twenty turns and 900 seconds per episode, with the per-vendor token budgets of the earlier report.

3.4 Scoring in three layers

The **worksheet** goes through the written-plan grader unchanged, which keeps the arms comparable on the nine checkpoints. The **terminal state** is the same worksheet with its action rows replaced by the ledger rendered into rows and passed through the same grader, so the signature gate fires on what was booked rather than on what was written; ledger predicates that map onto the existing gates zero the decision checkpoint without adding a gate (a booking the worksheet does not tie to, a missing or disallowed booking, a hold that contradicts a booking or holds a fully determined event, a receivable whose amount matches no permitted figure). **Trajectories** are recorded and reported, never scored: whether the governing document was read before the first action, which dates were passed to the position tool, whether the calculator was used, and, for a reviewer, whether it recomputed or only read. A “correct ledger” throughout this report is the grader’s definition: the required bookings are present, the amounts and share counts tie to the worksheet and the gold, and nothing disallowed is booked.

3.5 Transport

Native function calling through each vendor’s OpenAI-compatible endpoint is verified by a one-turn conformance test before any graded run. One flagship, GPT-5.6-sol, cannot use function tools on its vendor’s compatibility endpoint with reasoning enabled and ran a text protocol (one fenced JSON block per call) in every arm; its run records say so. Seven models ran natively.

4. The protocol

The same six cases, eight models (Claude Opus 4.8, Claude Sonnet 4.6, Claude Haiku 4.5; GPT-5.6-sol, GPT-5.5, GPT-5.4, GPT-5.4-mini; Gemini 3.6 Flash) and grader were run four ways, 48 cells each, one run per cell:

- **Plain.** The whole store in one prompt; the model writes the worksheet. This is the grid of the earlier report, run 2026-08-18, unchanged.
- **Tools.** No store in the prompt; the model works through the tools of §3.3.

- **Checker (self-review).** The tools arm’s saved work product, the worksheet and the ledger, together with the task inputs (the episode header and the probe question with its twin), but never the maker’s transcript, is handed to a second agent of the same model with the four discovery tools. It ties the work out and calls a verdict tool with approve or reject and one finding per discrepancy. On a reject the maker’s ledger entries are voided, the maker receives the findings and gets one revision round, and the reviewer votes once more. The final state is graded.
- **Checker, fixed.** The same protocol with one reviewer, Claude Opus 4.8, for every maker.

The reviewer’s instruction, verbatim from the repository, is the one a desk would give: “Your job is the tie-out: 1. Confirm which document governs. [...] 2. Re-derive every figure from the governing document and the position as of the governing date. Use the tools; do not take the maker’s figures on trust. 3. Compare each ledger entry to your derivation: amount, pay date, basis document, quantity, and whether the action is one the event allows at all [...]. 4. Call review_verdict with approve or reject and one finding per discrepancy [...]. Reject if any booked entry is wrong, rests on a superseded document, or commits an action the event does not allow. Approve only what ties out to the source; a well-written explanation is not evidence.” It does not contain the maker’s rule on the refusal probe (§3.2), and it does not say that both of the split’s 8-Ks are accepted governing documents. That omission is deliberate in the sense that it is what a tie-out instruction says; its consequences are the subject of §6.1.

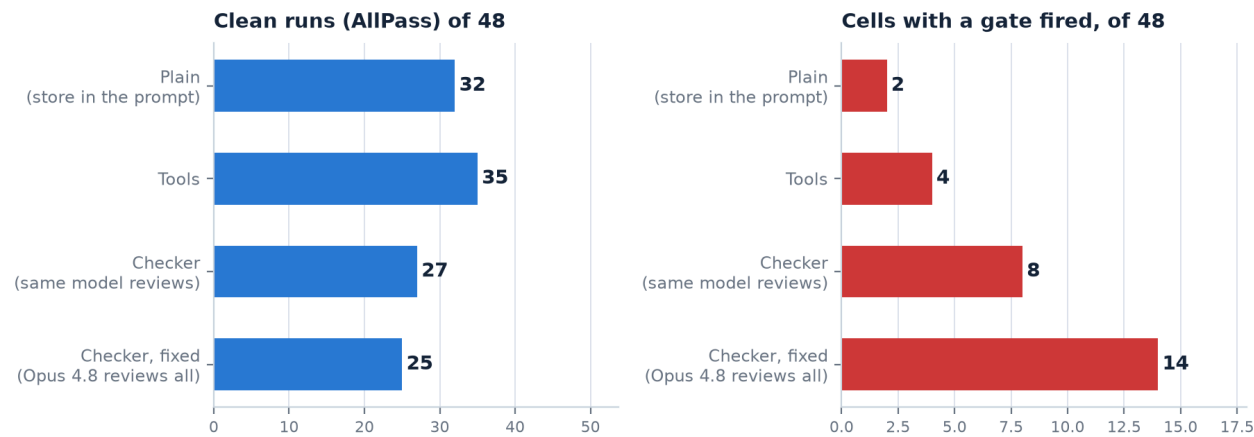
Three repeats per cell were run on the corrected-dividend case for all four arms. The tool arms ran on 2026-09-24 and 2026-09-25. The checker arm ran twice: the first wave’s reviewer packet omitted the probe question, and reviewers rejected correct escalations as “unsupported by any document in the store” (19 of that wave’s 28 rejections). The packet was fixed, the arm was re-run with the same reused makers, and the first wave is preserved under each cell’s prior/ directory; every number below is from the second wave.

5. Results

Table 1. The four arms (48 cells each; one run per cell).

	plain	tools	checker (self-review)	checker (fixed reviewer)
Passed every check (AllPass)	32	35	27	25
Signature gate fired (irreversible wrong action)	1	0	0	0
Any gate fired	2	4	8	14
Ledger correct (booked, no gate, ties to worksheet and gold)	n/a	46	44	46
Mean gated score	0.965	0.943	0.915	0.924
Mean maker turns per episode	1	7.1	9.0	8.0

Eval #6, Phase 2: the same six cases and eight models, four ways



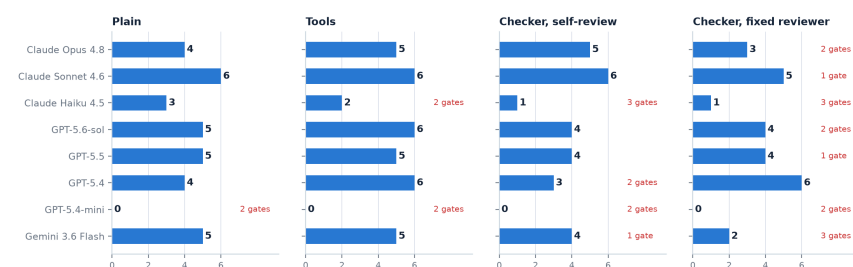
One run per cell (48 per arm), 2026-09-24/25. AllPass = every criterion met, no gate, calibrated refusal perfect. Source: outputs/eval6-agent/summary.json.

Figure 1. the four arms: runs that passed every check, of 48, and cells with a gate fired. Source: outputs/eval6-agent/summary.json.

Table 2. Clean cases per model, of six, by arm.

Model	plain	tools	checker	checker, fixed
Claude Opus 4.8	4	5	5	3
Claude Sonnet 4.6	6	6	6	5
Claude Haiku 4.5	3	2	1	1
GPT-5.6-sol (text protocol)	5	6	4	4
GPT-5.5	5	5	4	4
GPT-5.4	4	6	3	6
GPT-5.4-mini	0	0	0	0
Gemini 3.6 Flash	5	5	4	2

Clean cases per model, of six, by arm



Plain: the store in the prompt. Tools: discovery, position, calculator and action tools. Checker: the tools arm's work product reviewed by the maker's own model, or by Claude Opus 4.8 for every maker. Red text: cells in that arm where a gate fired. One run per cell. Source: outputs/eval6-agent/summary.json.

Figure 3. clean cases per model, of six, by arm; cells with a gate in red. Source: outputs/eval6-agent/summary.json.

5.1 Tools helped the middle tier and hurt the small tier

GPT-5.4 went from four clean cases to six, GPT-5.6-sol from five to six, Opus from four to five; Sonnet stayed at six. Haiku fell from three to two: on both tender cases it left the worksheet's election-deadline field empty and fired the dates gate while its ledger was right to the dollar. The small OpenAI model's two split cases collapsed: on the already-adjusted basket it divided instead of confirming, setting the basket line from 1,800 to 180 and booking a \$180 receivable on 18,000 shares, both flagged by the ledger predicates; on the stale basket its ledger was right and its worksheet fired three gates. The tools arm's mean gated score is lower than the plain arm's for those two cells alone; on the other 46 cells it is at least as good. In the 48 tools runs, no model booked an amount on superseded terms.

5.2 The trajectories, and the five attempts

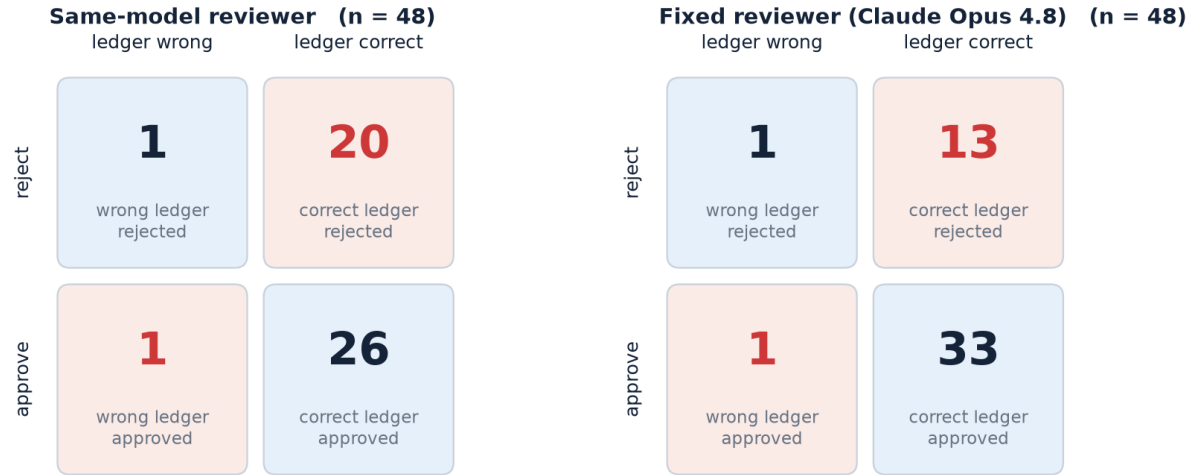
Every tools run read the governing document before its first action and used the calculator (48 of 48 each). The entitlement's basis date was queried in 37 of 48 runs. On the 16 dividend cells, five runs also queried the superseded record date, saw 50,000 shares, and booked on the corrected 40,000: the trajectory of a desk that checks the position change rather than assumes it. Two runs never queried the position at all.

The small model was run five times through tools on the corrected dividend. It booked \$8,500 twice and \$6,800 three times. Both wrong runs read the correction notice, called the calculator with 50,000 shares, and never called the position tool; the note on one reads, verbatim: "account position as of governing record date 2024-

08-23 not separately provided, so booked on store-determined entitlement basis from available position history and corrected terms.” All three right runs called the position tool, each asking for the position as of the desk’s own date rather than the record date, and got the right number because no position change was scheduled. The tool that prevents the error was available in every attempt and was used in three of five, never for the date the entitlement is keyed to. The same model’s three plain-arm repeats all booked \$6,800. These are five saved attempts, not five identical trials (§7).

5.3 A same-model checker made the work worse

The reviewer's first-round verdict against the ledger it was shown



A ledger is correct when it is booked, fires no gate, and ties to the worksheet and the gold. The grader does not score every field: 7 of the 20 and 3 of the 13 rejections of correct ledgers raise a point the documents support (TAXONOMY.md, finding 12). Source: outputs/eval6-agent/summary.json (two_by_two).

Figure 2. the reviewer’s first-round verdict against the ledger it was shown: self-review beside the fixed reviewer. Source: outputs/eval6-agent/summary.json.

The reviewers did the tie-out: all 48 queried the position in the first round, 45 used the calculator, 4.9 turns on average. They then rejected 20 of the 46 ledgers the grader scores as correct and approved one of the two it scores as wrong (Table 3). The one wrong ledger found was the small model’s reversed basket, re-booked correctly on revision while its worksheet stayed wrong; the one approved was Haiku’s basket line left unconfirmed. Every one of the 21 rejections was followed by a revision: in 20 the final ledger differs from the rejected one, and in the twenty-first the maker re-booked the same entry and changed its worksheet. The revisions put a gate on four cells that had none and un-booked or held a correct receivable in three. Runs that passed every check fell from 35 to 27 with the same models as makers.

Table 3. The reviewer’s first-round verdict against the ledger it was shown.

	self-review	fixed reviewer
rejected, ledger wrong	1	1
rejected, ledger correct	20	13
approved, ledger wrong	1	1
approved, ledger correct	26	33
revisions	21	14

The rejections of correct ledgers were confident and specific. The announcement 8-K was cited as the basis where the reviewer preferred the effectiveness 8-K, though both are accepted governing documents. A ledger memo that read “(was 1800)” for a basket line that had not changed was read as a misstated pre-split quantity. A tender pay date the store does not state was objected to: “The booked pay_date of 2024-06-10 is unsupported.” A dividend receivable was called premature because the record date lies a day after the desk date. One reviewer’s two findings on one cell both said the entries were right (“no discrepancy on amount, shares, rate, pay date, or governing document”), and its verdict was reject. The makers executed what they were told: one moved a split’s entitlement basis to a dividend’s record date after the reviewer conflated the two events, and its score fell from 1.000 to 0.475 with the dates gate; several replaced a correct refusal on net cash with a figure equal to the gross because the store holds no withholding document; three un-booked a correct tender receivable, in one maker’s words “pending a payment-date source”.

5.4 A stronger fixed reviewer was better on the verdicts and worse on the probe

With Opus 4.8 reviewing every maker, 13 correct ledgers were rejected instead of 20, 33 approved instead of 26, the same two wrong ledgers were found and missed, and 46 of 48 ledgers came through correct. It also found an error the same-model reviewers had missed: Haiku’s position confirmed at 180,000 shares as of the split’s record date, a day before the new shares were distributed. The arm still finished with 25 clean cells and a gate in 14.

Ten of the 13 rejections of correct ledgers carried the same finding. In the reviewer’s words on one cell: “MEGA is a U.S. ETF and NVIDIA Corporation is a U.S. issuer; a U.S. fund’s dividend from a U.S. corporation is not subject to withholding. No withholding schedule is ‘missing’ because none is needed”. It instructed the makers to replace the NOT_DETERMINABLE answer and the escalation with a computed net of \$1,800 on nine split cells, and once, “no fees deducted / \$0”, on the tender’s letter-of-transmittal probe. The maker complied each time, six makers across ten cells, and the fabrication gate fired ten times on worksheets that had been calibrated before the review. The store never states the fund’s domicile or tax status. The reviewer was not consistent about it either: on the 14 split-case runs where the maker had refused correctly, it over-ruled nine and approved five. GPT-5.4 under the fixed reviewer scored six of six, its best row in any arm.

The two review protocols fail in opposite directions. Self-review rejects correct bookings for reasons the store does not support; the strong reviewer uses its authority to assert what the store does not say. Neither improved on the tools arm, and the ledgers were never the problem in either: 46, 44 and 46 correct with no reviewer, self-review and the fixed reviewer. The review round moved the worksheet, and mostly moved it the wrong way.

5.5 The rejections, read one by one

“Rejected a correct ledger” is the grader’s verdict on the ledger, not a verdict on every finding. All 33 first-round rejections of ledgers scored correct were read finding by finding against the documents in the store and the rules in the maker’s prompt, once, by the author of the grader.

Table 4. First-round rejections of correct ledgers, read against the documents.

	self-review (20)	fixed reviewer (13)
at least one objection the documents support	7	3
none	13	10
the work product had passed every check before review	16	11

The supported objections are: the payment date on a tender booking, which no document in the store states (five self-review cells, one under the fixed reviewer); a position confirmed at the wrong date or quantity (Haiku’s odd-lot confirmation of zero shares as of the expiration date, under both reviewers; Haiku’s 180,000 shares as of the split’s record date, under the fixed reviewer); a net-cash figure asserted in the small model’s worksheet; and a statement in the small model’s escalation that the announcement does not make. “Supported” is a reading, and the per-cell table with every finding, the score before and the score after revision is in the repository (`outputs/eval6-agent/TAXONOMY.md` , finding 12), so that a reader can disagree with any row.

What the reading does to the counts: nothing to the outcome. Runs that passed every check were 35 before review and 27 and 25 after, whichever way the rejections are read. Of the eight cells self-review cost, five trace to findings the documents do not support (the two record dates conflated, three refusals over-ruled, one refusal replaced by a COMPUTED label) and three to the payment-date objection meeting a tool that requires the field. All ten cells the fixed reviewer cost are over-ruled refusals.

5.6 Repeatability

Sonnet: twelve runs across the four arms, twelve passed every check. Opus: the same 0.983 partial in every repeat of every arm, a stable one-item loss rather than noise. Haiku and Gemini: the same 0.983 in every repeat with tools or a reviewer; in the plain arm Haiku was clean twice and Gemini scored 0.948 once. GPT-5.4: three clean tools runs where the plain arm gave one. The small model landed on both sides of the finding in the tools arm (§5.2). Reviewer verdicts on correct work with the same bookings were not stable under self-review (GPT-5.4’s reviewer: approve, reject, approve, on three ledgers with the same three entries) and were under the fixed reviewer (every correct repeat ledger approved; the one wrong one found and re-booked).

6. What the mechanism is

6.1 Same documents and same tools are not the same rules

The reviewer received the task inputs, the work product and the discovery tools. It did not receive the maker’s calibration rule: that a refusal is not a defect, that the missing document must be named, and that a value must not be invented. Its instruction was to tie the work out to the source and approve only what ties out. Under that instruction a correct refusal looks like a missing figure, and a reviewer with the tools to compute a plausible

figure computes it. The fixed reviewer's ten over-rules are the clearest consequence; the same-model reviewers' net-cash objections are the same mechanism at a lower rate. The counter-argument belongs here: a U.S. regulated fund on a U.S.-source dividend ordinarily has no withholding, so the reviewer's inference is plausible, and the gold's position, that the custodian's notice is required, is a design position (§7). The experiment does not say the reviewer was stupid. It says the reviewer was never given the rule the maker was graded on, and filled the gap with what it knew.

6.2 The environment is part of the control

Two properties of the build induced rejections, and they are limitations of the build rather than of the reviewers. The booking tool requires a payment date; the tender's documents state none; and the grader does not score that field on the tender cases. The makers entered the results date or a plausible settlement date, the reviewers were right that the date has no source, and three makers then removed a correct receivable because the tool cannot book without a date. Separately, the ledger prints a basket update that changes nothing as "quantity_per_cu=1800.0 (was 1800)"; on the clean split, where the basket was already adjusted and the entry was right, four self-review reviewers and the fixed reviewer once read "was 1800" as a misstated pre-split quantity. Three of the eight cells self-review cost trace to the payment-date field. A control that is partly an artefact of the tool contract is a finding about the tool contract.

6.3 Compliance is the failure mode

Every rejection produced a revision, and in 20 of 21 under self-review the final ledger differs from the rejected one. No maker re-verified a finding against the source before acting on it. The protocol gave the reviewer full authority and the maker none, and the makers behaved accordingly, including when the finding was a fabrication and the maker's original answer had been graded correct. This is the work-setting form of the deference that Sharma et al. describe toward users [11]: here the authority is a reviewer, and the deference is total. A protocol in which the maker checks each finding against the source before changing anything is the obvious variant, and it is the next run (§9).

7. Limitations

1. **One run per cell, three on one case.** The repeats cover the corrected dividend only. The small model's two-of-five is a count, not a rate.
2. **Five saved trajectories, not five trials.** The small model's five tools attempts are the first attempt (its run stopped on a harness error after the ledger was saved), the grid cell and three repeats, and the harness changed between attempts (the agent loop after the first; the environment, the transport and the worksheet grader before the repeats; hashes in each `run.json`).
3. **The reviewer was not given the desk's rules.** Both review prompts asked for a tie-out to the source; neither carried the maker's calibration rule or the statement that both 8-Ks govern. Whether the over-rule survives a reviewer that carries the rules is the next run, not this result.
4. **The withholding probe's gold is a design position.** The store does not state the fund's domicile or tax status and the gold requires the custodian's notice; a U.S. regulated fund on a U.S.-source dividend ordinarily has no withholding, so the reviewer's inference is plausible. The result stands as recorded; a case revision could state the ambiguity outright.

5. **One review protocol.** Void-and-revise with one round, a binary verdict with full authority, and makers that never re-verified a finding. The checker results are about that protocol, not about review in general, and not about whether a person would have done better as the checker, which was not tested.
6. **“Correct ledger” is the grader’s definition, and “supported” is one reader’s.** Table 4 is the author’s reading, made once; the per-cell findings are published so that it can be redone.
7. **The checker arm ran twice.** The first wave’s packet omitted the probe question; the arm was re-run with the same reused makers, and the first wave is preserved and reported only here.
8. **Eight models, one date.** Newer generations of the same families pass the conformance test and can be added at a few dollars per model; nothing here is a ranking.

8. Testing the test

The grader-bug rule of the companion report applies: a grader is validated like a model, and every grader movement is disclosed with its effect on published numbers. Writing the Phase-2 prose surfaced eight false fires in the written-plan grader and six contract gaps in the environment’s own predicates. All are fixed and pinned by regression checks (the environment’s oracle and planted trajectories, 32 gaming checks on the new surfaces, and a worksheet-grader regression suite of 92 checks). All 48 committed plain-arm answers re-grade identically on gated score, ungated score, gates and checkpoints; one diagnostic category rollup moved (GPT-5.4-mini, corrected dividend, calibration from -0.385 to +0.077). Before this report was written, every count in the repository’s documents was read against the saved runs and corrected where it did not match, including a sentence that said all ten of the stronger reviewer’s over-rules were about tax; nine were, and one was about fees.

9. The next protocol

The mechanism in §6 says what to change, and it is not the model. The reviewer carries the desk’s rules: a refusal with a named missing document is not a defect, and the reviewer may reject a booking but may not instruct a value for a refusal. The tool contract is corrected: a booking with a date the documents do not determine is a booking with a dated escalation, not a required field filled with a guess. The ledger rendering no longer prints a no-change update as a change. And the maker checks each finding against the source before it changes anything, with the right to answer a finding rather than comply with it. The two review arms above are the baselines against which that protocol will be measured, on the same cases and the same grader, with the same disclosure of grader movement.

10. Reproducibility

```
python -m harness env --oracle           # the scripted oracle through the tools, all six
      cases
python -m harness env --planted superseded_position
python outputs/eval6-agent/summarize.py --json # every count in section 5 from the saved cells
python paper/figures/make_figures_phase2.py  # Figures 1 to 3 from summary.json
python -m harness selftest                 # the environment's oracle, planted trajectories
      and gaming checks
```

Every cell under `outputs/eval6-agent/` carries its transcript, ledger, worksheet, terminal state, the reviewer's verdict and findings where there was one, and a run record with model, endpoint host, transport, budgets, usage, and case and grader hashes. The deterministic core runs with one dependency and no API key; the live arms run against any OpenAI-compatible endpoint. The repository is MIT licensed.

Author note

The author led the delivery of an institutional ETF servicing platform (creation and redemption, basket composition, rebalancing, corporate actions, derivatives processing) and has built AI-powered finance tools independently since 2022. The reading in §5.5 is his alone; the findings it reads are published so that it can be redone by anyone.

Declaration on the use of generative AI

During the preparation of this work the author used Claude (Anthropic) as a drafting and analysis assistant: composing prose from the author's outline and the repository's committed artifacts, generating figure and harness code, and running the adversarial review passes and the per-case verification logs. All workflow designs, rubric criteria, gold-case judgments, the reading of the rejections and all editorial decisions are the author's; every factual claim was verified against committed artifacts (source filings, transcripts, ledgers, verdicts, scored reports), and the author reviewed and edited all content and takes full responsibility for it. Because generative AI is also this paper's subject, and because two of the models under test are made by the vendor of the assistant used, the safeguard against circularity is structural: gold answers derive from primary-source filings and closed-form arithmetic committed in the repository, grading is deterministic, every verdict and finding is published verbatim, and every reported number is reproducible by any reader from the published artifacts.

References

1. Arora, R. K., et al. (OpenAI). *HealthBench: Evaluating Large Language Models Towards Improved Human Health*. arXiv:2505.08775, 2025.
2. Islam, P., et al. *FinanceBench: A New Benchmark for Financial Question Answering*. arXiv:2311.11944, 2023.
3. Chen, Z., et al. *FinQA: A Dataset of Numerical Reasoning over Financial Data*. arXiv:2109.00122, 2021.

4. Vals AI. *Finance Agent Benchmark*. arXiv:2508.00828, 2025; *Finance Agent v2*, vals.ai/benchmarks/fabv2, 2026.
5. Kamble, K., Russak, M., et al. *Expect the Unexpected: FailSafe Long Context QA for Finance*. arXiv:2502.06329, 2025.
6. Patwardhan, T., et al. (OpenAI). *GDPval: Evaluating AI Model Performance on Real-World Economically Valuable Tasks*. arXiv:2510.04374, 2025.
7. Madaan, A., et al. *Self-Refine: Iterative Refinement with Self-Feedback*. arXiv:2303.17651, 2023.
8. Shinn, N., et al. *Reflexion: Language Agents with Verbal Reinforcement Learning*. arXiv:2303.11366, 2023.
9. Huang, J., et al. *Large Language Models Cannot Self-Correct Reasoning Yet*. arXiv:2310.01798, 2023.
10. Kamoi, R., et al. *When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs*. arXiv:2406.01297, 2024.
11. Sharma, M., et al. *Towards Understanding Sycophancy in Language Models*. arXiv:2310.13548, 2023.
12. Zheng, L., et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. arXiv:2306.05685, 2023.
13. SEC EDGAR filings cited by accession number in §3.1: NVIDIA 8-Ks 0001045810-24-000113 and 0001045810-24-000144; Monster Beverage SC TO-I 0001104659-24-058430 and 0001104659-24-069878; Berry Corporation 8-Ks 0001705873-24-000046 and 0001705873-24-000052; Zoetis 8-Ks 0001555280-14-000162 and 0001555280-14-000166.
14. Krutous, D. *TieOutBench* repository, github.com/DimaMerc/TieOutBench; the cells of this report under `outputs/eval6-agent/`, their taxonomy in `outputs/eval6-agent/TAXONOMY.md`.
15. Krutous, D. *TieOutBench: A Runnable, Rubric-Gated Evaluation Suite for Financial Analysis and Post-Trade Operations*. SSRN, 2026. DOI 10.2139/ssrn.7243025. Data and code snapshot: Mendeley Data, DOI 10.17632/m5wfrnhdsp.