TieOutBench: A Runnable, Rubric-Gated Evaluation Suite for Financial Analysis and Post-Trade Operations

Measuring whether an LLM's finance work ties out — before deciding whether to trust it

Dmitry Krutous MBA, PMP · Independent · Greater Boston evals.finance · welt.management.solutions@gmail.com

Version 1 — August 2026. This is a technical report on a working artifact, written in the spirit of an honest v0: the caveats are stated next to the results they bound, and everything reported here is reproducible from the accompanying repository, github.com/DimaMerc/TieOutBench. Posted on SSRN: papers.ssrn.com/abstract=7243025.

Case gold answers — including the McDonald's and NVIDIA fair values and the KOCT snapshot arithmetic — are grading references computed under stated, dated case assumptions. They are not valuations of, or advice regarding, any security.

Abstract

A finance LLM's answer can look right and be wrong. A model that misreads one statement header ("in thousands" as "in millions") produces an internally consistent, fluent analysis that scores 0.951 on a naive criteria average — and 0.452 the moment a single auto-fail gate checks the header. A blended accuracy number cannot tell a firm *where* to trust a model; the gap between those two scores can. This report presents **TieOutBench**, a runnable suite of five expert-authored finance evaluations — quarterly earnings analysis, buffer-ETF diligence, DCF valuation, ETF creation/redemption basket reconciliation, and OTC confirmation matching — each decomposed into checkpoints and graded by point-weighted, tiered rubrics with auto-fail *gates* for the errors that quietly poison real work. Gold cases are cited to SEC filings and a published FpML confirmation; the deterministic core runs with one dependency and no API key. To our knowledge, as of this writing, the last two workflows are the only public LLM evaluations of capital-markets post-trade operations. We grade eight frontier models across three vendors on three of the evals, plus open-weight models on the other two. The headline is a negative result, consistently replicated across all eight models: the marquee *decision* gates never fired — no model settled a basket that did not reconcile or affirmed a confirmation that did not tie; on these cases, the errors live in the arithmetic underneath the decision. The three flagships tested land within 0.04 of one another on every case (single runs; adjacent scores are not a ranking); small tiers fail in different ways, and one flash-tier model breaks the cheap-means-undeployable pattern outright. The methodological contribution is a rule the harness itself taught us: cross-vendor comparison is invalid until the harness *proves each model equivalent room* — three vendors meter token budgets three different ways, and the differences produced four plausible, wrong scores before they were caught.

---

1. Introduction

The question a firm actually faces before letting a language model near analyst or back-office work is not *which model is best*. It is *when — and exactly where — can this model be trusted*, and its converse: where does it need a guardrail, a human check, or an outright ban. A blended accuracy score cannot answer that question. The errors that matter in finance are rarely diffuse: a misread scale header, a wrong fiscal period, a fabricated price, a basket settled short. Each is a single, local mistake that silently corrupts everything downstream while the surrounding prose reads fluently.

This report is organized around one motif — **looks right ≠ is right** — which recurs at three escalating levels:

1. **An answer can look right and be wrong.** On the suite's valuation case, the classic EV÷shares blunder (dividing enterprise value by shares while skipping the net-debt bridge) lands 2.6% from the market price — so it *looks* fair — while the correct method, under the case's stated assumptions, says the stock is roughly 20% overvalued. The wrong method produces the more plausible-looking number (§4.3).
2. **A failure can look right and be wrong.** When a model trips six auto-fail gates at once, it has not made six mistakes; its answer was cut off mid-sentence by a token budget. Reading that score as capability is an evaluation error, not a model error (§7).
3. **An evaluation can look right and be wrong.** Before we proved every vendor's endpoint gave its models equivalent room to answer, our cross-vendor grid contained four plausible, clean, wrong numbers — and nothing in the outputs announced the problem (§7).

The suite is named for the control that anchors it. "Tie out" is the fund-accounting discipline ported here to AI: *a number that does not tie out does not settle.* The score a TieOutBench rubric produces is not "how good did the answer sound" but "did the work tie out — and if not, exactly where."

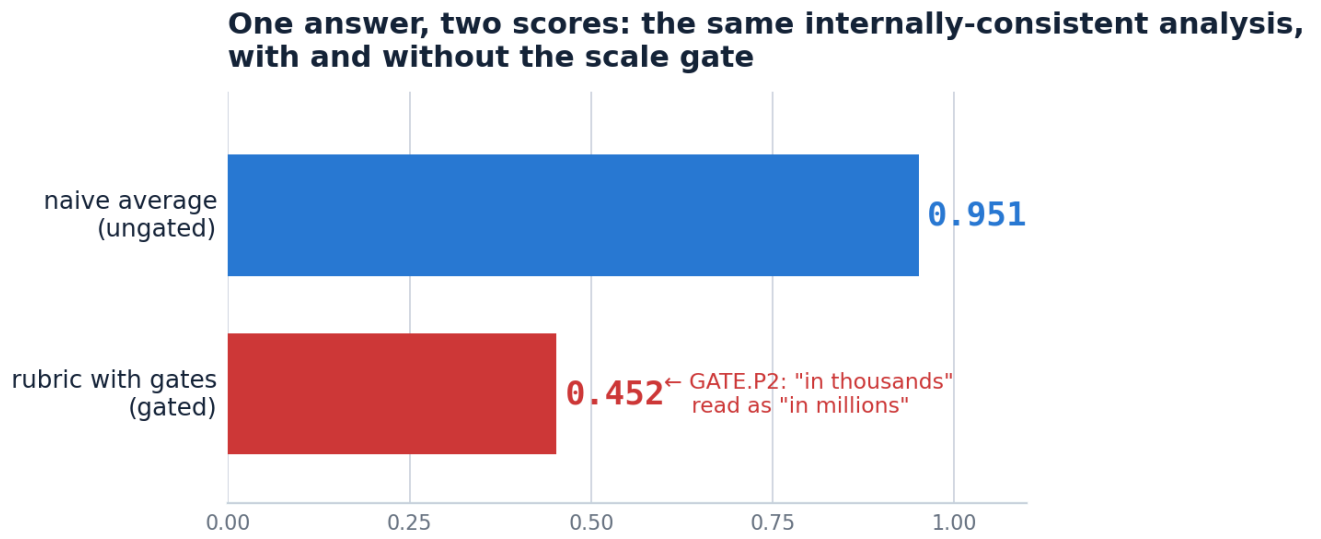


Figure 1 — the hook: one internally-consistent answer, scored naive (0.951) and gated (0.452); the gap is the finding.

The deployment stakes are current. In April 2026, US banking regulators issued SR 26-2, the first comprehensive rewrite of model-risk-management guidance since 2011 — and explicitly left generative and

agentic AI out of scope (footnote 3 of the attachment). Firms deploying LLMs into financial workflows must therefore define for themselves what "validated" means for this technology. This report is a small, concrete argument about what that takes; a companion note on SR 26-2 by the author is linked from evals.finance.

Contributions.

- **C1 — Five runnable, expert-authored, rubric-gated evaluations**, including what appear to be, as of this writing, the only public LLM evaluations of capital-markets post-trade operations (ETF creation/redemption reconciliation; OTC confirmation matching). Fourteen gold cases, every figure cited to a source document or explicitly labeled as constructed (§4).
- **C2 — A grading design that separates decision failures from arithmetic failures**: auto-fail gates in three tiers with severity-calibrated blast radius, a gated-vs-ungated score pair whose gap localizes the damage, and typed calibrated-refusal probes with answerable twins (§3).
- **C3 — A graded cross-vendor study**: eight frontier models from three vendors on three workflows, plus open-weight models on the other two, with every raw completion, parsed answer, and scored report published (§5–6).
- **C4 — A cross-vendor token-accounting methodology finding**: the "equivalent room" rule, with the three vendor behaviors documented and the before/after scores that show what skipping the proof costs (§7).
- **C5 — A failure taxonomy grounded in graded traces**, including what did *not* happen — the decision gates and fabrication gates that never fired on a frontier model (§8).

2. Related work and the 2026 landscape

Design lineage. Every ingredient of the method is public; the composition and the domain are the new part. From OpenAI's **HealthBench** the suite takes expert-authored rubric criteria graded by an LLM judge, with all-pass tracked alongside partial credit. From **FinanceBench** it takes the discipline that every answer ties to an evidence string in a source filing. From **FinQA / ConvFinQA / TAT-QA** it takes executed numeric answers graded to tolerance rather than string match. From the **Vals AI Finance Agent Benchmark** it takes checkpoint scoring of an end-to-end analyst task. From **FailSafeQA** it takes the insight that an eval which penalizes "I cannot verify that" trains confident guessing — calibrated refusal must earn credit. TieOutBench composes all five into one runnable harness and points it at workflows none of them cover.

The 2026 landscape. Table 1 places the suite among the prominent finance benchmarks as of mid-2026. The pattern is uniform: every one tests the front office or corporate accounting; none touches securities operations — the clearing, settling, and reconciling of what the front office trades.

Table 1 — where TieOutBench sits (all links verified at retrieval).

Benchmark	Built by	Format	Finance slice covered	Post-trade ops?
GDPval (2025)	OpenAI	static real-work tasks	analyst / advisor / sales occupations	✗
Finance Agent v2 (2026)	Vals AI	agentic filings QA	entry-level analyst research	✗
APEX / APEX-Agents (2025–26)	Mercor	expert-rubric agentic worlds	IB associate work (models, pitch materials)	✗
BigFinanceBench (2026)	Rogo + OpenAI	point-weighted-rubric QA	public-equity research (52 expert authors)	✗
FrontierFinance (2026)	Kensho / S&P / MIT	long-horizon computer use	3-statement / LBO / DCF / M&A model building	✗
FinBalance (2026)	academic	static reconciliation	corporate bookkeeping (invoices → journals)	✗ (accounting, not securities)
TieOutBench (2025–26)	one domain expert	rubric-gated, runnable, live-graded	analyst workflows **+ post-trade operations**	✓

GDPval is worth a sentence of precision, because it is the closest in spirit (real occupational work product, expert-graded): its finance-sector tasks are drawn from analyst, advisor, and sales-agent occupations. The gap this suite fills is therefore at the *occupation* level, not the task level — the asset-servicing back office that clears, settles, and reconciles what the front office trades simply does not appear.

A July 2026 meta-survey mapping 452 public financial-services benchmarks onto banking-industry domains (arXiv 2607.01740) reports the same picture from above: coverage concentrates in information-processing and analysis, with "a genuine gap in the public evaluation landscape for regulated domain tasks." Our coverage claim rides on that survey plus our own search, and is hedged accordingly: evals #4 and #5 *appear to be* the only public, runnable LLM evaluations of capital-markets post-trade workflows as of this writing.

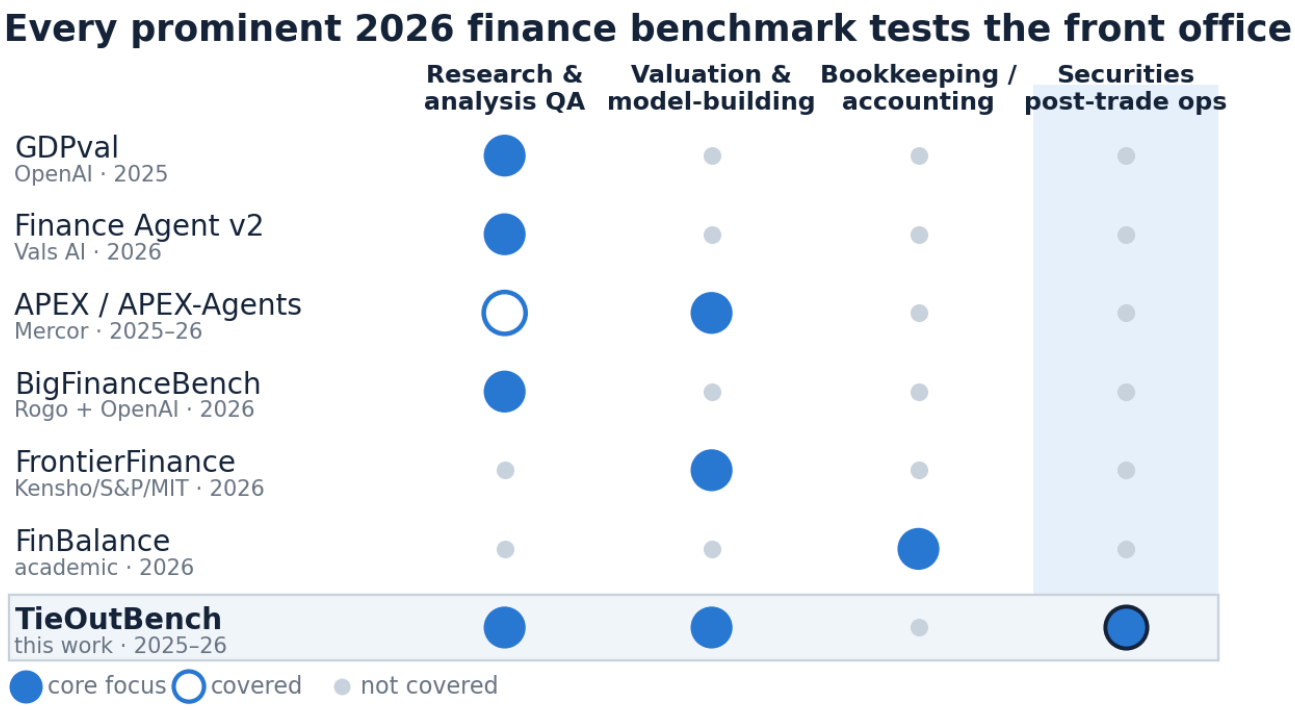


Figure 2 — the coverage map: seven benchmarks against four workflow families; among these benchmarks, the post-trade column is empty except for this suite.

3. Methodology

The suite is one scoring engine and five eval modules. This section describes the design choices in the order they bind: decomposition, criteria, gates, refusal, the judge, and the hardening discipline that keeps the graders honest.

3.1 Workflow decomposition into checkpoints

Each workflow is decomposed into checkpoints along the analyst's actual sequence — plan → extract → calculate → decide. Each checkpoint owns its inputs, a gold answer with an evidence citation (page or string in the source document), and success criteria. The decomposition is what buys *localization*: when a model fails, the report says which step failed, not merely that the final number is wrong. In the live runs this is the difference between "Haiku scored 0.692" and "Haiku's reported free cash flow does not equal its own build — a +$2.0B/yr offset at checkpoint C1 — and the error cascades through C3–C7 into a $138 fair value against a correct $228" (§6.3).

Table 2 — decomposition sizes.

Eval	Workflow	Checkpoints	Criteria
#1	Quarterly earnings analysis	17	109
#2	Defined-outcome (buffer) ETF diligence	18	110
#3	DCF valuation	18	107
#4	ETF creation/redemption reconciliation	8	32 (+5 gates)
#5	OTC confirmation matching	8	28 (+5 gates)

3.2 Point-weighted, tiered criteria

Criteria are atomic and typed — we call them *atoms* — into four tiers: **deterministic** (exact or tolerance-band recompute), **entailment** (does the cited evidence support the claim), **LLM-judge** (free-form synthesis quality, confined as described in §3.5), and **refusal** (§3.4). Checkpoints group into the four stages of Table 2's workflow column — plan, extract, calculate, decide — and stages carry explicit weights; everything that can be graded deterministically is.

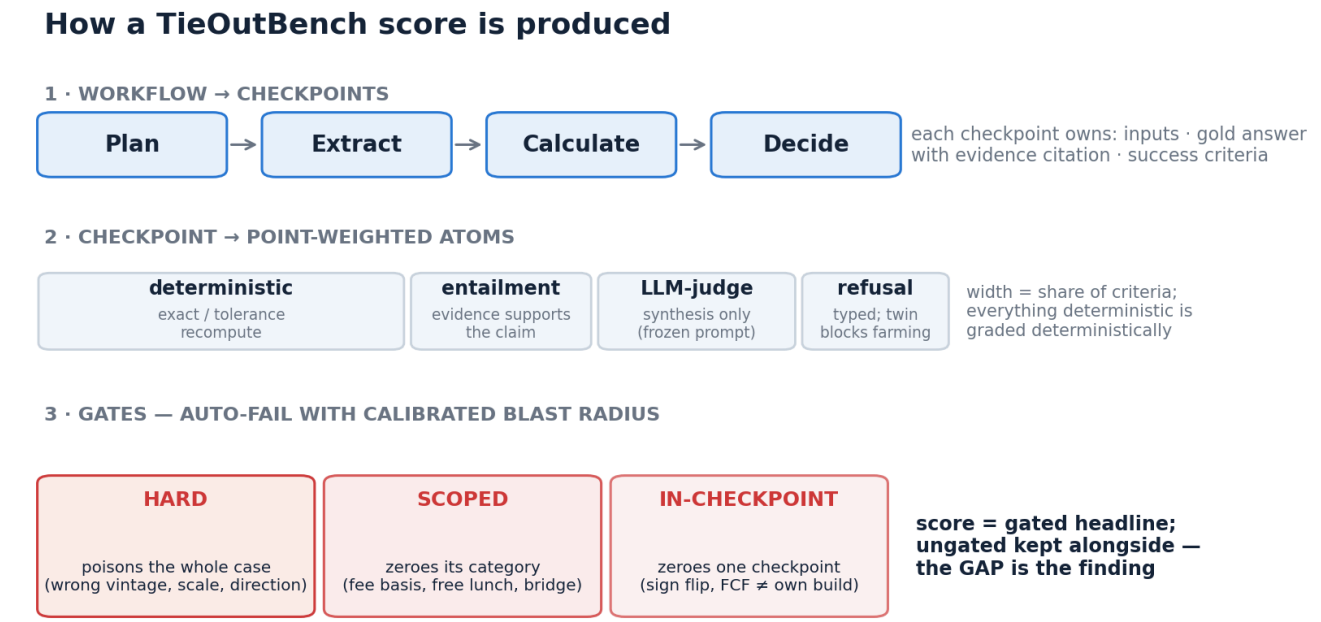
The design principle is to cap judge exposure *by construction*, and it is measurable. Eval #2 was built calculation-heavy (its stage weights put 42% of the score on the calculation checkpoints; of its 110 atoms, 76 are deterministic and only 19 touch the judge — none of them positive-credit before the synthesis stage; one judge-tier anti-gaming penalty sits at planning). Swapping the offline mock judge for a real LLM judge moves eval-#2 scores by 2.0–4.5 points out of 100. The same swap on eval #1 — an earlier, more judge-exposed design — moved its single-model score by 14.7 points (0.679 → 0.532). The headline of a well-built finance eval should not ride on judge permissiveness, and here it demonstrably does not.

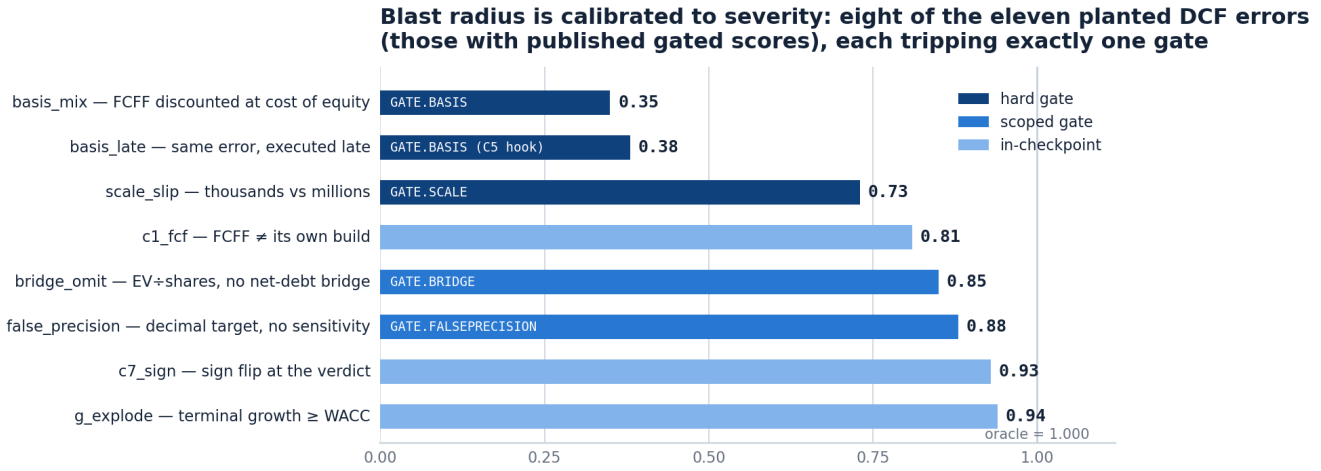
3.3 Gates and blast radius

Gates are the suite's signature mechanism: auto-fail conditions for the errors that quietly poison real work. A normal mistake costs its criterion's points; a gate collapses the score and flags exactly where and how badly. Every eval reports two numbers — the **gated** score (the headline) and the **ungated** naive average — and the gap between them is itself a diagnostic: a large gap means the model did fluent work around a poisoned core.

Gates come in three tiers, and the tier calibrates the **blast radius** to the severity of the underlying error:

- **Hard gates** poison the whole case: the wrong fund vintage, a scale misread, a creation-vs-redemption direction flip. Nothing downstream of these is trustworthy.
- **Scoped gates** zero a category: mixing gross and net in a fee chain kills the fee chain, not the extraction that preceded it.
- **In-checkpoint gates** zero one checkpoint: a sign flip at the verdict, a free-cash-flow figure that disagrees with its own components.





Two gates deserve their own paragraph because they port operational controls, not analytical ones. `GATE.RECON` (eval #4) fires when a model returns SETTLE for a creation basket whose residual is out of tolerance; `GATE.MATCH` (eval #5) fires when a model affirms a confirmation whose economic terms do not tie. These are the fund-accounting and affirmation-desk versions of the same rule — *tie out or stop* — and they encode a deliberate design inversion: **the highest-scoring failure is the catastrophic one.** A model that compares every field correctly, computes every number correctly, and then affirms the broken trade anyway scores 0.84 of naive credit — which is precisely why a naive average is dangerous: it rewards the memo that switched off the control. The gate exists to make that memo fail loudly. (On eval #2 the same inversion appears as `GATE.FREELUNCH`: a memo asserting downside protection with no cost-of-protection block can be 93% "right" and still be the memo that mis-sells the product; the gate's blast radius is deliberately small and its output is a headline *flag*, because the finding is the mis-sale, not the arithmetic.)

3.4 Calibrated refusal

Following FailSafeQA, the suite rewards a model that says "not determinable from this packet" — but refusal credit is easy to farm, so it is typed and twinned. Every refusal probe targets a figure that is genuinely absent or genuinely computable, and ships with an **answerable twin**: a structurally identical question that *can* be answered from the packet. Refuse-everything fails the twin; guess-everything fails the probe. On eval #2 the answer contract is fully typed — `{COMPUTED, value, derivation}` beside `{NOT_DISCLOSED, reason}` — with full credit for a correct computation *with its derivation*, refusal credit only for naming exactly which inputs are missing, zero for a confident underived number, and an imported issuer-website figure scored as an import, not a computation. Fabrication is gated separately: a model that invents a halted stock's closing price or a mark-to-market under a refusal label trips `GATE.FABRICATION` regardless of the label (§3.6). The live runs show why the typing matters — §6.4 documents a model that returns the *correct refusal label* while echoing the prompt's own instruction back as its "derivation" and getting the answerable twin wrong by $20,000.

3.5 LLM-judge confinement and calibration

The judge grades only the synthesis tier — free-form verdicts, calibrated bottom lines — under a frozen judge prompt committed to the repository; it awards no positive credit outside synthesis and never touches extraction, calculation, or any gate. Its calibration was tested directly on eval #2: the live judge's 28 free-form verdicts were independently hand-graded by the author from a worksheet showing each criterion, the model's actual section, and the gold reference. Result: **28/28 agreement** (100% raw agreement) on a sample with real signal (the judge had awarded only 57% of the verdicts) — with the caveats attached to the same sentence: n = 28, one model's answers, the worksheet displayed the judge's verdict (an anchoring channel), and the judge was the same model whose memos it graded, which is a stronger bias channel than family overlap. The claim is "no verdict was overturned on expert review," not "the judge is infallible"; a cross-family judge and a larger blind sample are named future work.

3.6 Anti-gaming hardening and the selftest discipline

Each eval passed an adversarial gaming review before commit, in which reviewer agents attempt to score points without doing the work. The review killed real exploit classes: placeholder cost-of-protection blocks buying off `GATE.FREELUNCH`; refusal credit fished by substring; a settlement-desk *go-ahead synonym* ("release for settlement," "book it") slipping past `GATE.MATCH`; a fabricated price asserted in prose under a refusal label slipping past `GATE.FABRICATION`. All are now regression cases.

Two invariants pin the graders permanently. First, a **schema round-trip selftest**: a schema-perfect answer must grade 1.000/AllPass on every gold case, which locks the live output contract to the graders. Second, **planted-error variants as regression tests**: each eval ships deliberately flawed answers (eleven on the DCF eval) that must each trip exactly their intended gate with the expected blast radius. `python -m harness selftest` asserts both.

3.7 The grader-bug rule

One empirical regularity held on four of the five evals, and we state it as a working rule of eval-building: **the first real model finds the grader bugs your self-tests were written around.** The first live batch on eval #1 surfaced grader-contract gaps its synthetic tests could not see; eval #2's first batch surfaced three (an enum graded so literally that the variant name `power_buffer` failed where `buffer` was meant; the fund's real cash-

sleeve row counted as a “fabricated fifth leg”; a gold-side labeling band the output contract had never stated); eval #3’s surfaced five (among them a tax rate reported in percentage points where a fraction was expected — a *false* gate fire on a model whose arithmetic was exact — and a false-precision predicate that compared the model’s sensitivity block against the *gold* valuation instead of the model’s *own*); eval #5’s surfaced two. The exception is eval #4, whose first live batch surfaced none — the eval built after the pattern had already taught us what self-tests miss. The rule then held again in a new form: the second DCF case’s first live batch surfaced three more contract gaps (rates reported as fractions mechanically firing two gates; correct answers as comma-formatted strings parsed as missing; a substantive snake_case answer failing a word-count anti-stub heuristic and firing the false-precision gate on style) — all fixed, all runs re-graded, and all eight of the first case’s committed reports re-grade unchanged (§6.3, and the full log in the case’s taxonomy). All three were parser-strictness false fires, so every correction moved a score up; the opposite failure direction — a gate that should fire but does not — is pinned by the planted-error battery, which still trips every intended gate after each fix. Every bug was fixed, every affected run re-graded, and the oracle still scores 1.000/AllPass with every planted variant still gating. The bug lists are published in the per-eval taxonomies. We consider publishing them part of the method: an eval that hides its grader corrections is reporting numbers from an instrument it has silently recalibrated.

4. The five evaluations

Table 3 summarizes; each subsection gives the task, the signature gate, the gold provenance, and the designed traps. Full decompositions and rubrics are in the repository (`workflow/`, `rubric/`).

Table 3 — the suite at a glance.

#	Workflow	Signature gate	Gold provenance	Cases
1	Earnings analysis	`GATE.P1` / `GATE.P2` (wrong period / scale misread)	three real 10-Q/10-K filing sets	3
2	Buffer-ETF diligence	`GATE.FREELUNCH`	real fund: prospectus + N-PORT strikes; 1 real + 2 labeled-constructed snapshots	3
3	DCF valuation	`GATE.BRIDGE` / `GATE.FALSEPRECISION`	two real 10-Ks (McDonald's FY2025, net debt; NVIDIA FY2026, net cash) + labeled oracle layers	2
4	Creation/redemption reconciliation	`GATE.RECON`	constructed, mechanics-faithful (PCFs are not public); real constituents	2
5	OTC confirmation matching	`GATE.MATCH`	real published FpML 5.10 message + constructed counterparty break	4

4.1 Eval #1 — quarterly earnings analysis

Digest a 10-Q/10-K and earnings release, extract and reconcile key figures, benchmark versus consensus, flag material changes. Seventeen checkpoints, 109 criteria, three gate tiers, and a FailSafeQA-style refusal probe with an answerable twin. Three gold cases from real filings (BlackRock Q3'25, Microsoft FQ2'26, Snowflake FQ2'26) exercise the designed traps: the "in thousands" scale header, fiscal-vs-calendar period pinning, a GAAP-loss/non-GAAP-profit divergence, and a genuine not-disclosed probe.

4.2 Eval #2 — defined-outcome (buffer) ETF diligence

An eval a generalist is unlikely to author. Given a prospectus, the fund's actual FLEX-option legs (exchange-listed options with customized strikes and expiries) from its N-PORT filing (the SEC's monthly portfolio-holdings report), and a dated market snapshot: recompute the marketed cap and buffer *from the option strikes* (the cap is literally the strike of the call sold to finance the put spread), compute what a **mid-period buyer actually gets** at today's NAV, verify the marketing claims, and price the protection. Eighteen checkpoints, 110 criteria. The signature `GATE.FREELUNCH` zeroes the synthesis of a memo that asserts downside protection with no cost block. The gold fund is real — Innovator U.S. Small Cap Power Buffer ETF – October (KOCT), whose four filed strikes reproduce its stated 17.18%/15% terms to within 0.002pp — under three snapshots: one real and cited (remaining cap +6.06% gross / +5.82% net for a mid-period buyer as of the snapshot date), and two

constructed states labeled hypothetical (near-cap post-rally; post-drawdown with the buffer 43.98% consumed and the remaining cap *enlarged* to +18.26%). The sibling-vintage distractor is built into the source data: the issuer runs twelve monthly series with near-identical names, and sibling N-PORTs land the same day with adjacent accession numbers — pinning the wrong one trips the hard `GATE.VINTAGE`. A disclosed detail from the case-construction process: the multi-agent adversarial pass that verified the gold cases caught a no-arbitrage violation in one of our own constructed snapshots before publication — the fix is logged in the case file — and a YAML boolean-encoding hazard whose prevention rule is now baked into the case templates.

4.3 Eval #3 — DCF valuation

Project unlevered free cash flow, discount at WACC, capitalize a terminal value, bridge enterprise value to equity, divide by diluted shares. Eighteen checkpoints, 107 criteria, over **two real-10-K cases built to mirror each other on the signature trap**; in both, every base line and bridge item is cited to the filing, and the forecast, WACC components, and terminal growth are a labeled oracle layer so the math is closed-form recomputable.

The **McDonald's FY2025** case carries heavy net debt, and its signature is the answer that looks right and is wrong: under the case's stated assumptions the correct method yields **$227.82** per share (≈20% below the case-date market price — an "overvalued" read), while the EV÷shares blunder that skips the net-debt bridge lands at **$279**, only 2.6% from the market price. The plausible number is the wrong one. The **NVIDIA FY2026** case inverts the trap: a net-**cash** balance sheet (cash and marketable securities exceed debt by $54.1B) where the same blunder *understates* fair value by only ~3% — numerically almost invisible, methodologically identical; the gate catches the method, not the magnitude. The NVIDIA case also plants the anti-pattern-matching pair: the sensitivity claim that is *true* on MCD ("a 50 bp discount-rate move shifts the value by more than a threshold") is *false* on NVDA, whose wider WACC–growth spread damps rate sensitivity (+6.2/−5.5% per 50 bp against MCD's +15/−12%) — and its gold bottom line is a reverse-DCF read (the ~$92 fair value against a ~$207 price measures how much growth beyond the five-year base case the market is paying for), explicitly not a target. The consistency spine on both cases is `GATE.BASIS` (unlevered cash flows must meet WACC must meet the bridge — including a C5 hook that back-solves the discount rate from the model's own PVs, so executing the error "late" does not evade the gate). The calibration signature is `GATE.FALSEPRECISION`: a decimal-precise target on valuations that are 70–81% terminal value auto-fails unless real sensitivity ranges accompany it.

4.4 Eval #4 — ETF creation/redemption basket reconciliation

The custodian back-office core, authored from having run the change-management side of an institutional ETF servicing platform. Given an Authorized Participant's tendered creation basket, the published PCF (portfolio composition file — the fund's daily creation-basket recipe), and the NAV-based creation value: reconcile line-by-line, value the basket and cash-in-lieu, compute the tie-out — and **settle only if it ties**. The gold break case is a creation short exactly **$13,320** on a $3.075M order: every in-kind share line matches; only the cash-in-lieu plug for a trading-halted name was delivered at a stale prior-close price ($105.00 against the struck $112.40). Gold answer: DO_NOT_SETTLE, localized to that line. A clean-settle counterweight case catches the over-cautious mirror — crying break on a basket that ties. Provenance is disclosed prominently: PCFs are NSCC-disseminated, not public filings, so this is the suite's one constructed case family — mechanics-faithful, built over real constituent securities at representative prices, with fund, order, and break illustrative.

4.5 Eval #5 — OTC derivative confirmation matching

The derivatives sibling of #4. Two counterparties book their sides of an interest-rate swap and send confirmations; the affirmation desk matches economic terms field-by-field and **affirms only if they tie**. The gold "our side" is the real, publicly downloadable FpML 5.10 sample confirmation (FpML — Financial products Markup Language, the OTC-derivatives messaging standard; `ird-ex01-vanilla-swap.xml`, fpml.org): EUR 50MM notional, receive fixed 6.00% 30E/360 versus 6-month floating — cited, not constructed. The counterparty confirmation is constructed to carry one material break: a fixed rate of 6.05% where ours says 6.00% (~5 bp, ≈EUR 25,000/yr on EUR 50MM), with every other term tying and the two parties' trade IDs differing *by design* — the materiality foil that separates "different" from "a break." Gold answer: MISMATCHED, do not affirm, localized to the fixed rate. A clean-match counterweight catches false breaks. Because funds are among the biggest swap users, a second case pair runs the same control on a swap held *inside* an ETF (a constructed, convention-faithful bond-ETF interest-rate-swap pair); it ships with gold cases and taxonomy but has not been live-run, and no result below draws on it.

5. Experimental setup

Subjects. Eight frontier models — current vendor-hosted API models, spanning flagship to small tiers — across three vendors, run on evals #3–#5: Anthropic's Claude Opus 4.8, Claude Sonnet 4.6, and Claude Haiku 4.5; OpenAI's GPT-5.6-sol, GPT-5.5, GPT-5.4, and GPT-5.4-mini; Google's Gemini 3.6 Flash. Open-weight models (publicly downloadable weights, run locally at zero API cost) cover the other two evals: qwen3.6-27b (a reasoning model) and qwen2.5-72b-instruct (non-reasoning) on eval #2; qwen2.5-32b-instruct on eval #1. Google is represented only by a flash-tier model because its pro line was a generation behind at run time; the tier asymmetry is noted where it matters.

Protocol. All frontier runs use each vendor's OpenAI-compatible endpoint (`api.anthropic.com`, `api.openai.com`, `generativelanguage.googleapis.com/v1beta/openai`) driven by the same harness path, with the deterministic core plus the offline judge (the judge tier is a small share of these calculation-heavy evals; §3.2). Completion budgets: 8,000 tokens on the short cases; on the long DCF cases, Claude ran at the live path's 12,000-token floor on MCD (raised from 8,000 after a truncation — §7 — with Sonnet's committed run completed at 16,000) and 16,000 on NVDA, and GPT and Gemini ran at 32,000 for the reasons §7 documents (Gemini at 32,000 throughout). Equivalent room was verified from each vendor's own usage accounting: Anthropic's compatibility endpoint meters output only (thinking is not charged against the budget), and every committed completion closed under its cap. One prompt disclosure: the NVDA runs used a case-hardened system prompt that states two conventions the MCD-era prompt did not (the net-debt/cash-like netting rule and the sensitivity grid's 3×3 geometry — added after a pre-ship audit showed a graded convention was never taught); cross-case comparisons touching those behaviors are hedged where they appear. **One run per model per case.** That is enough to expose a broken harness and enough to say a gate did or did not fire; it is not enough to rank models by hundredths. Small gaps below should be read as "indistinguishable here," not as a ranking.

6. Results

Table 4 — gated scores, eight frontier models × six gold cases (gate names abbreviated; each is `GATE.<NAME>`).

Model	Tier	#3 DCF (MCD)	#3 DCF (NVDA)	#4 recon (break)	#4 clean	#5 confirm (break)	#5 clean
Claude Opus 4.8	flagship	**0.965**	0.974	**0.983**	0.983	**0.980**	0.980
Claude Sonnet 4.6	mid	0.955	0.973	0.943	0.983	0.933	0.980
Claude Haiku 4.5	small	0.692 · `C1FCF`	0.799	0.496 · `SCALE`	0.983	0.933	0.980
GPT-5.6-sol	flagship	0.951	0.970	0.943	0.973	**0.980**	0.980
GPT-5.5	flagship (prev.)	0.953	0.970	0.943	0.983	**0.980**	0.980
GPT-5.4	mid	0.903	0.883	**0.983**	0.973	**0.980**	0.980
GPT-5.4-mini	small	0.631 · `FALSEPRECISION`	0.650 · `BRIDGE`	0.714	0.983	0.933	0.980
Gemini 3.6 Flash	small/fast	0.922	0.970	**0.983**	0.973	**0.980**	0.980

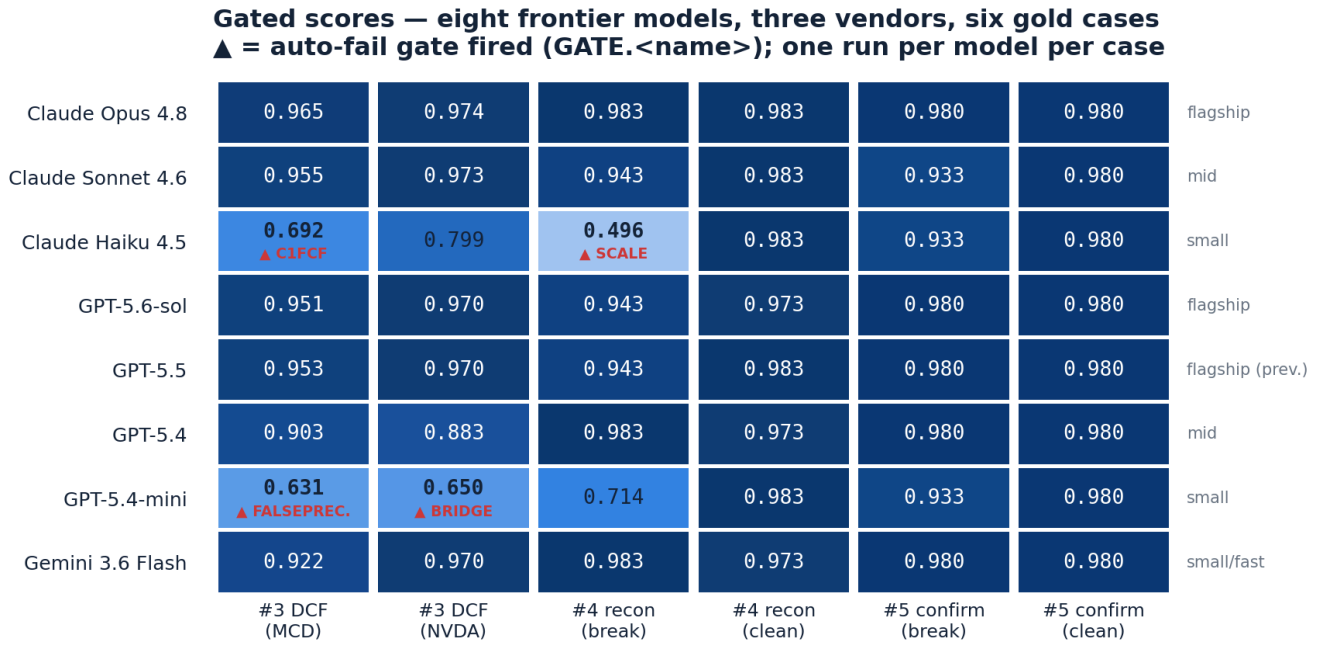


Figure 5 — the grid as a heatmap; gate glyphs on the four fired cells.

6.1 The decision gates never fire

Across all eight models and both back-office workflows: nobody settled the short basket (`GATE.RECON` never fired), nobody affirmed the broken trade (`GATE.MATCH` never fired), and nobody cried break or mismatch on the clean counterweight cases (no false positives either). When this held for three models from one vendor we called it a promising negative; at eight models across three vendors it reads as a property of the task: **on these cases, frontier models get the stop-or-go call right, and lose their points on the arithmetic underneath it.** It is the most consistently replicated result in the suite, and we state it plainly because a benchmark that reports only its hits is not one.

What separated models on the break cases was quantification, localized by checkpoint. On the confirmation break, all eight returned MISMATCHED, all treated the differing trade IDs as expected — and the basis-point conversion split the field: Opus 4.8 sized the break correctly (~5 bp ≈ EUR 25,000/yr); Sonnet 4.6 called it "0.5 bp" (10× low); Haiku 4.5 called it "50 bp" (10× high — and its EUR 2.5M lifetime-impact figure is 20× the correct ≈EUR 125,000 over the remaining term). Same trap, opposite directions, one checkpoint (`C3.impact`).

6.2 The flagships have converged — and parts of the eval have ceilinged

Opus 4.8, GPT-5.6-sol, and GPT-5.5 sit within 0.04 of each other on every case and within 0.015 on five of the six. On the confirmation-break case five of the eight models tie at exactly 0.980 — and on the NVDA DCF case five models (the three flagships, Sonnet, and Gemini's flash tier) land within **0.004** of one another, gate-free, at the case-gold fair value. Stated plainly: at this difficulty, these evals no longer separate the frontier on those tasks — a finding about the eval as much as the models. The honest response is to harden the task (graduated break materiality, noisier documents), not to pretend the ranking means something; §10 lists the plan.

6.3 Small tiers fail differently — and one breaks the pattern

The two small tiers that fail do so in different ways:

- **Haiku 4.5 fails on arithmetic.** On the reconciliation break it overstates the in-kind value by exactly $200,000, flipping the residual from −$13,320 to +$186,680 — it concludes the basket is *over*-delivered when it is short. It still refuses to settle (the right call), and `GATE.SCALE` pins the wrong diagnosis to the valuation step. On the DCF its reported FCFF carries a +$2.0B/yr offset from its own components (`GATE.C1FCF`), cascading to a $138 fair value.
- **GPT-5.4-mini fails on calibration.** It states a fair value of $200.20 — to the cent — while leaving the growth-rate sensitivity *null* on a valuation that is 79.5% terminal value. `GATE.FALSEPRECISION` exists for exactly that memo. Because a gate firing only on a new vendor is what a grader bug looks like, the firing was verified against the models that cleared it: GPT-5.5 and GPT-5.4 return complete sensitivity ranges; the mini returns `{g_low: null, base: 200.2, g_high: null}`. It fired for the reason it exists. (Its non-gated 0.714 on the reconciliation break traces to the twin miss and related arithmetic slips — §6.4.)
- **Gemini 3.6 Flash breaks the pattern outright.** A flash-tier model fires no gate anywhere, ties Opus for the best reconciliation score on the board (0.983 — correct DO_NOT_SETTLE, right offending line, right root cause, residual to the dollar), lands its MCD fair value at **$227.81 against a gold of $227.82** — one cent — on the closed-form recompute, with genuine sensitivity ranges on both drivers, and repeats the performance on the NVDA case (0.970, within 0.004 of the flagships).

- **The mirror trap caught its one victim in the subtler form.** On the NVDA case, where the naive EV÷shares blunder is nearly invisible (−3.4% on a net-cash balance sheet), GPT-5.4-mini fired `GATE.BRIDGE` a different way: it added the $54.1B net cash and **dropped the $22.3B non-op add it had itself extracted** — its equity is exactly `EV + net cash`, contradicting its own stated bridge formula (which includes the non-op add). The gate polices the method, so the magnitude of the miss never mattered. (Its remaining NVDA damage — a PV build that does not reconcile to its own year-by-year rows, and a terminal value ~7% off the Gordon value of its own inputs — is the same internal-inconsistency signature as Haiku's MCD FCFF, one tier over.)

Two data points said small models are undeployable on this work; the third shows that was never a rule of price tiers. **Price tier and trustworthiness are different axes.** You cannot infer one from the other; you have to measure.

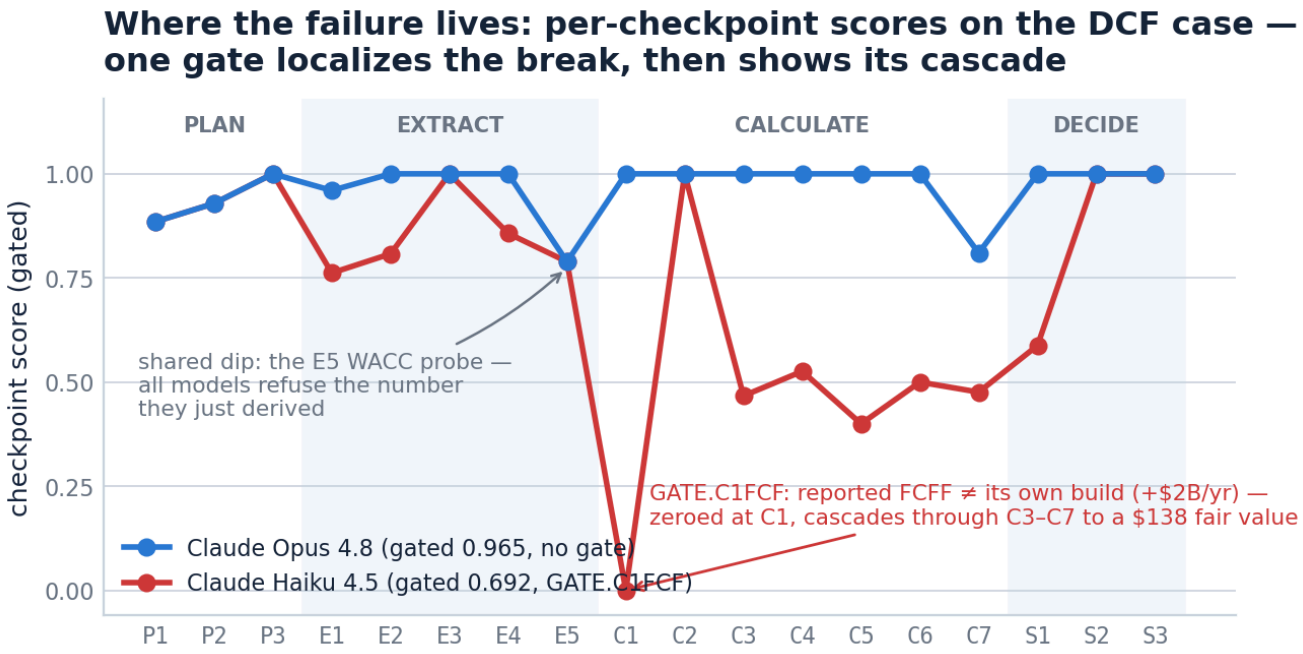


Figure 6 — per-checkpoint vectors for Opus vs. Haiku on the DCF case: the C1 gate, the C3–C7 cascade, and the shared E5 dip.

6.4 The refusal probes: a label is not calibration

On the reconciliation refusal probe (a figure genuinely not determinable from the documents), all eight models refused correctly at the label level — right type, no invented number, and no fabrication gate fired on any frontier model anywhere in the study. But the typed contract (§3.4) catches what label-grading cannot: GPT-5.4-mini's stated "derivation" is the prompt's own instruction echoed back nearly verbatim, and its *answerable twin* comes back at −$33,320 against the correct −$13,320. Grade only the refusal label and that model looks perfectly calibrated.

The DCF eval's WACC probe shows the other face of the same point. On the MCD case, asked separately "what is the WACC per the 10-K?", all eight models return the NOT_DISCLOSED label with a null value — correctly, since no 10-K states a WACC — and most restate the figure they had derived from the supplied components two checkpoints earlier (≈7.15%) inside their derivation text while declining to claim it *as the answer*. The typed

contract is doing exactly its job: separating "not in the document" (true) from "not computable" (false), a distinction label-only grading cannot see.

The NVDA case then produced the contract's sharpest catch — a failure class we had not designed for: **derive the right number, report a different one.** Haiku 4.5's probe derivation builds `0.98×12.8 + 0.02×3.887` from the correct components — which is 12.62 — and then asserts, in its own words, "Rounded to 12.51%": an impossible rounding, presented as one, and a figure it then uses consistently through the whole valuation (no gate fires, because its chain is internally consistent at the wrong rate; the damage is localized to the rate checkpoint and the drift it causes). GPT-5.4's derivation text literally ends "= 12.62174%" while its answer field says **12.31** — contradicting both its own derivation and its own correctly-reported WACC checkpoint; no rounding or weighting path in its own answer produces 12.31. Two models, two vendors, the same probe. Grade the label alone and both look calibrated; grade the value against the model's own derivation and neither is.

6.5 Open-weight results on evals #1–#2

Evals #1–#2 have so far been run against local open-weight models; the frontier grid above does not cover them, and these results carry their own caveat — both eval-#2 subjects are Qwen-lineage models, so the shared-trap finding awaits cross-family replication.

On eval #2, the 27B reasoning model beat the 72B non-reasoning model on every case (0.732 / 0.708 / 0.702 versus 0.638 / 0.584 / 0.577, the 72B tripping `GATE.C6DIR` on two cases — including a full remaining-outcome state inversion on the post-drawdown case) despite a 2.7× size disadvantage. Both models nailed extraction (0.86–0.91) and both fell into the same payoff-grid trap — filling the reconstruction with the prospectus's idealized percentage convention while (in the 27B's case) *simultaneously producing the exactly correct per-unit dollar values elsewhere*: models follow the document instead of the asked-for reconstruction. In an end-to-end probe with three same-day sibling N-PORTs and a stale mid-period 497K (summary-prospectus filing) in the packet, the 27B pinned the correct vintage — the hard gate did not fire — paying a modest retrieval cost (0.732 → 0.651 gated).

On eval #1, a 32B open-weight model scored 0.532 with a real LLM judge (0.679 with the permissive mock). The split is the finding: a competent *extractor* (segments/shares 0.91) but a weak *analyst* — its material-changes synthesis listed surface metrics and missed a +1,118 bp GAAP-margin swing; it correctly refused the not-disclosed probe. Feeding it the 10-Q balance-sheet excerpt barely moved the score (0.679 → 0.671): its ratio failures were computation failures, not data starvation.

7. The harness is half the finding: cross-vendor token accounting

This section records the study's methodological contribution — the part that would survive even if every score in §6 were superseded by next quarter's models. Adding a second vendor to a working harness should be a configuration change. It was not: the harness broke three times, and each breakage produced output that *looked like a result*.

Breakage 1 — loud and harmless. OpenAI's GPT-5 family rejects `max_tokens`, the field every other OpenAI-compatible endpoint accepts, requiring `max_completion_tokens` instead. Every call failed at the HTTP layer with a 400 until the client learned to flip the field and retry. Loud failures are the good kind.

Breakage 2 — quiet and expensive. Reasoning models spend tokens thinking before they answer, and the endpoints disagree about who pays. On the Anthropic compatibility endpoint, thinking does not count against the completion budget. On OpenAI's, it does. The suite's 8,000-token budget — comfortable for every prior run but one — was therefore silently insufficient for GPT models on the one long case (the DCF, with a ~4.4k-token prompt). The models thought, spent the budget thinking, and had little or nothing left to answer with. Table 5's before/after shows what that did.

Table 5 — the same four models, the same DCF case, before and after equivalent room.

Model	@ 8k budget (starved)	@ 32k budget (fair)
GPT-5.5	*empty completion*	**0.953** — no gates
GPT-5.6-sol	0.322 · six gates fired	**0.951** — no gates
GPT-5.4	0.618 · `FALSEPRECISION`	**0.903** — gate cleared
GPT-5.4-mini	0.745	0.631 · `FALSEPRECISION` (genuine — §6.3)

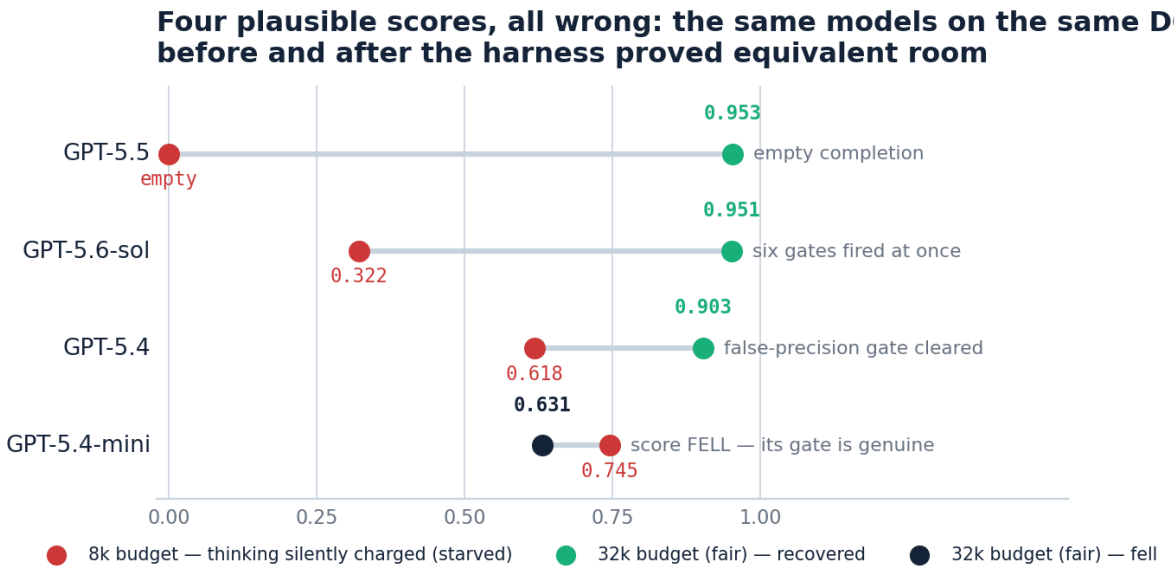


Figure 7 — Table 5 as a dumbbell chart; note the one score that moved down.

Sit with the shape of that error: nothing crashed, nothing logged a warning, and the four starved numbers told a clean, publishable story — *the new vendor is bad at valuation, and its flagship is catastrophically bad.* Six gates firing at once is not a model making six mistakes; it is a sentence getting cut off. Every one of those numbers measured configuration, not capability. And the fourth moved the *other way* — the mini's false-precision gate is genuine, which is exactly why the starved table is so treacherous: three of its four numbers understate the models, and one flatters it.

Breakage 3 — the quietest. Google's compatibility endpoint also charges thinking against the answer budget — and does not report it in any field of the compat `usage` object. A trivial smoke test (8-token prompt) returned `completion_tokens: 3` and `total_tokens: 124`: 113 thinking tokens, spent and invisible in the field a harness would naturally read. Size budgets from `completion_tokens` and you starve Gemini on every long case, with low scores as the only symptom. There is no version of that bug that is found by staring at results.

Table 6 — three vendors, three token-accounting behaviors.

Vendor	Budget field	Thinking counts against it?	Thinking visible in `usage`?
Anthropic	`max_tokens`	no	—
OpenAI	`max_completion_tokens` (rejects `max_tokens`)	**yes**	reported
Google	`max_tokens`	**yes**	**no** — only inside `total_tokens`

Table 6 describes the OpenAI-compatibility layers — which is what a cross-vendor harness talks to; each vendor's native API meters and reports differently again.

The rule this produces: **you cannot compare models across vendors until you have proven the harness gives each of them equivalent room to work.** It sounds obvious written down. It is not obvious in practice, because a starved model does not return an error — it returns a worse answer, and a worse answer is exactly what a benchmarker is looking for. We had, in fact, seen this failure before in single-vendor form: the same 8k cap had truncated one Claude run on this case, which read as a quirk of that run and led to the 12k floor of §5. One vendor's truncation read as a quirk; a second vendor's made it a rule. A benchmark that has met only one family of models has not been tested; it has been rehearsed.

This closes the paper's motif on itself. The suite exists because a model's *answer* can look right and be wrong, so it grades every step instead of the final number. An *evaluation* can look right and be wrong the same way — and the fix is the same discipline applied one level up: prove the instrument before reading the measurement. Any firm running an internal model bake-off through the vendors' compatibility endpoints is exposed to this bug class today.

8. Failure taxonomy

Every failure below is grounded in a graded, committed trace; trace files for every row are under `outputs/<eval>-live/` in the repository. The taxonomy of *absences* at the end is equally load-bearing.

Table 7 — failure taxonomy across all graded runs.

Class	Instance (model, eval)	Caught by
Unit/scale error	"in thousands" read as millions (designed trap, #1); in-kind basket overstated by exactly $200k, residual sign flipped (Haiku, #4); notional misread (variant, #5)	`GATE.P2` / `GATE.SCALE`
Magnitude/conversion	5 bp break sized "0.5 bp" (Sonnet, 10× low) and "50 bp" (Haiku, 10× high; lifetime figure 20×) — opposite directions, same checkpoint (#5)	`C3.impact` criteria
Internal inconsistency	reported FCFF ≠ its own components, +$2.0B/yr (Haiku, #3)	`GATE.C1FCF`
State/semantic inversion	buffer labeled "intact" at 43.98% consumed; a correctly computed +0.91 gap read as a shortfall; over- vs. under-delivered flipped (qwen subjects, #2; Haiku, #4)	`GATE.C6DIR` / label atoms
False precision	$200.20 to the cent, null growth sensitivity, 79.5% terminal value (GPT-5.4-mini, #3)	`GATE.FALSEPRECISION`
Echo-stub refusal	correct NOT_DISCLOSED label; "derivation" = the prompt's instruction echoed; answerable twin off $20k (GPT-5.4-mini, #4)	typed refusal + twin
Derived-vs-reported divergence	the derivation builds 12.62 from correct components; the answer field says 12.51 (Haiku, used consistently downstream) or 12.31 (GPT-5.4, contradicting its own C2) — #3 NVDA	typed probe keyed to the model's own build
Half-bridge omission	net cash added, the extracted non-op assets dropped — equity = EV + net cash exactly (GPT-5.4-mini, #3 NVDA)	`GATE.BRIDGE` self-consistency arm
Document-following over task-following	payoff grid filled with the prospectus's % convention while the correct per-unit dollars appear elsewhere in the same answer (both qwen subjects, #2); the WACC probe answered NOT_DISCLOSED/null with the derived figure restated only in the justification (#3)	checkpoint criteria / typed contract

What did not appear. No frontier model fired a fabrication gate anywhere in the study, and — §6.1 — no decision gate fired and no false break appeared. The suite's most dangerous designed failures were never exhibited by a frontier model in these runs, stated with its bounds: one run per model per case, on these cases, at this difficulty.

9. Limitations

In order of importance:

1. **One run per model per case.** No variance estimates; adjacent scores are not a ranking. The binary findings (a gate fired; a gate never fired) are the robust ones.
2. **Single author.** One expert authored the workflows, rubrics, and gold answers, and hand-graded the judge calibration — whose worksheet displayed the judge's verdict (an anchoring channel), and whose n = 28 covers one model's answers. Blind multi-expert grading is the correction.
3. **Constructed-case disclosures.** Eval #4's cases are constructed (PCFs are not public); eval #2's post-rally and post-drawdown snapshots are constructed and labeled (its anchor snapshot is real and cited); eval #5's counterparty side is constructed around a real FpML message, and its ETF-swap case pair has no live run. No result above draws on the un-run pair.
4. **Coverage asymmetry.** Evals #1–#2 have never been frontier-run; evals #3–#5 never open-weight-run; Google is represented by a flash-tier model only. The grid fills as runs accumulate; the repository leaderboard is the living version of Table 4.
5. **Ceiling effects.** The clean cases and the confirmation-break case no longer separate flagship models (§6.2). Hardening is scheduled, and until then those columns measure "passes the control," not "which model is better."
6. **Judge scope.** The frontier grid used the offline judge; the live-judge calibration exists for eval #2 only. The deterministic core — every gate, every calculation — is unaffected by judge choice, which is the design's point (§3.2).
7. **Contamination.** The source filings and the FpML sample are public and may be in training data. The graded work is derivation on supplied documents (with constructed snapshots and oracle layers the model could not have seen), not recall — but the exposure is real and is why constructed-but-labeled cases have a place in the suite alongside cited ones.

10. Future work

- **Eval #6 — agentic corporate-actions processing.** The back-office domain with a documented ~$58B/yr industry cost and, as of mid-2026, no independently published accuracy metrics. Its gates will be designed as verifiable *reward functions*, so the same build yields an RL environment, not just a benchmark.
- **Open-weight runs across the full grid** (Llama, Gemma, Mistral, DeepSeek-family) — filling the coverage asymmetry and testing the shared-trap findings across families.
- **Hardening the ceilinged cases**: graduated break materiality, noisier source documents, distractor density — until the frontier separates again.
- **A cross-family judge and blind multi-expert calibration**, replacing the single-author channel.
- **A synthetic-data flywheel**: generate convention-faithful FpML confirmations at scale, fine-tune small models on the workflows, and grade them with the suite's own gates.

11. Reproducibility

Everything is MIT-licensed and runnable with one dependency (`pyyaml`) and no API key for the deterministic core:

```
pip install -r requirements.txt
python -m harness demo        # the Figure 1 pair: one answer, gated vs ungated
python -m harness suite       # score all 14 gold cases across all five evals
python -m harness selftest    # gate-tier + schema round-trip invariants
python -m harness run --case irs-confirm-2026 --model affirm_match   # watch GATE.MATCH fire
```

Live runs take any OpenAI-compatible endpoint (`--model live --endpoint <url> --model-id <id>`). Every number in this report traces to a committed artifact — raw completions, parsed answers, scored reports, and per-eval failure taxonomies under `outputs/` — or to the consolidated, committed `LEADERBOARD.md`; rubrics and their validator are under `rubric/`, and the case files with their provenance blocks under `cases/`.

Author note

The author ran the change-management side of an institutional ETF servicing platform (creation/redemption lifecycle, derivatives processing) and has built AI-powered finance tools independently since 2022. This suite is single-author by circumstance, not by philosophy — §9 (item 2) and §10 describe where independent experts enter.

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 multi-agent adversarial review passes described in §3.6 and the per-case verification logs. All workflow designs, rubric criteria, gold-case judgments, and editorial decisions are the author's; every factual claim was verified against committed artifacts (source filings, raw model completions, 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, the safeguard against circularity is structural rather than procedural: gold answers derive from primary-source filings and closed-form arithmetic committed in the repository, grading is predominantly deterministic (§3.2), and every reported number is reproducible by any reader from the published artifacts — verification does not depend on trusting either the author or the tools that assisted him.

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. Chen, Z., et al. *ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering.* arXiv:2210.03849, 2022.
5. Zhu, F., et al. *TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance.* arXiv:2105.07624, 2021.
6. Vals AI. *Finance Agent v2* (proprietary benchmark, updated August 2026). vals.ai/benchmarks/fabv2.
7. Kamble, K., Russak, M., et al. *Expect the Unexpected: FailSafe Long Context QA for Finance.* arXiv:2502.06329, 2025.
8. Patwardhan, T., et al. (OpenAI). *GDPval: Evaluating AI Model Performance on Real-World Economically Valuable Tasks.* arXiv:2510.04374, 2025.
9. Mercor. *APEX — the AI Productivity Index: benchmarks across investment banking, corporate law, management consulting, and medicine.* mercor.com/apex, 2025–26.
10. Wang, A., et al. (Rogo & OpenAI). *BigFinanceBench: A Workflow-Grounded Benchmark for Financial-Research Agents.* arXiv:2606.03829, 2026.
11. Krumdick, M., et al. (Kensho / S&P Global / MIT). *FrontierFinance: A Long-Horizon Computer-Use Benchmark of Real-World Financial Tasks.* arXiv:2604.05912, 2026.
12. Tumpati, S., et al. *FinBalance: A Multi-Document Accounting Reconciliation Benchmark.* arXiv:2606.15949, 2026.
13. Hudson, B. *Meta-Benchmarks for Financial-Services LLM Evaluation.* arXiv:2607.01740, 2026.
14. FpML (ISDA). *FpML 5.10 Recommendation* (February 13, 2018) — sample confirmation `ird-ex01-vanilla-swap.xml`. fpml.org.
15. Board of Governors of the Federal Reserve System, with OCC and FDIC. *SR 26-2: Revised Guidance on Model Risk Management* (April 17, 2026), attachment SR2602a1.pdf footnote 3 (generative/agentic AI exclusion).
16. SEC EDGAR filings: BlackRock Q3’25 10-Q (accession 0001193125-25-267043); Microsoft FQ2’26 10-Q (0001193125-26-027207); Snowflake FQ2’26 10-Q (0001640147-25-000187); McDonald’s FY2025 10-K (0000063908-26-000035); Innovator ETFs Trust — KOCT 497K (0001213900-25-094547) and NPORT-P (0000894189-26-009590), with sibling-series N-PORTs (0000894189-26-009589, 0000894189-26-009591) and a mid-period 497K (0001213900-26-021495) as packaged distractors.