HEDGE-BENCH: CAN AGENTS REASON THEIR WAY TO BETTER PREDICTIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern AI agents are increasingly capable of mechanical financial-analysis tasks such as retrieving documents and updating spreadsheets. A harder challenge is forecasting an unknowable future from incomplete information. Public markets provide a natural testbed: analysts routinely predict discrete events from point-in-time information. We present `Hedge-Bench`, a benchmark of 92 real-world tasks derived from hedge fund analysts' reasoning traces and paired with subsequently resolved prediction questions. Agents reason from frozen information snapshots, and their forecasts are evaluated against realized outcomes. Across 11 frontier models and three trials per task, joint success rates range from 2.54 to 31.52%, requiring both a correct forecast and over 60% research coverage. Using GRPO with *turn-level credit assignment*, we train Muse-Glimmer on 165 environments and increase coverage on 22 held-out tasks from 17.17% to 33.36% at low reasoning effort and from 30.19% to 47.02% at extra-high effort. The trained model at low effort also exceeds the base model's extra-high-effort coverage while using 47.9% fewer accepted trajectory tokens. We also observe evidence of generalization: held-out forecasting accuracy at high effort improves by 10.06 percentage points, while external evaluations improve by 3.94 points on GDPval, 2.29 on GDP.pdf, 7.07 on Harvey LAB, and 4.96 on Frontier Finance under their respective metrics.

1 INTRODUCTION

Finance benchmarks evaluate agents on conducting the mechanical work within research workflows (Bigeard et al., 2025; Wang et al., 2026; Du et al., 2025), while forecasting benchmarks test predictions against subsequent events (Karger et al., 2025; Goel et al., 2026). Yet a correct prediction may lack sound reasoning, and a rigorous piece of analysis may reach the wrong conclusion. `Hedge-Bench` links both skills within the same task to investigate a central question: If we reward the kinds of reasoning demonstrated by expert analysts, could agents eventually surpass the best human analysts?

Making informed predictions about the future is core to the most consequential decisions institutions must make. "How should DraftKings account for prediction markets when forecasting revenue and weighing capital allocation over the next 24 months?" "If SpaceX's Starlink will launch 30,000+ satellites by 2030 with best-in-class unit economics, should Iridium Communications as a smaller incumbent try to compete with a like-for-like replacement?" "If software is becoming easier to reproduce, which companies become more attractive to serial acquirers like Bending Spoons?" These kinds of questions require reasoning from incomplete evidence about events that have yet to unfold.

In this paper, we introduce `Hedge-Bench`, a testing suite that evaluates whether agents can learn to reason within the finance domain and subsequently predict real-world outcomes. It pairs reasoning traces taken from real-world conversations between investment professionals with the broader predictions discussed, which have since been resolved in the real world. These research tasks include reading between the lines of what was said and explicitly not said in documents, computing valuation multiples, assessing normalized earnings power, identifying possible inflections, benchmarking against peers, and synthesizing opposing data points into a coherent investment view. We also introduce the Hedge-Bench-Public dataset, a set of 92 challenging tasks across 23 companies requiring extensive domain knowledge, hours of directed research work and multi-step problem solving that represent what the highest paid finance professionals actually spend time doing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

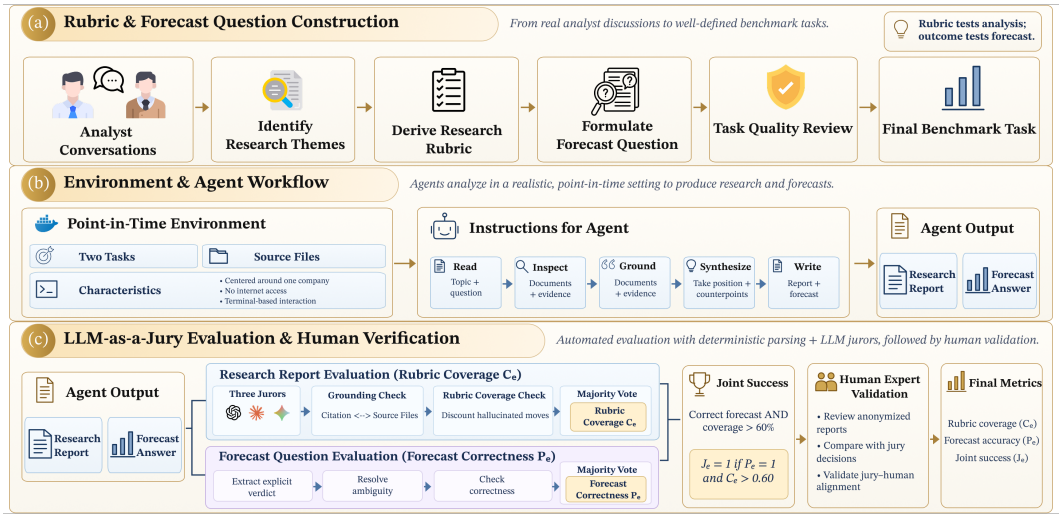

Figure 1: Overview of Hedge-Bench. (a) Task construction: Real analyst conversations are distilled into research rubrics and forecast questions, followed by task quality review. (b) Agent workflow: Agents operate in company-specific, point-in-time environments with source documents and terminal-based interaction, but no internet access, producing a research report and a forecast answer. (c) Evaluation: An LLM jury assesses report grounding and rubric coverage through majority-vote aggregation, while a separate pipeline extracts and evaluates forecast answers. Joint success requires both a correct forecast and rubric coverage above 60%. Human experts review anonymized reports to assess alignment between automated and human judgments.

Each environment pairs (1) an open-ended financial research task with (2) a forecasting question. The agent receives a terminal workspace containing primary documents cut off at a point in time, and then writes a research report and forecast. The analysts' diligence and often competing investment arguments define a hidden rubric of required comparisons and inferences, which we call analytical moves. Research coverage measures the percentage of moves credited by the jury after evidence checking; later events determine prediction correctness independently of the analysts' views.

We next use the benchmark to test whether training improves research completion. Group Relative Policy Optimization (GRPO) trains a policy using the relative scores of responses to the same task (Shao et al., 2024). A single trajectory score gives the same advantage to useful retrieval, repeated searches, and final synthesis. We instead have a judge read the completed trajectory and assign positive, negative, or zero credit to individual turns. We call this procedure **TurnCredit** and use it to train Muse-Glimmer into Muse-Glimmer-Hedge. We compare it first with outcome-only GRPO, which uses one score per trajectory, and then with a control that matches advantage magnitude. Evaluation on Hedge-Bench-Heldout tests performance on tasks excluded from training; Hedge-Bench-Private tests transfer to new companies. We discuss how Hedge-Bench and TurnCredit build on and extend prior work in Appendix A. Our contributions are:

1. **Benchmarking agents on both financial research and financial forecasting.** We construct 92 analyst-derived environments with separately resolved predictions (Section 2) and evaluate 11 frontier models with three trials per task (Section 4).

2. **Turn-level credit assignment improves research training.** After 21 updates, TurnCredit improves held-out research coverage over GRPO with a single trajectory score by 15.95 points at low effort and 21.69 points at medium effort. It also reduces repeated commands from 12.59% to 1.23% late in training (Section 5).

3. **Generalization across reasoning effort, companies, and tasks.** Training at high effort improves performance at all four tested efforts, on 53 tasks from new companies, and on four external professional evaluations (Section 6).

2 HEDGE-BENCH

In this section, we first describe the task construction pipeline, including how we extract evaluation rubrics from analyst conversations. We then explain how these rubrics are used to construct evaluation environments and forecasting questions. Next, we detail the tasks and resources provided in each environment. Finally, we summarize the specifications of the different `Hedge-Bench` partitions.

2.1 FROM ANALYST CONVERSATIONS TO RUBRICS AND FORECASTS

To source high-fidelity reasoning traces, we draw from a pool of more than 500 investment professionals employed full-time at established investment firms.

For each conversation, we connect two anonymous analysts at separate firms to engage in an open-ended discussion about whatever they deem relevant in their analysis of a specific public company. This format naturally elicits both collaborative reasoning and adversarial debate. Importantly, these discussions are not scripted responses to benchmark prompts. Figure 1 summarizes our task construction pipeline. We record and transcribe these conversations, which include discussions on calculations and underwriting assumptions, disagreements over the investment case, and open questions they are actively investigating. The conversations and the rubrics derived from them are proprietary and unpublished, drawing on the analysts' expertise in the domain and the individual security.

Rubric construction. We identify the main lines of inquiry in each analyst discussion as research themes. For each theme, the hidden rubric records the analytical moves made by the analysts such as counterarguments and inferences that advance the analysis beyond merely naming a topic. The task instruction exposes the themes but not the underlying moves, and every move must be supportable by the materials supplied to the agent.

For example, the Getty-Shutterstock task contains a theme on the threat of generative AI to stock imagery. Its four moves require the agent to identify the exposure of creative stock images, consider whether secular decline should compress the valuation multiple, assess potential creative-revenue erosion even if merger synergies are realized, and examine licensing or IP protection as a defensive moat. Merely identifying AI as a risk does not satisfy these moves. Semantically equivalent arguments receive credit, and agents are not penalized for reaching a different overall investment conclusion from the analysts.

Forecast construction. We separately formulate a forecasting question from an unresolved issue raised by the research. Each question specifies a concrete event and resolution horizon, with a subsequent disclosure, event, or deadline determining the ground-truth outcome. The forecasting question is distinct from the research rubric: the rubric measures whether the agent performs the required analysis, while the resolved outcome measures whether its forecast is correct.

The public release is selected from a broader collection of more than 5,112 candidate tasks gathered over nine months. We assess candidate difficulty, and two independent human reviewers verify task quality, such as whether the supplied sources support the analytical requirements and whether the forecasting question has a clear and resolvable decision boundary.

2.2 TASK AND ENVIRONMENT FORMULATION

Environment. Each evaluation environment is a self-contained, point-in-time workspace centered on one public company and research topic. It specifies the research themes, forecasting question, information cutoff, and required output. Agents do not receive the underlying analyst conversation, hidden move-level rubric, or future resolution evidence. Each environment runs in a Harbor container (Harbor Framework Team, 2026) with Internet access disabled.

Tasks. Each environment contains two paired tasks. First, the agent writes an analytical research report addressing the specified themes and supporting its claims with evidence. Second, it answers a concrete, falsifiable binary forecasting question. The research report and forecast are submitted together in `/app/answer.md`. We therefore treat each environment as one paired research-and-forecasting task instance.

Resources. Agents receive only the point-in-time materials provided under `/app/data/`. Depending on the environment, these include company and peer filings, earnings-call transcripts, press releases, financial statements, historical prices, and company profiles. Agents may inspect these materials through the terminal and perform arbitrary calculations, but cannot retrieve information beyond the environment's cutoff date. Figure 2 illustrates the structure of one such environment.

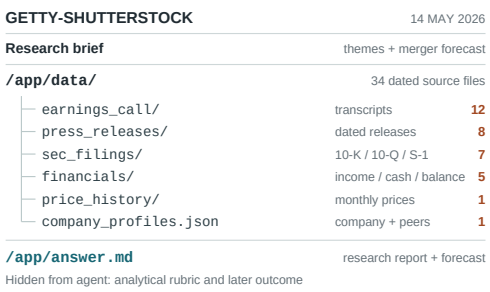

Figure 2: One Getty-Shutterstock environment at the May 14, 2026 cutoff. 34 point-in-time source files accompany the task.

The Getty-Shutterstock environment, for example, contains 34 source files and asks whether the proposed merger will close within nine months. Its 15 analytical moves cover business complementarity, regulatory execution, AI exposure, and the balance sheet. An agent may therefore find the financial rationale for the merger compelling while forecasting that regulatory barriers will prevent it from closing. Appendix C.1 provides the full task and resolution chronology.

2.3 Data splits

After constructing the task environments described above, we partition them according to their role in evaluation and policy training. `Hedge-Bench`-Public contains 92 tasks spanning 23 companies, with 49 Yes and 43 No forecasting outcomes. In total, these tasks contain 368 research themes and 1,150 analytical moves. Each task includes 3-5 themes and 9-20 moves (median 12), supported by 20-43 source files (median 31). Information cutoffs range from April 25, 2025 to May 22, 2026.

Table 1: Hedge-Bench task sets.

Set (tasks)	Role
Hedge-Bench-Public (92)	Public benchmark and frontier evaluation
Hedge-Bench-Train (165)	Policy training
Hedge-Bench-Heldout (22)	Held-out policy evaluation
Hedge-Bench-Private (53)	New-company transfer

For training, `Hedge-Bench`-Train contains 165 environments: 70 environments drawn from the public set and 95 additional private training-only environments. The remaining 22 public tasks form `Hedge-Bench`-Heldout and are excluded from policy updates, reward-prompt development, checkpoint selection, and training-configuration selection. `Hedge-Bench`-Private is a separate transfer split designed to test generalization to unseen companies: it contains 53 tasks spanning 11 companies that do not appear in RL training. Unlike `Hedge-Bench`-Heldout, which evaluates generalization to unseen tasks on familiar companies, `Hedge-Bench`-Private introduces entirely new companies and therefore measures cross-company transfer. Table 1 summarizes the four splits.

3 Experiment Setup

We introduce two evaluation settings: *full-environment* and *question-only*. We evaluate the research report and forecasting answer separately, then combine them into a joint success metric. We use an LLM-as-a-Jury framework to evaluate agent outputs and conduct a human expert study to assess alignment between jury decisions and those of financial professionals. Finally, we describe the evaluated agents and execution harness.

3.1 Evaluation and Scoring

LLM-as-a-Jury. We use three frontier models as independent jurors: Gemini 3.8 Flash, Claude Opus 5, and GPT-5.6 Sol. Each juror evaluates the agent output using only the ground-truth rubric and the agent's answer, with no access to external information or other jurors' decisions. Juror decisions are aggregated by majority vote (Verga et al., 2024).

Evaluation settings. We evaluate agents under two settings. In the *full-environment* setting, agents receive the complete task context in `/app/data/` and are required to produce both a research report and a forecast. In the *question-only* setting, agents receive only the forecasting question, without access to `/app/data/`, and are asked to provide the forecast alone. This ablation measures how much forecasting performance depends on task-specific evidence, while also probing how much relevant company knowledge is already encoded in the models' pretrained parameters.

Research report scoring. Each agent must support its analysis in the report with citations to sources in `/app/data/`. We score reports by the fraction of required rubric moves that are both analytically present and citation-grounded. Each juror performs two stages of evaluation. First, it verifies citation grounding by checking each cited claim against excerpts from the referenced source. Claims that are not supported by the cited evidence are flagged as hallucinated. If a majority of jurors flag the evidence supporting a rubric move as hallucinated, that move is discounted.

Second, each juror evaluates rubric coverage by comparing the full report against every required analytical move in the ground-truth rubric. Semantically equivalent arguments receive credit even when phrased differently, whereas generic mentions of the broader theme do not. A move is considered covered when a majority of jurors agree that the corresponding analysis is present and a majority does not flag its supporting evidence as hallucinated.

Formally, in an environment e with K_e rubric moves, for each move k, let h_{ejk} indicate whether juror j credits the move and t_{ejk} whether it flags the supporting evidence as hallucinated. Let $\mathbf{1}[\cdot]$ equal 1 when the condition holds and 0 otherwise. With majority threshold $m = 2$,

$$v_{ek} = \mathbf{1}\left[\sum_j h_{ejk} \geq m\right] \cdot \mathbf{1}\left[\sum_j t_{ejk} < m\right], \qquad C_e = \frac{1}{K_e}\sum_{k=1}^{K_e} v_{ek}, \qquad m = 2. \tag{1}$$

Thus, C_e is the fraction of required moves receiving valid majority credit. Appendix D.4 provides qualitative examples of accepted and rejected moves.

Human expert validation. Two financial professionals independently reviewed eight anonymized reports without access to model identities or jury scores. Across approximately 24 hours of review, the two experts made 197 individual judgments covering 110 distinct report-rubric-move pairs. Both experts gave definite decisions on 81 of these pairs. On this shared subset, the jury agreed with each expert on 77.8% of decisions, compared with 70.4% agreement between the experts. Under a stricter criterion requiring both the analytical move and factual support, jury-expert agreement was 75.5% and 77.0%. Appendix D.3 provides additional details.

Forecasting question scoring. For environment e, prediction correctness is

$$P_e = \mathbf{1}[\hat{y}_e = y_e], \tag{2}$$

where $\hat{y}_e$ is the agent's forecast and y_e is the resolved outcome. We extract the explicit verdict and use the jury to resolve conditional or ambiguous answers against the event definition. The question-only ablation uses the same forecasting question and label but provides no documents or Internet tools.

Joint success scoring. Joint success requires both a correct forecast and report coverage strictly above 60%:

$$J_e(\tau) = P_e\,\mathbf{1}[C_e > \tau], \qquad \tau = 0.60. \tag{3}$$

The threshold is applied before rounding; for example, 12- and 15-move rubrics require at least 8 and 10 credited moves, respectively. We additionally report continuous coverage and forecasting accuracy, with threshold sensitivity in Appendix E. For coverage and joint success, we average repeated trials within each environment and then average across environments. Forecasting accuracy is the fraction of valid responses that are correct. All three frontier-agent trials receive equal weight. Appendix D provides the detailed evaluation settings and individual-judge results.

3.2 Agents and Harness

As shown in Fig. 3, we evaluate 11 frontier proprietary and open-source agents on Hedge-Bench-Public, with three trials per agent and environment. Each environment runs in a Harbor container (Harbor Framework Team, 2026) using the Terminus-2 terminal-based workflow (Merrill et al., 2026). Agents use provider-specific reasoning controls; full execution settings are given in Appendix D.1.

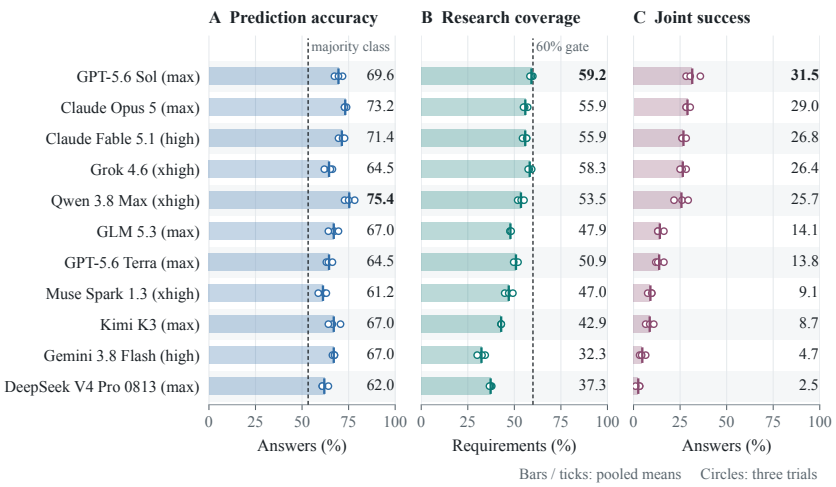

Figure 3: Each model has 92 environments and three trials (276 answers). Bars and vertical ticks show three-trial means; open circles show individual-trial means. Joint success requires a correct forecast and strictly more than 60% coverage. Dashed lines mark the 53.26% majority-class baseline and the 60% coverage threshold.

4 Evaluation Results

We evaluate 11 frontier models with three trials on each of the 92 Hedge-Bench-Public environments, and, separately, question-only predictions. We use the settings and scoring procedure in Section 3.1. As shown in Figure 3, joint success ranges from 2.54% to 31.52%, with GPT-5.6 Sol achieving the highest observed rate and six models scoring below 20%. Open-weight model families average 12.77% joint success, compared with 20.19% for proprietary families, despite similar forecasting accuracy, showcasing better research ability (Appendix L.4). Across all 92 tasks, Kimi K3 uses the fewest turns (28.2 per trajectory) and Qwen3.8 Max averages 111.6 turns, the highest across all 92 tasks and three trials (Table 41). Within the same model and task, longer trajectories do not consistently correspond to higher coverage or more accurate forecasts (Table 42). Longer final reports also need not achieve higher coverage (Appendix L.1). Saved trajectories also show repeated source retrieval and extended self-checking that can end with low coverage; In 1,622 of 3,036 reports, the jury withholds credit for analytical moves with inadequate supporting evidence, highlighting the importance of our sequential grading in preventing unsupported analysis from receiving credit (Appendix L.5). Appendix M gives model-by-model examples.

Research coverage and forecast accuracy can differ. Ideally, 100% coverage of both positive and negative rubric should result in a correct forecast, but imperfect coverage can skew the model reasoning to inaccurate forecasting. For example, Grok 4.6 has higher coverage than Qwen3.8 Max (58.25% versus 53.54%) but lower forecast accuracy (64.49% versus 75.36%). To separate this comparison from differences between models and environments, we also compare repeated trials within each model and environment. After centering within these pairs, model-specific coverage-correctness correlations range from -0.123 to 0.119, with none significant after Holm correction (Appendix L.3). Appendix L.2 qualitatively shows both outcomes on the same environments: a correct forecast with incomplete research, and a more detailed report with an incorrect forecast.

Full-environment addition helps in forecasting. Above, we found no significant correlation between coverage and forecast correctness within the same model and environment. Does that mean the models don't need the environment for prediction and have pretraining knowledge about the underlying securities? To answer this, we evaluated the models on all 92 forecast questions standalone in 3 trials (Table 16). Ten of eleven models have higher observed forecast accuracy with the full environment than with the question alone. Qwen3.8 Max rises from 59.06% to 75.36%, and Gemini 3.8 Flash from 56.16% to 67.03%. Muse Spark 1.3 is the exception, scoring 61.96% question-only and 61.23% with the environment. Also, performance is not confined to older outcomes: ten of eleven models achieve higher forecasting accuracy on questions resolved after March 2026 than on those resolved earlier (Table 17). Indicating most of the environment details are not in the models' knowledge.

5 POST-TRAINING

Next, we check whether training models on `Hedge-Bench` improves financial research. We post-train Muse-Glimmer-30B (Meta Superintelligence Lab, 2026) on Hedge-Bench-Train's 165 environments using LoRA adapters (Hu et al., 2022), high reasoning effort, and a Bash interface for one epoch (21 steps). Each of the 21 updates uses eight environments with eight rollouts per environment, for a batch size of 64 trajectories. The resulting policy is Muse-Glimmer-Hedge. Rather than combining forecast correctness and coverage into a fixed weighted reward, we score how well the report uses evidence to support its forecast.

5.1 TURNCREDIT: TRAINING WITH TURN-LEVEL CREDIT

In preliminary runs with outcome-only GRPO, the model's policy entropy collapsed. The number of turns also increased as training progressed (Figure 4), motivating turn-level credit to reinforce useful turns and penalize redundant ones, related to process supervision (Lightman et al., 2024). Per turn credit can be assigned in many ways, for example, sampling additional continuations from each turn to estimate advantage. However, this strategy substantially increases rollout cost. TurnCredit instead assigns credit after each rollout is complete, using an LLM judge (Falck et al., 2026).

For each environment, we sample eight trajectories, indexed by i. A Gemini 3.7 Flash judge scores how well each report uses evidence to support its forecast. We call the LLM judge twice for each report and average the two scores to obtain the trajectory's synthesis score s_i. The attribution judge reads the completed report with its synthesis score and written assessment, alongside the full agent trajectory and tool outputs. It also receives the task instruction and hidden rubric, together with the grading verdict identifying accepted and missed moves and forecast correctness. It judges early retrieval by its later use: useful evidence gathering receives positive credit, an unsuccessful but relevant search can remain neutral, and redundant retrieval or faulty synthesis receive negative credit. Figure 8 in Appendix G.2 illustrates the assigned credit on a VeriSign rollout.

We separate advantage scale from turn-level direction. First, we centre the scores of the eight sibling rollouts of a prompt without dividing by their standard deviation (Liu et al., 2025), $\Delta_i = s_i - \frac{1}{8}\sum_{i'=1}^{8} s_{i'}$. For turn j in trajectory i, the judge returns reasoning success $r_{ij}^{+} \in [0,1]$, reasoning failure $r_{ij}^{-} \in [0,1]$, and signed tool utility $u_{ij} \in [-1,1]$. We combine these into positive and negative weights w_{ij}^{+} and w_{ij}^{-}. For eligible token t, let $j(t)$ be its turn and A_{it} its advantage:

$$w_{ij}^{+} = \max(r_{ij}^{+}, \max(u_{ij}, 0)),\ w_{ij}^{-} = \max(r_{ij}^{-}, \max(-u_{ij}, 0)),\ A_{it} = |\Delta_i|(w_{ij(t)}^{+} - w_{ij(t)}^{-}). \quad (4)$$

Here, $|\Delta_i|$ sets the advantage scale, while turn weights determine which tokens are reinforced or suppressed. Each turn's coefficient is copied to its eligible assistant tokens; prompt tokens and tool observations are excluded. We use a token-level clipped GRPO objective, following DAPO (Yu et al., 2025; Schulman et al., 2017; Shao et al., 2024), implemented with slime (Zhu et al., 2025). Full training details and judge prompts appear in Appendices F and K.

5.2 RESULTS: TURN-LEVEL TRAINING IMPROVES FINANCIAL RESEARCH

We evaluate Muse-Glimmer and the trained model Muse-Glimmer-Hedge on the 22 Hedge-Bench-Heldout tasks, with eight trials per task at low, medium, high, and extra-high reasoning effort. Each condition contains 176 reports, for 1,408 overall, assessed by the three-juror procedure in Section 3.1. Coverage rises from 17.17% to 33.36% at low effort and from 30.19% to 47.02% at extra-high effort, with gains of 16.19-17.38 points across settings. Joint success also improves at every matched effort, by 7.39-18.18 points (Holm $p \leq 0.00089$). Thus, more reports both exceed 60% coverage and predict correctly. For example, at extra-high effort, Muse-Glimmer-Hedge examines Flywire's past healthcare underperformance and rollout risks, covering 10/12 moves versus 3/12 for Muse-Glimmer. Check Appendix I.1 for the qualitative comparison. Muse-Glimmer-Hedge at low effort also exceeds Muse-Glimmer at extra-high effort in coverage while using 47.9% fewer tokens (Figure 5).

We then compare TurnCredit with outcome-only GRPO, which uses the same synthesis score but applies each trajectory's advantage uniformly to its eligible tokens. TurnCredit improves coverage by 15.95 points at low effort and 21.69 at medium effort. Outcome-only GRPO gives little improvement over the base model at low effort and performs worse at medium effort (Table 2). The high-effort

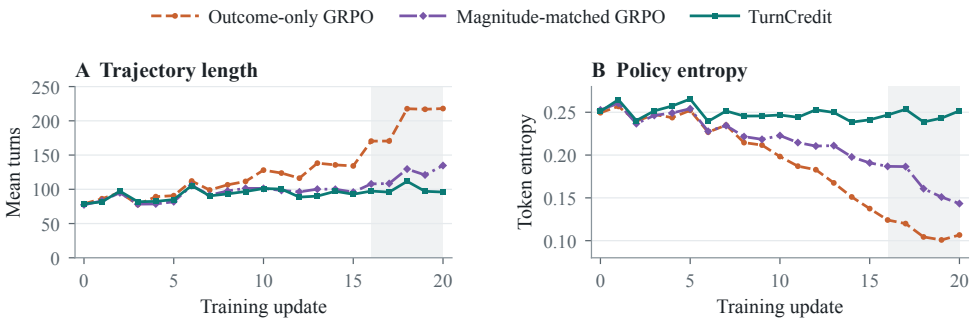

Figure 4: TurnCredit limits trajectory growth while maintaining policy entropy. Curves show all 21 updates without smoothing, with 64 trajectories per update. Shading marks updates 16–20.

outcome-only GRPO evaluation could not be completed because trajectories repeatedly exhausted the turn budget, even after it was increased.

As shown in Equation 4, TurnCredit also changes the credit magnitude, since the turn weights can reduce the absolute advantage applied to each token. To test whether such improvement can simply be attributed to advantage magnitude, we train a magnitude-matched outcome-only GRPO baseline which computes turn weights, then uses them only to globally rescale the trajectory-level advantages to the same mean absolute token magnitude. Muse-Glimmer-Hedge exceeds the magnitude-matched control by 11.67, 14.82, and 13.80 coverage points at low, medium, and high effort, respectively (all three differences remain significant after Holm correction, $p < 0.001$). Matching advantage magnitude does not reproduce the gain.

Table 2: Research coverage on Hedge-Bench-Heldout (%). Each condition contains eight trials on each of 22 tasks. Turn-level training exceeds magnitude-matched GRPO at all three efforts (Holm $p < 0.001$). Unscaled high-effort evaluation was not completed because of repeated turn/context exhaustion.

Policy	Low	Medium	High
Muse-Glimmer	17.17	28.64	29.51
Outcome-only GRPO	17.41	24.33	–
Magnitude-matched GRPO	21.69	31.20	33.02
Turn-level GRPO (Muse-Glimmer-Hedge)	**33.36**	**46.02**	**46.82**

Turn-level training also produces shorter, less repetitive training trajectories. Over later training updates (16–20; Figure 4), it averages 99.6 turns and 1.23% repeated commands, compared with 198.7 turns and 12.59% for outcome-only GRPO and 120.3 turns and 3.95% for the magnitude-matched control. Entropy remains stable compared to the outcome-only and magnitude-matched variants. Full held-out results and statistical tests are given in Appendix H.1; Appendix G provides further training diagnostics and the magnitude-matched control definition.

6 Generalization

We test whether the gains from training transfer across reasoning efforts and to other benchmarks. Although trained only at high effort, Muse-Glimmer-Hedge improves coverage in all four tested efforts (Figure 5). At low effort, it achieves 33.36% coverage with 28,662 trajectory tokens (47.9% shorter trajectories), versus 30.19% and 54,974 tokens for extra-high Muse-Glimmer. Low-effort Muse-Glimmer-Hedge exceeds all four Muse-Glimmer efforts in coverage on the fixed 22 tasks. All four comparisons remain significant after Holm correction ($p \leq 0.02496$; Table 25).

To avoid company information seen during RL training polluting the forecasting results, we separately evaluate on Hedge-Bench-Private: 53 tasks from 11 companies absent from RL training, with three trials per task and condition. Coverage improves by 13.26 points at low effort and 16.44 points at high effort. High-effort prediction accuracy increases from 38.36% to 48.43%, a 10.06-point gain. Both coverage gains and high-effort forecasting remain significant after Holm correction (Table 26). Low-effort Muse-Glimmer-Hedge also exceeds high-effort Muse-Glimmer coverage with 40.2% fewer trajectory tokens (Table 27). The 53 tasks have 28 Yes and 25 No outcomes (balanced data). Private environments are more difficult for our frontier reference; for reference, GPT-5.6 Sol at

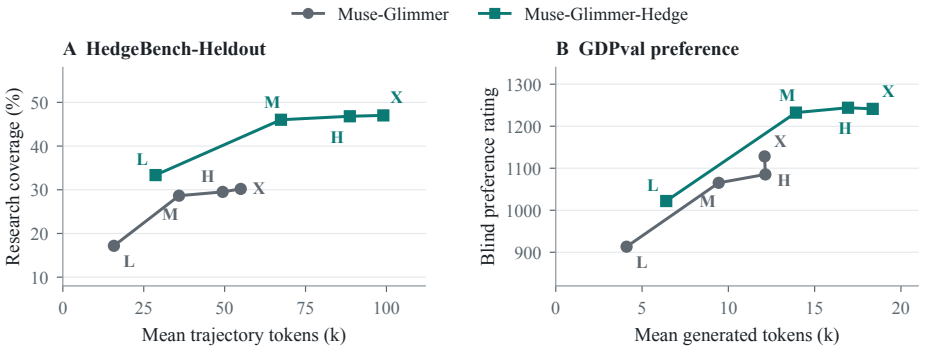

Figure 5: High-effort training improves performance across efforts. L, M, H, and X denote low through extra-high effort. Panels show Hedge-Bench-Heldout coverage (A) and GDPval pool-relative blind preference ratings (B).

maximum effort achieves 45.33% coverage and 35.85% prediction accuracy, versus 59.17% and 69.57% on Hedge-Bench-Public (Table 28). Appendix H.2 gives the reference comparison and answer rates; Tables 34 and 35 give the resolution dates.

Finally, we evaluate whether the learned research ability transfers to other professional benchmarks: GDPval (Patwardhan et al., 2025), GDP.pdf (Surge AI, 2026), Harvey LAB (Harvey AI, 2026), and Frontier Finance (Samaya AI, 2026), under their respective professional-task metrics. Appendix H.4.1 describes our Bash-based GDPval setup and preference pool; Appendix H.4.2 describes the Harvey execution and scoring adaptations. The GDP.pdf input rendering is specified in Appendix H.4.3, and Frontier Finance tools and judge in Appendix H.4.4. Table 3 reports rubric-based results; Appendix H.5 gives paired statistical tests.

Table 3: Professional-task transfer under each benchmark's metric (Appendix H). All seven gains remain significant after Holm correction. `MG` and `MG-H` are Muse-Glimmer and Muse-Glimmer-Hedge.

Evaluation	MG	MG-H	Gain
GDPval, low	74.19	77.93	+3.74
GDPval, medium	79.82	81.71	+1.90
GDPval, high	78.91	81.52	+2.61
GDPval, extra-high	78.92	82.86	+3.94
GDP.pdf, high	69.85	72.14	+2.29
Harvey LAB, high	64.19	71.26	+7.07
Frontier Finance, high	39.62	44.58	+4.96

On our Bash-based adaptation of the 220-task GDPval set, Muse-Glimmer-Hedge improves rubric completion at all four efforts. For example, in the same task, Muse-Glimmer-Hedge preserves the requested store-count definition, with 63.8% rubric completion versus 37.9%. It also retrieves a missing handbook instead of assuming its requirements, achieving 100% versus 50% with fewer turns (Appendix I). Blind preference ratings also increase, with paired Elo gains of 108.7–167.1 in all efforts (Figure 5). For Harvey LAB, we select four tasks from each of 25 areas of legal practice using a deterministic diversity procedure, giving 100 tasks (Appendix H.4.2). Its two judges separately measure gains of 7.13 and 7.02 points. GDP.pdf uses all 100 benchmark tasks with five attempts each (Appendix H.4.3), and Frontier Finance uses 189 matched queries (Appendix H.4.4). The gains remain significant under a joint Holm correction ($p \leq 0.01274$; Appendix H.5).

7 Conclusion

`Hedge-Bench` evaluates research completion and forecasting on 92 public analyst-derived tasks. Correct predictions often accompany incomplete analysis, motivating separate measures and their conjunction. Our evaluation of 11 frontier models reveals a significant gap relative to expert-derived research standards: joint success ranges from only 2.54% to 31.52% (Figure 3). TurnCredit improves coverage over trajectory-level GRPO and a magnitude-matched variant that proper credit assignment is crucial in these tasks. More importantly, the gains from post-training Muse-Glimmer on `Hedge-Bench` hold across reasoning efforts and extend to a wide range of other knowledge works (Table 3), demonstrating the value of `Hedge-Bench` as both a training and evaluation resource.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025. URL https://arxiv.org/abs/2507.19457.

Anthropic. Claude Fable 5.1, 2026a. URL https://platform.claude.com/docs/en/models/fable-5-1/overview. Model documentation, accessed September 9, 2026.

Anthropic. Models Overview, 2026b. URL https://platform.claude.com/docs/en/models/overview. Accessed September 9, 2026.

Artificial Analysis. GDPval-AA v2 leaderboard. Online evaluation resource, n.d. URL https://artificialanalysis.ai/evaluations/gdpval-aa. Accessed 2026-09-09; living leaderboard and methodology resource.

Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking LLMs on real-world financial research tasks. *arXiv preprint arXiv:2508.00828*, 2025. URL https://arxiv.org/abs/2508.00828.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3–4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324. URL https://doi.org/10.1093/biomet/39.3-4.324.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

DeepSeek AI. DeepSeek-V4-Pro Model Card, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro. Accessed September 9, 2026.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025. URL https://arxiv.org/abs/2506.11763.

Damon Falck, Samer Sabri, Anja Surina, Thom Foster, Anya Sims, Sam Devlin, Dylan Rogers, Tantum Collins, Kaloyan Aleksiev, Louis Kirsch, and Edward Hughes. Training AI scientists to replicate research. *arXiv preprint arXiv:2608.13331*, 2026. URL https://arxiv.org/abs/2608.13331.

Shubham Gandhi, Saurabh Goyal, Kiran Kate, and Yara Rizk. DRACO: Fine-grained credit assignment with dynamic rubrics for long-horizon agent training. *arXiv preprint arXiv:2609.04094*, 2026. URL https://arxiv.org/abs/2609.04094.

Getty Images Holdings, Inc. Form 8-k: Termination of the shutterstock merger agreement. U.S. Securities and Exchange Commission, July 2026. URL https://www.sec.gov/Archives/edgar/data/1898496/000121390026075721/ea029721801-8k_getty.htm. Filed July 7, 2026.

Shashwat Goel, Nikhil Chandak, Arvindh Arun, Ameya Prabhu, Steffen Staab, Moritz Hardt, Maksym Andriushchenko, and Jonas Geiping. FutureSim: Replaying world events to evaluate adaptive agents. *arXiv preprint arXiv:2605.15188*, 2026. URL https://arxiv.org/abs/2605.15188.

Google DeepMind. Gemini 3.8 Flash Model Card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/. Accessed September 9, 2026.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments. Software, 2026. URL https://doi.org/10.5281/zenodo.20953922.

Harvey AI. Harvey LAB: The legal agent benchmark. Public benchmark repository, 2026. URL https://github.com/harveyai/harvey-labs.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023. URL https://arxiv.org/abs/2311.11944.

Ezra Karger, Houtan Bastani, Yueh-Han Chen, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip E. Tetlock. ForecastBench: A dynamic benchmark of AI forecasting capabilities. In *International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/28507.

Srivatsa Kundurthy, Clara Na, Colton Moraine, Anoushka Mohta, Case Winter, George Fang, John Ling, Emma Strubell, and Zach Kirshner. BlueFin: Benchmarking LLM agents on financial spreadsheets. *arXiv preprint arXiv:2605.30907*, 2026. URL https://arxiv.org/abs/2605.30907.

Elaine Lau, Markus Dücker, Ronak Chaudhary, Hui Wen Goh, Rosemary Wei, Vaibhav Kumar, Saed Qunbar, Guram Gogia, Yi Liu, Scott Millslagle, Nasim Borazjanizadeh, Ulyana Tkachenko, Samuel Eshun Danquah, Collin Schweiker, Vijay Karumathil, Asrith Devalaraju, Varsha Sandadi, Haemi Nam, Punit Arani, Ray Epps, Abdullah Arif, Sahil Bhaiwala, Curtis Northcutt, Skyler Wang, Anish Athalye, Jonas Mueller, and Francisco Guzmán. BankerToolBench: Evaluating AI agents in end-to-end investment banking workflows. *arXiv preprint arXiv:2604.11304*, 2026. URL https://arxiv.org/abs/2604.11304.

Jiahui Li, Hanlin Zhang, Fengda Zhang, Tai-Wei Chang, Kun Kuang, Long Chen, and Jun Zhou. Optimizing language models with fair and stable reward composition in reinforcement learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10122–10140. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.565. URL https://aclanthology.org/2024.emnlp-main.565/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. URL https://arxiv.org/abs/2503.20783.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*, 2026. URL https://arxiv.org/abs/2601.11868.

Meta AI. Introducing Muse Spark 1.3, 2026. URL https://research.meta.ai/blog/introducing-muse-spark-1-3. Accessed September 9, 2026.

Meta Superintelligence Lab. Muse Glimmer Model Card, 2026. URL https://huggingface.co/meta-models/Muse-Glimmer-30B. Model release date: August 2026; accessed September 18, 2026.

Moonshot AI. Kimi-K3 Model Card, 2026. URL https://huggingface.co/moonshotai/Kimi-K3. Accessed September 9, 2026.

OpenAI. GPT-5.6 Sol Model, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Model documentation, accessed September 9, 2026.

OpenAI. GPT-5.6 Terra Model, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-terra. Model documentation, accessed September 9, 2026.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025. URL https://arxiv.org/abs/2510.04374.

Qwen Team. Qwen Studio: Models, 2026. URL https://chat.qwen.ai/legal-agreement/models. Accessed September 9, 2026.

Mahesh Ramesh, Kaousheik Jayakumar, Hemanth Ram, Pavan Thodima, Ramani Duraiswami, Dinesh Manocha, Aniket Rege, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. Playing with fire: What transfers when RL trains a language agent? In *ICML 2026 Workshop on RLxF: Reinforcement Learning from World Feedback*, 2026. URL https://openreview.net/forum?id=5VrEt3fthr.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. DocFinQA: A long-context financial reasoning dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 445–458. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-short.42. URL https://aclanthology.org/2024.acl-short.42/.

Samaya AI. Introducing FrontierFinance: A challenging benchmark for measuring frontier intelligence of finance agents, 2026. URL https://samaya.ai/blog/frontier-finance.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Surge AI. GDP.pdf, 2026. URL https://huggingface.co/datasets/surgeai/GDP.pdf.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024. URL https://arxiv.org/abs/2404.18796.

Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benchek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Austin Bridges, Jesse Boyle, Koby Twist, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents. *arXiv preprint arXiv:2601.14242*, 2026. URL https://arxiv.org/abs/2601.14242.

Alex Wang, Georg Meinhardt, Jacob Katz, Joseph H. Kim, Pratyush K. Chaudhary, Chase Blagden, and Eric Xu. BigFinanceBench: A workflow-grounded benchmark for financial-research agents. *arXiv preprint arXiv:2606.03829*, 2026. URL https://arxiv.org/abs/2606.03829.

xAI. Grok 4.6: Model Documentation, 2026. URL https://docs.x.ai/developers/grok-4-6. Accessed September 9, 2026.

Xiaomi MiMo Team. MiMo-V2.6-Pro-RL. https://huggingface.co/XiaomiMiMo/MiMo-V2.6-Pro-RL, 2026.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. FinBen: A holistic financial benchmark for large language models. *arXiv preprint arXiv:2402.12659*, 2024. URL https://arxiv.org/abs/2402.12659.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Z.AI. GLM-5.3 Overview, 2026. URL https://docs.z.ai/guides/llm/glm-5.3. Accessed September 9, 2026.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, Zhaoyang Liu, Bolin Ding, and Jingren Zhou. AgentEvolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025. URL https://arxiv.org/abs/2511.10395.

Yaocheng Zhang, Haohuan Huang, Zijun Song, Zijie Zhao, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. CriticSearch: Fine-grained credit assignment for search agents via a retrospective critic. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12272–12290. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.596. URL https://aclanthology.org/2026.findings-acl.596/.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3277–3287. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An LLM post-training framework for RL scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

APPENDIX CONTENTS

A Related Work . . . 15

B Future Directions . . . 15

C Qualitative Environment and Source Samples . . . 16
- C.1 Getty-Shutterstock Merger . . . 16
- C.2 WULF Financing and Lease Economics . . . 20

D Execution, Scoring, and Statistical Details . . . 20
- D.1 Trials and Valid Responses . . . 21
- D.2 Juror Decisions and Model Scores . . . 21
- D.3 Blind Expert Assessment . . . 22
- D.4 Qualitative Examples of Jury Decisions . . . 24
- D.5 Citation Frequency and Research Coverage . . . 25
- D.6 Citation-Processing Sensitivity . . . 25
- D.7 Panel and Aggregation Sensitivity . . . 26

E Joint Success and Question-Only Results . . . 26
- E.1 Question-Only Predictions . . . 27
- E.2 Resolution Timing . . . 28

F Training Configuration . . . 30
- F.1 Iterative Judge Prompt Tuning . . . 30
- F.2 Policy Loss and Token Weights . . . 30
- F.3 Reward Inputs and Attribution . . . 31
- F.4 Terminal Handling . . . 32

G Credit Assignment and Behavioral Evidence . . . 32
- G.1 Model Behavior . . . 32
- G.2 Examples of Credit Assignments . . . 33
- G.3 Magnitude-Matched Outcome-Only Training . . . 35

H Post-Training Evaluation Details . . . 36
- H.1 Hedge-Bench-Heldout . . . 37
- H.2 Hedge-Bench-Private . . . 38
- H.3 Joint Success . . . 39
- H.4 Professional-Task Protocols . . . 40
- H.5 Paired External Comparisons . . . 41
- H.6 Hedge-Bench-Private Outcome Chronology . . . 42
- H.7 Hedge-Bench-Public Outcome Chronology . . . 43

I Research Improvements in Held-Out and External Tasks . . . 45
- I.1 Hedge-Bench-Heldout: More Research Behind the Same Forecast . . . 45
- I.2 GDPval: Using the Right Denominator . . . 46
- I.3 GDPval: Recovering the Source Before Writing . . . 46

J Ethics . . . 47

K Full Training Reward and Attribution Prompts . . . 47
- K.1 Synthesis Objective: Full System Prompt . . . 47
- K.2 Turn-Level Attribution: Full System Prompt . . . 49

L Frontier Model Behavior . . . 52
- L.1 Report Length and Research Coverage . . . 52
- L.2 Same-Task Report Comparisons . . . 52
- L.3 Coverage and Correctness Within the Same Task . . . 53
- L.4 Open-Weight and Proprietary Model Families . . . 54
- L.5 Evidence-Support Penalties . . . 54
- L.6 Trajectory Length and Report Quality . . . 55

M Operational Failures and Model-Level Case Studies . . . 56
- M.1 Recurring execution patterns . . . 56
- M.2 Examples across all eleven models . . . 57
- M.3 Citation Validation and Evidence Support . . . 58

A Related Work

Existing financial benchmarks mostly probe document reasoning and professional deliverables. FinQA, TAT-QA, DocFinQA, and FinanceBench test reasoning over financial documents (Chen et al., 2021; Zhu et al., 2021; Reddy et al., 2024; Islam et al., 2023). FinBen includes forecasting and trading (Xie et al., 2024). Closer to our setup, Vals AI's Finance Agent Benchmark evaluates agentic capabilities with expert-answer-derived rubrics (Bigeard et al., 2025). BigFinanceBench scores auditable financial derivations (Wang et al., 2026). `Hedge-Bench` pairs analyst-derived research requirements with a separately resolved forecast. The main difference from existing benchmarks is that our rubrics come from analysts who had already taken positions in the underlying stocks and done their own diligence on those companies. The rubrics capture counterarguments and investment judgments from discussions between two analysts who know the company. `Hedge-Bench` combines that research with predictions to evaluate the models' research and forecasting ability together.

Apart from financial evaluation, `Hedge-Bench` also gauges the agentic ability of the models. Terminal-Bench is a well-known benchmark of agentic ability (Merrill et al., 2026). `Hedge-Bench` uses a Harbor environment, as Terminal-Bench does (Harbor Framework Team, 2026). APEX-Agents, BankerToolBench, and BlueFin evaluate cross-application work, investment-banking deliverables, and financial spreadsheets, respectively (Vidgen et al., 2026; Lau et al., 2026; Kundurthy et al., 2026). FrontierFinance evaluates open-ended investor research with source-attributed rubrics (Samaya AI, 2026). Our executable tasks require both a research report and a forecast. ForecastBench evaluates predictions submitted before events resolve, while FutureSim uses chronological event replay (Karger et al., 2025; Goel et al., 2026). We measure research coverage and prediction correctness together, distinguishing correct forecasts with incomplete analysis from reports with high research coverage that predict incorrectly.

There is no single consensus on how to assign per-turn credit in place of normal GRPO's trajectory-level advantage (Shao et al., 2024). Recent work has shown that fine-grained credit assignment can improve agents in several setups. Faraday uses turn-level credit to weight GRPO updates for scientific replication (Falck et al., 2026), while CriticSearch uses a retrospective critic to score individual search turns (Zhang et al., 2026). DRACO redistributes credit without changing the trajectory advantage's sign or total token weight (Gandhi et al., 2026); AgentEvolver instead combines signed action attribution with outcome rewards accumulated over later turns (Zhai et al., 2025). In TurnCredit, the judge's assessment of each turn determines whether its actions are reinforced or down-weighted, while the trajectory score determines the strength of the update. We test this formulation on financial research against outcome-only GRPO and a magnitude-matched control. Unlike DRACO, TurnCredit can assign opposite signs to turns in the same trajectory. CriticSearch is also retrospective, but we go beyond short-horizon tasks and show that TurnCredit improves long-horizon performance while using lower inference-time compute. We also demonstrate transfer across reasoning efforts with test-time scaling and generalization to other professional tasks.

B Future Directions

The current paper mostly uses discussions between two analysts about securities they have analyzed and hold positions in. In future, we can include discussions among more than two people on broader topics, such as whether AI will displace SaaS businesses, to build richer environments. We can also use the raw audio files to study how models understand 45-minute to one-hour research discussions. Comparing the original recording with its transcript on the same task would test whether hearing the discussion helps beyond reading its words. We can build RL environments around these recordings to improve audio understanding. We can also have computer-use agents perform Hedge-Bench tasks through financial websites or spreadsheets instead of a terminal.

We can also extend each environment over time, releasing new information and asking the agent to revise its forecast, similar to FutureSim (Goel et al., 2026). This would test whether it changes its position when the evidence changes. On the training side, we can extend TurnCredit to assign credit to individual reasoning steps within a turn, with an external judge providing finer-grained training feedback. We can also give the judge multiple trajectories for the same task and ask it to assign credit through direct comparison. The other trajectories provide a relative reference, which may make useful differences easier to identify than when each trajectory is judged separately. Similar

to MiMo-V2.6 which uses groupwise grading to compare rollouts and redistribute advantage toward higher-quality successful solutions (Xiaomi MiMo Team, 2026).

C Qualitative Environment and Source Samples

Over the span of nine months, we collected more than 5,112 candidate tasks and 20,448 analytical subtasks. From these, we handpicked tasks for the public release by balancing task difficulty and breadth of financial coverage. The 92 public environments cover valuation, growth, mergers and acquisitions, competitive positioning, operational execution, and risk. Each environment provides an open-ended research question, source materials, and hidden analytical requirements.

The 92 environments contain 368 research themes and 1,150 analytical moves. Of the tasks, 16 have three themes, 60 have four, and 16 have five. Of the 368 themes, 322 contain three moves and 46 contain four. The median task has 12 moves and 31 source files. Table 4 summarizes the source collection. The same source document can appear in multiple environments and is counted for each task that includes it.

Table 4: Source materials supplied in the 92 Hedge-Bench-Public environments.

Source category	Task-local file instances
Earnings-call transcripts	1,038
Financial statements	816
SEC filings	494
Press releases	424
Company profiles	92
Price histories	91
Investor presentations	12
Total	2,967

The Getty-Shutterstock merger and the WULF examples follow a task's source materials and hidden rubric to its forecast resolution. The examples also show what we score: whether the report completes the required analysis using evidence available at the cutoff date, and whether its forecast matches the later outcome.

An analytical requirement specifies the reasoning needed to address a research theme. For example, the WULF task compares colocation income with GPU ownership: a response must examine return predictability and asset life, rather than merely note demand for compute. The Getty task below combines business complementarity, regulatory conditions, AI exposure, and pro forma economics. In both examples, the required analysis is defined by the analyst's reasoning, whereas the ground-truth prediction is set by the real-world outcome.

C.1 Getty-Shutterstock Merger

Getty-Shutterstock merger environment asks:

> Consider the proposed GETY-Shutterstock merger, the rationale for it, and the approval process it has been navigating. Will the merger close within the next nine months?

The accompanying research assignment covers four themes: complementary profiles, regulatory hurdles, AI disruption, and pro forma viability and synergies. The rubric specifies 15 moves across these themes (4, 3, 4, and 4).

The instruction prompt. The following is the full agent-facing `instruction.md` from the current public-release version, reproduced verbatim. It uses the citation tool described in Appendix D.6; the reported evaluations used the earlier file-path citation protocol. The analytical rubric below is hidden from the agent.

Getty-Shutterstock: full instruction prompt

```
You are a financial analyst. Your task is to evaluate the provided
↪  topic for **Getty Images Holdings, Inc. (GETY)** as of
↪  **2026-05-14** based on the data provided in the environment,
↪  and ultimately make a concrete, falsifiable prediction.

## The Task

- **Primary Company:** Getty Images Holdings, Inc. (GETY)
- **Peer Companies:** Shutterstock, Inc. (SSTK)
- **Analysis Cutoff Date:** 2026-05-14

Develop a view on the topic below across the following themes and
↪  write your full analysis as a markdown file to `/app/answer.md`:

- **Topic:** Strategic Rationale of the Getty-Shutterstock Merger

1. Complementary Asset and Balance Sheet Profiles
2. Regulatory and Execution Hurdles
3. AI Disruption and Secular Decline Risk
4. Pro Forma Financial Viability and Synergies

## Prediction Verdict

Consider the proposed GETY-Shutterstock merger, the rationale for
↪  it, and the approval process it has been navigating. Will the
↪  merger close within the next nine months?

End your `/app/answer.md` with a section titled `## Prediction
↪  Verdict` that answers this question and commits to one outcome.
↪  Show your reasoning.

## Requirements

Your analysis must:

- Take a clear position and engage with the strongest
↪  counter-evidence in the data.
- Draw a concrete tradeoff, judgment, or conclusion out of your
↪  analysis. If management is vague or key data is unresolved,
↪  state that uncertainty directly instead of guessing.
- Use the provided themes to build a logical chain of reasoning
↪  that culminates in a final prediction.
- Organize your analysis by the themes above and end with your
↪  prediction verdict.
- Ground every specific factual claim with an inline citation
↪  immediately after the claim (see **Citation Format** below). A
↪  citation grounds only the sentence it appears in, so place each
↪  citation in the same sentence as the facts it supports.
↪  Analytical inferences need no citation, but every fact they
↪  rest on does.
- Preserve formatting. When writing `/app/answer.md`, use a quoted
↪  heredoc (`cat << 'EOF' > /app/answer.md` ... `EOF`) or another
↪  method that does not pass your text through shell expansion.
↪  Inside double quotes, bash silently deletes `$`-prefixed text
↪  (e.g. `$8.56B` becomes `.56B`). If a large heredoc stalls,
↪  append the analysis in a few smaller quoted heredocs (`cat <<
↪  'EOF' >> /app/answer.md`) or write a Python script to a file
↪  and run it. Never shorten the analysis to get the write through.
↪  After writing, re-read the file and confirm every figure
↪  survived, because the grader scores only what is literally in
↪  the file.
```

```
Claims you can't ground in `/app/data/` (numbers, events, entities) are discarded by the grader and won't count toward your score.

## Data Layout

`/app/data/` may contain any of the following (point-in-time as of 2026-05-14):

- `earnings_call/` | `.txt` earnings call transcripts for the primary company
- `financials/income_statement/` | quarterly `.json` income statements (revenue, margins, net income) for the primary company and peers
- `financials/balance_sheet/` | quarterly `.json` balance sheet (assets, liabilities, equity) for the primary company
- `financials/cash_flow/` | quarterly `.json` cash flow statements for the primary company and peers
- `press_releases/` | full-text wire press releases for the primary company as `.txt` files
- `sec_filings/10-k/` | annual SEC reports as `.txt` for the primary company (Risk Factors, Business Description, MD&A)
- `sec_filings/10-q/` | quarterly SEC reports as `.txt` for the primary company (interim MD&A, Risk Factors deltas)
- `sec_filings/s-1/` | IPO prospectus as `.txt` for the primary company (only present for recent IPOs)
- `company_profiles.json` | company profiles and market data for the primary company and peers
- `price_history/` | monthly `.json` closing price, market cap, and volume for the primary company

The titles of the files within each folder provide additional context. Peer files are provided for competitor comparisons or industry benchmarking where relevant. If a competitor is private or lacks standalone filings, evaluate competitive dynamics using GETY's disclosures. Not every environment will contain every source listed above. Work with the information that is present.

## Citation Format

Use `python3 /app/citation_tool.py` to select evidence from `/app/data`. The command checks the file and returns a citation ID such as `[[ev_0123456789abcdef01234567]]`. Put that exact ID immediately after the factual claim it supports, in the same sentence. Use multiple IDs when a comparison depends on multiple records. The grader accepts citation IDs issued during this task; handwritten file-path citations do not count.

- **Text:** `python3 /app/citation_tool.py text earnings_call/<file>.txt --lines 10-12 --speaker "Speaker Name"`. `--speaker` must match the turn header covering the lines; copy that name (`Sarah M. London:`, not a job title or `Speaker`). If the tool says `--speaker 'X' does not speak ... spoken by Y`, retry the same lines with `Y`. Multi-speaker windows are OK if you name one person who actually speaks in that range. Use the narrowest supporting line range. For other text files, omit `--speaker`.
```

```
- **JSON:** `python3 /app/citation_tool.py json
 ↪  financials/income_statement/<file>.json --where date=2025-09-30
 ↪ --keys revenue,grossProfit`. Use enough `--where` fields to
 ↪  select exactly one record. For a company profile, `--where
 ↪  symbol=CARR --keys marketCap` selects one company. For monthly
 ↪  prices, `--where date=2025-09-30 --keys close,marketCap` selects
 ↪  one row.

Check the returned excerpt against the full claim. A valid ID
 ↪  locates evidence; it does not make an unsupported statement
 ↪  true. Cite facts and numbers, then state your analytical
 ↪  inference separately.
```

The analytical rubric. The 15 move-level criteria below reproduce the current release rubric. The agent does not see them. Credit is based on the underlying analytical claim, not exact vocabulary or phrasing. Equivalent arguments that convey the same analytical idea satisfy the move. Every move must be supported by evidence in `/app/data/`. Moves relying on unsupported or fabricated claims receive no credit.

1. Complementary Asset and Balance Sheet Profiles. The merger pairs Getty's editorial and archival franchise with Shutterstock's cash generation. Management presents the combination as a path to further delever Getty and expand margins and cash flow.

(a) Identifies Getty's high debt load or interest burden as a primary risk to its standalone equity value.

(b) Contrasts Shutterstock's operating cash flow and period-end cash with Getty's higher interest expense and net debt.

(c) Characterizes Getty's editorial and archival segments as more resilient or premium compared to the more challenged creative or stock imagery segments.

(d) Infers that management positions the merger as a means to further delever Getty through combined cash flow and margins.

2. Regulatory and Execution Hurdles. The transaction faces significant scrutiny from international regulators, though the remedies required to close the deal are viewed as manageable rather than deal-breaking.

(a) Identifies specific regulatory bodies involved in the approval process, such as the DOJ or the U.K. Competition and Markets Authority (CMA).

(b) Analyzes the likelihood of the deal closing based on the nature of proposed remedies or the lack of an outright block from regulators.

(c) Notes the extended timeline of the merger process, describing it as delayed or prolonged.

3. AI Disruption and Secular Decline Risk. Generative AI represents a secular threat that could commoditize the stock imagery business, potentially offsetting any gains from merger-related cost cuts.

(a) Identifies AI-generated content as a direct threat to the marginal or disposable creative stock imagery market.

(b) Discusses the risk of multiple compression where the market refuses to pay a premium for earnings perceived to be in secular decline.

(c) Evaluates the potential for revenue erosion in the creative segment even if the company achieves its stated synergy targets.

(d) Mentions the importance of IP indemnification or licensing deals with AI platforms as a potential defensive growth vector.

4. Pro Forma Financial Viability and Synergies. The success of the combined entity depends on realizing operational synergies and stabilizing revenue to support a sustainable capital structure.

(a) Evaluates whether merger synergies and combined cash generation can lower Getty's standalone leverage on a pro forma basis.

(b) Identifies cost-cutting and operational synergies as the primary drivers for earnings growth in the absence of organic revenue expansion.

(c) Analyzes the necessity of the merger for Getty's survival, suggesting severe financial distress or substantial equity impairment if the deal fails to close.

(d) Notes that the combined entity must prove it can maintain a steady state of revenue to avoid further equity devaluation.

Table 5: Source inventory for the Getty-Shutterstock merger task at its May 14, 2026 information cutoff. The current release provides 34 files for Getty and Shutterstock.

Source family	Files	Contents
Earnings calls	12	Quarterly transcripts, 2023 Q2 to 2026 Q1
Press releases	8	February through May 2026 releases
Income statements	2	GETY and Shutterstock
Cash flow statements	2	GETY and Shutterstock
Balance sheet	1	GETY debt and liquidity
SEC filings	7	Two annual, four quarterly, one registration filing
Price history	1	69 dated observations
Company profiles	1	GETY and Shutterstock

Getty announced a conditional decision not to proceed on June 30, 2026, and formally terminated the agreement on July 7, which is 54 days after the task date (Getty Images Holdings, Inc., 2026). We use July 7 as the resolution date rather than the earlier statement of intent. The No outcome refers to termination of this proposal, not to whether the firms could ever reach another agreement, and is withheld from the research agent.

The regulatory requirement illustrates why research coverage is scored separately from prediction correctness. To satisfy this requirement, a report must assess the likelihood of the deal closing in light of proposed regulatory remedies or the absence of an outright block. Mentioning regulatory review alone is not enough. A model can correctly predict that the merger will not close while missing much of the required analysis. Conversely, it can explain the financial rationale in detail and still predict completion incorrectly.

C.2 WULF Financing and Lease Economics

The WULF examples use a March 17, 2026 information cutoff and illustrate the same distinction between retrieving facts and analyzing their implications. The tasks ask how GPU depreciation compares with colocation lease economics and whether a named counterparty is the end customer or an intermediary. They also ask how termination fees protect debt repayment if a deal fails, and how much value is created when powered capacity secures a signed lease.

The corresponding forecasts specify measurable events. One asks whether either of the next two quarterly reports will show at least $500 million of year-to-date common-stock issuance proceeds. Another asks whether a newly disclosed HPC lease will involve a counterparty other than Fluidstack or Core42 and at least 250 MW of critical IT load. Later financing and lease disclosures resolve these forecasts independently of the analytical requirements. These examples illustrate task construction; they do not add observations to the 92-task frontier evaluation.

D Execution, Scoring, and Statistical Details

This appendix explains how we collect and score responses. We then compare the jury with experts, examine individual grading decisions, and test how scoring choices affect the results. Coverage measures how much of the required analysis a report completes. The jury checks the cited evidence supporting that analysis, but does not verify every factual claim in the report. Prediction correctness is assessed separately, and joint success requires a correct forecast together with coverage strictly above 60%.

D.1 Trials and Valid Responses

We evaluate every model three times per task on the 92 Hedge-Bench-Public environments and weight the trials equally. In the full environment, the agent receives the research brief, task documents, and a terminal interface. In the question-only condition, it receives the forecast question without documents or internet access. We initially used the jury to extract Yes/No forecasts on Hedge-Bench-Public. After confirming agreement on the extracted verdicts across all models, we used a coding assistant (Codex) for subsequent extraction. We compare each extracted verdict with the resolved outcome. Question-only answers are scored directly from the final Yes/No response.

Across the 92 tasks and three trials, we set the effort level to High for Gemini and Fable; Max for Opus, Sol, Terra, GLM, Kimi, and DeepSeek; and extra-high for Muse Spark, Qwen, and Grok. We use each provider's reasoning controls where available. The same effort label does not imply the same amount of computation across providers. We resume or retry interrupted requests. We do not replace a valid completed answer because it received a low score. A juror output is valid only if it matches the rubric's themes and requirement identifiers. Every response across all 3,036 runs is reviewed by three jurors, ensuring every model and trial is scored under identical rules.

Tables 6 and 7 report pooled means and all three trials. Because tasks share company contexts across repeated trials, model rankings reflect observed score differences rather than statistically significant separations. For instance, GPT-5.6 Sol leads Claude Opus 5 by 2.53 percentage points in joint success, but this gap is not statistically significant.

Table 6: **All three frontier trials.** Joint success requires correct prediction and exact coverage $> 60\%$. T1–T3 track success across individual 92-task trials. Percentages cover the full set of 276 answers per model, and Effort notes the reasoning control used.

Model	Effort	T1	T2	T3	Joint (%)	Coverage (%)	Accuracy (%)
GPT 5.6 Sol	max	33	26	28	31.52	59.17	69.57
Claude Opus 5	max	26	26	28	28.99	55.89	73.19
Claude Fable 5.1	high	24	24	26	26.81	55.85	71.38
Grok 4.6	xhigh	26	24	23	26.45	58.25	64.49
Qwen3.8 Max	xhigh	20	24	27	25.72	53.54	75.36
GLM 5.3	max	12	15	12	14.13	47.90	67.03
GPT 5.6 Terra	max	11	15	12	13.77	50.86	64.49
Muse Spark 1.3	xhigh	9	7	9	9.06	47.00	61.23
Kimi K3	max	6	8	10	8.70	42.92	67.03
Gemini 3.8 Flash	high	6	3	4	4.71	32.35	67.03
DeepSeek V4 Pro 0813	max	3	3	1	2.54	37.26	61.96

Table 7: Full-environment results by trial. Coverage is a percentage; correct predictions are counts out of 92. Reasoning efforts match Table 6.

Model	Coverage T1	T2	T3	Correct T1	T2	T3
GPT 5.6 Sol	59.93	59.19	58.39	62	66	64
Claude Opus 5	57.18	55.56	54.92	67	68	67
Claude Fable 5.1	56.55	54.39	56.61	64	66	67
Grok 4.6	59.31	58.03	57.41	57	61	60
Qwen3.8 Max	51.79	53.64	55.18	67	72	69
GLM 5.3	47.73	47.69	48.29	62	59	64
GPT 5.6 Terra	50.96	51.98	49.65	58	59	61
Muse Spark 1.3	44.82	46.98	49.21	57	54	58
Kimi K3	42.84	42.94	42.98	61	59	65
Gemini 3.8 Flash	34.35	32.65	30.05	62	61	62
DeepSeek V4 Pro 0813	37.86	37.28	36.64	59	56	56

D.2 Juror Decisions and Model Scores

Experts define what analysis each task requires. Jurors check whether the report performs that analysis and whether its evidence supports it. The same requirements apply to every model and trial.

Jurors score those requirements rather than rank reports by overall preference. A hit means the report addresses a required move. A taint means the juror objects to the evidence supporting that analysis. Equation 1 combines the requirement-level decisions by accepting majority hits without majority taint.

Coverage assessment reads the whole report against every required move. For each cited sentence, the verifier compares the text up to its final citation with the referenced source excerpts. The code flags invalid file paths or source locations. If it cannot parse any citations in a report, it sends each coverage judge a warning about the missing citations. An evidence objection does not automatically give the whole report zero credit. The scoring rule also does not require a separate support vote for every factual premise. A report can satisfy a move by inference without an explicit citation, while the verifier checks its supporting premises. Text after the final citation and uncited premises are not independently checked. The score therefore measures whether the required analysis is present with verified citations, rather than exhaustive factual correctness.

Across all 3,036 responses, standalone untainted coverage is 61.96% for Opus, 43.39% for Gemini, and 37.60% for Sol. These are individual jurors' scores, not the performance of their corresponding generating models. Equation 1 combines the jurors' votes on each requirement, rather than averaging their overall coverage scores. Table 8 reports the full juror-by-model matrix. Table 13 shows how model rankings change when a juror is removed.

Table 8: Coverage awarded by each juror after its own evidence deductions (%). Each generating model contributes 276 reports, assessed by all three jurors.

Generating model	n	Gemini	Opus	Sol
GPT-5.6 Sol	276	52.89	73.60	48.74
Claude Opus 5	276	48.87	69.05	38.42
Claude Fable 5.1	276	50.22	68.33	41.91
Grok 4.6	276	51.93	69.31	43.12
Qwen3.8 Max	276	48.17	65.28	34.49
GPT-5.6 Terra	276	44.14	65.80	43.23
GLM 5.3	276	41.80	63.32	40.74
Muse Spark 1.3	276	41.48	60.89	37.68
Kimi K3	276	38.08	53.82	31.03
DeepSeek V4 Pro	276	32.38	49.89	30.39
Gemini 3.8 Flash	276	27.32	42.28	23.86

We also check whether jurors favor answers from their own models. For each answer, we subtract the other two jurors' average coverage from that juror's score, compare this difference on own-model versus other-model answers within each task, and average the results across tasks. The relative differences are $+0.71$ percentage points for Gemini, $+4.33$ for Opus, and $+0.63$ for Sol. This calculation uses three-juror responses and leaves the aggregation rule unchanged. Opus gives relatively more credit to answers from its own model. This comparison alone cannot tell whether that reflects self-preference or differences in the answers and how they are graded.

D.3 Blind Expert Assessment

The blind assessment compares jury decisions with expert judgments on selected reports. Two experts reviewed eight anonymized reports from seven models across six environments and six companies, drawn from a 22-report packet. For each model, we selected the report whose unrounded coverage was closest to 60%. Five of the reviewed reports came from this group. The other three came from a second group, sampled from a different eligible task for each model using seed 20260910. This includes decisions near the joint-success threshold alongside reports on other assignments. We shuffled report order and replaced model identities with anonymous identifiers. Neither expert saw model identities or automated scores.

The experts are financial professionals who independently assessed analytical presence and factual support. Each expert spent approximately 10-12 hours on the review, or 20-24 hours in total. Together, they made 197 judgments on 110 distinct rubric requirements. Some requirements were reviewed by both experts. Expert A assessed 102 requirements across seven complete reports. Expert B assessed

Table 9: Blind expert assessment. Expert A provided 102 ratings and Expert B provided 95. Seven partial ratings from Expert B are excluded from binary analytical comparisons. When comparing both analysis and evidence support, we exclude cases where the analysis is present but the expert could not determine whether it was supported. Agreement is measured on the same items in each row.

Comparison	Agree / compared	Agreement (%)	Cohen's κ
A analytical presence vs. final jury	78/102	76.5	0.534
B analytical presence vs. final jury	69/88	78.4	0.531
A vs. B analytical presence	57/81	70.4	0.404
Human consensus vs. final jury	51/57	89.5	0.763
A presence and support vs. final jury	71/94	75.5	0.516
B presence and support vs. final jury	67/87	77.0	0.505

Table 10: Agreement between each expert and the individual jurors or majority jury. The expert ratings include only definite decisions about whether the required analysis is present. Each column uses a fixed set of items: 102 for Expert A and 88 for Expert B. Juror decisions include their factual-support penalties.

Judge	Expert A: count (%)	Expert B: count (%)
Gemini 3.8 Flash	80/102 (78.4)	62/88 (70.5)
Claude Opus 5	73/102 (71.6)	73/88 (83.0)
GPT-5.6 Sol	71/102 (69.6)	63/88 (71.6)
Majority jury	78/102 (76.5)	69/88 (78.4)

95 across six complete reports and eight requirements in a seventh. Both gave definite ratings on 81 requirements. The primary comparison excludes Expert B's seven partial judgments. Blank entries are not counted as failures, and unreviewed items are excluded.

Table 9 compares final jury credit with the experts' definite analytical-presence decisions. The jury agrees with Expert A on 78/102 decisions and with Expert B on 69/88. On the shared 81 requirements, the experts agree on 57/81 (70.4%), while the jury agrees with each expert on 63/81 (77.8%). Expert B credits 22 requirements that Expert A rejects, whereas Expert A credits two that Expert B rejects. The experts agree on 57 requirements, and the jury agrees with them on 51. In the other six cases, it withholds four credits the experts award and awards two credits they reject. Agreement on this selected consensus subset is not an accuracy estimate for all requirements.

Disagreements usually came down to whether an implicit argument or related evidence was sufficient to meet a rubric requirement. For example, Expert B credited broad discussions of customer retention, whereas the jury insisted on the specific mechanism of expanding deployments within a fleet. Expert A was generally stricter than Expert B. Compared to Expert A, the jury was more lenient, awarding credit on 17 requirements that A rejected and withholding credit on seven that A approved. Against the more generous Expert B, the jury withheld credit on 17 requirements B approved and awarded credit on only two that B rejected.

We also evaluate agreement when requiring both analytical presence and verified factual support, excluding cases with unresolved support or partial judgments. This gives 94 comparisons for Expert A and 87 for Expert B, with 75.5% and 77.0% agreement, respectively (Table 9). The comparison uses the experts' support assessments and does not measure exhaustive citation-checking accuracy.

Two further comparisons show how support penalties and partial ratings affect agreement. Before factual-support penalties, the jury's analytical-presence votes agree with Expert A on 78/102 items and with Expert B on 71/88. The two additional agreements with Expert B concern analysis the jury recognized but did not credit because of supporting claims. We also test two ways to treat the seven partial judgments from Expert B. Counting all seven as no credit gives 75/95 agreement (78.9%). Counting them all as credit gives 70/95 (73.7%). Table 10 reports individual jurors alongside the majority.

D.4 Qualitative Examples of Jury Decisions

The saved assessments show why mentioning the required analysis does not always earn credit. Table 11 follows report claims through source evidence and juror objections to requirement-level decisions, covering unsupported premises, a guidance-period error, and supported inferences that retain credit.

In Kimi K3's Samsara sales-efficiency report, the answer states that "carrier bankruptcies and capex cuts among trucking customers should pressure the transportation vertical." The Opus coverage judge identifies the evidence gap explicitly: "this assertion is uncited." The report also attributes transportation growth to "a tightening freight market," whereas its cited passages establish vertical growth, efficiency, and safety, not the asserted freight-market conditions. All three judges, Opus, Sol, and Gemini, recognize the freight-recession analysis and mark requirement 3a as tainted. The recorded decision is therefore **three analytical hits, three evidence objections, and zero credit for the requirement**. These errors demonstrate how the jury separates analytical coverage from factual grounding: the report was penalized for both an uncited claim and a cited claim that the sources did not support.

Table 11: Evidence checks can withhold credit for present analysis or retain credit for supported inferences. Hits denote analytical-presence votes. Taints denote votes flagging supporting evidence. Both are counts out of three jurors. Samsara receives no credit despite three hits, whereas the supported Getty moves retain credit. The FND row reports the two inspected verifier decisions on a guidance-period mismatch, not a complete panel outcome.

Case	Claim and evidence check	Observed decision
Samsara: freight conditions	An uncited bankruptcy/capex premise accompanies a freight-market characterization not established by its citations.	3 hits, 3 taints; no credit.
FND: guidance period	Full-year comparable-sales guidance is applied to the first quarter.	Opus and Sol flag the period mismatch.
Getty: deleveraging	Cash-flow-supported deleveraging is consistent with the supplied call's stated merger rationale.	3 hits, 0 taints; credit.
Getty: cost synergies	The supplied call explicitly identifies cost synergies as the transaction rationale.	3 hits, 0 taints; credit.

An FND report shows a different error: the agent applied the -2% comparable-sales guidance floor to the first quarter. The cited call gives *full-year* guidance of -2% to $+1\%$, while its separate first-quarter discussion describes an initial model that was only slightly negative. Opus and Sol both flagged the period mismatch. They also rejected the claim that management's forecast necessarily put first-quarter sales below -2%. The verifiers therefore checked the citation's period and numerical implication, not just whether a relevant source was present.

To examine decisions that retained credit, we also inspected six majority-credited moves in three reports on the Getty-Shutterstock merger's strategic rationale. We selected two per report, prioritizing uncited factual statements and conclusions. This AI-assisted inspection was separate from the blind expert assessment. We found support for the core argument in four moves. Support for the other two remained unresolved. These selected examples show how the scoring works. They do not measure how often the jury makes errors.

For example, Sol's cash-flow-supported deleveraging argument received three analytical votes and no taint votes. Getty's supplied 2024 Q4 earnings call explicitly links the merger to deleveraging and increased cash flow (lines 61–62), supporting the inference. Gemini's cost-synergy argument likewise received three analytical votes and no taint votes: the supplied 2025 Q3 call identifies cost synergies as the transaction's rationale (lines 51–52). Its additional suggestions about cloud vendors and advertising auctions were not established by that passage, but were not necessary to satisfy the core move.

The unresolved cases concerned Fable's assertion of Shutterstock's historically lower leverage and Gemini's attribution of a below-$3.5\times$ leverage target to management. Both received three analytical votes and no taint votes. In the latter case, the nearby citation establishes Getty's estimated \$194 million cash-interest expense, not the claimed management target (2026 Q1 call, lines 155–158).

The inspected evidence did not establish either assertion's support or falsity. Together with the FND example, these cases distinguish detected factual errors, supported analytical inferences, and credited factual assertions requiring further source confirmation.

D.5 Citation Frequency and Research Coverage

We examine citation frequency to assess whether coverage primarily reflects the volume of citations in a report. The analysis includes 307,178 citation-unit occurrences across all 3,036 evaluated answers. A citation unit references one file and locator. Repeated references count separately, while unique-file counts measure source breadth. Counts use the scorer's paragraph normalization and citation parser. Table 12 reports all eleven models. Qwen averages 245.8 citations per answer and 53.54% coverage, whereas Sol averages 85.6 citations and 59.17% coverage. Across answers, citation count has only a weak positive correlation with final coverage ($r = 0.104$).

We test whether citation density is associated with coverage after accounting for report length and task composition. The regression includes task, model, and trial fixed effects, plus linear and squared terms for log prose length. A common tokenizer measures length, with citation strings excluded from prose length. Density is citations per 1,000 prose tokens. Standard errors cluster by task. We test citation density, log citation count, and the number of distinct cited sources against two outcomes: analysis recognized by a majority of jurors and final credited coverage. We apply Holm correction across all six tests.

Citation density has a small negative association with final coverage after adjustment (partial r = -0.101; Holm-adjusted $p < 0.001$). Ten additional citations per 1,000 prose tokens correspond to 4.12 percentage points lower coverage. The association with majority analytical presence before factual-support penalties is approximately zero (partial $r = -0.012$). Comparing repeated answers within the same model and task gives corresponding partial correlations of -0.094 and -0.003. The final-coverage association remains significant after correction ($p = 0.009$). Citation volume alone therefore does not explain research coverage, but these observational comparisons do not identify the effect of adding or removing citations.

Table 12: Citation behavior on all 3,036 frontier answers (276 per model). Cites is the average number of citations per answer, counting repeats. Files is the average number of distinct source-file paths cited. Density is the average number of citations per 1,000 final-answer tokens, measured with the same tokenizer for all models. The regression uses a different denominator: prose tokens with the citation text removed. For each verifier and answer, we calculate the fraction of checked citations involved in an evidence objection. Objections reports the average of these fractions. It does not measure factual errors established by an independent reviewer.

Model	Effort	Cites	Files	Density	Objections (%)	Coverage (%)
GPT 5.6 Sol	max	85.6	9.8	14.79	8.30	59.17
Grok 4.6	xhigh	133.9	16.0	13.75	17.60	58.25
Claude Opus 5	max	144.5	19.2	14.21	18.88	55.89
Claude Fable 5.1	high	99.6	15.8	12.86	18.91	55.85
Qwen3.8 Max	xhigh	245.8	20.6	14.65	21.33	53.54
GPT 5.6 Terra	max	60.2	7.8	13.60	5.91	50.86
GLM 5.3	max	52.3	9.6	14.01	5.75	47.90
Muse Spark 1.3	xhigh	86.4	9.4	15.59	11.31	47.00
Kimi K3	max	60.6	10.8	13.21	32.56	42.92
DeepSeek V4 Pro 0813	max	34.5	7.6	13.08	11.25	37.26
Gemini 3.8 Flash	high	109.5	15.0	13.76	44.48	32.35

D.6 Citation-Processing Sensitivity

To assess sensitivity to citation parsing, we checked explicit file and locator references. This identified 179 reports across Hedge-Bench-Public, Hedge-Bench-Heldout, and Hedge-Bench-Private for targeted review. We reran citation verification and dependent coverage with the same three jurors on 155 reports whose source excerpts could be matched unambiguously. Twenty-four retained their original grades. No answer, rubric, or forecast label changed. The check changed 78 coverage

scores: 67 on Public, nine on Heldout, and two on Private. Nineteen reports crossed the strict $> 60\%$ threshold (16, two, and one, respectively). On the 3,036 Public reports, pooled joint success changed from 527 to 531, leaving the main comparison intact. This check measures score sensitivity to citation parsing.

We have implemented and tested a citation tool that the public release will use in place of free-form file-path citations. The tool validates the selected source location and returns a citation ID tied to the excerpt. The grader reconstructs that excerpt from the frozen task sources, avoiding the free-form citation extraction issues described above. The jury continues to assess whether the evidence supports the claim.

D.7 Panel and Aggregation Sensitivity

To test whether model orderings depend on panel composition or aggregation, we recompute scores for all 3,036 responses. Results average the 276 responses within each model. Removing each juror in turn retains Spearman correlations of 0.964–0.973 for coverage ranks and 0.736–0.909 for joint-success ranks across the eleven models (Table 13). With two jurors, both must recognize a move for it to receive credit. Removing a juror can also remove an evidence objection and restore credit. Scores can therefore change in either direction.

We also test a variant where we count a move as successful only if at least two jurors independently find it present and untainted. This contrasts with our original rule, which takes separate majority votes on presence and taint. Coverage and joint-success rankings retain correlations of 0.991 and 0.982, respectively. Coverage ordering is more stable than the thresholded joint endpoint under panel removal.

We compare each variant's model ranking with the original three-juror ranking using Spearman correlation, accounting for ties.

Table 13: Model-ranking sensitivity to juror removal and aggregation. Entries are tie-aware Spearman correlations across eleven models relative to the original three-juror rule, with all variants scored on the same 3,036 responses. Joint success requires a correct prediction and coverage strictly above 60%.

Alternative panel or rule	Coverage	Joint success
Without Gemini	0.964	0.909
Without Opus	0.973	0.736
Without Sol	0.973	0.782
Majority of individually untainted hits	0.991	0.982

E Joint Success and Question-Only Results

In this appendix, we first vary the coverage threshold used for joint success. We then compare full-environment and question-only forecasts, and examine results by outcome date. All analyses use saved responses from the Hedge-Bench-Public evaluation.

Joint success combines a correct forecast with a minimum level of completed analysis. We calculate it from saved prediction verdicts and exact accepted-move counts, using the same rule for every model and weighting the three trials equally. A report earns credit for a whole number of moves. Two different percentage thresholds can therefore require the same number of credited moves. With 12 moves, both $> 60\%$ and $\geq 62.5\%$ require eight credits. With 15 moves, both require ten.

Table 14 varies the coverage threshold while continuing to require a correct prediction. The table also reports results for the weighted rule $0.6P + 0.4C \geq 0.85$, where P is binary correctness and C is fractional coverage. Passing this rule requires a correct forecast and at least 62.5% coverage.

The component scores also differ at the response level. Across all 3,036 responses, 853 (28.10%) exceed the 60% coverage threshold. Their prediction accuracy is 531/853 (62.25%), compared with 1,519/2,183 (69.58%) for the remaining reports (Table 15). In this sample, reports above the coverage threshold are not more likely to predict correctly. This pooled comparison mixes models

Table 14: Joint-success sensitivity across all three trials. Every threshold additionally requires a correct prediction; all columns use 276 answers per model.

Model	$> 50\%$	$> 60\%$	$\geq 62.5\%$	$> 70\%$
GPT 5.6 Sol	47.46	31.52	31.16	13.41
Claude Opus 5	43.12	28.99	27.90	14.13
Claude Fable 5.1	39.13	26.81	25.72	13.04
Grok 4.6	40.58	26.45	26.09	14.86
Qwen3.8 Max	40.22	25.72	24.64	13.04
GLM 5.3	25.36	14.13	13.41	4.35
GPT 5.6 Terra	22.46	13.77	13.04	5.80
Muse Spark 1.3	22.10	9.06	8.70	2.90
Kimi K3	20.29	8.70	8.33	3.26
Gemini 3.8 Flash	10.87	4.71	4.71	2.54
DeepSeek V4 Pro 0813	7.25	2.54	2.54	1.45

and tasks. We retain both endpoints because they measure different aspects of a report: completing expert-defined analysis and making an accurate forecast.

Table 15: Research coverage and prediction correctness across 3,036 full-environment responses: eleven models, 92 Hedge-Bench-Public tasks, and three trials. Cells count responses, not distinct tasks. The coverage gate uses exact credited-move fractions. The 531 responses with coverage $> 60\%$ and a correct prediction satisfy joint success.

	Correct prediction	Incorrect prediction	Total
Coverage $> 60\%$	531	322	853
Coverage $\leq 60\%$	1,519	664	2,183
Total	2,050	986	3,036

E.1 Question-Only Predictions

The question-only condition measures prediction accuracy without research documents or internet access. Table 16 includes all three trials, with 276 valid responses per model and 92 per trial. Accuracy is total correct divided by 276, equal to the mean of the three trials. These responses contain no research reports and therefore have neither coverage scores nor joint-success rates. Figure 6 compares prediction accuracy across both conditions. Full-environment runs allow multi-turn tool interaction, so the score difference cannot be attributed to document access alone.

Table 16: Question-only prediction accuracy over all three trials. Each trial contains 92 valid answers per model, and the aggregate uses all 276. Effort denotes the configured reasoning setting. These results show forecasting without the research documents. Because interaction budgets also differ from the full-environment runs, the comparison does not isolate the effect of document access.

Model	Effort	Q1	Q2	Q3	Aggregate (%)
GPT 5.6 Sol	max	61/92	61/92	59/92	65.58
Claude Opus 5	max	61/92	55/92	60/92	63.77
Claude Fable 5.1	high	64/92	61/92	61/92	67.39
Grok 4.6	xhigh	56/92	52/92	56/92	59.42
Qwen3.8 Max	xhigh	55/92	53/92	55/92	59.06
GLM 5.3	max	53/92	54/92	60/92	60.51
GPT 5.6 Terra	max	58/92	60/92	54/92	62.32
Muse Spark 1.3	xhigh	56/92	59/92	56/92	61.96
Kimi K3	max	54/92	56/92	62/92	62.32
Gemini 3.8 Flash	high	50/92	54/92	51/92	56.16
DeepSeek V4 Pro 0813	max	52/92	48/92	56/92	56.52

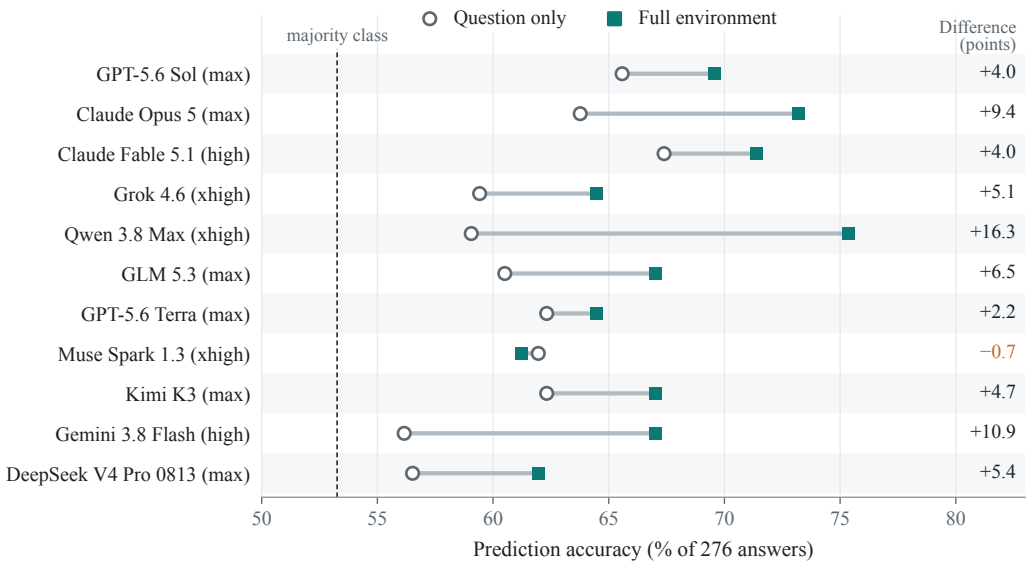

Figure 6: **Ten models have higher observed prediction accuracy in the full environment.** Each point aggregates three trials on all 92 Hedge-Bench-Public tasks (276 predictions per model and condition). Open circles show question-only accuracy, and teal squares show full-environment accuracy. The right column gives full-environment minus question-only accuracy in percentage points, and the dashed line marks the 53.26% majority-class baseline. Labels identify models and configured effort. Because the two conditions differ in tool interaction, the comparison does not isolate the effect of document access.

E.2 Resolution Timing

We examine resolution timing in two ways: a common calendar split and model-specific subsets defined by documented knowledge cutoffs. The calendar split in Table 17 divides Hedge-Bench-Public at March 31, 2026, retaining all three trials on 23 earlier and 69 later tasks. Table 35 gives the complete chronology. Ten of eleven models score higher on the later subset. GPT-5.6 Sol rises from 57.97% to 73.43%, for example, so later-resolving tasks are not uniformly harder. This comparison uses different tasks on either side of a common calendar boundary, not a shared training-data cutoff. Recorded resolution dates refer to disclosures, events, or the end of a resolution window, rather than the earliest date of public knowledge.

The model-specific analysis instead evaluates outcomes disclosed after each model's documented knowledge cutoff. The tasks in each subset depend on the model's cutoff, rather than random sampling. We therefore report the number of tasks, the Yes/No balance, and accuracy for each subset. We determine recency from the date of outcome disclosure rather than when the task was assembled. The fixed historical benchmark permits comparison on the same questions and observed outcomes. In contrast, prospective forecasting requires waiting for new events to resolve. We complement the historical set with question-only controls, documented-cutoff comparisons, and monthly private additions from ongoing analyst-call collection.

Table 19 lists the documented cutoffs. These include February 16, 2026 for GPT-5.6 Sol and Terra, March 2026 for Gemini 3.8 Flash, May 2026 for Claude Opus 5, and June 2026 for Claude Fable 5.1 (OpenAI, 2026a;b; Google DeepMind, 2026; Anthropic, 2026a). We compare these cutoffs with public outcome disclosures, not task dates or fiscal-period ends. When only a month is specified, the outcome must occur after its final day to enter the post-cutoff subset.

Public disclosures establish decisive outcome dates for 39 tasks. If a forecast requires a condition to hold in every report, one report that violates it resolves the forecast as No. If the forecast requires the condition to hold in either report, one qualifying report resolves it as Yes. Table 18 includes only outcomes disclosed after each model's documented cutoff. Each model is evaluated on a different subset, and the Opus and Fable subsets are especially small. These scores should not be read as a

Table 17: Full-environment prediction accuracy on Hedge-Bench-Public before and after a common resolution-date boundary. The split is through March 31, 2026 versus April 1, 2026 onward. Each model contributes three trials on 23 earlier tasks (69 predictions) and 69 later tasks (207 predictions). Counts and percentages use all attempts in each subset. The subsets contain different tasks, and recorded resolution dates refer to disclosures, events, or window cutoffs rather than the earliest date of public knowledge.

	Through March 31		After March 31	
Model	Correct / 69	Accuracy (%)	Correct / 207	Accuracy (%)
Claude Opus 5	48	69.57	154	74.40
Qwen 3.8 Max	54	78.26	154	74.40
Claude Fable 5.1	44	63.77	153	73.91
GPT-5.6 Sol	40	57.97	152	73.43
Gemini 3.8 Flash	39	56.52	146	70.53
GLM 5.3	42	60.87	143	69.08
Kimi K3	45	65.22	140	67.63
GPT-5.6 Terra	40	57.97	138	66.67
Grok 4.6	40	57.97	138	66.67
DeepSeek V4 Pro	37	53.62	134	64.73
Muse Spark 1.3	41	59.42	128	61.84

ranking on the same tasks. This analysis evaluates documented post-cutoff outcomes, though we cannot verify the contents of proprietary training data.

Table 18: Prediction accuracy on tasks whose outcomes were confirmed by public sources after the model's documented knowledge cutoff. Month-only cutoffs use the final day of the month. Eligible subsets differ by model, and each task performs three trials per setting.

Model	Tasks	Full correct/total	Full (%)	Q-only correct/total	Q-only (%)
GPT-5.6 Sol	34	65/102	63.73	52/102	50.98
GPT-5.6 Terra	34	49/102	48.04	46/102	45.10
Grok 4.6	36	57/108	52.78	48/108	44.44
Gemini 3.8 Flash	32	52/96	54.17	47/96	48.96
Claude Opus 5	10	23/30	76.67	22/30	73.33
Claude Fable 5.1	7	15/21	71.43	16/21	76.19

Table 19: Provider-documented knowledge cutoffs used to define model-specific post-cutoff subsets, checked September 9, 2026. "Not stated" means the cited official source does not specify a cutoff, and release dates are not substituted. Month-only cutoffs use the month's final day. These are provider knowledge cutoffs, not independently audited training-data boundaries.

Model	Documented cutoff	Official source
GPT-5.6 Sol	February 16, 2026	OpenAI (2026a)
GPT-5.6 Terra	February 16, 2026	OpenAI (2026b)
Grok 4.6	February 1, 2026	xAI (2026)
Gemini 3.8 Flash	March 2026	Google DeepMind (2026)
Claude Opus 5	May 2026	Anthropic (2026b)
Claude Fable 5.1	June 2026	Anthropic (2026a)
Qwen 3.8 Max	Not stated	Qwen Team (2026)
GLM 5.3	Not stated	Z.AI (2026)
Kimi K3	Not stated	Moonshot AI (2026)
DeepSeek V4 Pro	Not stated	DeepSeek AI (2026)
Muse Spark 1.3	Not stated	Meta AI (2026)

F Training Configuration

F.1 Iterative Judge Prompt Tuning

We improved the per-turn credit assignment by tuning judge prompts by inspecting failures in short training probes. We used coding agents to review sampled reports and individual credit and blame assignments and revised the objective or attribution instructions. They then reassessed saved trajectories before the next experiment. We call this human-directed, coding-assisted procedure meta-prompting. GEPA similarly uses trajectory feedback to refine prompts (Agrawal et al., 2025). We fixed the judge instructions through this iterative process before the reported training run. The run used 21 updates to cover the 165 environments once, with three extra task groups to fill the final batch. We compare the resulting training method with outcome-only and magnitude-matched controls on held-out tasks that were not used to develop the prompts.

Our frontier evaluation found that higher research coverage did not consistently imply higher forecast accuracy (Table 15). In preliminary runs, fixed mixtures of binary forecast correctness and research coverage did not consistently improve both metrics. This motivated our use of a synthesis-aware reward, alongside prior work on reward interference (Ramesh et al., 2026) and reward composition (Li et al., 2024).

We compare a reconstructed weighted reward, $0.8C + 0.2P$, with the holistic objective, where C is coverage and P the forecast score. Across the same 384 trajectories from 48 tasks, the reconstructed reward's correlation with binary correctness was 0.4904, compared with 0.9074 for the holistic objective. This comparison checks whether its scores follow the intended reward rule: The higher correlation shows that the holistic reward tracks forecast correctness more closely, as intended. A report can cover much of the required research while reaching the wrong forecast, as illustrated in Appendix L.2. The synthesis judge therefore assesses the strongest relevant evidence on both sides, the connection between analysis, conclusion, and the forecast. A wrong prediction must score below 0.5 (Appendix K). A correct prediction can also score below 0.5 if the research or synthesis is materially incomplete. High scores require both substantial accepted coverage and strong synthesis.

During prompt tuning, inspection by coding agents led to three attribution rules. First, assess evidence gathering separately from later inference: sound retrieval is not blamed simply because the forecast is wrong. Second, distinguish reasonable exploration that finds no match from repetition that adds no information. Third, assess a report-writing turn by its claims and conclusions, not by the act of writing. These rules address failures observed in the sampled trajectories. Once fixed, the instructions are applied retrospectively by the reward judge.

F.2 Policy Loss and Token Weights

The policy loss applies each turn's assigned advantage to its eligible assistant tokens. Let a_{it} be a sampled assistant token and h_{it} its preceding conversation and token history, including tool observations. The ratio $\rho_{it} = \pi_\theta(a_{it} \mid h_{it})/\pi_{\text{roll}}(a_{it} \mid h_{it})$ compares the current policy with the policy that generated the rollout. The advantage is A_{it} from Equation 4. With eligibility mask m_{it} and clipping width $\epsilon = 0.2$, the policy surrogate is

$$\mathcal{L} = -\frac{\sum_{i,t} m_{it} \min\{\rho_{it} A_{it}, \text{clip}(\rho_{it}, 1-\epsilon, 1+\epsilon) A_{it}\}}{\sum_{i,t} m_{it}}. \tag{5}$$

The denominator includes eligible tokens with zero attribution. Each eligible token in a turn receives the same coefficient, without division by turn length, so a longer credited turn contributes more terms to the loss. The configuration uses no learned value critic and no reference-policy KL penalty.

The judge decides whether each action receives positive or negative weight. The absolute difference between the trajectory score and its group mean sets the overall scale. The token weights therefore need not sum to zero within a group. These turn weights change the training objective. They are not an unbiased estimate of the policy gradient obtained from the original trajectory reward.

For turn j of trajectory i in task group g, r^+_{gij} and r^-_{gij} are the judge's reasoning success and failure weights, each in $[0, 1]$. Signed tool utility is $u_{gij} \in [-1, 1]$. Check Algorithm 1 for credit assignment and policy update.

Algorithm 1 TurnCredit reward construction and policy update.

Require: Tasks $x_1, \ldots, x_8$, rollout policy π_{roll}, current policy π_θ, training-only rubrics and outcomes, reward judge J, clipping width $\epsilon = 0.2$.

for $g = 1, \ldots, 8$ **do**
  Sample $\{\tau_{gi}\}_{i=1}^{8} \sim \pi_{\text{roll}}(\cdot \mid x_g)$ using the Bash agent.
  for $i = 1, \ldots, 8$ **do**
    if τ_{gi} has a terminal, empty, or missing answer **then**
      Apply terminal score -0.5 and the applicable loss mask (Appendix F.4).
    else
      Grade the final answer against the hidden rubric and resolved outcome.
      Draw two synthesis judgments; average valid scores to obtain s_{gi}.
      Draw two turn attributions using the completed trajectory and synthesis diagnoses.
      Validate and aggregate the success, failure, and tool-utility channels separately.
    end if
  end for
  if the group has terminal siblings **then**
    Set $\bar{s}_g$ to the nonterminal-score mean (zero if none).
  else
    Set $\bar{s}_g = \frac{1}{8} \sum_{i=1}^{8} s_{gi}$.
  end if
  For each ordinary rollout, set $\Delta_{gi} = s_{gi} - \bar{s}_g$.
  for each ordinary trajectory i and assistant turn j **do**
    For a report-writing turn, set $u_{gij} \leftarrow 0$; retain its reasoning weights.
    $w^{+}_{gij} \leftarrow \max(r^{+}_{gij}, u_{gij}, 0)$; $w^{-}_{gij} \leftarrow \max(r^{-}_{gij}, -u_{gij}, 0)$.
    Assign $A_{git} = |\Delta_{gi}|(w^{+}_{gij} - w^{-}_{gij})$ to each eligible token t in turn j.
  end for
end for
Mask prompt and tool-observation tokens; keep zero-attribution eligible tokens in the denominator.
Update π_θ with the eligible-token-mean clipped loss in Equation 5.

Table 20: Optimization and sampling settings for retrospective turn-credit training. The loss averages over eligible tokens, including those with zero attribution.

Setting	Value
Schedule	21 updates; 8 tasks per update; 8 rollouts per task
Policy batch	64 trajectories; microbatch 1; one optimizer step per rollout batch
Optimizer	Adam; constant learning rate 10^{-5}; $\beta_1 = 0.9$, $\beta_2 = 0.98$; weight decay 0
Precision and clipping	BF16; gradient clipping 1.0; policy-ratio clipping $[0.8, 1.2]$
Adapter	LoRA rank 16, scaling 32; attention Q/K/V/output and MLP up/gate/down projections
Loss reduction	Eligible-token mean; no group-standard-deviation normalization
Regularization	Reference-policy KL coefficient 0; entropy coefficient 0
Policy sampling	High reasoning effort; temperature 1
Reward draws	Two objective draws, followed by two attribution draws

F.3 Reward Inputs and Attribution

Algorithm 1 uses two judging stages, both using Gemini 3.7 Flash with different prompts. The report judge scores the completed report and assesses how its analysis supports the forecast. The attribution judge then uses the full trajectory together with that score and assessment to assign reasoning credit or blame to individual turns and score their tool use. We request two assessments at each stage. Each turn-level assignment includes the turn number and an explanation; turns with zero weight are omitted. Appendix K gives the full system prompts and input-message fields

Both stages run after generation and leave the completed trajectory unchanged. The objective receives the task instruction, hidden grading rubric, final answer, authoritative grading verdict, and objective context. Attribution receives those inputs plus the averaged objective score, synthesis diagnoses, and ordered turn records. Each record contains its index, assistant content, tool calls, and subsequent tool

outputs. The judge can therefore assess an early action using later turns and the final answer. The acting policy sees neither the grading verdict nor the hidden rubric.

We average valid objective draws and validate and aggregate each attribution channel separately. If a required attribution field is missing or malformed, we ask the judge to repair its response, with up to three repair calls. We do not fill in the missing field by assigning uniform credit. Judge temperature inherits the provider default.

The attribution judge prompt distinguish evidence gathering from its later use. A relevant search with no match is neutral; retrieval that supports later analysis and verification that identifies or repairs a real defect can earn credit. Repeated calls receive negative tool utility when they add no information. Sound evidence gathering is not blamed solely because later synthesis reaches a wrong conclusion. Report-writing turns receive no tool-utility weight but remain eligible for reasoning credit or blame.

Unlisted turns receive zero weight. We aggregate reasoning and tool-utility channels independently across usable draws, separating positive and negative tool utility before aggregation. Equation 4 takes the elementwise maximum of the positive maps and, separately, the negative maps to avoid additive double counting. Token positions identify which assistant tokens belong to each turn. The loss mask excludes tokens that are not training targets. There is no additional turn-length normalization. Appendix G.2 illustrates these assignments with a qualitative example.

F.4 Terminal Handling

In the final configuration, malformed tool calls and empty terminal outputs receive a reward of -0.5. The same penalty applies when the agent stops without a valid `answer.md`. These failures bypass the report and attribution judges and receive negative weight across their eligible assistant tokens.

Terminal failures are excluded when computing the group baseline. The baseline is the mean score of the remaining nonterminal rollouts, or zero if none remain. Failed rollouts therefore do not lower the baseline for the other trajectories.

Budget limits are handled separately. Rollouts that reach a turn or token limit, or exhaust the context window, without a valid answer are excluded from the policy loss. If a valid answer already exists, we retain it for scoring; any assistant suffix that caused the truncation is excluded from the loss.

G Credit Assignment and Behavioral Evidence

G.1 Model Behavior

We first compare training behavior under outcome-only and turn-level supervision, then examine saved credit assignments and a separately trained magnitude-matched control. Both forms of supervision use the same group-centered, synthesis-aware objective. Outcome-only training assigns signed Δ_i to every eligible token; turn-level training assigns action-specific signs and magnitudes through Equation 4. The initial comparison therefore includes differences in update scale as well as credit assignment.

The high-effort outcome-only evaluation remained incomplete because trajectories repeatedly reached turn or context limits despite the higher budgets described in Appendix H.1. We report its completed low- and medium-effort evaluations.

Similarly, outcome-only trajectories are longer and more repetitive during training. Over updates 16–20, unscaled outcome-only GRPO averages 198.7 turns, compared with 99.6 for turn-level training, and repeats 12.59% of commands compared to 1.23%. Magnitude matching reduces (compared to Outcome only) these to 120.3 turns and 3.95% repeated commands (Table 21). Compared with unscaled outcome-only GRPO, turn-level coverage is higher in 39 of the 40 matched environment groups across these five updates.

Magnitude-matched trajectories remain shorter than unscaled outcome-only trajectories, but their turn count rises late in training while token entropy falls (Figure 10B–C). From update 16 to 20, mean turns increase from 107.9 to 134.8 while entropy falls from 0.187 to 0.143. Both outcome-only controls use more turns late in training. TurnCredit keeps its turn count comparatively stable while using more tokens per turn, alongside higher research coverage (Figure 10). Fewer turns do not

necessarily mean fewer tokens: magnitude matching uses fewer trajectory tokens than turn-level training over these updates. Table 21 compares all three policies during training.

Table 21: Training behavior over updates 16–20. Means use 320 trajectories per policy. Trajectory tokens include accepted assistant output and tool observations; repeated commands are exact repeats.

Policy	Mean turns	Mean trajectory tokens	Repeated commands (%)
Outcome-only GRPO	198.7	70,365	12.59
Magnitude-matched GRPO	120.3	55,141	3.95
TurnCredit	99.6	63,800	1.23

Our inspection suggests one reason for this behavior. Among outcome-only trajectories included in the loss, 609 were nonterminal and had positive centered advantage. Of these, 348 covered less than half the rubric. Uniform positive advantage reinforces every eligible action in these trajectories because their score exceeds the group baseline. Turn-level attribution can instead penalize faulty synthesis in an above-baseline trajectory and reward useful retrieval in a below-baseline one. We examine both cases below.

G.2 Examples of Credit Assignments

For each saved example, we compare the assigned turn-level advantage with the scalar coefficient that outcome-only training would broadcast to the same trajectory.

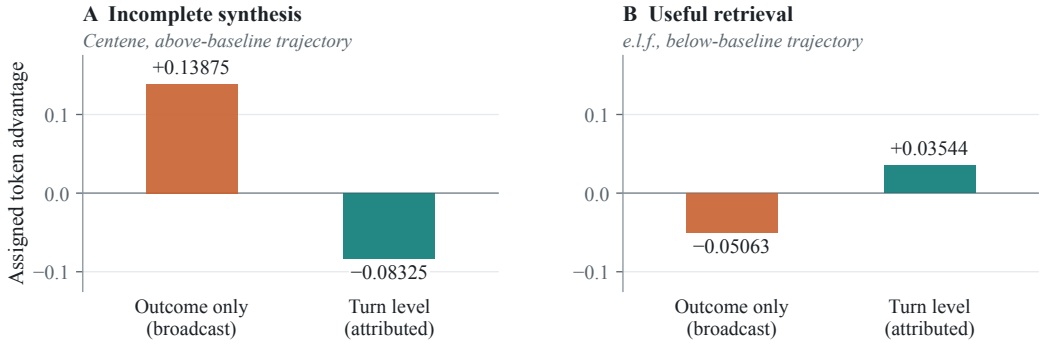

Figure 7: **Turn attribution can reverse the sign of episode-level credit for an individual action.** In A, incomplete synthesis receives negative advantage despite an above-baseline trajectory score. In B, useful retrieval receives positive advantage despite a below-baseline score.

Figure 7A shows a turn receiving negative credit despite an above-average report score. In the Centene example from batch 5, the forecast is correct, but the report covers only 14.3% of the rubric. Its score of 0.400 exceeds the group mean of 0.26125, giving $\Delta = +0.13875$. The judge still penalizes the answer-writing turn for missing analysis and citation failures, assigning a weight of -0.6 and an advantage of -0.08325.

Figure 7B shows the reverse. In the e.l.f. example from batch 6, the report scores below its group mean ($\Delta = -0.050625$). A retrieval about international growth still provides useful evidence, so the judge gives that turn a positive weight of 0.7 and an advantage of $+0.0354375$. The useful step is rewarded even though the overall score is below average.

The VeriSign rollout in Figure 8 shows how useful research can receive positive credit even when the forecast is wrong. The agent predicted that VeriSign would not disclose delegation of the `.web` domain in either of its next two quarterly reports; the recorded outcome was Yes. The judge described the forecast reasoning as "a plausible estimation based on post-hearing briefing schedules" and credited the reconstruction of litigation history. The episode score was $s_i = 0.38$ against a sibling baseline of 0.34125, giving $\Delta_i = +0.03875$.

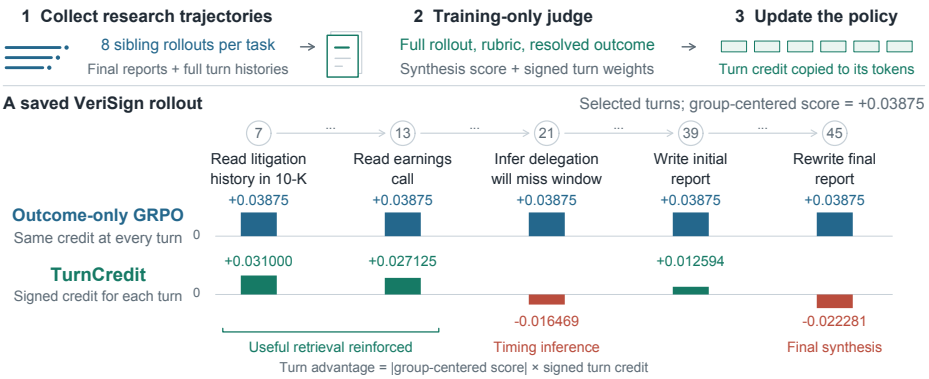

Figure 8: TurnCredit uses the completed rollout to distinguish useful retrieval from faulty inference and synthesis. The bars show token coefficients for selected turns of a saved VeriSign trajectory, compared with uniform outcome-only credit on that same trajectory.

Table 22 shows what the agent did at selected turns, with excerpts from its report and the credit each turn received. Reading the filing and earnings call received positive credit; inferring that procedural delays would extend beyond the forecast window received negative credit. The initial report write was credited, while the final rewrite received negative synthesis attribution. Thus the incorrect forecast did not give every research action negative weight.

Table 22: **Credit within a wrong-forecast trajectory.** Selected turns from one VeriSign rollout. Quoted report excerpts are verbatim; the remaining descriptions condense the saved turns. Turns are numbered from one; $c_{ij} = w_{ij}^{+} - w_{ij}^{-}$ combines reasoning attribution and tool utility. The last column is the coefficient applied to eligible assistant tokens.

Turn	Condensed action or output	c_{ij}	$\lvert\Delta_i\rvert c_{ij}$
7	Reads lines 600–620 of the 2025 10-K to reconstruct the litigation history surrounding `.web`.	$+0.800$	$+0.031000$
13	Reads lines 190–210 of the Q2 2025 earnings call for further context on `.web`.	$+0.700$	$+0.027125$
21	Reasons that post-hearing briefings in the first half of 2026 would be followed by a panel decision and further delegation delays. Infers that delegation is unlikely within the next two quarterly reports.	-0.425	-0.016469
39	Writes the initial report: “VeriSign frames the .web pursuit as a competitive initiative, not an optional experiment.” Chooses No, while acknowledging that a favorable decision could arrive quickly.	$+0.325$	$+0.012594$
45	Rewrites the report to add missing citations and shorten factual statements: “VeriSign describes the .web effort as a competitive initiative.” The submitted forecast remains No.	-0.575	-0.022281

The same pattern appears across 635 early training trajectories containing 11.65 million eligible tokens. In trajectories scoring above their group mean, 15.05% of tokens receive negative advantages. Conversely, 12.28% of tokens in below-average trajectories receive positive advantages. Commands that inspect the written answer usually receive zero advantage, covering 97.80% of their eligible tokens.

The training updates also differ in scale. On the first batch, outcome-only supervision produces a gradient norm 6.11 times that of TurnCredit. Does TurnCredit give repeated actions less weight simply because its advantages are smaller overall? To check this, we rescale the advantages within each saved trajectory to match the total absolute weight of outcome-only supervision. Repeated commands still receive only 22.2% of their outcome-only weight; answer inspections receive 4.4%.

Thus, even at the same total magnitude, TurnCredit assigns less weight to these actions. The next section tests magnitude matching during training.

G.3 Magnitude-Matched Outcome-Only Training

We train a magnitude-matched outcome-only policy to test whether smaller advantages explain TurnCredit's gains. Starting from the same base model, we train for 21 updates on 165 environments. Each update uses eight environments with eight rollouts each, giving a batch of 64 trajectories.

The reward judge computes a report score and turn-level weights for each rollout. We use these weights only to calculate one scale factor for the whole batch. This factor rescales the outcome-only advantages to match the mean absolute token advantage that TurnCredit would assign to the same rollouts:

$$s_B = \frac{\sum_{i,t} m_{it}|\widetilde{A}_{it}|}{\sum_{i,t} m_{it}|\Delta_i|}, \qquad A_{it}^{\text{control}} = s_B \Delta_i, \qquad \widetilde{A}_{it} = |\Delta_i|(w^+_{ij(t)} - w^-_{ij(t)}). \tag{6}$$

Here, m_{it} is one for tokens included in the policy loss and zero otherwise. The sums cover the complete 64-trajectory batch. The turn weights $w^+_{ij(t)}$ and $w^-_{ij(t)}$ follow Equation 4, with tool utility set to zero on answer-writing turns. Every eligible token within a trajectory receives the same rescaled advantage. The control therefore matches advantage magnitude while retaining outcome-only credit assignment. The training uses high-reasoning effort, Adam with learning rate 10^{-5} and betas $(0.9, 0.98)$, token-normalized loss, policy clipping at 0.2, gradient clipping at 1.0, and no KL or entropy bonus. Everything same as TurnCredit and Outcome only GRPO training.

Results: TurnCredit exceeds the magnitude-matched control by 11.67–14.82 coverage points across the three evaluated reasoning efforts, with all gains significant after Holm correction ($p < 0.001$; Table 23 and Figure 9). Reducing advantage magnitude alone therefore does not reproduce the improvement. At low effort, TurnCredit reaches 33.36% coverage with 28,662 trajectory tokens, versus 33.02% and 77,134 tokens for magnitude-matched high effort, once again demonstrating efficient performance of TurnCredit training.

Table 23: **Coverage and trajectory length for all evaluated training configurations.** Each row contains the average of 176 reports scored by the same jury.

Effort	Policy	Coverage (%)	Mean trajectory tokens
Low	Muse-Glimmer	17.17	15,782
Low	Outcome-only GRPO	17.41	28,246
Low	Magnitude-matched GRPO	21.69	22,038
Low	Turn-level GRPO (Muse-Glimmer-Hedge)	**33.36**	28,662
Medium	Muse-Glimmer	28.64	35,902
Medium	Outcome-only GRPO	24.33	87,310
Medium	Magnitude-matched GRPO	31.20	58,632
Medium	Turn-level GRPO (Muse-Glimmer-Hedge)	**46.02**	67,402
High	Muse-Glimmer	29.51	49,387
High	Magnitude-matched GRPO	33.02	77,134
High	Turn-level GRPO (Muse-Glimmer-Hedge)	**46.82**	88,683

Figure 10 compares all 21 training updates. Magnitude matching reduces repetition relative to unscaled outcome-only training, but its entropy still falls and its trajectories lengthen late in training. Over updates 16–20, TurnCredit reaches 45.43% average training coverage, compared with 33.64% for magnitude matching and 25.09% for unscaled outcome-only training. TurnCredit also averages 11.4 tool commands after the first answer write; the corresponding means are 20.2 for magnitude matching and 32.6 for unscaled outcome-only training.

We evaluate the models using the settings and recovery procedure in Appendix H.1. Table 23 reports eight trials on each of the 22 held-out environments. We use the same within-environment permutation test, with Holm correction across the three effort-matched comparisons.

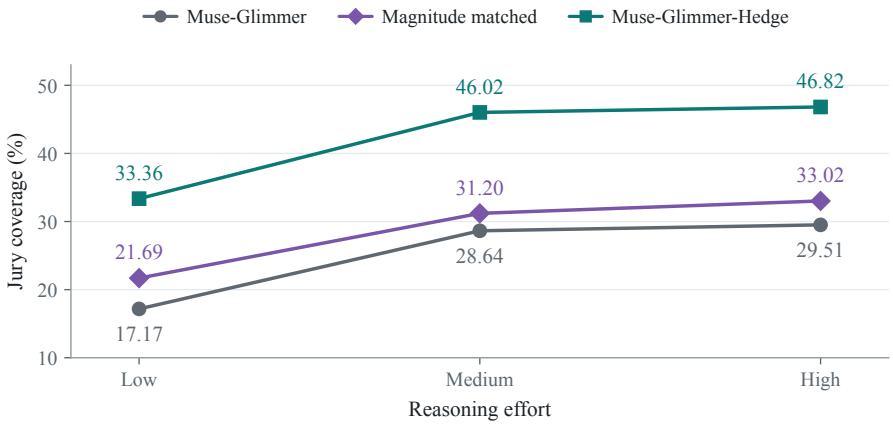

Figure 9: **The magnitude-matched policy has lower coverage than turn-level training at each evaluated effort.**

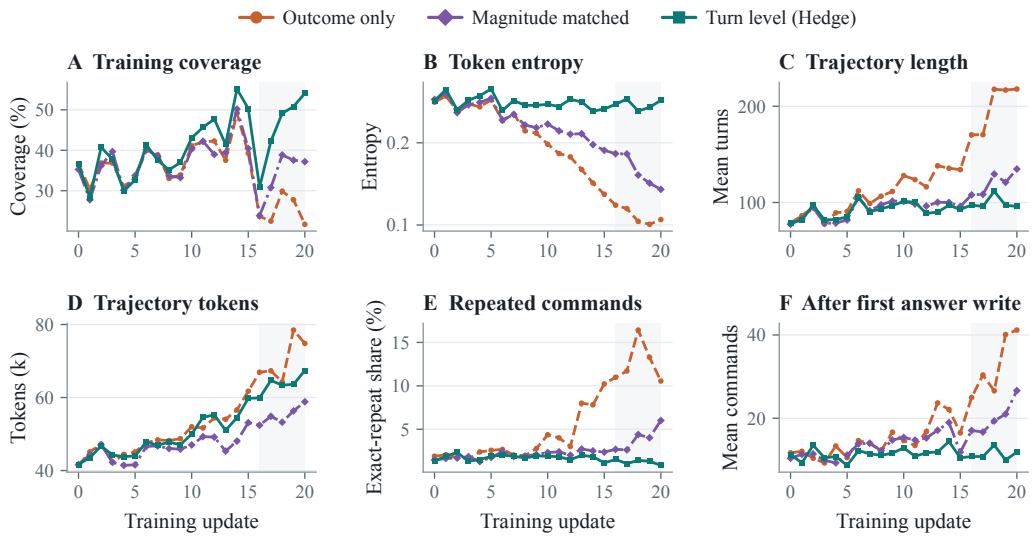

Figure 10: **Turn-level training maintains coverage and entropy while limiting repeated and post-write actions.** Hedge denotes Muse-Glimmer-Hedge, the turn-level policy. The six panels show all 21 updates without smoothing, with 64 trajectories per update. Shading marks updates 16–20. Training coverage is the training judge's diagnostic. Post-write commands are tool commands after the first answer write; trajectory tokens include accepted assistant output and tool observations.

H Post-Training Evaluation Details

This appendix gives the evaluation protocols and full post-training results for Hedge-Bench (Tables 24 and 26) and external benchmarks (Tables 32 and 33). Hedge-Bench-Train contains 165 environments: 70 from Hedge-Bench-Public and 95 private environments. The remaining 22 public environments form Hedge-Bench-Heldout and do not overlap with training. Hedge-Bench-Private contains 53 environments from 11 target companies absent from training. Hedge-Bench-Heldout tests new environments; Hedge-Bench-Private tests transfer to new target companies.

H.1 HEDGE-BENCH-HELDOUT

We evaluate Muse-Glimmer and Muse-Glimmer-Hedge on the same 22 Hedge-Bench-Heldout environments, with eight trials per environment at low, medium, high, and extra-high reasoning effort. This gives 1,408 answers: $22 \times 8 \times 2$ policies $\times 4$ efforts. Each condition uses the official Muse chat template and parser with a Bash tool. Sampling uses temperature 1, top-p 1, and no top-k filtering. The limits are 16,384 generated tokens per turn, a 128,000-token model context, and 240 turns. Throughout this appendix, accepted trajectory tokens count agent output and tool observations. They exclude the initial prompt.

Claude Opus 5, GPT-5.6 Sol, and Gemini 3.8 Flash independently verify cited evidence and assess research coverage. Every answer has three valid assessments, combined using the majority rule in Equation 1. We average coverage over trials within each environment, then give every environment equal weight. Forecast correctness is scored separately.

All trained models use the same initial settings and scoring procedure. Trajectories that reach a turn or context limit continue from the existing conversation with higher limits: up to 480 turns for magnitude-matched GRPO, and up to 960 turns with a 512,000-token context and 65,536 generated tokens per turn for unscaled outcome-only GRPO. Malformed generations are retried from the same conversation state.

Table 24 compares coverage and trajectory length across 8 settings, each with 176 scored answers. Training increases coverage by 16.19 percentage points at low effort, 17.38 at medium, 17.31 at high, and 16.83 at extra-high. Low-effort Muse-Glimmer-Hedge exceeds extra-high-effort Muse-Glimmer by 3.17 points while using 47.9% fewer accepted trajectory tokens.

Table 24: Hedge-Bench-Heldout coverage increases at every matched effort.

Effort	Muse-Glimmer coverage (%)	Muse-Glimmer-Hedge coverage (%)	Muse-Glimmer tokens	Muse-Glimmer-Hedge tokens
Low	17.17	33.36	15,782	28,662
Medium	28.64	46.02	35,902	67,402
High	29.51	46.82	49,387	88,683
Extra-high	30.19	47.02	54,974	99,037

The significance tests compare low-effort Muse-Glimmer-Hedge with each Muse-Glimmer effort on the same 22 environments. Let C^V_{er} and C^B_{er} be their coverage scores for environment e and trial r. We average the eight trials within each environment, then give every environment equal weight. The test statistic is the difference between the policy means:

$$\widehat{\Delta} = \frac{1}{22}\sum_{e=1}^{22}\left(\frac{1}{8}\sum_{r=1}^{8} C^V_{er} - \frac{1}{8}\sum_{r=1}^{8} C^B_{er}\right).$$

Within each environment, we randomly divide the two policies' 16 scores into groups of eight. Each of 200,000 permutations preserves environment identity and group size. The two-sided Monte Carlo p-value is $(b+1)/(200{,}000+1)$, where b counts permuted absolute differences at least as large as $|\widehat{\Delta}|$. We apply Holm correction across the four Muse-Glimmer-Hedge-low versus Muse-Glimmer comparisons (Table 25).

Table 25: Low-effort Muse-Glimmer-Hedge exceeds Muse-Glimmer at all four reasoning efforts on Hedge-Bench-Heldout. Base denotes Muse-Glimmer and Hedge denotes Muse-Glimmer-Hedge. Each condition has eight trials per environment (176 answers). Two-sided within-environment permutation tests use 200,000 draws; p-values are Holm-adjusted over these four comparisons.

Base effort	Hedge-low coverage gain	Adjusted p
Low	+16.19	2.0×10^{-5}
Medium	+4.73	0.00178
High	+3.85	0.00920
Extra-high	+3.17	0.02496

H.2 HEDGE-BENCH-PRIVATE

H.2.1 EVALUATION DESIGN AND AGGREGATE RESULTS

We evaluate Muse-Glimmer and Muse-Glimmer-Hedge at low and high effort on 53 environments from 11 companies absent from RL training, with three trials per environment: 159 reports per condition and 636 in total. The same three-model jury assesses every report using Equation 1, giving 1,908 assessments. We first average each environment's three coverage scores, then average across the 53 environments. Forecast correctness is scored separately.

Coverage improves by 13.26 points at low effort and 16.44 points at high effort. High-effort prediction accuracy increases by 10.06 points (Table 26). Coverage improves on 46 environments at low effort, with four regressions and three ties. At high effort, 45 improve, three regress, and five tie. Company-average coverage improves for all 11 companies at both efforts. Low-effort Muse-Glimmer-Hedge also exceeds high-effort Muse-Glimmer in coverage and prediction accuracy while using 40.2% fewer accepted trajectory tokens (Table 27).

With equal weight on each effort, prediction accuracy rises from 125/318 (39.31%) to 147/318 (46.23%). Averaging the six attempts per environment gives 18 prediction improvements, four regressions, and 31 ties; pooled balanced accuracy rises from 41.31% to 47.96%. We extract explicit Yes/No verdicts. Ambiguous prose is reviewed against the exact forecast question, and missing answers count as incorrect.

Hedge-Bench-Private contains 28 Yes and 25 No outcomes, so a constant Yes forecast scores 52.83%. The class balance is similar to Hedge-Bench-Heldout, which has 10 Yes and 12 No outcomes and a 54.55% majority baseline. Training improves accuracy over Muse-Glimmer on Hedge-Bench-Private, but pooled accuracy remains below the constant-label baseline. Hedge-Bench-Private is genuinely more difficult for models compared to Hedge-Bench-Heldout. Sol also performs worse on Hedge-Bench-Private than on Hedge-Bench-Public, with prediction accuracy falling from 69.57% to 35.85% (Table 28). The next subsection examines this comparison.

Table 26: **Company-held-out research and forecasting.** Each policy and effort has 159 reports: three trials on each of 53 environments from 11 companies. Coverage uses the three-model jury. Two-sided within-environment permutation tests compare repeated generations on the fixed environments.

Metric	Effort	Muse-Glimmer (%)	Muse-Glimmer-Hedge (%)	Gain (points)	Holm p
Coverage	Low	15.36	28.62	+13.26	0.00002
Coverage	High	22.23	38.67	+16.44	0.00002
Prediction	Low	40.25	44.03	+3.77	0.25165
Prediction	High	38.36	48.43	+10.06	0.00351

Table 27: **Cross-effort comparison on Hedge-Bench-Private.** Means over three trials per environment, including all 159 reports per condition.

Configuration	Mean trajectory tokens	Coverage (%)	Accuracy (%)
Muse-Glimmer high	45,797	22.23	38.36
Muse-Glimmer-Hedge low	27,393	28.62	44.03

H.2.2 FRONTIER REFERENCE AND OUTCOME TIMING

GPT-5.6 Sol Max provides a frontier reference on the same 53 tasks (Table 28). Sol contributes one report per task and 159 valid assessments from the same three-model jury. All models use the same forecast questions and outcome definitions.

Sol reaches 45.33% coverage and 35.85% prediction accuracy on Hedge-Bench-Private, compared with 59.17% and 69.57% on Hedge-Bench-Public: decreases of 13.84 coverage points and 33.72 prediction points. Sol predicts Yes 16 times and No 37 times, correctly forecasting 5 of 28 Yes

Table 28: **Public and private research difficulty.** All rows use full-environment reports and the three-model jury.

Model and effort	Set	Reports	Coverage (%)	Accuracy (%)
GPT-5.6 Sol Max	Hedge-Bench-Public	276	59.17	69.57
GPT-5.6 Sol Max	Hedge-Bench-Private	53	45.33	35.85
Muse-Glimmer Low	Hedge-Bench-Private	159	15.36	40.25
Muse-Glimmer-Hedge Low	Hedge-Bench-Private	159	28.62	44.03
Muse-Glimmer High	Hedge-Bench-Private	159	22.23	38.36
Muse-Glimmer-Hedge High	Hedge-Bench-Private	159	38.67	48.43

outcomes and 14 of 25 No outcomes. Its lower performance concerns both research completion and forecast choice. On this set, Muse-Glimmer-Hedge improves over Muse-Glimmer on both endpoints.

The public and private cohorts overlap in outcome timing; Tables 34 and 35 list all 53 private and 92 public dates. Of the public tasks, 69 resolve after March 31, 2026. Sol scores 73.43% on these later tasks and 57.97% on the 23 earlier tasks (Table 17). The performance difference therefore does not simply contrast an entirely old public set with a recent private one.

H.2.3 Prediction Breakdown and Statistical Tests

To understand why both models score below the majority-class baseline, we examine their Yes/No predictions. Both favor No, although 28 of the 53 environments resolve to Yes. Table 29 reports the proportion of Yes predictions alongside accuracy. This shows how training changes the tendency to predict No; the breakdown by outcome below shows whether the changed predictions are correct.

At high effort, both policies give usable verdicts on all 159 attempts. With three trials per environment, each policy makes 84 forecasts on the 28 environments with Yes outcomes and 75 forecasts on the 25 environments with No outcomes. On Yes outcomes, Muse-Glimmer-Hedge makes 17 correct forecasts compared with 7 for Muse-Glimmer, an improvement of ten. On No outcomes, Muse-Glimmer-Hedge makes 60 correct forecasts compared with 54 for Muse-Glimmer. At high effort, training improves accuracy on both Yes and No outcomes. The gain is not just a simple shift away from No predictions. At low effort, correct Yes outcomes rise from 3 to 12, while correct No outcomes fall from 61 to 58.

Table 29: **Hedge-Bench-Private answer rates and forecast composition.** Each row contains 159 attempts (53 environments, three trials).

Policy	Effort	Usable predictions	Predicted Yes (%)	Accuracy (%)
Muse-Glimmer	Low	159/159	10.69	40.25
Muse-Glimmer-Hedge	Low	158/159	17.61	**44.03**
Muse-Glimmer	High	159/159	17.61	38.36
Muse-Glimmer-Hedge	High	159/159	20.13	**48.43**

To test these gains, we use the within-environment permutation test described in Section H.1, with three trials per policy. Each environment has six scores and $\binom{6}{3} = 20$ possible divisions into groups of three. Holm correction covers coverage and prediction at both efforts. Adjusted values are $p = 0.00002$ for both coverage comparisons, $p = 0.00351$ for high-effort prediction, and $p = 0.25165$ for low-effort prediction. The test measures variation across independent generations on these 53 environments, holding the checkpoints and grading results fixed.

H.3 Joint Success

Table 30 reports joint success: coverage strictly above 60% and a correct forecast in the same report. Each success rate uses all reports at that effort. We apply the within-environment permutation test from Section H.1 to binary success outcomes, with eight trials per policy on Hedge-Bench-Heldout and three on Hedge-Bench-Private. Holm correction covers these six comparisons.

Joint success increases significantly at every effort on Hedge-Bench-Heldout. On the harder Hedge-Bench-Private (Section H.2), low-effort training improves research coverage significantly, while forecasting accuracy improves without statistical significance. Only one of the 159 low-effort Muse-Glimmer-Hedge reports combines a correct forecast with coverage above 60%, compared with none for Muse-Glimmer, so the joint gain is not significant. At high reasoning effort, the Hedge-Bench-Private joint gain becomes significant. Thus five settings show significant joint gains.

Table 30: **Post-training joint research and prediction success.** Success requires $C > 0.60$ and a correct forecast for the same answer. Holm correction covers all six fixed-task comparisons.

Set	Effort	Muse-Glimmer J (%)	Muse-Glimmer-Hedge J (%)	Gain (pp)	Holm p
Hedge-Bench-Heldout	Low	1/176 (0.57)	14/176 (7.95)	+7.39	0.0009
Hedge-Bench-Heldout	Medium	2/176 (1.14)	34/176 (19.32)	+18.18	< 0.0001
Hedge-Bench-Heldout	High	2/176 (1.14)	32/176 (18.18)	+17.05	< 0.0001
Hedge-Bench-Heldout	Xhigh	4/176 (2.27)	31/176 (17.61)	+15.34	< 0.0001
Hedge-Bench-Private	Low	0/159 (0.00)	1/159 (0.63)	+0.63	1.0000
Hedge-Bench-Private	High	1/159 (0.63)	9/159 (5.66)	+5.03	0.0228

H.4 Professional-Task Protocols

We test whether training on Hedge-Bench improves performance on other professional tasks. This section describes the evaluation setup and scoring for GDPval, Harvey LAB, GDP.pdf, and Frontier Finance, comparing Muse-Glimmer with Muse-Glimmer-Hedge on each benchmark.

H.4.1 GDPval

We evaluate 220 public GDPval tasks with a Bash-based agent under an adapted protocol, not an official leaderboard submission. We report two scores. For weighted rubric completion, Gemini 3.7 Flash scores each deliverable independently against the task's rubric, using provider-default reasoning. For blind pairwise preference, a judge compares two deliverables for the same task without knowing the models' identities. The preference pool includes 28 configuration pairs on 220 tasks and eight configurations compared with 185 available human references, giving 7,640 comparisons. Each comparison samples one judge from GPT-5.5, Gemini 3.1 Pro, and Claude Opus 4.8. We fit a Bradley-Terry model (Bradley & Terry, 1952), counting ties as half-wins and fixing the human reference rating at 1000. Confidence intervals use a task-clustered sandwich estimator. These ratings depend on the comparison pool and differ from the independently maintained GDPval-AA rating (Artificial Analysis, n.d.). Both models are evaluated at low, medium, high, and extra-high effort. Rubric completion includes all 220 tasks at every setting.

Table 31: GDPval operating points shown in Figure 5. Rubric completion uses all 220 tasks. Blind ratings belong to the eight-configuration comparison pool described above, with the human reference fixed at 1000. Tokens refer to rubric trajectories.

Model	Effort	Generated tokens (rubric answers)	Rubric (%)	Blind rating
Muse-Glimmer	Low	4,103	74.19	913.3
Muse-Glimmer	Medium	9,441	79.82	1,065.3
Muse-Glimmer	High	12,143	78.91	1,085.3
Muse-Glimmer	Extra-high	12,100	78.92	1,128.2
Muse-Glimmer-Hedge	Low	6,400	77.93	1,022.0
Muse-Glimmer-Hedge	Medium	13,915	81.71	1,232.4
Muse-Glimmer-Hedge	High	17,424	81.52	1,244.0
Muse-Glimmer-Hedge	Extra-high	18,364	82.86	1,241.3

H.4.2 Harvey LAB

We evaluate both models on 100 legal tasks from Harvey LAB, with four tasks from each of 25 areas. From 2,010 task configurations, we exclude diligence and firm-knowledge tasks. We choose tasks with different types of work and output file formats. We also vary the number of source documents

and rubric criteria. Selection is deterministic: within each area, we add tasks one at a time, choosing the task that adds the most variety in these properties. A fixed seed breaks ties. The sample contains 763 source documents and 5,684 criteria. It is balanced by area rather than randomly sampled from the repository's task distribution.

Generation uses high reasoning with the official six tools and provided document skills. We use a 200-turn limit and Modal filesystem isolation in place of Podman. Judges assess each criterion from the requested deliverable without model identity, giving 400 report assessments and 22,736 criterion decisions. The pooled score counts every criterion equally, so tasks with more criteria have more weight. We also report each judge's results separately. Under strict all-criteria success, both judges record zero successful Muse-Glimmer tasks and one successful Muse-Glimmer-Hedge task.

H.4.3 GDP.PDF

We evaluate all 100 GDP.pdf tasks (Surge AI, 2026) with five attempts per task and model. The official Muse renderer places the prompt and PDF page images in context. Page resolution is adjusted to fit the available context. Both policies use the same representation for each task. Gemini 3.5 Flash grades the answer against the supplied rubric. Criterion-weighted coverage includes all 500 attempts. It rises from 4,453/6,375 (69.85%) to 4,599/6,375 (72.14%). Strict success, requiring every criterion, rises from 58/500 (11.6%) to 71/500 (14.2%). Across matched task-attempt pairs, the trained model has higher coverage in 197 cases, Muse-Glimmer in 121, and 182 tie.

H.4.4 FRONTIER FINANCE

We use the released questions and rubric pipeline (Samaya AI, 2026) with Finance Agent v2 tools for web search, SEC filings, and market data. Both policies have the same tools and scoring procedure, without an elapsed-time cutoff. Gemini 3.8 Flash through OpenRouter grades each answer using the external task rubric.

For each query, we calculate the fraction of rubric criteria satisfied by the answer, then average across queries with equal weight. This is the benchmark's macro-qualification score. The paired comparison uses 189 queries, excluding one query for which Muse-Glimmer-Hedge reached the turn limit without producing an answer. On these queries, the score rises from 39.62% for Muse-Glimmer to 44.58% for Muse-Glimmer-Hedge, a 4.96-point gain. On paired comparison, there are 94 trained-model wins, 48 Muse-Glimmer wins, and 47 ties.

GDPval, GDP.pdf, Harvey LAB, and Frontier Finance all compare Muse-Glimmer with Muse-Glimmer-Hedge. Each benchmark retains its own score definition.

H.5 PAIRED EXTERNAL COMPARISONS

Table 32 reports gains at all four GDPval efforts and on GDP.pdf, Harvey, and Frontier Finance. Differences are computed before rounding. Each comparison uses the same task IDs for Muse-Glimmer and the trained model. GDPval scores include all 220 tasks, including unsuccessful submissions. Frontier Finance uses 189 shared queries.

We estimate uncertainty by resampling tasks, keeping both models' scores and all attempts for a task together with their judge decisions. We draw 200,000 paired bootstrap samples at the task level. Sampling preserves the observed counts within GDPval's 44 occupations, GDP.pdf's ten domains, and Harvey's 25 practice areas. Frontier Finance resamples the shared queries without stratification. GDPval and Frontier Finance average task-level fractions. GDP.pdf and Harvey recompute the total passed criteria divided by the total criteria in each sample. GDP.pdf therefore has 100 task clusters, all five attempts per task remain in the denominator. The intervals hold the answers and grading results fixed.

For rubric gains, two-sided permutation tests swap the complete Muse-Glimmer and trained score vectors within tasks, keeping all attempts and judge decisions together. We use 200,000 permutations and the correction $(b + 1)/(200{,}000 + 1)$, where b counts absolute permuted differences at least as large as observed. For Elo differences, we account for covariance between ratings in the Bradley-Terry fit, using a task-clustered sandwich estimator and two-sided normal Wald tests. Holm correction

covers the seven rubric and four Elo comparisons. All eleven remain significant; the intervals in Tables 32 and 33 are marginal 95% intervals.

Table 32: **Rubric scores improve in all external comparisons.** Base denotes Muse-Glimmer and Hedge denotes Muse-Glimmer-Hedge. Scores are percentages; gains and intervals are percentage points under each benchmark's metric. Holm p Holm correction covers these four Elo comparisons (Table 33) together with the seven rubric-score comparisons.

Evaluation	Tasks	Base	Hedge	Gain	95% interval	Holm p
GDPval low	220	74.19	77.93	+3.74	[2.17, 5.40]	0.00021
GDPval medium	220	79.82	81.71	+1.90	[0.65, 3.13]	0.01274
GDPval high	220	78.91	81.52	+2.61	[1.27, 3.96]	0.00190
GDPval extra-high	220	78.92	82.86	+3.94	[2.55, 5.43]	0.000035
GDP.pdf high	100	69.85	72.14	+2.29	[0.76, 3.84]	0.01274
Harvey LAB high	100	64.19	71.26	+7.07	[2.78, 11.33]	0.00922
Frontier Finance high	189	39.62	44.58	+4.96	[2.18, 7.66]	0.00162

Table 33: **Blind preference favors Muse-Glimmer-Hedge at every GDPval effort.** Base and Hedge denote Muse-Glimmer and Muse-Glimmer-Hedge. Uncertainty in the Elo difference accounts for the covariance between the fitted ratings and groups the 7,640 comparisons by their 220 tasks.. The human reference is fixed at 1000. Holm p uses the joint eleven-comparison family.

Effort	Base	Hedge	Gain	Paired SE	95% interval	Holm p
Low	913.3	1022.0	+108.7	17.4	[74.6, 142.8]	3.22×10^{-9}
Medium	1065.3	1232.4	+167.1	18.7	[130.4, 203.7]	4.03×10^{-18}
High	1085.3	1244.0	+158.7	17.5	[124.4, 192.9]	1.18×10^{-18}
Extra-high	1128.2	1241.3	+113.1	17.8	[78.3, 148.0]	1.69×10^{-9}

The rubric gains also have positive lower bounds under unstratified paired resampling. For Harvey, the separate judge gains are +7.13 points for Claude Sonnet 4.6 (95% interval: +2.75 to +11.43) and +7.02 for GPT-5.5 (+2.77 to +11.27). Direct same-effort blind comparisons favor Muse-Glimmer-Hedge on 149, 154, 172, and 156 of 220 GDPval tasks at low through extra-high effort, versus 70, 65, 46, and 62 Muse-Glimmer wins; the remaining comparisons are ties. These direct outcomes agree with the fitted Elo contrasts.

H.6 Hedge-Bench-Private Outcome Chronology

Table 34 reports the outcome-evidence dates for all 53 private prediction questions. D marks a public disclosure date. W marks the end of the period over which a market-price condition was evaluated. Dates run from November 6, 2025 through September 3, 2026 and overlap the public benchmark's resolution period. Tasks receive anonymous identifiers ordered by date; issuer names, task descriptions, forecast wording, and answer labels are withheld to preserve the private evaluation set. The same outcome definitions and labels are used for every evaluated model.

Table 34: **Outcome chronology for all 53 Hedge-Bench-Private tasks.** Anonymous task IDs retain timing information without exposing the task-to-answer mapping. D: disclosure; W: market-window endpoint.

Task ID	Date	Type	Task ID	Date	Type
HB-P-001	2025-11-06	D	HB-P-028	2026-07-16	D
HB-P-002	2025-11-12	D	HB-P-029	2026-07-16	D
HB-P-003	2026-01-22	D	HB-P-030	2026-07-21	D
HB-P-004	2026-01-27	D	HB-P-031	2026-07-21	D
HB-P-005	2026-02-03	D	HB-P-032	2026-07-21	D
HB-P-006	2026-02-05	D	HB-P-033	2026-08-04	D
HB-P-007	2026-02-24	D	HB-P-034	2026-08-04	D
HB-P-008	2026-03-05	D	HB-P-035	2026-08-05	D

Table 34 continued

Task ID	Date	Type	Task ID	Date	Type
HB-P-009	2026-03-09	D	HB-P-036	2026-08-06	D
HB-P-010	2026-03-18	D	HB-P-037	2026-08-06	D
HB-P-011	2026-03-19	D	HB-P-038	2026-08-06	D
HB-P-012	2026-04-21	D	HB-P-039	2026-08-06	D
HB-P-013	2026-04-21	D	HB-P-040	2026-08-06	D
HB-P-014	2026-04-21	D	HB-P-041	2026-08-06	D
HB-P-015	2026-05-07	D	HB-P-042	2026-08-06	D
HB-P-016	2026-05-07	D	HB-P-043	2026-08-06	D
HB-P-017	2026-05-07	D	HB-P-044	2026-08-06	D
HB-P-018	2026-05-07	D	HB-P-045	2026-08-06	D
HB-P-019	2026-05-14	D	HB-P-046	2026-08-06	D
HB-P-020	2026-05-14	D	HB-P-047	2026-08-06	D
HB-P-021	2026-05-14	D	HB-P-048	2026-08-06	D
HB-P-022	2026-05-27	D	HB-P-049	2026-08-10	D
HB-P-023	2026-05-27	D	HB-P-050	2026-08-11	D
HB-P-024	2026-05-27	D	HB-P-051	2026-08-13	D
HB-P-025	2026-05-27	D	HB-P-052	2026-08-17	W
HB-P-026	2026-07-14	D	HB-P-053	2026-09-03	W
HB-P-027	2026-07-16	D			

H.7 HEDGE-BENCH-PUBLIC OUTCOME CHRONOLOGY

Table 35 lists all 92 public tasks and their recorded outcome dates. These are the dates used for the common calendar split in Table 17: 23 tasks fall through March 31, 2026 and 69 afterward. The date-basis column distinguishes source-corroborated disclosures (D), disclosure intervals (I), question-window endpoints (W), an event date (E), and a candidate disclosure date (C). The analysis based on each model's knowledge cutoff uses the separately verified outcome dates described in Appendix E.2.

Table 35: **Recorded outcome chronology for all 92 public tasks.** Task themes identify the public environments. D: source-corroborated disclosure; I: disclosure interval; W: window endpoint; E: event date; C: candidate disclosure date.

No.	Issuer	Task theme	Date	Basis
1	APO	Capital allocation buybacks vs dividends	2026-05-06	D
2	APO	Private credit risk and underwriting quality	2026-05-06 to 2026-05-07	I
3	APO	SRE margin compression and bottoming	2026-05-06	D
4	APO	Sum-of-the-parts valuation methodology	2026-08-04	D
5	CARR	Capital allocation and share buybacks	2026-07-28	D
6	CARR	Carrier's valuation relative to peers	2026-02-05	D
7	CARR	Competitive moats and distribution channels	2026-07-29	D
8	CARR	Data center growth vs core cyclicality	2026-02-05	D
9	CARR	Earnings accretion vs multiple compression	2026-07-28	D
10	CARR	Residential market recovery and catalysts	2026-07-28	D
11	CNC	Capital adequacy and dilution risk	2026-02-17	D
12	CNC	Feasibility of Medicaid book repricing	2026-07-28	D
13	CNC	Impact of expiring EPTC subsidies	2026-07-28	D
14	CNC	Investment income and macro tailwinds	2026-02-06	D
15	CNC	Relative risk-reward of managed care names	2026-07-28	D
16	CNSWF	AI disruption of vertical software	2026-05-12	D
17	CNSWF	Customer psychology and switching costs	2026-05-12	D
18	CNSWF	M&A runway and capital deployment	2026-05-12	D
19	FI	Argentina's impact on growth modeling	2026-05-05	D
20	FI	Competitive dynamics of the core banking duopoly	2026-05-05	D
21	FI	Fiserv's management strategy and restructuring	2026-08-06	D
22	FI	Valuation multiples and long-term earnings power	2026-05-05	D

Table 35 continued

No.	Issuer	Task theme	Date	Basis
23	FLYW	Healthcare segment viability and performance	2026-07-31	W
24	FLYW	Immigration headwinds and student demand	2026-07-31	W
25	FLYW	Strategic rationale for the travel segment	2026-05-05 to 2026-05-06	I
26	FLYW	Valuation and sum-of-the-parts analysis	2026-05-05 to 2026-05-06	I
27	FND	Competitive moat vs diversified retailers	2026-04-30	D
28	FND	Flooring vs non-discretionary home repair	2026-07-30	D
29	FND	Housing mobility as a demand catalyst	2026-04-30	D
30	FND	Macro signals and false bottoms	2026-04-30	D
31	FND	Unit growth algorithm and store targets	2026-04-30	D
32	FND	Valuation multiples and historical mean reversion	2026-04-30	D
33	GDOT	Customer concentration and obsolescence risk	2026-06-23	D
34	GDOT	Execution risk vs management track record	2026-06-23	D
35	GDOT	Small-cap liquidity and float dynamics	2026-07-31	W
36	GETY	AI's impact on creative stock vs editorial	2026-06-21	D
37	GETY	Multiple compression in less-bad value trades	2026-06-30	W
38	GETY	Strategic rationale of the Getty-Shutterstock merger	2026-07-07	D
39	GOOGL	Google Network and Performance Max impact	2026-02-04	D
40	GOOGL	Regulatory breakup risks and structural remedies	2026-06-30	W
41	GOOGL	Search market share and competitive threats	2025-07-23	D
42	GRAL	Downside protection and data value	2025-10-21	E
43	GRAL	Medicare reimbursement and political catalysts	2026-02-03	D
44	IOT	Competitive landscape and platform advantage	2026-03-05	D
45	IOT	Growth durability vs valuation ceiling	2025-12-04	D
46	IOT	Long-term disintermediation by OEMs and autonomous tech	2026-03-31	D
47	IOT	Margin inflection and free cash flow potential	2026-03-05	D
48	IOT	Sales efficiency and macro-induced deceleration	2026-03-05	D
49	IRDM	Capital allocation and the capex holiday	2026-06-30	W
50	IRDM	Future constellation relaunch vs spectrum sale	2026-02-12	D
51	IRDM	IoT growth potential and technical limitations	2026-04-23	D
52	IRDM	L-band utility in defense and government	2026-02-12	D
53	IREN	Colocation vs GPU ownership models	2026-05-07	D
54	IREN	IREN-Microsoft contract economic viability	2026-06-01	D
55	IREN	Power availability vs development execution	2026-05-07	D
56	IREN	Software moats in AI infrastructure	2026-05-05	D
57	IREN	Valuation methodologies for AI infrastructure	2026-06-01	D
58	LKNCY	Competitive impact of Starbucks pricing	2026-04-29	D
59	LKNCY	Delivery margin recovery and subsidies	2026-04-29	D
60	LKNCY	US expansion and unit economics	2026-04-29	D
61	LULU	Competitive threat of Vuori and Alo	2025-09-04	D
62	LULU	Expansion into niche sports categories	2025-10-27	D
63	LULU	North American growth vs international potential	2026-06-04	D
64	LULU	Supply chain and margin headwinds	2026-06-04	D
65	LULU	Valuation rerating and market sentiment	2025-09-04	D
66	PLBY	Adult asset licensing and Byborg deal	2026-05-11	D
67	PLBY	Brand equity and growth levers	2026-05-11	D
68	PLBY	China JV and asset monetization	2026-05-11	D
69	PLBY	Deleveraging path and Honey Birdette sale	2026-05-11	D
70	PLBY	Operational leverage and cash conversion	2026-05-11	D
71	PLBY	Strategic pivot and capital allocation history	2026-05-11	C
72	PLBY	Valuation and forward EBITDA projections	2026-05-11	D
73	RBLX	AI as a competitive threat vs enabler	2026-04-30	D
74	RBLX	Infrastructure strategy and capex efficiency	2026-02-05	D
75	RBLX	User base maturation and aging up	2026-07-30	D

Table 35 continued

No.	Issuer	Task theme	Date	Basis
76	SNAP	Efficacy of activist influence via Irenic	2026-08-03	D
77	SNAP	Potential for acquisition or terminal value	2026-08-31	W
78	SNAP	Viability of Spectacles and hardware strategy	2026-06-16	D
79	SOC	County permit delays and takings claims	2026-03-15	W
80	SOC	Federal supremacy vs state regulation	2026-03-30	D
81	SOC	Impact of new California legislation	2025-12-31	W
82	TKO	Impact of streaming on rights monetization	2026-05-06	D
83	TKO	Revenue synergies and the flywheel model	2026-02-25	D
84	TKO	Strategic M&A and capital allocation	2026-08-02	W
85	TKO	Valuation and forward return profile	2026-08-03	D
86	WEN	Competitive positioning and fresh marketing	2026-05-08	D
87	WEN	Franchisee financial health and systemic risk	2026-05-08	D
88	WEN	Investment asymmetry and the Peltz factor	2026-05-20	D
89	WU	Capital allocation and shareholder returns	2026-07-30	D
90	WU	Digital disintermediation and network moat	2026-07-30	D
91	WU	Resiliency of cash payout networks	2026-04-24	D
92	WU	Valuation multiples and earnings trajectory	2026-04-24	D

I Research Improvements in Held-Out and External Tasks

What changes in the research when coverage improves? We compare reports from Muse-Glimmer and Muse-Glimmer-Hedge on the same assignments, first in Hedge-Bench-Heldout and then in GDPval. The excerpts below retain the models' wording; code excerpts come from their tool calls. Hedge-Bench scores credited analytical moves, whereas the GDPval examples report the fraction of rubric items passed. These selected pairs illustrate the differences behind those scores.

I.1 Hedge-Bench-Heldout: More Research Behind the Same Forecast

The April 13, 2026 Flywire assignment asks whether the company will announce a divestiture, spin-off, or formal strategic review of its healthcare business by July 31, 2026. We compare the third trial from each model at extra-high reasoning effort. Both correctly predict No, but research coverage rises from 3 of 12 moves (25.00%) for Muse-Glimmer to 10 of 12 (83.33%) for Muse-Glimmer-Hedge.

The base report mainly summarizes recent management commentary about healthcare growth and the Cleveland Clinic rollout. The trained report also looks back at earlier underperformance, then asks whether the rollout can address the business's weaknesses.

Verbatim report excerpts: Flywire

Muse-Glimmer: recent growth

> The health care payment processing ramp is a material contributor to near-term top-line outperformance, while the lower-margin profile is managed through scale and cross-sell into higher-margin software and patient affordability solutions.

Muse-Glimmer-Hedge: earlier underperformance

> In Q4 2023 the analyst asked directly: "maybe just on the healthcare, Mike, healthcare down 1%. How do we think about the reacceleration of that business?" (`earnings_call/flyw_2023_q4_earnings_call.txt`; speaker=John Davis; lines=567-569). Management responded with conviction but acknowledged disappointment: "we're not satisfied with the results that we saw in 2023" (`earnings_call/flyw_2023_q4_earnings_call.txt`; speaker=Rob Orgel; lines=571-573).

Muse-Glimmer-Hedge: rollout risk

> If the patient financial experience phase slips beyond H1 2026, margin headwinds persist and the reference account narrative weakens.

The trained report follows healthcare's weak 2023 performance through its modest return to growth in 2024. It compares that record with the stronger growth in travel and education, rather than treating recent healthcare wins as sufficient evidence of a recovery. The jury credits all three moves under segment growth and contribution analysis, compared with none for the base report.

Both reports describe the phased Cleveland Clinic implementation. The trained report goes further by examining the slow rollout and the implementation work that could delay its contribution. The jury credits all three partnership-execution moves, compared with one for the base report. Overall, the trained report gains seven credited moves and exceeds the 60% coverage threshold. The unchanged forecast therefore hides a substantial difference in the research supporting it.

I.2 GDPVAL: USING THE RIGHT DENOMINATOR

The inventory assignment asks the agent to calculate the percentage of stores that are out of stock for each product. Its instruction defines an active store as follows: "A store should be considered active if it appears in the dataset for that item (if a store number is returned, it is considered active) and also if it has an out-of-stock percentage." Both models run at extra-high effort and take 35 turns, but they translate this definition into different calculations.

Verbatim tool-code excerpts: counting active stores

Muse-Glimmer: excludes rows without days-on-hand data

```
mask_active = sub['Avg Days On Hand']!='-'
active = sub[mask_active]
num_stores = active['Store Number'].nunique()
```

Muse-Glimmer-Hedge: counts the store numbers returned for the product

```
num_stores = sub['Store Number'].nunique()
```

The base model adds a filter that the task did not request. It counts only 63–302 active stores per product, compared with 728-1,232 for the trained model. This changes the denominator of the out-of-stock percentage and therefore the inventory assessment built on it. Rubric items passed increase from 37.9% to 63.8%. The two trajectories have the same number of turns; the improvement is in what is calculated. Generated tokens increase from 12,719 to 18,483.

I.3 GDPVAL: RECOVERING THE SOURCE BEFORE WRITING

The compliance assignment requests a PDF with test questions and exception statements for two specified provisions of the VA Servicer Handbook. Both models initially have difficulty retrieving the handbook. The base model proceeds from assumed requirements. The trained model retrieves an archived copy and reconstructs the paragraph numbering before drafting its tests.

Verbatim trajectory excerpts: handling an unavailable source

Muse-Glimmer: after unsuccessful retrieval

> Given time constraints, perhaps we can assume the content based on typical requirements.

Muse-Glimmer-Hedge: documenting the recovered source

> Assumptions recorded: Handbook version used is VA Servicer Handbook M26-4 Chapter 9 dated August 12, 2024, retrieved via archive.org as the live VA site redirects to KnowVA.

This difference carries into the deliverables. The base model asks about 12 consecutive payments to a bankruptcy trustee. The trained model instead checks the handbook's bankruptcy conditions during the trial payment plan and the required Chapter 7 discharge disclaimer in the modification agreement. For example, its second test asks whether the servicer included "the VA-prescribed bankruptcy discharge disclaimer language verbatim in the executed VASP loan modification agreement".

Rubric items passed rise from 50.0% to 100%. The trained model takes 41 turns and generates 9,828 tokens, compared with 61 turns and 14,335 tokens for the base model. Recovering the source produces a more complete deliverable with a shorter trajectory in this pair. Together, the GDPval examples show improvements in calculations and source use outside the financial environments used for training.

J Ethics

Release Scope and Privacy: As part of our broader data collection process, we worked with a pool of 500+ investment professionals who are employed full-time at established investment firms. For onboarding, each contributor signed a Contributor's License Agreement prior to participating in the audio calls. This agreement grants us a perpetual, royalty-free license to their contributions and affirms our mutual commitment to compliance protocols (e.g. creating guardrails against material nonpublic information). We also removed any personally identifiable information and worked directly with several compliance teams of these funds.

We will release the 92 Hedge-Bench-Public environments, training and evaluation code, experiment configurations, synthesis and turn-attribution prompts, response schemas, and the trained adapter. Across all tasks, this release will include 2,967 source-file copies representing 730 distinct file contents. This release excludes the audio files, the transcribed audio and all metadata pertaining to the experts. Other environments, including monthly additions, remain private.

AI Use Statement: We used AI coding assistants to help implement evaluation and analysis tools, inspect trajectories and reward assignments, and revise the manuscript. The turn-level reward development procedure is described in Section 5; model-based scoring is described in Section 3.1.

K Full Training Reward and Attribution Prompts

The two training prompts are shown below, with internal version references expanded for clarity. The synthesis judge scores the completed report and explains its weaknesses. The attribution judge then uses that assessment and the full trajectory to assign weights to individual turns.

Appendix F describes how we developed the instructions through short training probes and coding-assisted review of saved assignments. The judge assigns credit and blame after seeing the completed trajectory and resolved outcome.

Each system prompt is paired with a JSON user message. For the objective, this contains `task_instruction`, `hidden_grading_rubric`, `final_answer_md`, `authoritative_grader_verdict`, and `objective_context`. Attribution receives these fields together with `objective_score`, `objective_synthesis_diagnoses`, `turn_count`, and `agent_turns`. Each turn is therefore assessed in the context of the completed trajectory. The returned turn indices map to eligible assistant-token spans as described in Equation 4.

The JSON schemas enforce the response fields and bounds shown below. Both stages use the training reward judge and aggregation procedure specified in Appendix F.

K.1 Synthesis Objective: Full System Prompt

Synthesis Objective: System Prompt

```
You grade the final answer of one completed financial research
↪  task.
```

```
Judge one holistic objective: gather and correctly use the evidence required by
the rubric, synthesize it into a rational answer to the exact prediction
question, and commit to the evidence-supported outcome the task requests.

The payload includes an authoritative citation-aware grader verdict, including
accepted and missed rubric moves and deterministic prediction correctness.
Treat those results as facts. Do not independently inflate rubric coverage
based on headings or broad discussion. Do not calculate a weighted blend or a
separate prediction bonus. Judge whether the answer succeeds as one complete
research-and-prediction product.

The authoritative normalized coverage and accepted/missed moves are hard
evidence about completion of the requested analysis. Inspect them before
scoring. A correct outcome cannot compensate for missed required moves, and
good prose cannot turn a missing move into coverage. The holistic score should
rise only when BOTH required evidence coverage and evidence-to-prediction
synthesis improve.

The main distinction you add is synthesis quality. Check whether the answer:
- answers the exact event, date horizon, and threshold;
- uses the strongest relevant evidence on both sides;
- draws the conclusion that follows from its own evidence rather than merely
  repeating management language;
- commits to a clear outcome consistent with the evidence and uncertainty;
- avoids unsupported confidence and semantic or temporal mistakes.

Use the full 0.0 to 1.0 range. A wrong binary prediction must be below 0.5. A
thoughtful near-boundary miss can approach 0.5; a confident prediction based on
a clear reasoning error should be much lower. A correct binary outcome is not
automatically strong and can also score below 0.5 when coverage, grounding, or
synthesis is poor. High scores require both strong accepted coverage and strong
prediction synthesis.

Use these qualitative anchors, not a weighted formula:
- Below 0.5: wrong prediction, or materially incomplete/unsupported analysis.
- 0.5 to 0.74: correct or plausible prediction but important required moves,
  evidence, or synthesis remain missing.
- 0.75 to 0.89: a clear majority of required moves are accepted and the
  prediction follows from the evidence, with only limited remaining defects.
```

```
- 0.90 to 1.0: near-complete accepted coverage and strong
↪  synthesis. Reserve
  1.0 for no material remaining defect.

For example, 0.50 normalized coverage means half of the required
↪  analytical
moves were not accepted. A correct prediction alone does not make
↪  that a 0.75+
answer.

If objective_score is below 1.0, decisive_error must name the
↪  largest remaining
coverage, synthesis, or prediction weakness, even when the
↪  prediction is correct.
Use "none" only for a genuinely complete 1.0 answer.

Return JSON only:
{
  "schema_version": 8,
  "objective_score": 0.0,
  "synthesis_assessment": "name the most important accepted and
  ↪  missed requirements, then state whether the conclusion
  ↪  follows",
  "prediction_assessment": "briefly assess the binary prediction
  ↪  and whether it follows from the evidence",
  "decisive_error": "the main evidence-to-conclusion error, or
  ↪  none"
}
```

K.2 Turn-Level Attribution: Full System Prompt

Turn-Level Attribution: System Prompt

```
You assign sparse signed causal attribution for one completed
↪  research trajectory.

The objective score, authoritative rubric verdict, recorded
↪  outcome, and
objective diagnoses are fixed facts. Do not rescore the trajectory,
↪  compare it
with sibling rollouts, or predict which attribution channel
↪  training will use.

Perform two separate audits: reasoning causality and tool evidence
↪  utility.
The final_answer_md is already graded through the fixed holistic
↪  objective
score. In the reasoning audit, report-writing turns can receive
↪  reasoning
credit or blame under the rules below. Do not reward or penalize a
↪  trajectory
merely because it is long or short.

SUCCESS: Would removing or degrading this turn materially weaken
↪  targeted
evidence collection, accepted rubric coverage, verification,
↪  rational
synthesis, or an evidence-derived prediction?

FAILURE: Did this turn materially cause a missed requirement,
↪  unsupported
```

```
claim, premature stopping, misuse or omission of relevant evidence, faulty
synthesis, or a wrong event, horizon, threshold, or outcome?
Would correcting this turn materially improve the final answer? Every failure
attribution must use failure_kind="answer_error".

REASONING TURN MAP RULES:
- Always perform both audits, regardless of score or prediction correctness.
- Wrong or low-scoring trajectories may retain success credit for sound work.
- Correct or high-scoring trajectories may receive failure blame for remaining
  coverage, synthesis, or prediction defects.
- The same turn may appear in both maps only if it materially helped one part
  of the objective and harmed another.
- Do not put tool evidence utility in these two turn maps; tool utility has its
  own field below. A report-writing turn may receive reasoning credit or blame
  only for the concrete synthesis, omission, unsupported claim, or prediction
  it commits. Protocol syntax remains a separate deterministic diagnostic.
- Never blame sound evidence gathering because later synthesis failed.
- Never blame a unique, reasonable search merely because it returned no match.
  Repeated reads, citation checks, failed commands, and report corrections are
  neutral unless the selected turn itself makes or commits a concrete answer
  error. A verification that finds and fixes a real defect can receive success
  credit.
- Attribute an omission only to a turn that caused premature stopping, ignored
  available evidence, or committed the omission in the final synthesis.
- If the submitted binary outcome is wrong, the failure map must contain at
  least one genuinely causal turn. Find the earliest decisive reasoning turn
  that formed or accepted the bad inference or outcome and the later
  report-synthesis turn that committed it, when these are distinct.
- A forecast-decision turn qualifies only when its actual assistant content
  chooses, accepts, or materially justifies the outcome or direction.
  Planning to assess the forecast, gathering or citing evidence, and discussing
  uncertainty without making the decision do not qualify. Never invent causal
  content that is absent from the selected turn.
- Report writing is not itself a failure. Blame a report turn only for the
  incorrect prediction, faulty synthesis, unsupported claim, or material
  omission it commits. Do not blame a final chat message that merely confirms
```

```
  completion.
- Evidence collected but ultimately unused is neutral in the
↪  reasoning maps.
  Routine navigation, generic planning, formatting, malformed tool
  ↪  syntax, and
  final confirmation receive zero unless the turn itself commits a
  ↪  concrete
  reasoning error.
- Use sparse attribution. Never distribute credit or blame
↪  uniformly. There is
  no target number of nonzero turns.
- Values represent causal strength and must be in (0,1]. Omit zero
↪  values.
- A channel may be empty only when genuinely no model-controlled
↪  turn passes
  its causal test. Score or correctness alone is not a valid empty
  ↪  reason.

Rubric coverage is not synthesis. The forecast must answer the
↪  exact event,
horizon, and threshold; weigh the strongest evidence on both sides;
↪  and choose
an outcome consistent with the evidence. Use the objective
↪  diagnoses to locate
each identified weakness in the reasoning or report turn that
↪  caused it.

TOOL EVIDENCE UTILITY:
- Audit each assistant turn that has tool_calls against all
↪  evidence available
  before that turn and the following_tool_outputs it produced.
- Positive utility means the call added new task-relevant evidence,
↪  a new
  calculation, or a verification that resolved a real uncertainty.
- Negative utility means the call was demonstrably redundant with
↪  evidence
  already available, irrelevant to the task, or avoidably failed
  ↪  and produced
  no usable information. A repeated call can be negative even if
  ↪  its syntax is
  valid.
- A unique reasonable search that returns no match is neutral.
↪  Never punish
  sound exploration merely because later synthesis ignored its
  ↪  evidence.
- Omit neutral calls. Omit turns that write or modify answer.md.
- utility is signed in [-1,1] excluding zero. The utility applies
↪  to every
  semantic command token in that tool-call turn; fixed wrapper
  ↪  syntax is masked
  separately.

Return JSON only:
{
  "schema_version": 8,
  "turn_count": N,
  "success_attributions": [
    {"turn_index": 0, "success_credit": 0.8, "reason": "specific
    ↪  causal contribution"}
  ],
  "success_empty_reason": null,
  "failure_attributions": [
```

```
    {"turn_index": 1, "failure_blame": 0.7, "failure_kind":
    ↪  "answer_error", "reason": "specific causal defect in the
    ↪  reasoning or final answer"}
  ],
  "failure_empty_reason": null,
  "tool_call_attributions": [
    {"turn_index": 2, "utility": -0.4, "evidence_delta":
    ↪  "duplicates evidence already available from turn 0",
    ↪  "reason": "specific utility judgment"}
  ]
}

Copy turn_count exactly. Indices must be unique within each list
↪  and between 0
and turn_count-1. The same index may occur once in each list. When
↪  a list is
nonempty, its empty_reason must be null. When empty, empty_reason
↪  must provide a
specific explanation.
```

L Frontier Model Behavior

These analyses use the same 3,036 reports and jury assessments as the main evaluation (Section 4). Every model contributes three reports for each of the same 92 tasks.

L.1 Report Length and Research Coverage

Table 36 compares report length with the two evaluation scores. Tokens are counted in the final report with the same `o200k_base` tokenizer for every model, including citation text. These counts measure written output rather than full trajectory length or inference cost. Each model contributes the 276 reports used for scoring.

Qwen3.8 Max writes the longest reports on average, at 16,777 tokens, with 53.54% coverage. GPT-5.6 Sol reaches 59.17% coverage with 5,713 report tokens. Gemini 3.8 Flash writes 7,887 tokens on average but covers 32.35% of the requirements. Longer reports therefore do not necessarily cover more of the required analysis. After subtracting each task's three-trial mean, the within-model correlation between log report length and coverage ranges from -0.009 to 0.194 across models.

Table 36: Report length and scores on Hedge-Bench-Public. Each row contains 276 scored reports. Report-token means are rounded to the nearest token, and coverage and accuracy are percentages.

Model	Mean report tokens	Coverage	Accuracy
GPT-5.6 Sol	5,713	59.17	69.57
Claude Opus 5	10,188	55.89	73.19
Claude Fable 5.1	7,713	55.85	71.38
Grok 4.6	9,703	58.25	64.49
Qwen3.8 Max	16,777	53.54	75.36
GLM 5.3	3,761	47.90	67.03
GPT-5.6 Terra	4,402	50.86	64.49
Muse Spark 1.3	5,521	47.00	61.23
Kimi K3	4,566	42.92	67.03
Gemini 3.8 Flash	7,887	32.35	67.03
DeepSeek V4 Pro 0813	2,627	37.26	61.96

L.2 Same-Task Report Comparisons

We examine two pairs of reports. Within each pair, the same model answers the same task: one forecast is correct but has low coverage, while the other is wrong but exceeds 60% coverage. Table 37 gives the scores, and the saved reports and juror explanations show which parts of the analysis differ.

Table 37: Contrasting trials on the same research assignment. Trial numbers refer to the three trials reported in the paper. Coverage counts requirements that receive final jury credit.

Assignment	Model	Trial	Coverage	Prediction
Fiserv: Argentina growth	Muse Spark 1.3	1	3/14 (21.43%)	Correct (Yes)
		3	10/14 (71.43%)	Incorrect (No)
Flywire: valuation	Claude Fable 5.1	1	3/12 (25.00%)	Correct (No)
		2	8/12 (66.67%)	Incorrect (Yes)

The Fiserv question asks whether Q1 2026 Argentina revenue will be lower year over year in as-reported dollars, before currency adjustment. Trial 1 distinguishes "as-reported dollars before currency adjustment, not constant-currency organic" and predicts Yes. It receives limited research credit. For requirement 1a, the Sol juror notes that the report does not identify Argentina's historical annual growth above 50% or connect it to the consolidated 9–12% organic-growth target. Trial 3 documents Argentina growth of 257% in 2023, 329% in 2024, and 56% year to date in 2025, with all three jurors crediting requirement 1a. Its final answer nevertheless predicts that revenue "will NOT be lower," reasoning from a normalized growth comparison. The resolved as-reported revenue falls from $194 million to $174 million. The higher-coverage report better documents the historical growth mechanism but misses the subsequent reported-dollar outcome.

The Flywire question asks whether the next quarterly report will disclose revenue dollars or a margin for an individual vertical. Trial 1 predicts No and receives 3/12 final credits. Trial 2 predicts Yes and receives 8/12. Trial 2 earns unanimous credit for relating a low forward EBITDA multiple to double-digit growth. The Gemini juror cites roughly 7× 2026 EV/adjusted EBITDA and guidance for 15–21% revenue-less-ancillary-services growth. The forecast instead turns on whether acquisition-level Sertifi revenue counts as a vertical disclosure. Trial 1 argues that "Sertifi is an acquired company, not the travel vertical," while Trial 2 treats its revenue as attributable to that vertical. The resolution records no qualifying vertical revenue-dollar or margin disclosure. In Trial 1, a majority of jurors recognize five other requirements but withhold credit because of evidence objections. The forecast is correct even though parts of the report are inadequately supported.

For investment research, the report makes the assumptions behind a position available for examination, while the forecast records whether the specified event occurs. These cases show why Hedge-Bench measures both. Financial returns would additionally depend on prices and position sizing, which are not outputs of this evaluation.

L.3 Coverage and Correctness Within the Same Task

To separate differences between models from differences between tasks, we analyze the three trials of each model separately. For model m, task e, and trial t, let C_{met} be coverage in percentage points and $Y_{met} \in \{0, 1\}$ indicate a correct prediction. We subtract each model–task pair's trial mean, giving $\widetilde{C}_{met} = C_{met} - \overline{C}_{me}$ and $\widetilde{Y}_{met} = Y_{met} - \overline{Y}_{me}$. Table 38 reports their correlation and the within-task slope,

$$b_m = 1000 \, \frac{\sum_{e,t} \widetilde{C}_{met} \widetilde{Y}_{met}}{\sum_{e,t} \widetilde{C}^2_{met}},$$

expressed in accuracy percentage points per ten coverage points. This slope captures an observational association across repeated trials rather than a causal effect of increasing coverage.

For each model, we permute correctness labels among the three trials within each task, keeping that task's number of correct predictions fixed. Under the test's null hypothesis, the correctness labels can be reassigned among the three trials of the same task. We use 100,000 permutations with seed 20260923 and a two-sided test of the within-task covariance, then apply Holm correction across the eleven models. All 92 tasks enter each calculation. The table also reports N_{mixed}, the number of tasks with both correct and incorrect predictions across trials.

The within-task correlations range from -0.123 to 0.119. None are statistically significant after Holm correction. This result separates the two measured outcomes: how much of the required research appears in the report, and whether its forecast matches the realized event. The full-environment versus

question-only comparison asks whether access to the research environment improves forecasting. This analysis asks a different question: among repeated full-environment reports, are higher-coverage reports more likely to predict correctly?

Table 38: Coverage–correctness association within each model and task. N_{mixed} counts tasks with both correct and incorrect trials. b_m is the accuracy-point difference per ten coverage points, and p is the two-sided permutation value.

Model	N_{mixed}	r	b_m	p	Holm p
GPT-5.6 Sol	11	0.056	1.25	0.449	1.000
Claude Opus 5	9	0.001	0.01	1.000	1.000
Claude Fable 5.1	8	−0.080	−1.15	0.293	1.000
Grok 4.6	5	−0.056	−0.60	0.431	1.000
Qwen3.8 Max	17	0.016	0.31	0.846	1.000
GLM 5.3	14	0.037	0.86	0.605	1.000
GPT-5.6 Terra	14	−0.123	−2.83	0.077	0.846
Muse Spark 1.3	16	0.017	0.34	0.825	1.000
Kimi K3	24	0.119	2.97	0.102	1.000
Gemini 3.8 Flash	11	0.094	1.06	0.240	1.000
DeepSeek V4 Pro 0813	21	0.029	0.86	0.686	1.000

L.4 Open-Weight and Proprietary Model Families

We group Qwen3.8, GLM 5.3, Kimi K3, and DeepSeek V4 Pro as families with publicly released weights, and compare them with the remaining seven proprietary families. This grouping reflects weight availability. All scores come from the evaluated endpoints and settings in Section 3.1. In particular, Qwen3.8 Max is the hosted version based on the released Qwen3.8 checkpoint, rather than a separately evaluated local deployment. Table 39 gives equal-model averages on all 92 tasks.

Forecast accuracy is nearly identical between open-weight (67.84%) and proprietary families (67.34%). In contrast, research coverage shows a clear gap: open-weight families average 45.41% compared with 51.34% for proprietary families, leading to joint success rates of 12.77% and 20.19%, respectively. Individual models overlap substantially: Qwen3.8 Max reaches 25.72% joint success and the highest forecast accuracy in the full comparison. These averages describe the models we evaluated.

Table 39: Weight-access groups on Hedge-Bench-Public. Means weight each model equally, with 276 reports per model. All scores are percentages.

Model group	Models	Reports	Coverage	Accuracy	Joint
Open-weight families	4	1,104	45.41	67.84	12.77
Proprietary families	7	1,932	51.34	67.34	20.19

L.5 Evidence-Support Penalties

We measure how often the jury removes credit from a requirement that the report addresses but does not adequately support. For report r, let H_r count majority-hit requirements and D_r count those that also receive a majority evidence objection. With K_r rubric requirements, final coverage is $100(H_r - D_r)/K_r$. Table 40 reports how often a report loses credit ($D_r > 0$) and what fraction of addressed requirements lose credit ($\sum_r D_r / \sum_r H_r$). It also gives the mean reduction in coverage ($100 D_r/K_r$). These counts show where the jury found missing or inadequate support, including claims it could not verify. They are not an independently measured rate of hallucinations across all claims.

The final saved assessments remove 3,723 of 22,377 majority-hit requirements across 1,622 reports. Gemini 3.8 Flash has the largest mean deduction, at 32.74 coverage points, whereas GPT-5.6 Terra and GLM 5.3 have deductions of 2.12 and 1.99 points. This explains part of the difference between discussing the requested topics and earning research credit. The same-task examples in Appendix L.2 show how evidence objections affect particular reports.

Table 40: Evidence objections that remove research credit. Report rate and removed-move rate are percentages, and the final column is the mean coverage deduction in percentage points.

Model	Reports	Rate	Moves	Rate	Deduction
GPT-5.6 Sol	99/276	35.87	137/2183	6.28	4.08
Claude Opus 5	195/276	70.65	396/2319	17.08	11.55
Claude Fable 5.1	169/276	61.23	328/2252	14.56	9.56
Grok 4.6	172/276	62.32	334/2332	14.32	9.57
Qwen3.8 Max	210/276	76.09	504/2355	21.40	14.65
GLM 5.3	52/276	18.84	68/1719	3.96	1.99
GPT-5.6 Terra	60/276	21.74	71/1825	3.89	2.12
Muse Spark 1.3	118/276	42.75	195/1817	10.73	5.70
Kimi K3	203/276	73.55	462/1939	23.83	13.46
Gemini 3.8 Flash	261/276	94.57	1121/2243	49.98	32.74
DeepSeek V4 Pro 0813	83/276	30.07	107/1393	7.68	3.13

L.6 Trajectory Length and Report Quality

We analyze all 3,036 trajectories from the 92 Hedge-Bench-Public tasks, with three trials per task for each of the 11 models. Table 41 reports mean output tokens and assistant turns per trajectory alongside the corresponding scores. Trajectory output-token counts come from each provider and include reasoning tokens where those are recorded. Final-report lengths in Table 36 instead use the same tokenizer for every model. These counts reflect tokens generated across the trajectory, rather than cumulative prompt tokens or context-window usage.

Table 41: Trajectory statistics across all 92 Hedge-Bench-Public tasks, with 276 reports per model. Output tokens and turns are per-trajectory means. Coverage and accuracy are percentages, and deduction is the mean evidence-support penalty in percentage points.

Model	Output tokens	Turns	Coverage	Accuracy	Deduction
GPT-5.6 Sol	63,670	55.1	59.17	69.57	4.08
Claude Opus 5	112,683	73.7	55.89	73.19	11.55
Claude Fable 5.1	41,705	40.6	55.85	71.38	9.56
Grok 4.6	121,434	62.8	58.25	64.49	9.57
Qwen3.8 Max	290,237	111.6	53.54	75.36	14.65
GLM 5.3	123,392	45.0	47.90	67.03	1.99
GPT-5.6 Terra	59,324	57.2	50.86	64.49	2.12
Muse Spark 1.3	142,876	57.5	47.00	61.23	5.70
Kimi K3	22,116	28.2	42.92	67.03	13.46
Gemini 3.8 Flash	90,455	75.2	32.35	67.03	32.74
DeepSeek V4 Pro 0813	61,713	31.1	37.26	61.96	3.13

We then compare longer and shorter trials of the same model on the same task. For each model, we subtract the task's three-trial mean from $\log(1 + \text{output tokens})$ and from each outcome before calculating the correlation. Table 42 relates length to final coverage, forecast correctness, and the evidence-support deduction. Two-sided permutation tests shuffle outcomes only within a task's three trials (100,000 draws), and Holm correction covers all 33 model–metric comparisons. None of these associations is statistically significant after correction. Across models, coverage correlations range from -0.254 to 0.113, and forecast-correctness correlations from -0.190 to 0.065. Longer trajectories therefore do not consistently correspond to better reports or forecasts within the same model and task.

Table 42: Within-model, within-task correlations with log trajectory output length across all 92 tasks. A positive deduction correlation means that longer trials lose more coverage to evidence objections. None is significant after Holm correction across the 33 comparisons.

Model	Coverage	Correctness	Deduction
GPT-5.6 Sol	−0.254	−0.017	0.114
Claude Opus 5	0.018	0.023	0.121
Claude Fable 5.1	0.113	0.061	0.002
Grok 4.6	−0.080	−0.029	0.091
Qwen3.8 Max	0.021	−0.190	0.047
GLM 5.3	0.040	−0.077	−0.100
GPT-5.6 Terra	0.020	0.065	0.002
Muse Spark 1.3	0.086	−0.110	−0.050
Kimi K3	0.085	0.056	−0.089
Gemini 3.8 Flash	0.040	−0.092	−0.069
DeepSeek V4 Pro 0813	0.027	−0.044	−0.036

M Operational Failures and Model-Level Case Studies

The final report does not show the steps an agent took to produce it. We inspect the saved trajectories for all 3,036 Hedge-Bench-Public reports across 92 environments, three trials, and 11 models. We first count observable execution patterns, then examine individual traces alongside their final jury scores. The examples below distinguish errors that the agent repairs from problems that remain in the submitted analysis. They are selected case studies, not estimates of how often each model makes a particular reasoning error.

M.1 Recurring execution patterns

Table 43 separates rejected command responses from responses that the harness repairs automatically. A rejection is recorded only when the harness explicitly returns a response-parsing error. We count an automatic repair when the harness emits a repair warning. That warning does not mean the command failed. These counts reflect both the model and the harness used for each run. Parser handling differed across runs.

Table 43: Observed execution patterns across all 92 environments. Each model has 276 trajectories. Entries report the number and percentage of trajectories meeting each condition. Repeated reads refers to at least five extra read calls that exactly repeat earlier calls. Long/low coverage denotes at least 100 recorded assistant turns and less than 20% final coverage. These conditions flag trajectories for inspection. A trajectory can meet more than one condition, and a flag does not by itself establish a failure.

Model	Rejection	Auto-repair	Repeated reads	Long/low coverage
Claude Fable 5.1	0 (0.0%)	4 (1.4%)	0 (0.0%)	0 (0.0%)
Claude Opus 5	0 (0.0%)	24 (8.7%)	0 (0.0%)	0 (0.0%)
DeepSeek V4 Pro	0 (0.0%)	8 (2.9%)	72 (26.1%)	0 (0.0%)
Gemini 3.8 Flash	61 (22.1%)	215 (77.9%)	0 (0.0%)	9 (3.3%)
GLM 5.3	1 (0.4%)	41 (14.9%)	214 (77.5%)	0 (0.0%)
GPT-5.6 Sol	0 (0.0%)	0 (0.0%)	71 (25.7%)	0 (0.0%)
GPT-5.6 Terra	0 (0.0%)	8 (2.9%)	44 (15.9%)	0 (0.0%)
Grok 4.6	0 (0.0%)	25 (9.1%)	0 (0.0%)	0 (0.0%)
Kimi K3	0 (0.0%)	14 (5.1%)	0 (0.0%)	0 (0.0%)
Muse Spark 1.3	0 (0.0%)	1 (0.4%)	154 (55.8%)	0 (0.0%)
Qwen3.8 Max	0 (0.0%)	2 (0.7%)	0 (0.0%)	8 (2.9%)

Gemini receives 1,961 rejected responses across 61 tasks, accounting for 9.45% of its 20,753 recorded assistant turns. Fifty-nine trajectories contain at least five rejections. The rejections occur in the first scored trial, whereas the second and third trials instead trigger automatic-repair warnings, totaling 10,040 turns across all three trials. Keeping these events separate avoids counting successful repairs as lost execution steps.

For repeated reads, we compare exact command strings after trimming surrounding whitespace. The screen includes shell read commands such as `cat`, `sed`, and `grep`, and excludes writes and checks of `answer.md`. It does not capture equivalent commands written differently or reads implemented in Python. A repeated read can be useful, particularly when earlier output was truncated. The GLM and DeepSeek examples below show agents repeatedly requesting excerpts that were lost from the terminal view, making output management part of the research problem.

The long/low-coverage screen identifies eight Qwen trajectories across eight tasks and nine Gemini trajectories across eight tasks. A broader descriptive threshold finds 119 Qwen trajectories with at least 100 turns that do not exceed 60% coverage. These cases motivate inspecting what the extra turns accomplish, rather than treating trajectory length as research completion. Appendix L reports the corresponding aggregate relationships with coverage and forecast accuracy.

M.2 Examples across all eleven models

Table 44 presents one inspected trajectory for each model. Turn numbers index assistant turns starting from 1. Coverage and correctness describe the final report. Several agents recover from an early programming mistake and continue their research successfully.

Table 44: Selected model-level behaviors in Hedge-Bench-Public. Each row identifies the company and research topic, with the scored trial in parentheses. All examples come from the saved trajectories used for the reported evaluation.

Model and task	Observed behavior and final outcome
Claude Fable 5.1 FND valuation (trial 1)	At turn 3, the agent slices a price-history dictionary as though it were a list. It inspects the keys on turn 4. A string-formatting error interrupts its financial table on turn 7, but the agent corrects the expression on turn 8. The agent promptly corrects these errors rather than repeating them. The 51-turn report receives 40.00% coverage and makes an incorrect forecast.
Claude Opus 5 LULU supply chain (trial 2)	A multiline Python command fails because it contains literal newline escapes (turn 32). The agent writes a script file instead and successfully prints the peer financial data on turn 33. Despite this recovery, the final report covers only 3/12 credited moves after 69 turns and 183,429 completion tokens, and the forecast is incorrect. Repairing execution errors does not substitute for completing the substantive analysis.
DeepSeek V4 Pro SNAP hardware (trial 1)	The agent requests the same 10-Q range on five consecutive turns (19–23). It repeatedly notes that earlier output was truncated, but continues requesting batches of excerpts and financial records. The run finishes in 32 turns with 25.00% coverage and a correct forecast. The repeated work concerns recovering source output, not changing the forecast.
Gemini 3.8 Flash IRDM constellation (trial 1)	Thirty-six of 134 turns receive command-format rejection feedback. The agent later reports that its own structural validator passes 149 citation occurrences. The jury nevertheless removes six of eight analytically present moves for inadequate support, leaving 2/12 credited moves (16.67%). The forecast is incorrect. Appendix M.3 examines this distinction.
GLM 5.3 SOC regulation (trial 1)	One 10-K range is requested 11 times, including every turn from 29 through 34. The agent says it has identified the main argument but still needs exact excerpts without terminal truncation. It continues bundling reads even while trying to obtain the missing passages. The report finishes after 61 turns with 33.33% coverage and a correct forecast.
GPT-5.6 Sol TKO revenue synergies (trial 2)	Its report-writing script relies on exact phrase matches. At turn 26, a locator fails, but by turn 28 the agent identifies a phrase split across physical lines. Further locators fail, and running the unchanged writer on turn 30 triggers the same error. The agent then switches to shorter anchors and multiline matching. The 53-turn report receives 50.00% coverage and an incorrect forecast.
GPT-5.6 Terra LULU niche sports (trial 3)	After a broad search produces too much output, the agent tries to restore the terminal prompt by sending `C-c` followed by a newline. The shell interprets this as a command and returns "command not found" (turn 5). Narrower searches then continue. The report completes in 40 turns with 33.33% coverage and a correct forecast.

Table 44, continued.

Model and task	Observed behavior and final outcome
Grok 4.6 APO private credit (trial 2)	The shell misinterprets a multiline extraction command on turn 15. The agent switches to a script file, but piping its large output through a truncating command produces a broken-pipe message. It writes an intermediate results file and reads excerpts from it instead. The 77-turn report receives 50.00% coverage and a correct forecast.
Kimi K3 FND competitive moat (trial 2)	Verification exposes incorrect peer figures and a mismatched fiscal date in the draft (turns 30–31). The agent corrects the numbers and date, and later removes a trading-volume claim attached to the wrong period. This is observable factual self-correction. The final report still receives only 1/12 credited moves (8.33%). Four are present in the text, but three lose credit for evidence support. Its forecast is correct.
Muse Spark 1.3 LULU geographic growth (trial 2)	The same 10-Q range is read 26 times, and another is read 15 times. Across turns 17–41, the agent repeatedly returns to collecting exact locators or a complete evidence set. The report eventually takes 90 turns and 317,488 completion tokens. Six moves are present, but four lose credit for support, leaving 16.67% coverage. The forecast is incorrect.
Qwen3.8 Max LULU supply chain (trial 1)	The agent repeatedly returns to final source checks before writing its report at turn 90. The completed trajectory uses 113 turns and 323,704 completion tokens. Its report contains 251 citation occurrences and all four theme headings, but the jury credits only 1/12 analytical moves (8.33%). The forecast is incorrect. Here, extensive output does not supply the required analysis.

M.3 Citation Validation and Evidence Support

The Gemini IRDM example separates two checks that can look similar in a trace. An agent can verify that a cited file exists and that the requested lines fall within it. The jury instead asks whether those lines support the analytical move. Gemini's own validation passes 149 citation occurrences, yet only two moves retain credit after evidence checking. The deductions therefore matter even in a report that has undergone extensive self-checking, as quantified across the full evaluation in Appendix L.5.

Qwen's TKO strategic-M&A report (trial 3) shows a related pattern without command-format rejections. It repeatedly announces that evidence collection is complete, then returns to source and locator checks. It first writes the report on turn 120 and continues checking through the 197-turn trajectory, using 302,360 completion tokens. Its validator passes 253 citation occurrences, but the jury removes three of eight present moves, leaving 41.67% coverage. The forecast is correct. The agent keeps checking its work, but the final analysis is still incomplete or inadequately supported. Unlike the repeated-read cases, it is not simply running the same command again.

Qwen's LULU supply-chain report (trial 1) illustrates a different dynamic. The report loses no majority-present moves to evidence deductions, but only one required move is present in the first place. The text includes the requested headings and abundant citations, but those properties do not establish that it performs the analyst's comparisons and inferences. Kimi's FND case provides another distinction. Its own checks genuinely correct draft numbers, but the final report still falls short of the rubric. Observing a useful repair is therefore different from observing a complete research argument.

An additional Gemini IRDM-IoT trajectory (trial 3) illustrates how self-checking can introduce its own execution work. After a quoting error leaves the shell in an unfinished string, the agent tries to interrupt it by sending the literal characters `\x03`. Bash reads them as a command, not an interrupt. Later, three consecutive attempts to run or repair the citation validator fail with syntax errors (turns 75–77). The trajectory nevertheless completes in 101 turns with 40.00% coverage and a correct forecast. We found this exact failed-interrupt pattern in one report. It illustrates a problem managing the shell, but it is not one of the recurring patterns counted in Table 43.