OPENPREDICTOR: HARNESSING GENERAL-PURPOSE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

General-purpose prediction must translate open-ended questions into resolvable targets and combine evidence with appropriate estimators across heterogeneous answer types. Language models offer a flexible interface, yet reasoning and tool access alone do not ensure that target semantics are preserved throughout execution. We present OpenPredictor, a unified harness that coordinates this process through explicit task and evidence contracts. Requests become typed targets with declared answer semantics, information cutoffs, and resolution rules. These contracts guide evidence acquisition and restrict estimator selection to compatible methods. The workflow separates model-generated hypotheses from registered numerical operations, retaining units, label identities, and computational dependencies through aggregation and decision construction. Mechanical checks and a model critic guide bounded repair, while linked records make decision derivations inspectable. On FutureX Past (Zeng et al., 2026), the complete workflow achieves substantial gains over direct answers on several backbones and lower numeric error on jointly answered tasks, although set and ranking accuracy remain challenging. These findings support contract-conditioned execution as a reusable approach to heterogeneous prediction. We will open-source OpenPredictor soon.

1 INTRODUCTION

Predicting events, numerical quantities, and future rankings supports science and planning. These tasks require different estimators and formats but share the need for defined targets, evidence, and resolvable answers, motivating a common prediction framework.

Language models interface with these requests through retrieval, reasoning, and aggregation (Halawi et al., 2024; Alur et al., 2025), while structured belief states organize evidence and forecast updates (Murphy, 2026). Open-ended forecasting extends beyond binary events (Chandak et al., 2025; Zeng et al., 2026); agent frameworks organize actions, observations, and modular computation (Yang et al., 2024; Wang et al., 2025; Khattab et al., 2024).

Integrating these capabilities requires preserving a prediction's meaning across language, evidence, and computation. Numbers may be probabilities or measurements; entity lists may be sets or rankings. Relevant evidence can measure the wrong quantity; retrospective evaluation also risks temporal leakage (Paleka et al., 2026). An executable method may estimate the wrong target, while a cited explanation leaves its numerical derivation unspecified. Explicit interfaces must therefore connect target semantics to evidence, admissible methods, and answer construction.

We introduce OpenPredictor, a six-stage prediction framework (Figure 1). *Frame* (S0) converts the request into typed targets with units, horizons, answer semantics, and resolution rules. *Plan* (S1) selects routes, candidate methods, and evidence requirements. *Gather* (S2) retrieves sources under the information cutoff and separates source text from observations with explicit values, units, and vintages. A pre-reasoning gate checks temporal admissibility, evidence adequacy, and method compatibility. *Reason* (S3) generates hypotheses and tool estimates, executes compatible methods, and selects distinct successes using holdout scores when available. Registered operators pool aligned numerical values or named distributions, or fuse rankings. Calibration uses compatible outcome data when supported; otherwise the estimate is retained for decision construction.

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.

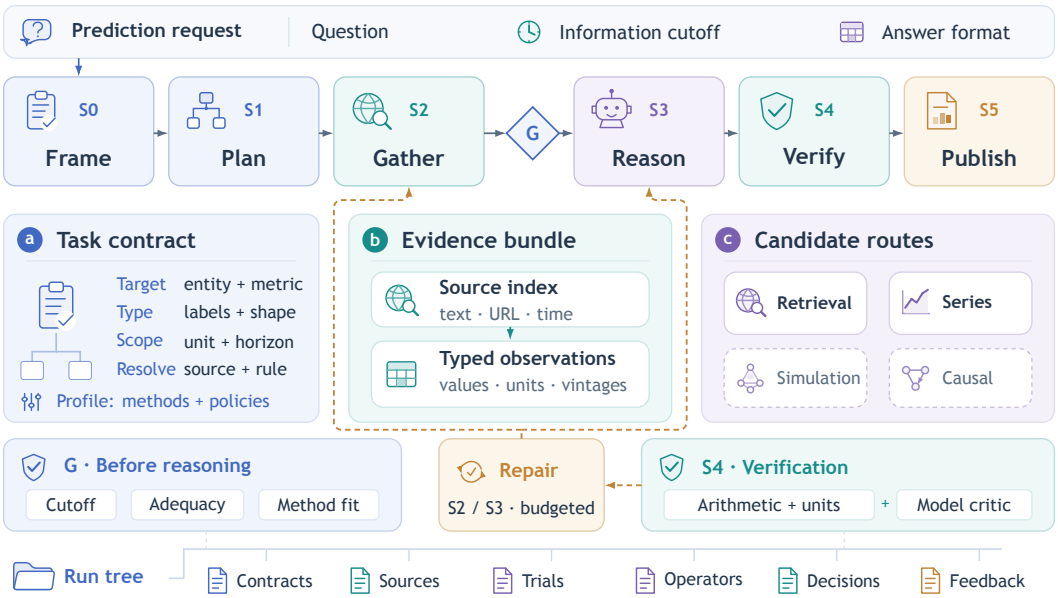


Figure 1: **Execution architecture with explicit checks and recorded state.** Gate G checks the cutoff, evidence adequacy, and method compatibility before S3; S4 combines numerical checks with a model critic. Budgeted repair returns to S2 or S3, while the run tree retains typed artifacts and feedback. Dashed route boxes are implemented but not evaluated; benchmark runs skip S5 report rendering.

Verify (S4) combines arithmetic, unit, and dependency checks with a model critic. Findings can return execution to S2 or S3 under a bounded repair budget; changed inputs invalidate dependent computations before re-evaluation. *Publish* (S5) optionally renders a report bound to the recorded decisions; benchmark mode exports answers without report rendering. A persistent run tree links sources, candidate outputs, operations, and feedback to each decision. Task profiles specialize this shared execution graph for different target and estimator families, providing the framework's general-purpose organization.

Evaluation on FutureX Past (Zeng et al., 2026) retains final submissions and execution failures across six answer types. It exercises retrieval and series; simulation, causal, and multi-target interfaces are implemented but unevaluated. Our contributions are:

- We introduce a unified prediction harness that connects typed task contracts, evidence requirements, compatible estimators, and recorded decisions. A shared execution graph combines model-mediated candidate generation with registered numerical operations and dependency-aware repair, preserving units, labels, and computational lineage across heterogeneous workflows.
- We evaluate the complete workflow across multiple backbones and heterogeneous answer types. The results show improved deterministic scores on several backbones and lower numeric error on shared answered tasks. Analyses of output coverage, structured-answer quality, and operational failures characterize the framework's benefits and remaining limitations.

2 Related Work

Forecasting with language models. Retrieval, reasoning, and aggregation are established forecasting ingredients. Halawi et al. (2024) combine historical news with trimmed-mean aggregation; heterogeneous LLM ensembles support judgmental forecasts (Schoenegger et al., 2024). AIA Forecaster uses agentic research, supervision, and recalibration (Alur et al., 2025); BLF maintains linguistic belief states and studies aggregation and calibration through ablations (Murphy, 2026). Open-

Forecaster trains open-ended forecasters on historical news (Chandak et al., 2025). Our focus is a common representation connecting heterogeneous answer semantics, evidence, compatible estimators, and decision provenance, rather than a new aggregation or calibration algorithm.

Agent harnesses and executable workflows. ReAct interleaves reasoning and actions (Yao et al., 2023); AutoGen programs agent conversations (Wu et al., 2024), and CodeAct uses executable actions (Wang et al., 2024). SWE-agent and OpenHands structure agent–environment interaction (Yang et al., 2024; Wang et al., 2025). DSPy provides declarative module signatures (Khattab et al., 2024), while AFlow searches code-represented workflows (Zhang et al., 2025). Component studies separate planning, action space, and context management (Fan et al., 2026), and distinguish execution success from terminal acceptance (Zhang et al., 2026). OpenPredictor specializes these interfaces to prediction: task contracts constrain evidence and estimator compatibility within a fixed graph with bounded repair.

Executable computation and provenance. PAL and Program of Thoughts delegate arithmetic to programs (Gao et al., 2023; Chen et al., 2023). Our registered operators additionally check units, labels, and horizons before combining predictions. Provenance models represent entities, activities, and derivations (Moreau & Missier, 2013). Linked inputs and transformations make our decisions inspectable independently of their explanations, without certifying the premises.

Temporal evaluation and heterogeneous outputs. ForecastQA and Autocast connect forecasts to dated information (Jin et al., 2021; Zou et al., 2022). ForecastBench evaluates unresolved probabilities (Karger et al., 2025); FutureX covers choices, rankings, and numeric outcomes (Zeng et al., 2026). Agentic Time Machine approximates a past web environment with planning, parallel solvers, and aggregation (Chai et al., 2026). Our retrospective filter addresses a related information boundary. Temporal leakage and benchmark construction can undermine apparent gains (Paleka et al., 2026); these results are therefore not prospective leaderboard claims. Probability consistency (Paleka et al., 2025) and proper scoring rules (Gneiting & Raftery, 2007) address questions beyond our hard-answer L1/L2 endpoint.

3 OpenPredictor

Given a request q and information cutoff c, OpenPredictor constructs typed targets, acquires evidence, executes compatible estimators, and records decision dependencies. Task profiles specialize the six-stage graph in Figure 1.

3.1 Task representation and execution policy

Framing produces a directed acyclic task graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \dots, n_t\}$. At node i, the contract $t_i = (q_i, \tau_i, e_i, \nu_i, u_i, h_i, \rho_i, \psi_i)$ specifies the question, answer type, entity, metric, unit, horizon, resolution rule, and scoring rule. Counterfactual targets additionally require an identification specification. A profile π_i declares compatible routes and methods, evidence requirements, and aggregation, calibration, and decision policies; r_i is the selected route. Stages check these fields; the benchmark scorer is separate from ψ_i.

The estimator value space $\mathcal{Z}_i$ contains event probabilities in $[0, 1]$, aligned numerical arrays in $\mathbb{R}^{d_i \times H_i}$ for d_i channels and H_i horizon steps, or categorical distributions in the simplex $\Delta(\mathcal{L}_i)$ over finite labels $\mathcal{L}_i$. Rankings preserve ordered labels. The decision policy maps estimates to answer formats, such as probabilities to yes/no answers; the resolution rule defines the scored outcome. Type correctness alone does not establish target correctness.

Let R_k contain contracts, evidence, outputs, feedback, and remaining budgets at step k. Following agent–environment interfaces (Yang et al., 2024), write $\mathcal{H} = (\Phi, \mathcal{A}, \mathsf{U}, \mathsf{E})$ for context construction, admissible actions, state update, and export. A model-mediated step is

$$a_k \sim P_\theta(\cdot \mid \Phi(q, c, R_k)), \qquad R_{k+1} = \mathsf{U}(R_k, a_k, o_k), \tag{1}$$

where P_θ is the model's sampling distribution and o_k includes results and failure feedback. Violations of $\mathcal{A}(t_i, R_k)$ trigger failure or bounded repair; registered numerical operations execute without

sampling. This controller assumes neither a learned policy nor Markov state. Independent subtasks can run concurrently; dependencies require ordered execution and explicit composite computation. Up to ten subtasks are supported; each evaluated question has one submission target.

3.2 Evidence acquisition under task constraints

Planning associates targets with methods, observations, and applicable holdouts. Gathering separates indexed source text from typed observations binding entity, metric, value, unit, source, and temporal scope. Measurements, aliases, proxies, and context remain distinct. Observation time differs from vintage: a measurement before c may have been revised after c.

Before reasoning, adequacy checks enforce observation, source, unit, vintage, and temporal requirements. Strict execution may stop; the evaluated best-effort policy can retain limitations and continue with a judgmental estimate. Deduplication, retrieval budgets, and failed-domain limits bound acquisition.

The cutoff constrains search, page filtering, and observation purging. A retrospective filter additionally uses resolved-answer terms, excluded from the prediction plan, to suppress revealing pages. Retrospective evaluation still risks contamination (Sainz et al., 2023; Paleka et al., 2026). Section B.4 details model-memory and vintage limitations here.

3.3 Candidate generation and method selection

Trials generate hypotheses and tool outputs through retrieval, series forecasting, simulation, or causal estimation; mixed dispatch combines retrieval and series. Candidates need not be independent. The admissible estimator set is

$$\mathcal{M}_i = \mathcal{M}_{\pi_i} \cap \mathcal{M}_{\tau_i} \cap \mathcal{M}_{r_i} \cap \mathcal{M}_{\text{exec}}, \tag{2}$$

where the sets impose profile, target-type, route, and resource constraints. Compatibility precedes ranking: lower error cannot justify a different target or horizon. Successful outputs retain units, axes, versions, and sources; failed methods are excluded and equivalent serialized outputs collapsed.

Let $\mathcal{J}_i$ index distinct successes and $\mathcal{S}_i \subseteq \mathcal{J}_i$ those with usable horizon-matched holdouts. When $\mathcal{S}_i \neq \varnothing$, the default rule for $j \in \mathcal{S}_i$ is

$$J_{ij} = \ell_{ij}^{\text{val}} + \lambda_s \sigma_{ij} + \lambda_c \kappa_{ij}, \qquad w_{ij} = \frac{(J_{ij} - J_i^{\min} + \epsilon)^{-1}}{\sum_{j' \in \mathcal{S}_i} (J_{ij'} - J_i^{\min} + \epsilon)^{-1}}, \tag{3}$$

where ℓ_{ij}^{val} is validation MAE, σ_{ij} a nonnegative stability penalty, κ_{ij} a resource proxy, $J_i^{\min} = \min_{j \in \mathcal{S}_i} J_{ij}$, and $\epsilon > 0$. Only scored outputs receive weight when $\mathcal{S}_i$ is nonempty; otherwise distinct successes receive $1/|\mathcal{J}_i|$. An empty $\mathcal{J}_i$ yields no estimate. Version-compatible historical weights may override this rule. Section F.2 gives coefficients and scale assumptions. Selection weights and aggregation coefficients are recorded separately.

3.4 Typed conclusion and computational provenance

Following executable numerical reasoning (Gao et al., 2023; Chen et al., 2023), let $Z_i = (z_{ij})_{j \in \mathcal{K}_i}$ contain selected outputs. Aggregation A_{π_i}, calibration C_{π_i}, and decision policy D_{π_i} produce

$$b_i = A_{\pi_i}(Z_i, \alpha_i), \qquad \widetilde{b}_i = C_{\pi_i}(b_i; \mathcal{D}_i^{\text{cal}}), \qquad \widehat{y}_i^{\text{int}} = D_{\pi_i}(\widetilde{b}_i), \tag{4}$$

where $\mathcal{K}_i$ indexes selected members, $\mathcal{D}_i^{\text{cal}}$ contains compatible calibration data, and $\widehat{y}_i^{\text{int}}$ is the internal decision. Mean-based pooling of aligned values uses

$$b_i = \sum_{j \in \mathcal{K}_i} \alpha_{ij} z_{ij}, \qquad \alpha_{ij} \geq 0, \quad \sum_{j \in \mathcal{K}_i} \alpha_{ij} = 1. \tag{5}$$

Here α_i need not equal selection weights w_i. With common units and coordinates, convex pooling preserves a convex value domain. Fixed coefficients bound output perturbation by the largest input perturbation in the same norm. This excludes adaptive selection and discrete decisions; label stability also requires a sufficient top-two margin. Median and trimmed mean are separate operators. Section F.3 gives assumptions and derivations.

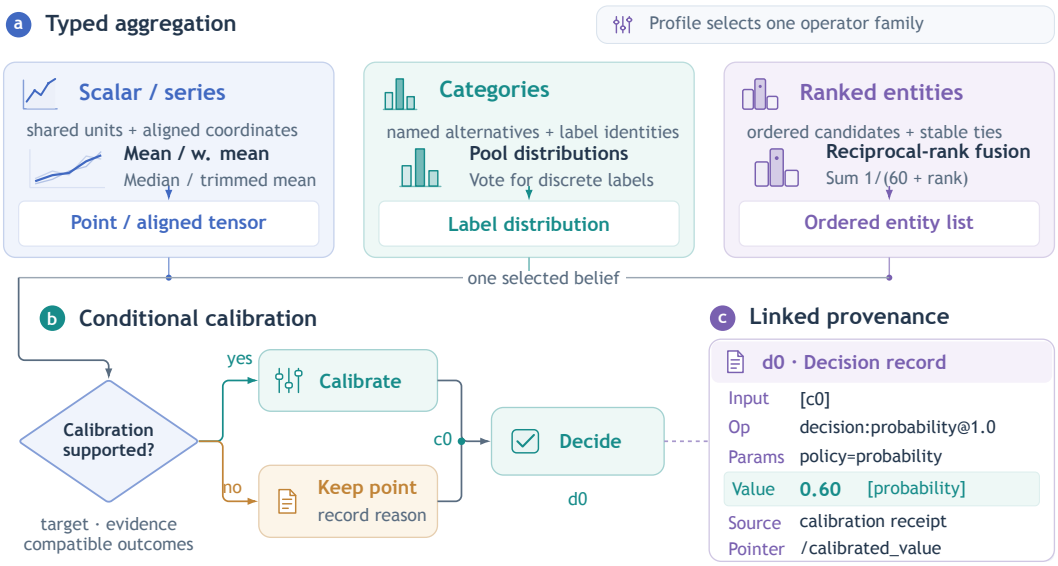


Figure 2: **Typed conclusion with conditional calibration and node-level records.** A profile selects the aggregation family; ranking uses reciprocal-rank fusion (Cormack et al., 2009). The selected belief follows the supported-calibration or point-preserving branch before decision. Support requires an applicable target, adequate evidence, and compatible outcomes. The linked decision record illustrates input, operation, parameter, value, and source bindings; its IDs and value are schematic, not experimental results.

For ranked lists L_{ij}, where $L_{ij}[r]$ is the rth entry, reciprocal-rank fusion (Cormack et al., 2009) assigns entity e the score

$$g_i(e) = \sum_{j \in \mathcal{K}_i} \sum_{r=1}^{|L_{ij}|} \frac{\mathbf{1}\{L_{ij}[r] = e\}}{60 + r}. \tag{6}$$

Here $\mathbf{1}$ is the indicator. Descending scores, label-based tie-breaking, and top-k selection preserve identities without guaranteeing correct order.

Calibration requires compatible resolved outcomes (Guo et al., 2017; Angelopoulos & Bates, 2023). Without applicable support, the workflow records the reason and retains b_i. The evaluated receipts contain no out-of-sample observations, establishing no calibration benefit. Optional debate reconstructs aggregates from recorded anchors (Appendix F); report and action synthesis can remain model-mediated.

The computation DAG stores each node's value, unit, operation version, ordered inputs, parameters, and source. Leaves distinguish observations, tool outputs, and judgments. Fixed inputs and deterministic operations permit replay to declared precision; checks validate references, units, arithmetic, and root agreement. Replay records derivation, not factual correctness (Moreau & Missier, 2013).

3.5 Verification and Task-Specific Export

Mechanical checks and a model critic assess substantive support and request bounded gathering or reasoning repair. A changed subtask set $U \subseteq \mathcal{V}$ invalidates $\mathcal{I}_{\mathcal{G}}(U) = U \cup \bigcup_{i \in U} \mathrm{Desc}_{\mathcal{G}}(i)$, including all strict descendants. The controller regenerates these outputs in dependency order and refreshes shared evidence and decisions (Figure 12). This maintains dependency consistency, not monotonic forecast improvement. Feedback-based revision follows agent practice (Shinn et al., 2023), without a guarantee of intrinsic self-correction (Huang et al., 2024). Our benchmark wrapper allows four episode attempts.

At terminal step K, the submission is $\widehat{y}_i = \mathsf{E}(R_K)_i$. Export can consult trial hypotheses, retain an earlier format-valid candidate, or apply recorded cross-question adjustments; thus $\widehat{y}_i$ can differ from $\widehat{y}_i^{\mathrm{int}}$. Evaluation uses the submission and retains missing answers in the denominator. Appendix G

illustrates an execution. Optional publication binds report values to analysis; benchmark mode skips it.

4 EVALUATION

We evaluate complete-system gains, answer quality, and execution costs. Table 1 retains the full metric comparison; the figures examine how those aggregate results arise. All analyses reuse completed predictions.

4.1 DATA, EXECUTION, AND SCORING

Corpus and conditions. We use 282 resolved FutureX Past instances (Zeng et al., 2026) from four nonconsecutive bundles: 2026-06-W5, 2026-07-W1, 2026-07-W3, and 2026-08-W1. They contain 74 L1, 79 L2, 51 L3, and 78 L4 tasks and 215 distinct IDs, joined by (identifier, resolution date). The six answer types comprise 53 yes/no, 92 choices, 15 sets, 11 entities, 62 rankings, and 49 numeric answers; these types are distinct from benchmark levels (Appendix B).

Direct makes one request with the question and answer format, without tools or retries. OpenPredictor uses `deep` mode and our wrapper's `escalate` budget: three trials, a six-source structured-retrieval ceiling, a critic, and up to four attempts. Each task has one submission target; vision and report rendering are disabled. Four pairs yield 2,256 task slots, including failures, across DeepSeek V4 Flash, Claude Opus 4.6, Gemini 3.6 Flash, and GPT-5.6 Terra. Full runs use the day before resolution as cutoff; Direct has no retrieval. Retrieval, inference, and recovery vary together, so these are configured-system comparisons, not ablations or equal-cost tests (Section B.3).

Endpoints and uncertainty. For level $\ell \in \{1, 2\}$, let $\mathcal{I}_\ell$ contain n_ℓ questions. For backbone m and condition $a \in \{\mathrm{D}, \mathrm{F}\}$ (Direct or full), $s^a_{mi} \in [0, 1]$ is the submitted answer's score and $S^a_{m\ell} = 100\, n_\ell^{-1} \sum_{i \in \mathcal{I}_\ell} s^a_{mi}$. The primary endpoint and paired difference are

$$S^a_{m,\mathrm{det}} = \frac{S^a_{m1} + S^a_{m2}}{2}, \qquad \Delta_m = S^{\mathrm{F}}_{m,\mathrm{det}} - S^{\mathrm{D}}_{m,\mathrm{det}}. \tag{7}$$

L1/L2 use FutureX's deterministic scorer (Zeng et al., 2026); our analysis assigns zero to missing or invalid answers. For 80 non-numeric instances (39 L3, 41 L4), the same DeepSeek V4 Pro judge scores normalized answers without system identifiers; missing answers remain zeros. This endpoint is internally comparable but is not the official leaderboard judge. Normalization and hidden identities do not eliminate judge bias (Zheng et al., 2023). For the 49 numeric instances, let $\mathcal{N}^a_m$ contain returned numeric values $\widehat{y}^a_{mi}$ with reference $y_i \neq 0$. We report $100\,\mathrm{median}_{i \in \mathcal{N}^a_m}(|\widehat{y}^a_{mi} - y_i|/|y_i|)$ and $n = |\mathcal{N}^a_m|$. This conditional error excludes non-answers and is not averaged into Equation (7); the official four-level outcome-variability normalizer is unavailable.

Coverage counts parseable scored outputs over all 282 tasks. Our paired bootstrap (Efron, 1979) uses 5,000 replicates, resampling weeks and then questions within week and level. Two-sided sign-flip tests use 10,000 draws, with Holm correction (Holm, 1979) across four backbones. These intervals describe question-sampling uncertainty, not independent reruns. Only four weeks are available; all 153 primary-endpoint IDs are distinct. Figure 5 and Appendix C compare week-based and task-only intervals.

4.2 COMPLETE-SYSTEM GAINS AND THEIR HETEROGENEITY

Backbone and level differences. DeepSeek, Gemini, and GPT improve by 27.17, 22.98, and 12.43 primary-score points, respectively, with positive week-bootstrap intervals and Holm-adjusted $p_{\mathrm{H}} < 0.001$, $p_{\mathrm{H}} < 0.001$, and $p_{\mathrm{H}} = 0.013$ (Table 1 and Figure 5). The workflow changes the observed backbone ordering: GPT has the highest Direct score, 57.54, while DeepSeek reaches the highest full score, 79.55. Thus Direct performance alone does not predict the benefit from the additional workflow. The gains also differ by level: Gemini's L2 score rises from 39.42 to 73.21, a much larger change than its L1 increase from 54.05 to 66.22; DeepSeek improves strongly on both. These are heterogeneous system gains, without identifying responsible modules or statistically resolved backbone rankings.

Table 1: **Direct answers versus OpenPredictor across four backbones.** Our results on 282 FutureX Past questions (Zeng et al., 2026). L3/L4 report non-numeric scores; numeric errors include returned-answer counts. The primary column equally weights L1/L2; scores use a 0–100 scale.

		Deterministic ↑		Judge ↑		Primary ↑	Numeric ↓
System	Valid (%)	L1	L2	L3	L4	L1/L2	Err. (%) [n]
DeepSeek V4 Flash							
Direct	89.01	59.46	45.30	10.43	4.96	52.38	16.34 [48]
OpenPredictor	98.23	82.43	76.67	28.89	15.45	79.55	2.92 [49]
Claude Opus 4.6							
Direct	85.46	50.00	49.21	8.55	2.44	49.61	18.55 [34]
OpenPredictor	66.31	51.35	59.61	7.69	3.74	55.48	3.45 [32]
Gemini 3.6 Flash							
Direct	95.04	54.05	39.42	9.91	9.35	46.74	17.92 [49]
OpenPredictor	95.39	66.22	73.21	25.88	9.59	69.72	2.06 [49]
GPT-5.6 Terra							
Direct	80.14	60.81	54.27	5.98	7.32	57.54	10.87 [47]
OpenPredictor	98.58	74.32	65.62	13.16	15.45	69.97	1.79 [49]

Availability constrains realized performance. Claude's increase from 49.61 to 55.48 has an interval spanning zero. Its coverage is 66.31%, and all 95 exhausted tasks terminate with a quota error at framing. Those tasks cannot benefit from downstream reasoning or repair. On the 95 primary tasks answered in both conditions, its difference is +27.92 points, but this selected subset does not estimate the missing outage-free outcome. Retaining failures therefore measures a different, operationally relevant quantity from conditional answer quality.

Numeric estimates and structured predictions. On the exact returned-task intersections, median numeric error falls from 16.34% to 3.23% for DeepSeek, 32.47% to 3.42% for Claude, 17.92% to 2.06% for Gemini, and 10.87% to 1.79% for GPT (Figure 5). The direction persists when answer support is held fixed, so changed non-answer rates alone do not explain these reductions. Support still ranges from 22 to 49 tasks, and medians do not characterize tail errors. The non-numeric judge endpoint behaves differently: Gemini's L3 score improves substantially while its L4 score changes little, and Claude's L3 decreases. Successful numerical estimation therefore coexists with unresolved entity selection and ordering errors.

Variation across time bundles. DeepSeek, Gemini, and GPT improve on every included bundle (Figure 7). On the newest bundle, their Direct scores are only 31.7, 34.2, and 31.7, while workflow gains reach +45.0, +33.3, and +25.0 points. This pattern is compatible with external evidence helping where an unaided answer is weak, but gains are not uniformly monotone over time. Topics, difficulty, provider availability, and possible model memory co-vary. The four retrospective bundles establish neither a memory effect nor prospective temporal robustness.

4.3 What improves, and what remains difficult?

Coverage versus conditional quality. For a fixed backbone, let V_i^a indicate a scorer-valid answer, $v_\ell^a = n_\ell^{-1} \sum_{i \in \mathcal{I}_\ell} V_i^a$, and μ_ℓ^a be mean score among valid answers, when nonempty. Zero credit otherwise implies $S_\ell^a/100 = v_\ell^a \mu_\ell^a$ and

$$\frac{S_\ell^{\mathrm{F}} - S_\ell^{\mathrm{D}}}{100} = \underbrace{(v_\ell^{\mathrm{F}} - v_\ell^{\mathrm{D}})\mu_\ell^{\mathrm{D}}}_{\text{coverage term}} + \underbrace{v_\ell^{\mathrm{F}}(\mu_\ell^{\mathrm{F}} - \mu_\ell^{\mathrm{D}})}_{\text{conditional-quality term}} . \tag{8}$$

The decomposition in Figure 3 averages these terms with the primary endpoint's equal-level weights. DeepSeek's +27.17 points comprise +1.72 coverage and +25.45 conditional quality; GPT's +12.43 comprise +0.42 and +12.01. Gemini gains despite a negative coverage term, -2.63, offset by +25.60

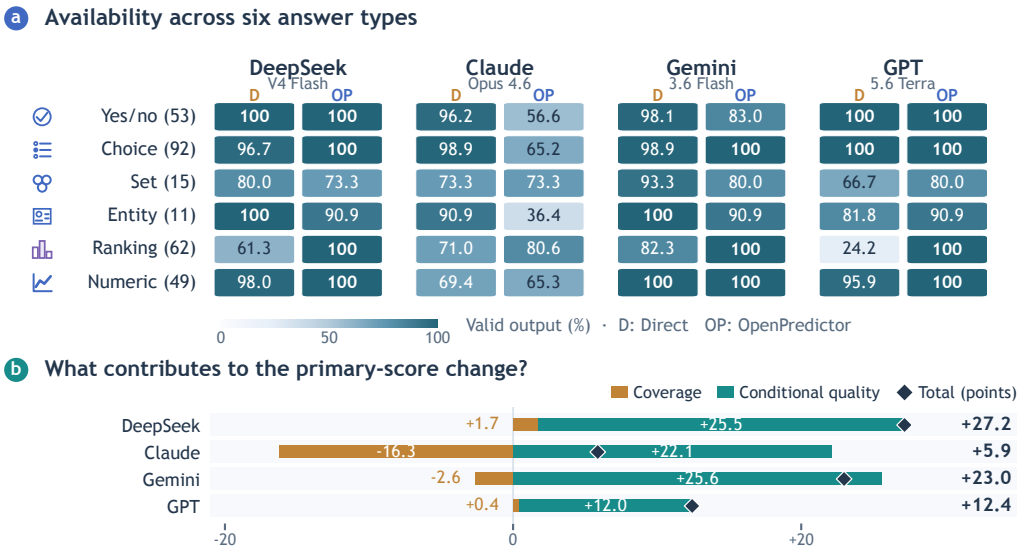


Figure 3: **Answer availability and the sources of score changes.** (a) Scorer-valid outputs by backbone and type; parentheses give all tasks of each type, including failures. D denotes Direct and OP denotes OpenPredictor. (b) Exact L1/L2 decomposition into coverage and conditional-quality terms, with diamonds marking the sum before rounding. Validity is not correctness; the decomposition is descriptive and does not identify component effects.

quality points. Claude combines a -16.25 coverage term with +22.13 quality points. The main improvements therefore exceed what additional valid outputs alone explain. Jointly reporting coverage and conditional quality follows selective prediction (Geifman & El-Yaniv, 2019). Our identity is descriptive: valid subsets differ, and Direct's quality is the anchor, so its terms do not identify causal component effects.

Output structure and semantic quality. The heatmap in Figure 3 shows why whole-corpus coverage must be distinguished from the primary endpoint. GPT's ranking coverage rises from 15/62 to 62/62, accounting for 47 of its 52 additional valid outputs; 59 of the 62 ranking tasks are outside L1/L2. This large structural improvement can therefore coexist with the small +0.42 primary coverage term. DeepSeek and Gemini also return all 62 rankings and 49 numeric answers. Complete structure still leaves difficult content: the median per-backbone score on 59 judged rankings rises only from 2.26 to 7.08. Deterministic single-choice score rises from 46.43 to 75.60, while entity-set score falls from 65.69 to 39.89 on 11 tasks (Figure 6). These patterns motivate checking entity membership, cardinality, and order separately from parseability. Sparse entity and deterministic-ranking strata cannot support broad type-level rankings.

Task composition helps explain this separation: yes/no and choice questions comprise 137 of 153 primary tasks, versus 16 sets, entities, or rankings. Equal level weights do not balance answer types, limiting generalization to open-ended outputs. Figure 6 exposes variation hidden by backbone medians. Larger structured-output strata and separate membership and ordering diagnostics would distinguish format recovery from better predictions.

Evidence volume and execution scope. The retained-page analysis (Figure 8) is non-monotone: one-to-three-page instances gain +31.3 points, versus +18.2 for four or more pages. The latter group also has a higher Direct baseline, so document volume alone is a poor explanation of the contrast. Even zero-retained-page instances gain +17.2 points, but average 7.62 searches and 25.13 hits; earlier attempts, snippets, and tools may supply information. These selected, instance-weighted strata are not a retrieval ablation or the primary equal-level estimator. Final-attempt records exercise retrieval, series, and mixed routes, without simulation or causal execution. All 977 calibration receipts contain zero out-of-sample observations. Accordingly, the observed gains cannot be presented as evidence for those broader routes or learned calibration.

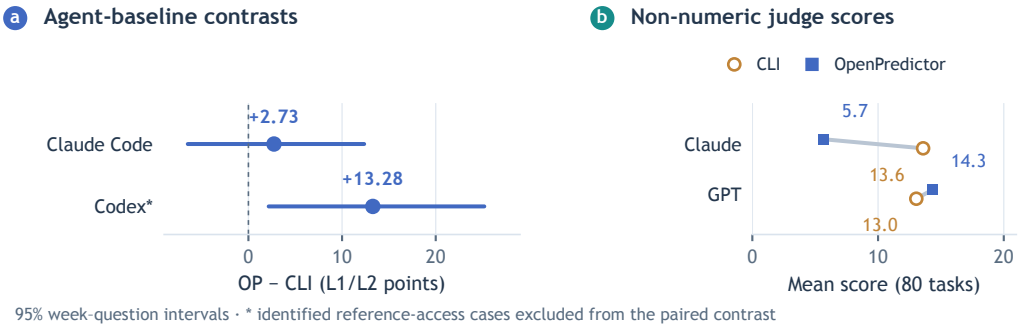


Figure 4: **Comparison with general-purpose agent workflows.** (a) Paired L1/L2 differences with 95% week–question bootstrap intervals; the Codex contrast excludes identified reference-access cases and remains descriptive. (b) Non-numeric L3/L4 scores on all 80 tasks, including non-answers. The full metric breakdown appears in Table 3.

Resource implications. Full median summed attempt times are 430.5, 636.1, 662.8, and 397.3 seconds, versus 1.4–16.1 seconds for Direct (Figure 9). Similar means of 2.24–2.42 attempts conceal different distributions: Claude has 142 one-attempt tasks and 130 four-attempt tasks, whereas GPT has 155 two-attempt tasks. The mean thus hides concentrated recovery effort. Gathering consumes 49.0–64.9% of recorded input tokens; Gemini spends 26.8% on planning. These profiles inform future budget controls, without identifying stage-level accuracy effects. Incomplete token logs are lower bounds, not bills; joint accuracy–cost evaluation remains necessary (Kapoor et al., 2025). Retries contribute one terminal submission, retaining failures. Appendix G illustrates a successful execution.

4.4 Comparison with general-purpose agents

Agent CLIs can retrieve and execute tools, providing a stronger reference than Direct. Their nominal backbones and cutoffs match OpenPredictor, while prompts, retrieval, and compute can differ. Claude Code (Anthropic, n.d.) scores 52.75; OpenPredictor's paired difference is +2.73 [-6.48, +12.36] points (Figure 4). The interval includes zero. The judged endpoint favors Claude Code, 13.58 versus 5.67, even though both have 25 non-answers: the CLI earns partial credit on 13 tasks versus one for OpenPredictor. This reversal shows why a deterministic gain cannot be treated as uniform superiority over answer types (Figure 10).

Codex CLI (OpenAI, n.d.) scores 56.76. Its records show reference-answer access in a small subset, so the comparison remains descriptive. Excluding identified cases gives +13.28 [+2.16, +25.19] points but does not establish full isolation. Across all primary tasks, 39 receive positive score only from OpenPredictor and 15 only from Codex; positive score includes partial credit. On judged tasks, both produce 17 positive-score answers despite reference-length matches of 91.2% versus 43.8%. Better output shape alone therefore does not resolve semantic quality. Appendix E and Table 3 retain the audit and full metrics. Controlled information access and matched computational budgets are needed before attributing these contrasts to the harness itself.

5 Conclusion

OpenPredictor organizes general-purpose prediction around explicit targets, admissible evidence, compatible methods, and recorded decisions. Its central design is a common execution contract linking evidence acquisition, method selection, typed numerical operations, and computational provenance. The coverage–quality decomposition makes the sources of aggregate score changes explicit without assigning causal effects to individual modules. Completed experiments show substantial total-workflow gains on three backbones and lower numeric error on shared answered tasks across four. They also expose quota sensitivity, weak entity-set and ranking quality, and limits of retrospective agent comparisons. These findings motivate prediction systems whose accuracy, output coverage, and computational provenance are evaluated together.

AI USE STATEMENT

Generative AI tools were used as writing and programming assistants during the preparation of this work. Writing assistance included drafting and revising text to improve language, clarity, and organization. Coding assistance included implementing, debugging, and refining software and analysis scripts. These tools supported the writing and development process; the authors remain responsible for reviewing and validating AI-assisted text and code, verifying the reported claims and results, and ensuring the accuracy and integrity of the final manuscript.

REFERENCES

Rohan Alur, Bradly C. Stadie, Daniel Kang, Ryan Chen, Matt McManus, Michael Rickert, Tyler Lee, Michael Federici, Richard Zhu, Dennis Fogerty, Hayley Williamson, Nina Lozinski, Aaron Linsky, and Jasjeet S. Sekhon. AIA forecaster: Technical report, 2025. URL https://arxiv.org/abs/2511.07678.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Anthropic. Claude Code: Overview. Official documentation, n.d. URL https://code.claude.com/docs/en/overview. Accessed 26 September 2026.

Jingyi Chai, Bingyang Zheng, Xiangrui Liu, Hao Lu, Zihang Zhou, Tianchen Wang, Kemeng Zhang, and Siheng Chen. Agentic time machine as an infrastructure for future-event forecasting, 2026. URL https://arxiv.org/abs/2606.21013.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Scaling open-ended reasoning to predict the future, 2025. URL https://arxiv.org/abs/2512.25070.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.12588.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114. URL https://doi.org/10.1145/1571941.1572114.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

Morten W. Fagerland, Stian Lydersen, and Petter Laake. The McNemar test for binary matched-pairs data: Mid-p and asymptotic are better than exact conditional. *BMC Medical Research Methodology*, 13:91, 2013. doi: 10.1186/1471-2288-13-91.

Run-Ze Fan, Zihao Zhang, Simin Ma, Yebowen Hu, Shouju Wang, Kaiqiang Song, Fei Liu, Hamed Zamani, and Xiaoyang Wang. An empirical study of harness design for coding agents, 2026. URL https://arxiv.org/abs/2609.20804.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10764–10799, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2151–2159, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. Approaching human-level forecasting with language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2402.18563.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8b4add8b0aa8749d80a34ca5d941c355-Abstract-Conference.html.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3 edition, 2021. URL https://otexts.com/fpp3/. Section 5.10; online version accessed 26 September 2026.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. ForecastQA: A question answering challenge for event forecasting with temporal text data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4636–4650, 2021. doi: 10.18653/v1/2021.acl-long.357. URL https://aclanthology.org/2021.acl-long.357/.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025. URL https://collaborate.princeton.edu/en/publications/ai-agents-that-matter/.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, 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://arxiv.org/abs/2409.19839.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/f1cf02ce09757f57c3b93c0db83181e0-Paper-Conference.pdf.

Luc Moreau and Paolo Missier (eds.). *PROV-DM: The PROV Data Model*. World Wide Web Consortium, 2013. URL https://www.w3.org/TR/2013/REC-prov-dm-20130430/. W3C Recommendation, 30 April 2013.

Kevin Murphy. Agentic forecasting using sequential bayesian updating of linguistic beliefs, 2026. URL https://arxiv.org/abs/2604.18576.

OpenAI. Codex CLI. Official documentation, n.d. URL https://developers.openai.com/codex/cli. Accessed 26 September 2026.

Daniel Paleka, Abhimanyu Pallavi Sudhir, Alejandro Alvarez, Vineeth Bhat, Adam Shen, Evan Wang, and Florian Tramèr. Consistency checks for language model forecasters. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5107a33432b9a5cffafd0a53ef6c6a18-Abstract-Conference.html.

Daniel Paleka, Shashwat Goel, Jonas Geiping, and Florian Tramèr. Pitfalls in evaluating language model forecasters. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b9d69c72e4587ddf3fa6352890aefc4f-Abstract-Conference.html.

Belinda Phipson and Gordon K. Smyth. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010. doi: 10.2202/1544-6115.1585.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722/.

Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, Rafael Valdece Sousa Bastos, and Philip E. Tetlock. Wisdom of the silicon crowd: LLM ensemble prediction capabilities rival human crowd accuracy. *Science Advances*, 10(45):eadp1528, 2024. doi: 10.1126/sciadv.adp1528.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 50208–50232, 2024. URL https://proceedings.mlr.press/v235/wang24h.html.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a4b6ad6b48850c0c331d1259fc66a69c-Paper-Conference.pdf.

Anderson M. Winkler, Gerard R. Ridgway, Matthew A. Webster, Stephen M. Smith, and Thomas E. Nichols. Permutation inference for the general linear model. *NeuroImage*, 92:381–397, 2014. doi: 10.1016/j.neuroimage.2014.01.060.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Conference on Language Modeling*, 2024. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2023/08/LLM_agent.pdf.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, Yixiao Tian, Jinpeng Wang, Zaiyuan Wang, Yang Yang, Lingyue Yin, Mingren Yin, Zhenwei Zhu, Tianle Cai, Zehui Chen, Jiecao Chen, Yantao Du, Xiang Gao, Jiacheng Guo, Liang Hu, Jianpeng Jiao, Xiangsheng Li, Jingkai Liu, Shuang Ni, Zhoufutu Wen, Ge Zhang, Kaiyuan Zhang, Xin Zhou, Wenhao Huang, Jose Blanchet, Xipeng Qiu, and Mengdi Wang. FutureX: An advanced live benchmark for

LLM agents in future prediction. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/21315abf210cef71317e887e9cda78b3-Paper-Conference.pdf.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5492ecbce4439401798dcd2c90be94cd-Abstract-Conference.html.

Yukun Zhang, Kemu Xu, and Yishen Chen. How do agent harnesses create value? planning information and release control in stateful LLM agents, 2026. URL https://arxiv.org/abs/2609.20474.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.05685.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. Forecasting future world events with neural networks. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022. URL https://arxiv.org/abs/2206.15474.

A SUMMARY

This appendix collects the protocol, statistical definitions, implementation details, and archived evidence supporting the main results. Supplementary analyses reuse the same 282 FutureX Past questions and four backbone pairs, without new prediction or judge calls.

Appendices B and C specify task identity, scoring, information access, estimands, and uncertainty. Appendices D and E examine output types, execution diagnostics, and agent-baseline validity. Appendices F and G connect the mathematical operators and repair logic to a correct prompt-to-result case. Appendix H describes offline checks and reproduction; Appendix I closes with limitations and research directions. Post-hoc subsets and illustrative cases remain distinct from whole-corpus comparisons, and computational properties are not prediction guarantees.

B EVALUATION PROTOCOL

B.1 CORPUS, TASK IDENTITY, AND OUTPUT STRATA

Table 2: Corpus composition. Each (ID, resolution date) is a distinct instance; 215 IDs produce 282 instances. Deterministic scoring covers 153 instances, non-numeric L3/L4 scoring covers 80, and numeric error covers 49.

Bundle	Tasks	L1	L2	L3	L4
2026-06-W5	84	20	31	14	19
2026-07-W1	67	18	19	9	21
2026-07-W3	71	24	9	15	23
2026-08-W1	60	12	20	13	15

We select four completed FutureX Past bundles (Zeng et al., 2026); Table 2 reports our selected corpus. Development bundles and one bundle with inconsistent task accounting are excluded. This is an availability-based corpus rather than an independently registered random sample. The resolved dates do not make the retrospective runs prospective forecasts.

Questions are distinguished by both identifier and resolution date: 215 identifiers generate 282 instances. The analysis checks matching task sets, complete denominators, and agreement between per-question and aggregate scores. Non-answers remain explicit observations. All reported results use the completed evaluation snapshot; subsequent reruns do not replace these observations.

The post-hoc answer-shape classifier uses the following precedence: explicit yes/no format, ranked-list request, numeric reference, multi-element reference, public lettered options, and otherwise an entity answer. It is used only to stratify evaluation. The prediction-time answer contract is inferred from the public prompt; it does not use reference values from this analysis classifier. Thus a reporting shape need not equal the model's internal target type.

B.2 SCORING INSTRUMENTS

L1/L2 use the released FutureX matching rules (Zeng et al., 2026). Our analysis scores missing or invalid predictions as zero. The primary score equally weights the two level means, rather than pooling 74 L1 and 79 L2 tasks.

Non-numeric L3/L4 scoring comprises 39 L3 and 41 L4 instances. The same DeepSeek V4 Pro judge normalizes/extracts answers and scores them without system identifiers. Its extraction-based validity is therefore not purely a syntax check. The judge receives the reference required for scoring; it is distinct from the prediction model and from the official hosted leaderboard judge. Identical treatment enables an internal comparison, but does not remove judge bias (Zheng et al., 2023).

The remaining 12 L3 and 37 L4 instances are numeric. Their references are nonzero, permitting absolute relative error. Error is conditional on a returned numeric value and is accompanied by its valid count. The outcome-variability normalizer needed for the official numeric score is unavailable. Neither a numeric leaderboard score nor an official four-level aggregate is imputed.

B.3 Conditions, model configuration, and budgets

Direct uses one provider request with the public question and output format, without tools, retrieval, or retries. OpenPredictor uses the `deep` runtime with our wrapper's `escalate` budget: three reasoning trials, a six-source structured-selection ceiling, critic enabled, vision disabled, and no full report rendering. Gathering ceilings are 16 tool iterations, 12 search calls, and 18 fetch calls. These are configured limits, not measured usage. Gathering repair is capped at two and verifier repair at one. A provider transport call allows at most two attempts; the batch independently allows up to four full-episode attempts. Those retry levels must not be conflated.

The backbones are DeepSeek V4 Flash, Claude Opus 4.6, Gemini 3.6 Flash, and GPT-5.6 Terra. Hosted model names do not identify immutable model weights; the corresponding request configurations are retained in the evaluation artifacts. The full-versus-direct contrast evaluates each system as configured, including retrieval, inference budget, and recovery. It is not a matched-cost or isolated-component comparison.

B.4 Temporal information policy

The prediction input contains the public question, answer schema, and a cutoff set to the calendar day before resolution. Queries, page filtering, and typed observation purging use that cutoff. A separate retrospective filter also consumes resolved-answer terms and resolved-event signals to suppress revealing pages. Input checks verify that these labels are not included in the prediction plan, but do not establish complete isolation throughout execution.

This filtering uses information unavailable at the historical cutoff. It can alter the distribution of legitimate pre-cutoff evidence as well as remove revealing pages. Its success is not an independent leakage audit. Model memory remains a contamination concern (Sainz et al., 2023; Paleka et al., 2026); undated sources and revised historical values are additional limitations here. A historical Wikipedia revision or frozen benchmark context can preserve a specific version. A current financial series merely truncated to an old date need not preserve the values originally published. The experiments therefore approximate a historical information set and do not establish vintage fidelity or a source-count-based leakage rate.

C Statistical estimands and sensitivity analyses

C.1 Level-balanced paired inference

We use the score and index conventions of Section 4.1. For backbone m, define the paired difference $d_{mi} = s^{\mathrm{F}}_{mi} - s^{\mathrm{D}}_{mi}$. The observed primary contrast is

$$\Delta_m = 50 \sum_{\ell=1}^{2} \overline{d}_{m\ell}, \qquad \overline{d}_{m\ell} = \frac{1}{n_\ell} \sum_{i \in \mathcal{I}_\ell} d_{mi}. \tag{9}$$

The factor 50 combines equal weighting of the two levels with the 0–100 reporting scale. This is neither an instance-pooled nor an equal-week estimator. All original instances, including failures, remain in the denominators.

Let $\mathcal{W}$ contain the four observed week bundles and $\mathcal{I}_{w\ell}$ the instances in week w and level ℓ. Our paired resampling adapts the bootstrap principle (Efron, 1979). For each replicate, draw four weeks independently with replacement. For each selected occurrence of w, resample $|\mathcal{I}_{w\ell}|$ paired instances with replacement within each level. Pool these resampled multisets by level and recompute Equation (9), using the replicate's level-specific sample sizes. Direct and full scores remain paired. The reported 95% interval uses the empirical 2.5% and 97.5% endpoints of $B = 5{,}000$ replicates. It does not average four week-specific contrasts.

For the two-sided sign-flip test (Winkler et al., 2014), independently draw $\xi_i^{(b)} \in \{-1, +1\}$ with equal probability and compute

$$T_m^{(b)} = 50 \sum_{\ell=1}^{2} \frac{1}{n_\ell} \sum_{i \in \mathcal{I}_\ell} \xi_i^{(b)} d_{mi}, \qquad p_m = \frac{1 + \sum_{b=1}^{B_{\mathrm{sf}}} \mathbf{1}\{|T_m^{(b)}| \geq |\Delta_m|\}}{B_{\mathrm{sf}} + 1}, \tag{10}$$

with $B_{\text{sf}} = 10{,}000$, the plus-one Monte Carlo correction (Phipson & Smyth, 2010), and numerical tolerance for ties. This test assumes joint sign invariance under the null; paired observations alone do not guarantee that assumption. Both procedures use seed 17. For valid marginal p-values, Holm adjustment (Holm, 1979) controls family-wise error across four backbones, separately for the combined endpoint and each level. For sorted values $p_{(1)} \le \cdots \le p_{(4)}$, it is

$$p_{\text{H},(r)} = \min\left\{1, \max_{1\le j\le r}(5-j)p_{(j)}\right\}, \qquad r \in \{1,\dots,4\}. \tag{11}$$

Figure 5 displays the primary comparisons and both resampling intervals.

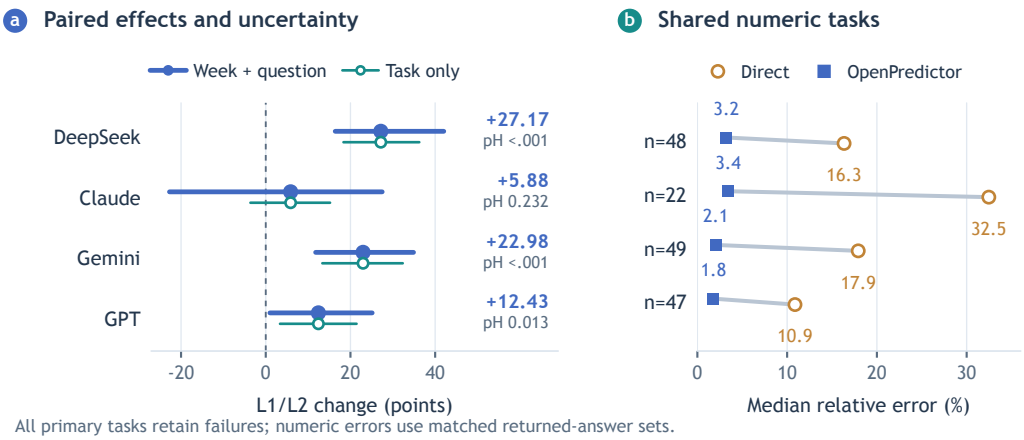


Figure 5: **Paired uncertainty and matched-answer numeric sensitivity.** (a) The same L1/L2 difference with 95% intervals from week–question and task-only resampling. Annotations give the difference and Holm-adjusted paired sign-flip p-value. (b) Median absolute relative error on exactly the same returned numeric tasks in both conditions. Rows follow the backbone order in (a); counts are out of 49. Conditional numeric errors do not represent missing answers.

Only four week clusters are available. All 153 primary-endpoint IDs are distinct; repeated IDs occur outside L1/L2. The task-bootstrap sensitivity resamples paired instances directly and recomputes the same level-balanced estimator, without week-level resampling. It is not a correction for repeated primary-endpoint IDs. These intervals describe uncertainty from the specified resampling scheme, not variation from independently rerunning the systems.

C.2 Coverage and Conditional Quality

For a fixed backbone, suppress m and let $\mathcal{V}_\ell^a = \{i \in \mathcal{I}_\ell : V_i^a = 1\}$. When $|\mathcal{V}_\ell^a| > 0$, define

$$v_\ell^a = \frac{|\mathcal{V}_\ell^a|}{n_\ell}, \qquad \mu_\ell^a = \frac{1}{|\mathcal{V}_\ell^a|}\sum_{i\in\mathcal{V}_\ell^a} s_i^a, \qquad \frac{S_\ell^a}{100} = v_\ell^a \mu_\ell^a. \tag{12}$$

The last equality follows from zero scores outside $\mathcal{V}_\ell^a$. Adding and subtracting $v_\ell^{\text{F}}\mu_\ell^{\text{D}}$ gives Equation (8). Equal-level averaging yields $\Delta_m = C_m + Q_m$. Restoring the backbone index, the contributions are

$$C_m = 50\sum_{\ell=1}^{2}(v_{m\ell}^{\text{F}} - v_{m\ell}^{\text{D}})\mu_{m\ell}^{\text{D}}, \qquad Q_m = 50\sum_{\ell=1}^{2} v_{m\ell}^{\text{F}}(\mu_{m\ell}^{\text{F}} - \mu_{m\ell}^{\text{D}}). \tag{13}$$

Figure 3 displays the two contributions for all four backbones, with their sum marked separately. The archived figure data retain the unrounded values and their exact agreement with the primary score difference.

This identity is anchored at Direct's conditional quality. Reversing the anchor redistributes the interaction term without changing Δ_m. If a valid-output set is empty, its score and coverage are zero but

its conditional quality is undefined; the decomposition is then not reported. The connection to risk–coverage analysis (Geifman & El-Yaniv, 2019) is descriptive: coverage here includes operational failures rather than a learned abstention policy. Because the valid subsets can differ, neither C_m nor Q_m is a causal component effect.

The common-output L1/L2 sensitivity restricts both conditions to $\mathcal{V}^{\mathrm{D}}_{m\ell} \cap \mathcal{V}^{\mathrm{F}}_{m\ell}$ at each level. Likewise, numeric errors in Figure 5 are recomputed on $\mathcal{N}^{\cap}_m = \mathcal{N}^{\mathrm{D}}_m \cap \mathcal{N}^{\mathrm{F}}_m$, with each condition using that same set. These conditional comparisons address unequal returned-answer support, not the unobserved performance of missing answers. An empty intersection has no reported median; errors are not imputed.

The matched primary sets contain 148, 95, 142, and 152 tasks for DeepSeek, Claude, Gemini, and GPT. Their conditional differences are +27.01, +27.92, +28.01, and +12.52 points. The much smaller Claude set illustrates how conditioning on returned answers can change both composition and the estimand; it does not replace the whole-corpus comparison.

C.3 Task-level outputs and repeated attempts

Each instance contributes its terminal submitted answer once, including any retained candidate or consistency adjustment. Attempts do not create additional evaluation observations. A missing answer retains zero credit on deterministic and non-numeric L3/L4 scores; numeric error instead uses its reported returned-answer set. Figure 8 summarizes this accounting.

Retry histories contain 211 tasks with at least two nonempty answers, of which 19 repeat the same recorded string throughout. Retries follow failures and may change evidence or feedback. These selected pairs therefore do not estimate independent run-to-run stability or provide a one-sided bound on it.

D Supplementary empirical analyses

D.1 Answer types and per-backbone coverage

Figure 3 reports every backbone separately, with the same shape denominators and 0–100% color scale. Each heatmap cell equals the number of valid scored outputs divided by all tasks of that shape. Validity does not establish semantic correctness. The underlying 48 counts/rates are saved in the figure's analysis sidecar and checked against the score-derived aggregate.

Figure 6 displays quality within separate scoring regimes. Large markers summarize four per-backbone statistics; small markers retain the variation among those backbones. These medians differ from the individually displayed availability rates in Figure 3. L1/L2 and judge quality cannot be mixed merely because their displayed scales both run from 0 to 100. The entity and deterministic-ranking strata have only two and three instances.

D.2 Bundle composition and recency

The chronological bundle analysis in Figure 7 reports Direct scores beside full-minus-direct differences, with n counting L1/L2 tasks. Direct is lower on the newest bundle than the oldest for all four backbones. DeepSeek, Gemini, and GPT have positive contrasts in all included bundles, but the contrasts are not uniformly monotone with recency. Topic, difficulty, score composition, provider availability, and possible model memory all co-vary. These four bundles cannot identify a memory effect or temporal robustness. Claude's two newest bundles include quota failures.

D.3 Operational failures and execution-order subsets

Returned and no-answer counts in Figure 8 sum to 282 for each backbone. Returned counts describe the terminal submission, while validity in the main tables is determined by the scorer. Consistency adjustments are included in those submissions rather than reported as a separate system.

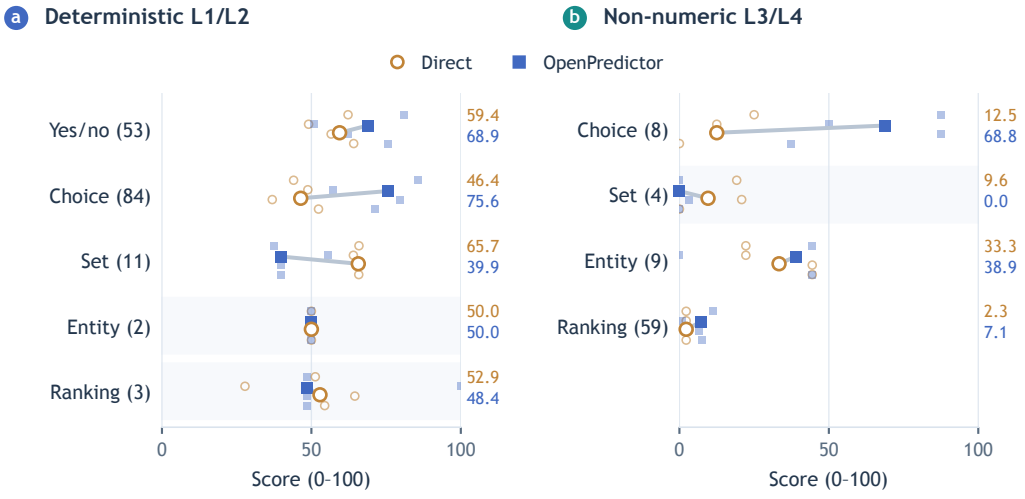


Figure 6: **Answer quality under separate scoring regimes.** Large markers and annotations show medians of four per-backbone statistics; small markers show the individual backbones. Parentheses give stratum sizes, with sparse strata shaded. Deterministic and judged scores retain failures within their respective denominators; their common display scale does not make them interchangeable. Numeric sensitivity is shown separately in Figure 5.

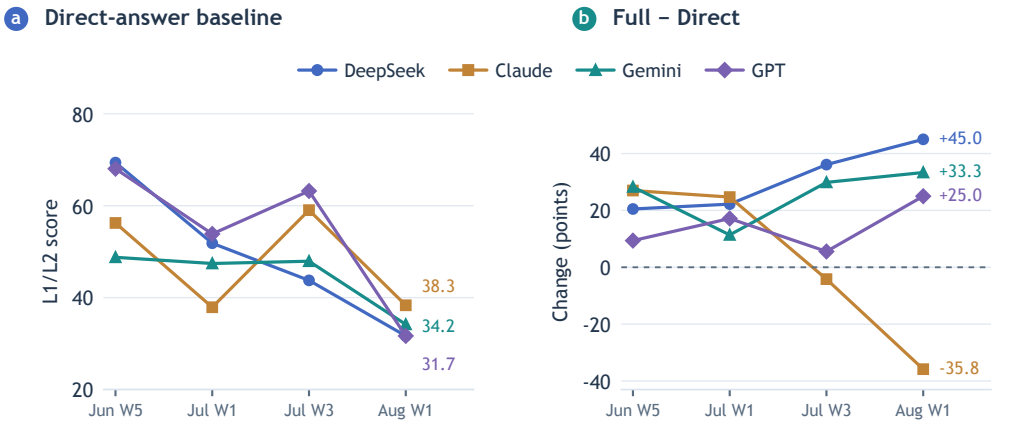


Figure 7: **Baseline scores and workflow gains across chronological bundles.** Each point uses the same L1/L2 level-balanced statistic within its bundle; all failures remain included. Lines connect the four observed bundle categories, which are not equally spaced in calendar time. Differences in topic, difficulty, provider availability, and possible model memory prevent a causal recency interpretation.

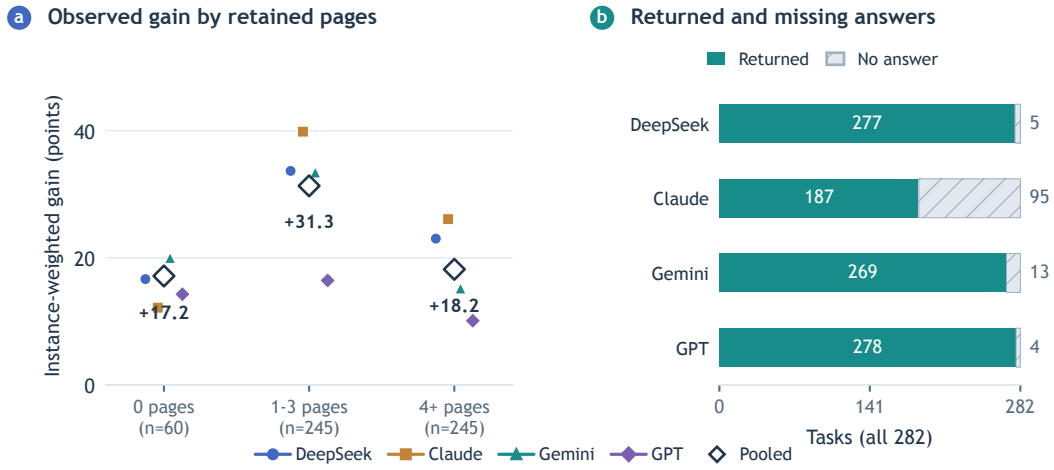


Figure 8: **Retained-page strata and terminal answer accounting.** (a) Instance-weighted L1/L2 gains within final-attempt retained-page strata; small markers show each backbone and outlined diamonds show the pooled value. The strata contain 60, 245, and 245 accepted model–task pairs. They are selection-conditioned observations, not a retrieval ablation. (b) Terminal returned and no-answer counts sum to 282 for each full campaign; right annotations give the no-answer counts. Returned answers can still fail the scorer's validity check.

The 95 unanswered Claude tasks terminate with quota failures during task framing. These counts describe observed execution outcomes, not the answers that would have been produced with uninterrupted availability.

An execution-order analysis identifies an initial group of 159 Claude tasks with no terminal failure. Its 96 deterministic instances yield 86.04 for OpenPredictor and 55.49 for Direct (+30.54 points). This is an execution-order subset, not an outage-free counterfactual or a randomized comparison.

The subset disproportionately represents earlier question bundles, so its composition differs from the full corpus. All headline results retain the original 282 tasks and all 95 quota failures; this subset is used only to illustrate the interaction between availability and observed performance.

D.4 Recorded resource usage

Per-task full runtime sums the recorded durations of its attempts, excluding queue time. Concurrent tasks and model calls prevent summing those durations into campaign wall time. Recorded input/output counters omit some framing, routing, and failed requests; they are lower bounds, not bills. Judge calls are excluded. Token counts are interpreted within each backbone because counting conventions can differ.

The stage-share block uses all run records retained before campaign finalization, whereas the per-task token block uses attempts in the terminal task histories. The two scopes need not reproduce the same totals. Gathering accounts for 69.2M recorded input tokens (64.9%), compared with 31.1M (29.2%) for reasoning. Gemini assigns 26.8% to planning. These are execution-specific profiles; they do not identify which component caused a score improvement. Joint reporting of accuracy and cost remains necessary (Kapoor et al., 2025).

D.5 Retained-page strata are not a retrieval ablation

The left panel of Figure 8 shows the retained-page strata, including each backbone and the pooled instance-weighted result.

The page analysis pools 550 accepted L1/L2 model–task instances and takes an instance-weighted mean, not the primary equal-level estimator. It joins the scored export to document counts from the

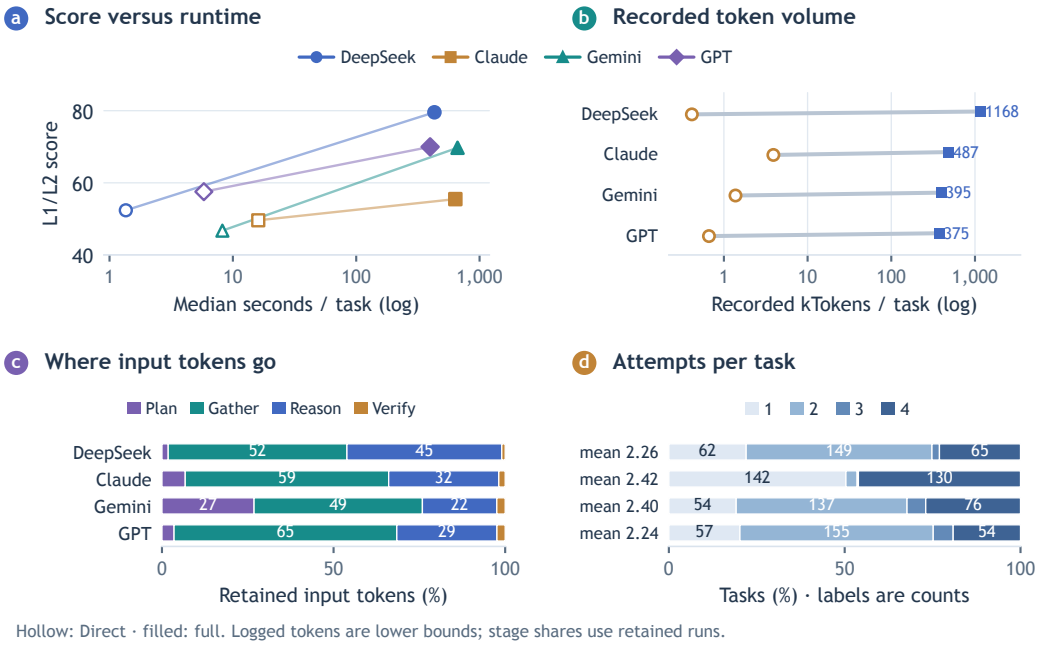


Figure 9: **Accuracy and recorded resource usage.** (a) Primary score versus median runtime; full runtime sums attempt durations and excludes queue time. (b) Recorded input plus output tokens per manifest task. (c) Input-token shares across retained stage records. (d) Attempt-count distribution over all 282 tasks. Logarithmic axes in (a,b) preserve the large scale differences. Token counters are incomplete, differ in convention between backbones, and exclude judge calls; these plots do not establish equal-cost efficiency.

final attempt. The export may have been retained from an earlier attempt, so the final record need not describe every information source used by that answer.

The 60 zero-retained-page instances have a difference of +17.2 points, with positive differences in every backbone. However, this group averages 7.62 search calls and 25.13 search hits per instance, and includes missing final-attempt evidence artifacts. It is not a no-retrieval condition. Search snippets, structured tools, earlier attempts, and model knowledge may remain relevant. A document count cannot show that retrieved information was irrelevant.

The one-to-three-page group has the largest observed difference (+31.3), compared with +18.2 for four or more pages. Retrieval volume is chosen by the system and can depend on task difficulty, evidence availability, and earlier failures. Conditioning on that volume and on successful export introduces selection. A controlled ablation would vary retrieval while holding the remaining workflow and budget fixed; evidence replay would separately test behavior with frozen evidence. Neither experiment was run.

D.6 Non-numeric L3/L4 results

Figure 4 displays the mean judge scores; Figure 10 shows how full credit, partial credit, zero-score returned answers, and non-answers contribute to those results.

Claude Code's mean score is 13.58, compared with 5.67 for OpenPredictor on the same backbone. Both arms have 25 unanswered instances, with 18 shared non-answers. Their score distributions differ: the CLI has 13 partial and 6 perfect scores, versus 1 and 4 for OpenPredictor. Perfect denotes full credit, partial denotes strictly positive credit below the maximum, and nonzero includes both. No-answer instances receive zero. One perfect answer contributes 1.25 points to the 80-instance mean. The GPT pair scores 14.33 for OpenPredictor and 13.04 for Codex; both return 17 positive-score answers, with 8 and 7 full-credit answers, respectively. The baseline audit in Appendix E applies to these comparisons.

Table 3: **General-purpose agent CLIs versus OpenPredictor.** Results on the same 282 questions with matched nominal backbones, using the metrics in Table 1. Section 4.4 describes the baseline comparison.

		Deterministic ↑		Judge ↑		Primary ↑	Numeric ↓
System	Valid (%)	L1	L2	L3	L4	L1/L2	Err. (%) [n]
Claude Opus 4.6							
Claude Code	68.09	50.00	55.51	13.85	13.33	52.75	13.80 [26]
OpenPredictor	66.31	51.35	59.61	7.69	3.74	55.48	3.45 [32]
GPT-5.6 Terra							
Codex	99.65	63.51	50.01	13.16	12.93	56.76	13.82 [49]
OpenPredictor	98.58	74.32	65.62	13.16	15.45	69.97	1.79 [49]

The subset contains 59 ranked-list questions with median reference length 3. Across all 80 judged instances, returned-answer length matches the reference in 67.5% and 91.2% of OpenPredictor's Claude and GPT answers, compared with 47.5% for Claude Code and 43.8% for Codex. Length matching is only a coarse structural diagnostic: it does not check entity identity, ordering, or the adequacy of the predicted set. These observations do not validate the judge or imply that either system has solved structured prediction.

E AGENT-BASELINE EVALUATION

E.1 COMPARISON SETTING AND INFORMATION ACCESS

Claude Code (Anthropic, n.d.) and Codex CLI (OpenAI, n.d.) use the same 282 questions, corresponding nominal backbones, and per-question cutoffs as OpenPredictor. Submitted predictions are checked against the retained execution records and evaluated with the same scoring protocol. Retrieval choices, prompts, computation, and execution environments can differ, so these are comparisons of configured workflows rather than matched-budget algorithmic effects.

Review of the Codex runs identified reference-answer access in a small subset of tasks. Its results are therefore descriptive, with an additional sensitivity analysis excluding the identified cases. Exclusion does not establish that all remaining runs are free from information leakage. No such access was detected in the available Claude Code records, although this does not prove complete isolation. Task-level audit details and the exclusion set are retained in the accompanying evaluation artifacts.

E.2 NONZERO-SCORE AGREEMENT

Figure 10 binarizes a deterministic score as strictly positive versus zero. Positive score includes partial credit and is not synonymous with an entirely correct answer. The four-way counts preserve all 153 deterministic instances. Against Codex, the discordant counts are 39 versus 15, with two-sided exact binomial $p = 0.001$; against Claude Code they are 22 versus 16, with $p = 0.418$.

Let n_{10} and n_{01} count tasks with positive score only for OpenPredictor and only for the CLI, respectively, and let $d = n_{10} + n_{01}$. The exact conditional McNemar test (Fagerland et al., 2013) assumes independent, equally likely discordance directions under the null: $X \mid d \sim \mathrm{Binomial}(d, 1/2)$, with two-sided value

$$p_{\text{exact}} = \min\left\{1,\ 2\sum_{j=0}^{\min(n_{10},n_{01})}\binom{d}{j}2^{-d}\right\}. \tag{14}$$

The value is one when $d = 0$. This exact test can be conservative (Fagerland et al., 2013); it discards score magnitude and does not replace week-cluster inference. These descriptive counts use the complete task set; the exclusion sensitivity uses a restricted set. Both remain subject to the information-access limitations described above.

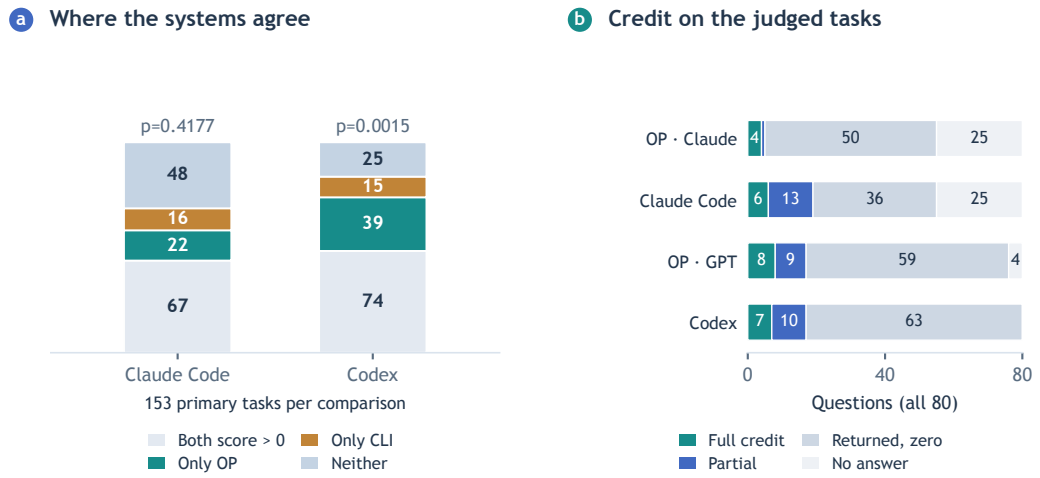


Figure 10: **Task-level score agreement and credit distributions.** (a) Counts over all 153 deterministic tasks; positive score includes partial credit. Annotations give the exact two-sided discordance p-value. (b) Full credit, partial credit, returned zero-score answers, and non-answers over all 80 judged tasks. These evaluation categories do not refer to internal verification verdicts. The information-access qualification for Codex applies to both panels.

F Implementation and Mathematical Details

F.1 Task Contracts, Evidence, and Execution State

The context/action/state separation in Equation (1) follows agent-interface and runtime abstractions (Yang et al., 2024; Wang et al., 2025); declarative modules and explicit workflows provide related structures (Khattab et al., 2024; Zhang et al., 2025). The record R_k may retain the full execution history. Failure can consume budget and add feedback without producing a prediction; no sufficient Markov state is assumed. Planning, context, and terminal acceptance concern distinct aspects of execution (Fan et al., 2026; Zhang et al., 2026).

The profile registry contains 17 entries and four generator implementations: retrieval, series, simulation, and causal. Mixed dispatch chooses between retrieval and series across trials. Each profile defines admissible targets, routes, methods, normalization, evidence requirements, and output policies. Registry coverage is an implementation property, not accuracy evidence for every application.

For a finite nonempty label set $\mathcal{L}_i$, the simplex used in Section 3.1 is

$$\Delta(\mathcal{L}_i) = \left\{ p : \mathcal{L}_i \to [0,1] \;\middle|\; \sum_{e \in \mathcal{L}_i} p(e) = 1 \right\}. \tag{15}$$

Named distributions are aligned by labels rather than array position. Numerical arrays instead require common channel and horizon coordinates and compatible units. A label sequence represents an ordered answer; a set-valued answer has different matching semantics even when its serialized elements agree. Internal value spaces and external scoring rules are therefore specified separately.

The evidence matrix records observations with source references, entity, metric, unit, observation time, data vintage, and target relation. Adequacy checks compare actual observations against declared requirements. Best-effort continuation records insufficient support; it does not convert a proxy into a direct measurement or override blocking temporal and compatibility checks.

F.2 Admissible Methods and Default Selection

The registry requires an executable implementation with compatible target, route, profile, and resource declarations. A plan resolves its baseline and candidate methods within a common semantic family. Fallbacks undergo the same compatibility checks.

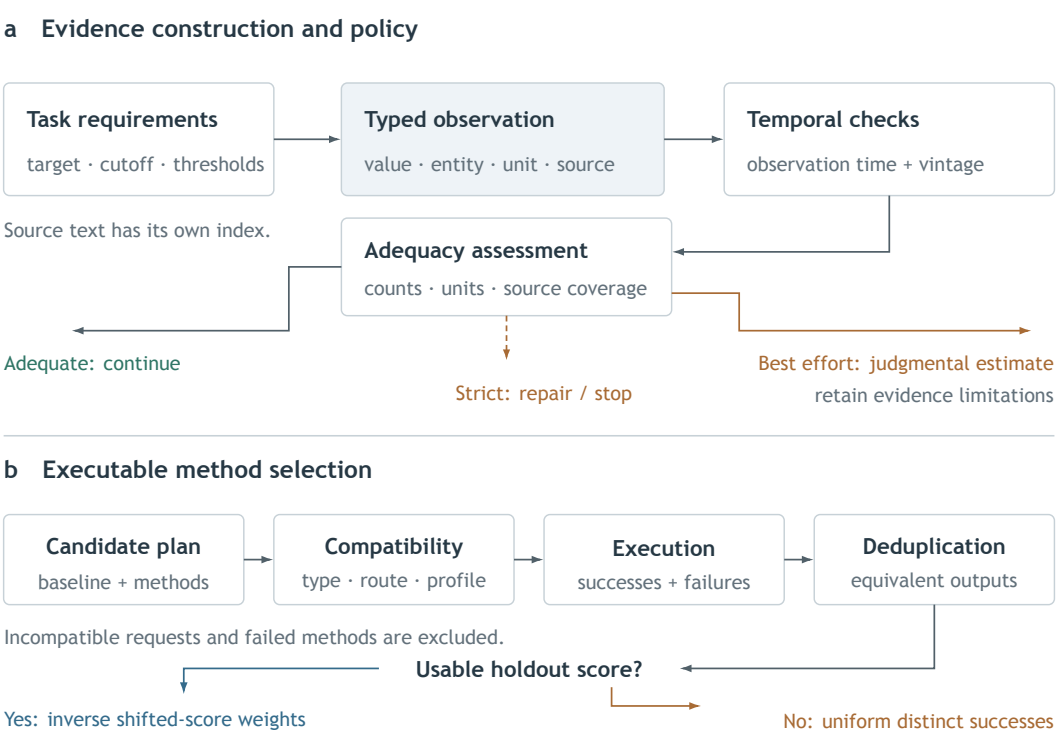


Figure 11: **Evidence validation and method selection.** The evidence path separates source text, typed observations, temporal checks, and policy-dependent adequacy. The method path enforces executable compatibility before selecting successful distinct outputs. The lower alternatives specify the default tournament; version-compatible historical weights may override them.

Holdouts use rolling-origin evaluation (Hyndman & Athanasopoulos, 2021). We require a usable score ℓ_{ij}^{val} to be finite, nonnegative MAE from at least one window matching the final horizon. The implementation uses $\lambda_s = 0.05$, $\lambda_c = 0.001$, and $\epsilon = 10^{-6}$ in Equation (3); missing stability penalties are zero. The resource proxy is a configured nonnegative value or a resource-class coefficient plus reserved memory. It is not observed price or latency. These are fixed numerical trade-off coefficients on the reported score scales, not a dimensionless objective invariant to rescaling errors or costs.

Successful outputs are deduplicated by complete serialized content, with declared preference, resource proxy, and a deterministic tie-break selecting one representative. Let $\mathcal{J}_i$ index these representatives and $\mathcal{S}_i$ the scored subset. Then $\mathcal{K}_i = \mathcal{S}_i$ if $\mathcal{S}_i \neq \varnothing$ and $\mathcal{K}_i = \mathcal{J}_i$ otherwise. In the latter case, $w_{ij} = |\mathcal{J}_i|^{-1}$ for $j \in \mathcal{J}_i$, provided this set is nonempty. If $\mathcal{J}_i = \varnothing$, no aggregate is produced. Deduplication prevents repeated representations from receiving separate weight; it does not make different method outputs independent.

For a nonempty $\mathcal{S}_i$, finite J_{ij}, and $\epsilon > 0$, every denominator in Equation (3) is positive, so $w_i \in \Delta(\mathcal{K}_i)$. The implementation includes a numerical floor below ϵ. The heuristic is invariant to adding a common constant to all objectives, but not to arbitrary rescaling with ϵ held fixed. Version-compatible historical weights can override the default. The effective coefficients α_i are those used by the chosen aggregator, which may apply uniform weights regardless of w_i.

F.3 Typed aggregation and perturbation bounds

Assume $\mathcal{K}_i \neq \varnothing$, all z_{ij} share a finite-dimensional normed vector space and coordinate semantics, and $\alpha_i \in \Delta(\mathcal{K}_i)$ is fixed. Then Equation (5) lies in $\mathrm{conv}\{z_{ij} : j \in \mathcal{K}_i\}$. For categorical inputs $p_{ij} \in \Delta(\mathcal{L}_i)$, nonnegativity and $\sum_{e \in \mathcal{L}_i} \sum_{j \in \mathcal{K}_i} \alpha_{ij} p_{ij}(e) = \sum_{j \in \mathcal{K}_i} \alpha_{ij} = 1$ preserve the simplex. Write $Z_i' = (z_{ij}')_{j \in \mathcal{K}_i}$ for perturbed inputs and $A_{\alpha_i}(Z_i) = \sum_{j \in \mathcal{K}_i} \alpha_{ij} z_{ij}$. For any norm, the triangle

inequality gives

$$\|A_{\alpha_i}(Z_i) - A_{\alpha_i}(Z_i')\| \le \sum_{j \in \mathcal{K}_i} \alpha_{ij} \|z_{ij} - z_{ij}'\| \le \max_{j \in \mathcal{K}_i} \|z_{ij} - z_{ij}'\|. \tag{16}$$

This is a non-expansiveness property of fixed-weight pooling under the maximum input norm. It does not assert continuity of method selection, weight re-estimation, or discrete decoding.

For a categorical aggregate b_i on at least two labels, let $e^\star$ be its unique maximizer and $\gamma_i = b_i(e^\star) - \max_{e \neq e^\star} b_i(e)$. For $\eta \ge 0$, if $\|b_i - b_i'\|_\infty \le \eta$, then $b_i'(e^\star) - b_i'(e) \ge \gamma_i - 2\eta$ for every competing label. Thus $\gamma_i > 2\eta$ is sufficient for unchanged argmax. The statement is conditional on a unique maximizer and common labels; it is not a calibration or end-to-end robustness guarantee.

The registered mean uses equal weights; weighted mean normalizes nonnegative supplied weights, reverting to equal weights when their sum is zero. Median is coordinate-wise. For $N_i = |\mathcal{K}_i| \ge 5$, trimmed mean removes $\zeta_i = \max\{1, \lfloor 0.1 N_i \rfloor\}$ values from each tail in each coordinate; for $N_i < 5$ it uses the mean. Coordinate-wise trimming need not correspond to one shared vector of member weights and does not inherit every convex-domain property of mean pooling. Named distributions are normalized, aligned over their label union with zero mass for absent labels, and pooled. Discrete labels use deterministic voting.

Equation 6 sums reciprocal ranks over all listed positions (Cormack et al., 2009). For a list without duplicates, this reduces to one contribution per present entity; absent entities contribute zero. Descending scores and label-based tie-breaking determine the ordering. The top-k decision returns its requested prefix. This is a registered operator, not evidence that every evaluated ranking used it or that the order is correct.

F.4 Calibration, numerical records, and replay

Calibration in Equation (4) is conditional on compatible held-out outcomes. Without such support, or when inapplicable, the effective transformation is $\widetilde{b}_i = b_i$, with the reason recorded. All 977 final-attempt receipts here lack out-of-sample observations. Internal confidence is not a substitute for empirical calibration (Guo et al., 2017; Angelopoulos & Bates, 2023).

Let $\mathcal{G}_i^{\mathrm{comp}}$ be the acyclic computation graph for target i, distinct from the subtask dependency graph $\mathcal{G}$. A node v stores value x_v, unit, ordered parents $\mathrm{pa}(v)$, operation version ω_v, parameters ϑ_v, and source location. For an internal deterministic node,

$$x_v = F_{\omega_v}\big((x_u)_{u \in \mathrm{pa}(v)}; \vartheta_v\big). \tag{17}$$

Fixing leaves, versions, parameters, seeds where used, and numerical precision determines internal values by topological evaluation. Recorded model judgments are inputs to replay, not deterministic predictions of a fresh model call. Missing inputs or unrecorded state invalidate the replay premises. Provenance describes derivation, not truth (Moreau & Missier, 2013).

Optional debate proposes a probability $p^\star$ supported by bound anchors. For an admissible pair $a_1, a_2 \in [0,1]$ with $|a_1 - a_2| > \delta_{\mathrm{tol}}$, reconstruction uses

$$\beta = \Pi_{[0,1]}\left(\frac{p^\star - a_2}{a_1 - a_2}\right), \qquad p_{\mathrm{rec}} = \beta a_1 + (1-\beta) a_2, \tag{18}$$

where $\Pi_{[0,1]}(x) = \min\{1, \max\{0, x\}\}$. Near-coincident anchors use $\beta = 1/2$. A proposal is accepted only with valid evidence bindings and $|p_{\mathrm{rec}} - p^\star| \le \delta_{\mathrm{tol}}$; otherwise the previous aggregate remains. The default tolerance is 10^{-9}. No debate-specific accuracy gain is claimed.

F.5 Repair, invalidation, and publication

Feedback identifies the target and stage requiring revision. Gathering repair checks whether evidence changed before rebuilding reasoning outputs; reasoning repair directly regenerates those outputs and their numerical conclusion. For changed subtasks U, $\mathcal{I}_{\mathcal{G}}(U)$ contains the nodes that must be refreshed. The controller recomputes affected layers in topological order and updates shared evidence and decisions. This dependency rule prevents mixing old and revised states but does not prove improved accuracy. The evaluated single-target questions do not establish multi-target gains.

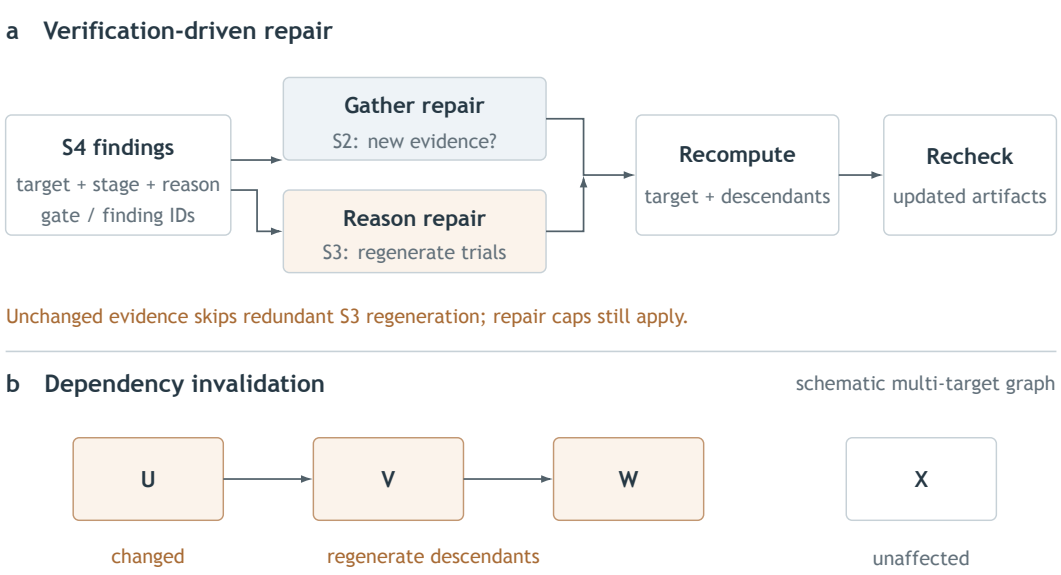


Figure 12: **Stage-specific repair and dependency refresh.** Gather repair checks for changed evidence before regenerating reasoning. Reason repair clears stale S3 outputs directly. Changed upstream decisions invalidate transitive dependents, which are recomputed in dependency order. The schematic graph illustrates implemented multi-target behavior, not an evaluated benchmark task.

Mechanical checks address finite values, compatible units, graph references, arithmetic, and root agreement; the critic assesses substantive support. The export head can use hypotheses or retain an earlier candidate, so $\widehat{y}_i = \mathsf{E}(R_K)_i$ may differ from $\widehat{y}_i^{\mathrm{int}}$. Replaying the internal graph is not equivalent to replaying the entire submission policy. Publication binds report values to recorded analysis when enabled; action/report synthesis may remain outside the strictly numerical interface.

G A PROMPT-TO-RESULT CASE STUDY

We connect the pipeline's prompt contracts to one correct L1 prediction from the retained campaign. This post-hoc example illustrates execution, not representative accuracy. The task input and results are archived; stage prompt excerpts come from the accompanying source templates, since the run does not retain serialized provider messages. All trace times below are UTC on 2026-09-20; internal verification is distinct from the benchmark score.

From the task prompt to a correct event prediction

Task prompt (archived excerpt). "Will Apple release Homepod Mini Successor by June 30?"

Required ending. `\boxed{Yes}` or `\boxed{No}`.

DeepSeek V4 Flash; cutoff 2026-06-29; L1.

Submission	Reference	Deterministic score
No	**No**	**100 / 100**

Prompt contracts and recorded execution

Pipeline prompt excerpt	Recorded execution
S1: Plan "Select only method ids listed in the routing contract."	Binary target; a historical base-rate baseline and a judgmental estimator; horizon-matched holdout planned. 08:49:34 UTC

From the task prompt to a correct event prediction (continued)

S2: Gather "Never use an observation or issue vintage after the supplied as_of."	9 evidence-index items and 12 typed observations; the retained matrix marks evidence adequate. This is the run's assessment. 08:53:05 UTC
S3: Reason "establish the outside-view base rate / reference class first and state it;" … "run a pre-mortem and widen if fragile;"	3 reasoning trials produce 12 hypotheses. The base-rate and judgmental estimates are 0.020000 and 0.024278, respectively (rounded). 08:54:12 UTC
S4: Verify "Preserve any deterministic blocking failures (never upgrade them to pass)."	Mechanical gates pass; the critic records `pass` and no blockers. An unresolved-conflict warning is retained. 08:54:24 UTC

Recorded numerical conclusion.
Let p_{base} and p_{judg} denote the two method estimates.

$$\widehat{p}^{\,\text{int}}_{\text{Yes}} = \frac{p_{\text{base}} + p_{\text{judg}}}{2} \approx 0.022139.$$

No usable holdout score is returned; both method weights are 0.5. The two-member trimmed-mean operator reduces to a mean; the unavailable-calibration receipt preserves this value ($n_{\text{OOS}} = 0$). The submission head returns `No`.

Provenance and scope. Task input, values, and timestamps are archived; prompt excerpts quote implementation templates rather than recorded API messages. The full trace and provenance are retained in the accompanying artifacts.

H Artifact checks and reproduction

H.1 Offline mechanism checks

A selected set of 41 existing regression tests passed without prediction or judge API calls. They cover task contracts, stage transitions, quantitative operations, output construction, and temporal filtering. Test records are retained with the implementation. These checks establish expected behavior on controlled inputs, not prediction accuracy, natural failure rates, or independent leakage estimates.

The mathematical checks separately exercise pooling stability, simplex preservation, selection support, horizon matching, rank fusion, and anchor-based reconstruction. Synthetic cases also test the coverage–quality identity and the resampling and multiplicity procedures. Formula and denominator checks reuse the archived scores. These checks establish implementation correspondence without adding forecast observations to any result table.

H.2 Analysis lineage and available materials

The analysis follows the keyed lineage

$$\text{corpus} \longrightarrow \text{submissions and execution records} \longrightarrow \text{scores} \longrightarrow \text{tables and figures.}$$

The archived materials include predictions, evidence, method outputs, decisions, and execution histories. Analysis checks link each reported result to its task and retained output, verify the scoring denominators, and reproduce aggregate statistics without additional model or judge calls.

The source package includes active LaTeX, the unmodified conference style, standalone PDF figures, editable vector sources, analysis scripts, validated aggregates, the selected case study, and citation-verification records. Building the paper needs no experimental API key. Recomputing the aggregate from raw runs requires the separately retained engine and campaign artifacts; the source package is not a complete raw-run archive.

The source package is compiled independently, with checks for consistency among the manuscript, tables, figures, and references. Detailed file organization and execution instructions are provided with the artifacts.

Archived responses support analysis replay, while fresh calls to hosted models can differ. Releasing the raw runs additionally requires removing identifying metadata without changing scored predictions or their provenance.

I Limitations and Research Directions

Scope of generality. The software accommodates several target and method families, but this evaluation covers six output forms from one benchmark and predominantly retrieval-based prediction. It does not establish performance on causal effects, simulation, long-horizon time series, or composite multi-target requests. The set and ranking results show that a valid output format is only one part of predictive competence. Dedicated benchmarks are needed before claiming gains for the additional routes.

Information and identification. Models are evaluated after the events resolved. Date filtering, label-informed page suppression, and input sanitization constrain some external information paths, but model memory, undated pages, revised observations, and undeclared local access remain risks. The baseline evaluation also highlights the need to isolate reference answers from prediction-time inputs. No independent leakage-rate estimate is available. Prospective submission with a recorded timestamp, and isolated historical evidence, would provide stronger evidence. Contamination audits must be benchmark-specific (Sainz et al., 2023), and forecast calibration requires outcomes beyond internal confidence labels (Guo et al., 2017).

What the comparisons identify. The full-versus-direct contrasts combine retrieval, inference, retry policy, and numerical processing. Quota failures make provider availability a substantial confound for Claude. A matched evidence and compute study is needed to isolate the contribution of task contracts, method selection, or verification. The common-answer intersections, execution-order subsets, and alternate bootstrap are diagnostics rather than substitutes for these experiments. Judge biases (Zheng et al., 2023), four sampled weeks, and one realized campaign per condition further limit statistical generalization.

Reliability beyond replay. A reproducible computation can still start from an incorrect or unsupported candidate. Mechanical checks cannot certify source truth, causal assumptions, or probability calibration; passing implementation tests does not estimate their natural error rates. Independent-run answer stability is also unmeasured: of the 211 tasks whose retry ledger retains two or more non-empty predictions, only 19 repeat the same answer. Retries follow failures and can change evidence or feedback, so this selected comparison provides neither an independent-run estimate nor a one-sided bound. Future work should pair prospective evaluation with blinded audits of accepted and rejected evidence and expand the method-specific benchmarks while preserving the same recorded execution contract.