BENCHMARKING FORECASTING AGENTS ON YESTERDAY'S TOMORROWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Digital agents can now gather and organize information, reason over it, and use tools to complete complex tasks. Forecasting puts all of these abilities to the test: it requires integrating incomplete evidence, inferring causal drivers, and committing to calibrated probabilistic judgments. Existing benchmarks are poorly suited to measuring and improving this ability. Live benchmarks provide feedback only after events resolve, so every evaluation cycle waits on the calendar; fixed-corpus evaluations return results immediately but confine research to pre-collected documents. We introduce FABENCH, a scalable backtesting environment for **F**orecasting **A**gents. An automated pipeline discovers events, generates questions, and resolves outcomes for any past forecast date, so the benchmark can be renewed to stay ahead of model training cutoffs; its first release poses 287 questions across nine domains. Agents research each question freely in an archive environment that combines an offline archive with live web search screened to prevent temporal leakage. Evaluating agents built on eight recent models, we find that retrieval improves the Brier score of seven of them in every retrieval setting and lifts the leading agents to the market consensus on the 54 questions linked to Polymarket, with the best three scoring slightly above it. Research traces reveal recurring failures in how agents weigh and apply evidence, leaving ample room for improvement. We release our code, data, and archive environment as a foundation for community research on forecasting agents.

1 INTRODUCTION

Large language models have advanced at a remarkable pace, and the agents built on them now carry out a growing share of everyday knowledge work: writing and debugging code (Jimenez et al., 2024), conducting deep research across many sources (Wei et al., 2025), and operating computers through their interfaces (Xie et al., 2024). On some of these tasks, agents now approach the level of human experts (Patwardhan et al., 2026). A gap remains, however, on complex, long-horizon tasks (Kwa et al., 2025) that require an agent to gather large amounts of information, reason about how a situation will unfold, and commit to a decision. Forecasting is the sharpest instance of this challenge. To predict a future event, an agent must integrate incomplete and often conflicting evidence, infer the causal drivers behind it, and compress that understanding into a calibrated probability. It is also a demanding test of decision-making under uncertainty (Tetlock & Gardner, 2015; Mellers et al., 2015): whether an agent can forecast future events as well as human experts, which LLM forecasters have begun to approach (Halawi et al., 2024; Schoenegger et al., 2024; Gu et al., 2026).

Progress toward this goal is hard to measure, because existing forecasting benchmarks fall into two camps with complementary weaknesses. Live benchmarks pose questions about events that have not yet occurred, typically drawn from prediction markets or forecasting platforms (Karger et al., 2025; Zeng et al., 2026; Yang et al., 2026). Since no information about the outcome exists anywhere, they cannot leak outcomes; but feedback arrives only when the event resolves, often weeks or months after the forecast is made (Paleka et al., 2026). Every evaluation cycle is gated by the calendar, the question set is dictated by whatever happens to be pending, and scores from different periods cannot be compared because the questions differ (Kucinskas et al., 2025). Fixed-corpus benchmarks take the opposite approach: they select events that have already resolved, assemble a corpus of documents published before each question's forecast date, and ask the agent to forecast from that corpus alone (Zou et al., 2022; Wildman et al., 2025; Liptay et al., 2026). Feedback is immediate

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.

and evaluation is reproducible. A main problem, however, is that the agent can research only what the benchmark's creators chose to collect, so evidence outside that corpus stays out of reach however relevant it is. And because the question set is fixed, such benchmarks are static: they saturate (Kiela et al., 2021), and their outcomes eventually enter model training data (White et al., 2025; Paleka et al., 2026).

We introduce FABENCH, a scalable backtesting environment for forecasting agents that removes this trade-off. Our approach treats forecasting as backtesting: questions are posed at historical time points whose outcomes are now known, so feedback is immediate and verifiable, while the agent searches an archive and the live web, both limited to what was published before that time point, so the research itself remains open-ended. FABENCH has two components (Figure 1): a workflow that generates questions, and an archive environment in which agents research them.

FABENCH currently contains 287 forecasting questions across nine domains: finance, economy, technology, health, sports, culture, geopolitics, weather and politics. Rather than curating them by hand, we build a workflow that discovers candidate events from recurring data releases, news and prediction markets, writes well-specified questions with unambiguous resolution criteria, and resolves each question against its recorded outcome. The workflow takes the forecast date as input, so it can be re-run for new dates and extended to new domains, and the benchmark grows on demand and can be refreshed to stay ahead of model training cutoffs. All current questions are posed at July 1, 2026, after the pretraining cutoffs of all the models we evaluate (Section E).

We use FABENCH to evaluate agents built on eight frontier models. Retrieval raises the Brier Index of every model, by 3.4 points on average, and brings the leading agents on par with the market consensus, a strong baseline (Wolfers & Zitzewitz, 2004; Alur et al., 2025), which every model trails without retrieval. Yet on about a quarter of the questions, agents still do worse on average than a constant 0.5 forecast, with or without retrieval. Because the environment records every step of the agent's research alongside time-stamped evidence, we can trace these failures to their sources. We identify three error modes: favoring unofficial reports over newer official statements, following a public consensus that proves wrong, and applying correct evidence to the wrong case.

2 FABENCH

2.1 FORECASTING TASK AND EVALUATION

We are interested in forecasting consequential real-world events. To score forecasts without waiting, we pose each question at a past forecast date t_0 about an event that has since resolved. Formally, a question is a yes-or-no statement q with a resolution rule r, which names the source that determines the outcome $y \in \{0, 1\}$. For example, a question posed on July 1, 2026, asks whether the first estimate of UK GDP growth for 2026 Q2 will exceed the 0.6% first published for Q1; it resolves from the Office for National Statistics release of August 13.

Given a question, a forecasting agent returns the probability $p \in [0, 1]$ that $y = 1$. To forecast as someone at t_0 would, the agent may use only $\mathcal{I}(t_0)$, the information published before t_0. The archive environment of Section 3 provides this information. What counts is when information is published, not when events occur: in the example, the quarter has ended by t_0, but its GDP estimate is not yet public. Besides retrieval, a model may carry later information from training, which no environment can remove; t_0 must therefore follow the training cutoffs of the evaluated models.

We score each forecast p against the outcome y with the Brier score (Brier, 1950),

$$\mathrm{BS} = \frac{1}{N} \sum_{i=1}^{N} (p_i - y_i)^2, \tag{1}$$

where N is the number of questions with a valid forecast. It measures accuracy and is strictly proper: a forecaster minimizes its expected score only by reporting its actual belief (Gneiting & Raftery, 2007). For readability, we report it as the Brier Index (Kucinskas et al., 2026),

$$\mathrm{BI} = 100 \left(1 - \sqrt{\mathrm{BS}}\right), \tag{2}$$

on which a constant forecast of 0.5 scores 50 and a perfect forecaster scores 100.

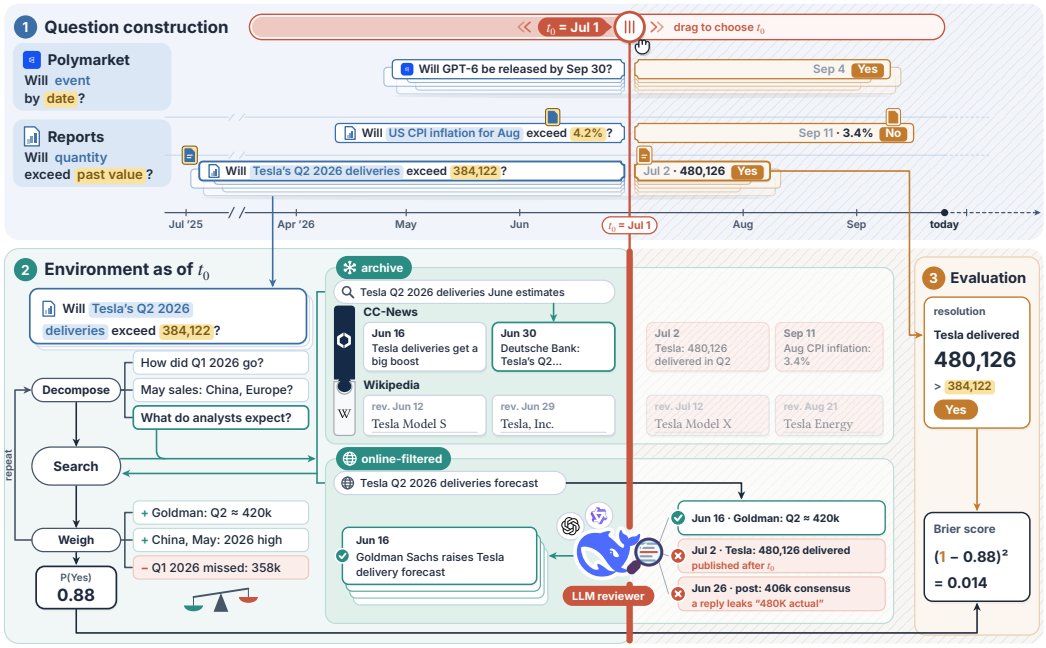


Figure 1: Overview of FABENCH. The upper part generates questions, and the lower part is the archive environment in which an agent researches a question and gives a probability, scored against the outcome at the bottom right. The red line marks the forecast date t_0: content to its left was published before t_0, and content to its right after t_0.

A good Brier score, however, does not show whether the probabilities can be taken at face value. A forecaster that separates Yes from No events well can score well while being overconfident, yet a decision that weighs a probability against costs needs the probability itself to be right. To measure this calibration, we also report the expected calibration error (Guo et al., 2017),

$$\mathrm{ECE} = \sum_{b=1}^{10} \frac{n_b}{N} \left| \bar{p}_b - \bar{y}_b \right|, \tag{3}$$

where ten equal-width probability bins group the forecasts, and bin b holds n_b of them with mean $\bar{p}_b$ and Yes rate $\bar{y}_b$; lower is better.

2.2 Question Generation

To evaluate agents on events that matter, and to renew the questions as models advance, we generate questions with an automatic workflow that takes t_0 as input (panel 1 of Figure 1). Renewal is necessary because a set expires once models are trained past its t_0. The workflow draws events from two complementary sources and, in each, keeps those that many people follow.

Recurring releases. The first source covers releases such as the earnings of S&P 100 companies, the decisions of major central banks and the official statistics of major economies. They meet both needs: many people follow them, and their published schedules and fixed definitions allow questions to be posed at any t_0. For a given t_0, an LLM agent selects releases due after t_0 from calendars published before it. A template then turns each release into a question, with a threshold computed by a fixed rule from values published before t_0, as in the GDP example; the fixed rule keeps the threshold independent of the outcome.

Emerging events. Many consequential events, however, happen only once, such as elections, drug approvals and product launches, and they form the second source. To find them, the agent searches news and event calendars published before t_0 for pending events. Prediction markets also show

which events people follow, so we draw on Polymarket as well. We search its markets on consequential topics and turn those open at t_0 into questions. Each such question keeps the market's wording and resolution rule, as in "Will GPT-6 be released by September 30, 2026?" The market consensus at t_0, its price just before t_0, is then a forecast of the same question, which gives a human baseline (Section 4.2).

Validation and resolution. To ensure validity, a second agent checks that each question the agents write resolves unambiguously, was still open at t_0 and reveals nothing published after t_0. Once the resolution date passes, each outcome is recorded from the named source with its evidence and kept out of the agent's input. Questions whose outcome cannot be established are dropped.

First release. Our first release poses 287 questions at t_0 = July 1, 2026, 254 written by agents and 33 taken from Polymarket, in nine domains: finance, economy, politics, technology, health, sports, culture, geopolitics and weather (Tables 3 and 4). Building it also involved manual steps, and some questions were written with outcomes visible; Section B lists both. The workflow is not tied to this date: an earlier run at March 1, 2026, produced 232 agent-written and 300 Polymarket questions (Section C).

2.3 Comparison with Existing Benchmarks

Table 1: Comparison with representative forecasting benchmarks. ✓ yes; ◐ partly; ✗ no.

	Benchmark	Instant scoring	Outcome-blind	Open research	Answer-blind	Calibration
Live	ForecastBench (Karger et al., 2025)	✗	✓	◐	✓	◐
	Prophet Arena (Yang et al., 2026)	✗	✓	◐	✓	✓
	FutureX (Zeng et al., 2026)	✗	✓	✓	✓	✗
Backtest	Autocast (Zou et al., 2022)	✓	✓	✗	◐	◐
	Daily Oracle (Dai et al., 2025)	✓	✗	✗	◐	✗
	BTF-2 (Liptay et al., 2026)	✓	✓	◐	◐	✓
	ATM (Chai et al., 2026)	✓	✓	✓	✗	✗
	FABench	✓	✓	✓	✓	✓

Table 1 compares FABench with representative benchmarks; Section B notes where our first release departs from the properties marked there, and Section A reviews further work.

Live benchmarks, such as ForecastBench, Prophet Arena and FutureX, cannot leak outcomes, but every score waits for events to resolve, and systems evaluated at different times answer different questions. Other backtests score at once, but they either choose events after the fact, as Daily Oracle does by generating questions from articles that report the outcome, or confine forecasting to a pre-collected corpus, as Autocast does.

BTF-2 is closest to ours in how it selects and scores questions. Its corpus, however, is crawled for each question while the question is still open, so a set for a new period must be planned and collected before that period, and, as its authors note, models trained later need such a new set. FABench instead builds a set after the fact for any past t_0 that its news archive covers, because the news archive is collected independently of the questions and web pages are screened against t_0. A new model can therefore be evaluated once events after its training cutoff have resolved, without a collection planned in advance.

Agentic Time Machine (ATM) also lets agents research the live web, but its filter reads each question and its known answer, so what the agent sees can depend on the outcome it must forecast. FABench screens each page by its publication time alone, so the evidence cannot be tilted toward or away from the outcome (Section 3.2). ATM also reuses existing questions and scores point predictions, whereas FABench generates questions for a chosen t_0 and scores probabilities, which also reveals calibration.

3 ARCHIVE ENVIRONMENT

To let the agent research a question as a forecaster at t_0 could, using only information published before t_0 (Section 2.1), we build an archive environment (panel 2 of Figure 1). In this environment, the agent decides what to search for and which documents to read, and the environment decides what its search and read tools return. These tools should deliver as much relevant information as possible while leaking nothing published after t_0. An offline archive of pages captured before t_0 meets the second requirement by construction but holds only what was collected. Live web search reaches far more sources, but its pages must be screened one by one. We therefore combine the two, so that one search draws on the archive's dated news and reference text and on pages that only the web can reach.

For a request of N results, the archive supplies its top $\lceil N/2 \rceil$ documents, and the web supplies the pages that pass screening among its top $\lfloor N/2 \rfloor$ candidates; withheld pages are not replaced. The fixed shares keep archive documents, which cannot leak, in every result list, so the agent never depends on screened web pages alone. A URL found in both keeps only its archive copy, whose capture date needs no review. Either source can also be searched alone, which Section 4.3 uses to isolate what each contributes.

3.1 OFFLINE ARCHIVE

Contents. To forecast well from an archive, a forecaster needs two kinds of information: background on the entities and topics of a question, and the latest developments before t_0. For background, we use Wikipedia. Because it stores every revision with its time, we take the last revision before t_0 of the articles on the entities and topics in the questions (Section D). For recent developments, we use CC-News, Common Crawl's continuous crawl of news sites (Nagel, 2016), and keep the English pages captured before t_0. A page captured before t_0 was public by then, so the archive needs no page-by-page review and cannot leak through a reviewer's mistakes. For t_0 = July 1, 2026, it holds about 20.1 million news captures and 684 Wikipedia revisions.

Retrieval. When the agent searches the archive, BM25 retrieves candidates from both parts, and a rule-based scorer removes duplicates and reranks them. All term statistics come from documents before t_0, so no signal from after t_0 shapes the ranking. A search result shows the beginning of each document, and the read tool returns its full text (Section D).

Comparison with BTF-2. BTF-2 also gives agents an offline corpus, but it searches with RetroSearch, which orders results by live Google search before dropping pages outside the corpus (Liptay et al., 2026; Wildman et al., 2025). A live ranking reflects what is relevant today, so pages that became prominent because of later events can rise to the top; its developers acknowledge this as a possible leak. The live ranking also changes over time, so the same query can return different results in different runs. Our ranking is computed from the archive alone, so it carries no information from after t_0 and returns the same results for the same query in every run.

3.2 SCREENED WEB SEARCH

Search and cleaning. The archive, however, misses sites and pages that CC-News did not capture. To reach them, the search tool also asks a live search engine for results dated before t_0. This date limit only narrows the candidates, because an old URL may now show updated content. We therefore fetch each web page before the agent sees it and judge it by the text it shows now. To keep market prices out of the agent's view, we reject pages on known prediction-market sites. From the other pages, we strip navigation and advertisements, and we replace Wikipedia pages with their last revision before t_0, which needs no review.

Temporal review. Every other page goes to an LLM reviewer that sees neither the question nor its answer, so that its decision depends on timing alone (Section D). The reviewer first removes leftover blocks, such as recommendation lists and live widgets, whose newer headlines would otherwise count against an old article. It then judges whether the remaining text was public before t_0. What counts is publication, not when events occur, so a forecast or plan published before t_0 passes. When timing is unclear, the reviewer can search the web to check, and what it finds never reaches the agent.

Delivery. Based on this judgment, a page is withheld entirely if it contains information from after t_0 or its timing stays unclear. Withholding the whole page, rather than cutting out the later passages, avoids passing text whose remaining context could still reveal later events. For a page that passes, the read tool returns the cleaned text, and a search result shows an excerpt of it. The search engine's own titles, snippets and dates never reach the agent, because they are generated today and can reflect later events. Every page is logged with its decision, so that decisions can be audited; Section D gives the reviewer's instructions. In the `online-filtered` runs, the reviewer withheld 16.0% of 302,561 pages. Reviewing 3,087 pages at both our low reasoning effort and high effort gave the same decision for 95.7% of them; only 14 (0.5%) passed at low effort yet were judged at high effort to disclose later information. Unlike the archive, screened web search cannot be repeated exactly: a live search engine may no longer return some pages that were available at t_0, and its results change over time, so runs made at different times can see different pages; the logs make each run auditable, not repeatable.

Comparison with Agentic Time Machine. Judging timing without the answer distinguishes our reviewer from the filter of Agentic Time Machine, which sees the answer. Such a filter can withhold a legitimate forecast for matching the outcome, as it did with an award-nomination forecast published two days before the announcement (Chai et al., 2026). A forecaster at t_0 would have weighed exactly this kind of evidence, so removing it tilts the agent's evidence away from the correct answer. Our reviewer passes such forecasts. It also withholds pages of unclear timing, which that filter passes; in one of its cases, an undated post quoting a price from another day passed and misled the agent. The cost of our choice is that some legitimate pages of unclear timing are lost.

4 Experiments

4.1 Experimental setup

Environment and data. We evaluate every model in four information settings: `none`, without tools; `archive` and `online-filtered`, with only the archive or only screened web search; and `full`, our standard setting, with both (Section 3). All 287 questions come from the release of Section 2.2, posed at t_0 = July 1, 2026; under `full`, every model forecasts all of them.

Compared models. We compare **(1)** three proprietary models: GPT-6 Astra, GPT-6 Sol and Claude Opus 5.5; **(2)** five open-weight models: DeepSeek V4.1 Flash (DS V4.1 Flash), Kimi K3, GLM 5.3, Qwen3.8-27B and Qwen3.6-35B-A3B; and **(3)** JEV 1.13.0 (Almeida, 2026), a model trained to return calibrated probabilities, which answers without retrieval. As references, we use the constant forecast of 0.5 and the market consensus on Polymarket (Section 4.2). Most models were released after t_0, but without retrieval the best of them scores 55.0 BI, only 5 above the constant forecast, so none appears to know the outcomes from training (Table 8).

Implementation and metrics. Every model receives the same prompt, searches the same sources and makes at most 100 tool calls per question, and forecasts each question once per setting; ChatGPT models run in the Codex CLI and Claude in Claude Code with their built-in search replaced by our tools, and the other models in our agent loop (Sections D and E). We measure accuracy with the BI and calibration with the ECE (Section 2.1).

4.2 Comparison among models and with prediction markets

Comparison among models. To compare the models in the standard setting, Figure 2 shows their BI and ECE under `full`. Five models, three proprietary and two open-weight, form a leading group at 57 to 58 BI, while the weakest model and JEV, which does not retrieve, score near the constant forecast. Even the best model removes only about 31% of the squared error of the constant forecast.

The ranking depends less on what a model knows than on how much it researches. Accuracy without tools is unrelated to the gain from retrieval (Spearman $\rho = 0.14$ across the eight models), whereas the number of pages a model reads per question tracks it closely ($\rho = 0.79$; Tables 2 and 9). Kimi K3 and DeepSeek V4.1 Flash read the most pages and gain the most, which lifts them into the leading

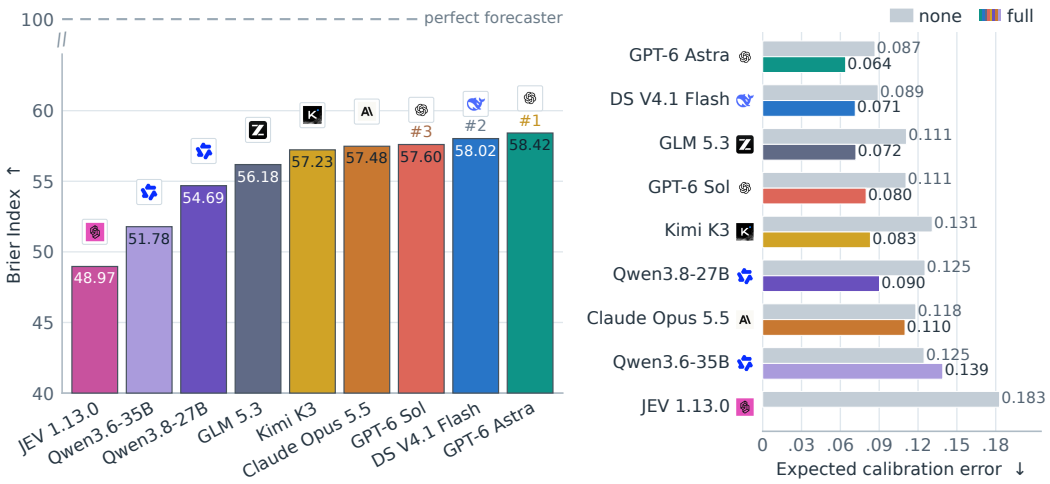


Figure 2: BI under `full` (left) and ECE under `none` and `full` (right) of each model. Lower ECE is better; JEV 1.13.0 does not retrieve, so it appears with its `none` score.

group, whereas Claude Opus 5.5, the most accurate model without tools, reads less than one page per question and gains only 2.1 BI.

Calibration does not follow accuracy: Claude Opus 5.5 scores within 1 BI of GPT-6 Astra, yet its ECE is 0.110 against 0.064. Retrieval lowers ECE for 7 of the 8 models, but not for Qwen3.6-35B, whose ECE rises from 0.125 to 0.139 although its BI improves.

Comparison with prediction markets. To compare the models with human forecasters, we use the 54 questions linked to Polymarket: 33 taken from Polymarket and 21 generated questions that match a contract. The market consensus, the market's price about one minute before t_0, is a forecast made at the same cutoff as the models' and a strong baseline: prior work has found it hard to beat (Karger et al., 2025; Alur et al., 2025; Mostafa et al., 2026), turning forecasts into trading gains is harder still (Wang et al., 2026), and roughly 69% of Polymarket users lose money (Akey et al., 2026), which suggests that forecasts at the market's level may already outperform most participants.

Figure 3 compares each model with the market in the four settings. Under `full`, the leading agents perform at about the market's level: six models lie between 1.8 BI below and 2.7 BI above the market. Without retrieval, every model falls below the market, by 1.8 to 10.8 BI. The market price aggregates what traders learned before t_0, and retrieval lets an agent gather the same public information itself, which plausibly explains why the gap closes only with retrieval. Prior work has instead found LLM forecasters trailing liquid markets (Alur et al., 2025) or beating a market only shortly after it opens (Mostafa et al., 2026).

4.3 Comparison of retrieval settings

To test whether each source adds useful information, and whether combining them adds more, Table 2 compares the four settings for each model. Retrieval improves nearly every model in every setting: averaged over models, `archive`, `online-filtered` and `full` add 2.8, 3.7 and 3.4 BI. The archive alone, which cannot leak by construction, already yields about four fifths of the gain of `full`, so most of the improvement reflects information beyond what the models learned in training rather than leaked outcomes.

The three retrieval settings, however, lie close together. `full` is within 1.1 BI of `online-filtered` for every model, and the three settings differ by at most 1.5 BI. This pattern suggests that screened web search adds little information from after t_0 beyond the archive, which predates t_0 by construction. It also suggests that, on average, neither source lacks much of what the other provides, although the archive falls short in three domains (Section 4.4).

Table 2: BI of each model in the four information settings. `none`, `archive` and `online-filtered` are scored on the 233 questions that every model answered in all four settings and `full` on all 287; gains over `none` are in parentheses and use the 233 questions; in each column, bold marks the highest value and underlining the second highest, separately for scores and gains.

Model	`none`	`archive`	`online-filtered`	`full`
GPT-6 Astra	<u>54.52</u>	**57.84** (+3.32)	**58.91** (+4.39)	**58.42** (+3.88)
DeepSeek V4.1 Flash	53.88	<u>57.80</u> (<u>+3.92</u>)	<u>58.69</u> (+4.82)	<u>58.02</u> (<u>+4.59</u>)
GPT-6 Sol	53.16	56.61 (+3.45)	58.10 (<u>+4.95</u>)	57.60 (+4.29)
Claude Opus 5.5	**54.96**	56.17 (+1.21)	57.05 (+2.09)	57.48 (+2.11)
Kimi K3	51.74	56.78 (**+5.04**)	57.68 (**+5.94**)	57.23 (**+5.73**)
GLM 5.3	53.34	55.93 (+2.60)	56.74 (+3.41)	56.18 (+3.42)
Qwen3.8-27B	52.67	55.24 (+2.57)	55.70 (+3.03)	54.69 (+2.00)
Qwen3.6-35B-A3B	50.96	50.84 (−0.12)	51.72 (+0.76)	51.78 (+1.13)

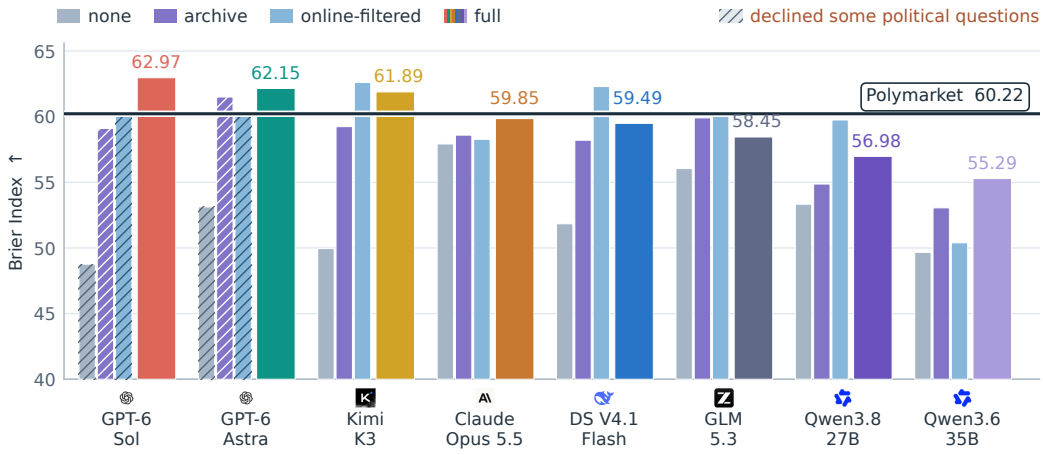


Figure 3: BI of each model on the 54 questions linked to Polymarket. The line marks the market consensus at t_0, numbers give the BI under `full`, and hatched bars omit declined political questions.

4.4 Performance across domains

Forecasting performance differs widely across domains: under `full`, the BI averaged over models ranges from 49.1 on Culture and 49.8 on Weather to 63.2 on Health. Retrieval, too, helps unevenly, and Figure 4 breaks its gains down by domain under `archive`, which searches the archive alone, and under `full`, which adds screened web search. Overall, retrieval raises the BI in most model–domain pairs: 48 of 72 under `archive` and 59 of 72 under `full`. Adding web search matters most in Technology, Health and Culture, where the gain averaged over models rises from about 1.7 to 3.5 BI under `archive` to about 4.0 to 6.3 BI under `full`. Finance and Economy gain about 2.7 to 4.5 BI in either setting.

One explanation is that questions in these three domains often ask whether an announced event, such as a model release, a drug approval or a film opening, happens as planned. The decisive sources are then company, regulator and industry pages, which web search reaches but a news archive may not contain (Section 3.2). News, by contrast, closely covers the scheduled releases behind finance and economy questions.

Retrieval can also lower accuracy. In both settings, every model loses BI in at least one domain, by up to 5.6 BI under `archive` (Qwen3.6-35B on Geopolitics) and 4.8 BI under `full` (Kimi K3 on Geopolitics). Geopolitics and Politics gain little or lose on average, plausibly because their coverage before t_0 consists largely of commentary and competing predictions about open developments. Forecasting performance should therefore be reported together with the domains and the setting in which it was measured.

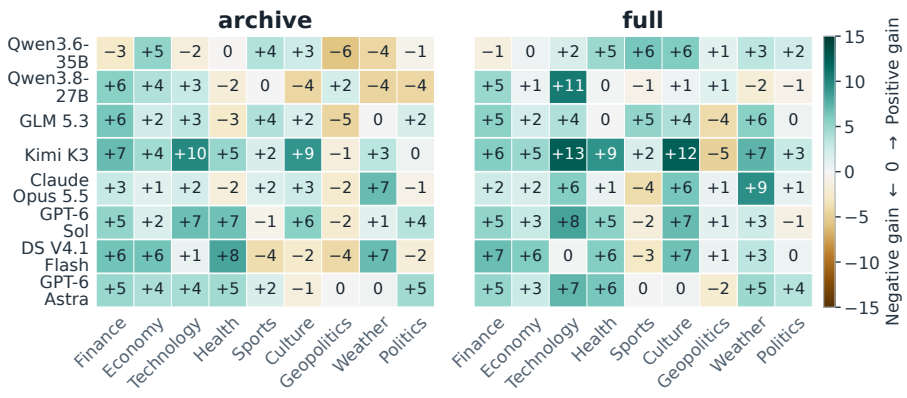


Figure 4: Gain in BI from retrieval, by model and domain. Gain over `none` under `archive` and `full`, rounded to whole BI; domains contain 13 to 77 questions.

4.5 Failure analysis

To find failures that aggregate scores hide, we classify each question by whether its Brier score, averaged over models, exceeds 0.25, the score of the constant forecast, without retrieval and with `full`. Retrieval corrects 42 questions and misleads 11, while 76, about a quarter of the benchmark, stay above 0.25 in both settings; on 28 of them, no model beats the constant forecast. Traces of representative questions show why (Section F).

Retrieval helps when decisive evidence is already public. In the traced corrections, the decisive evidence, such as a central bank's split vote and guidance, was published before t_0 and replaced the trend that models extrapolate without tools.

Retrieval misleads when informal signals meet official guidance. Several models let a concrete but unofficial report, such as a launch window relayed on social media, outweigh a newer official statement, which suggests that they weigh how specific a signal is more than who issued it and when. Adding the archive reduces this error in the traced case; most misled questions, 7 of 11, concern finance.

Shared errors come from the public record itself. When the consensus at t_0 is wrong, as with box-office forecasts that all overestimated a film's opening, retrieval returns that consensus and cannot correct it; models treat the few dissenting forecasts as tail risk. In the traced cases, models also match evidence to a question by topic rather than by the exact event, so an authoritative fact about one case, such as the production timeline of one coin, is applied to another.

5 Conclusion

We introduced FABench, a scalable backtesting environment for forecasting agents that combines automated question generation with open-ended research under historical information constraints. By pairing resolved events with an offline archive and temporally screened web search, it provides immediate, verifiable feedback without restricting agents to a pre-collected corpus. Our evaluation of latest models on 287 questions across nine domains shows that retrieval generally improves forecasting accuracy and calibration, but its benefits vary across models and domains. Several agents score on par with the market consensus on the questions linked to Polymarket contracts, while our case studies reveal failures in how retrieved evidence is weighed and applied. These findings motivate improving not only information access, but also the reasoning that turns evidence into probabilistic judgments. With renewable questions, open-ended research, and immediate outcome feedback, FABench provides a foundation for faster evaluation and future outcome-based training of forecasting agents.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing the manuscript and creating figures. LLMs were also used for event discovery, benchmark question construction, selection, and validation, as well as cleaning and temporal screening of retrieved web content. The authors reviewed and revised the AI-assisted text and figures for accuracy and consistency with the methods and results. The authors take responsibility for the final content of this work, including all AI-assisted text, claims, and research artifacts.

ETHICS STATEMENT

This work develops a benchmark for evaluating forecasting agents using questions and evidence drawn from publicly available sources. The selection of events and reliance on English-language news may introduce coverage biases. Forecasts can be inaccurate or miscalibrated, and performance on this benchmark should not be interpreted as evidence that a system is suitable for high-stakes deployment. To support research integrity, we document temporal screening procedures and trace audits, including their limitations. Responsible use and redistribution of the underlying sources require attention to attribution, licensing, and privacy.

REPRODUCIBILITY STATEMENT

We will release every question set of FABENCH on Hugging Face, starting with the first release and the set posed at March 1, 2026, together with their resolution keys and the materials needed to rebuild the retrieval archive: the manifests of the CC-News captures and the identifiers of the Wikipedia revisions. The code is released with the data: a few commands build the archive environment and evaluate an agent in any of the four information settings, with the prompts, tool definitions and reviewer instructions of Sections D and E. New question sets built with the same pipeline will feed a live leaderboard, so that agents can be compared on questions that postdate their training. The full research traces of the reported runs, over 100 GB, will be shared with teams that request them.

REFERENCES

Pat Akey, Vincent Grégoire, Nicolas Harvie, and Charles Martineau. Who Wins and Who Loses in Prediction Markets? Evidence from Polymarket. CEPR Discussion Paper DP21615, Centre for Economic Policy Research, 2026. URL https://cepr.org/publications/dp21615.

Diogo Almeida. Introducing system one models & Jev. TypeSafe AI blog, 2026. URL https://typesafe.ai/blog/introducing-system-one-models-and-jev.

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. arXiv:2511.07678.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

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. arXiv: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. arXiv:2512.25070.

Hui Dai, Ryan Teehan, and Mengye Ren. Are LLMs prescient? A continuous evaluation using daily news as the oracle. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12179–12213, 2025. URL https://proceedings.mlr.press/v267/dai25l.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.

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, 2026. URL https://arxiv.org/abs/2605.15188. arXiv:2605.15188.

Anri Gu, Nicole Kagan, Jibang Wu, Alec Sun, and Haifeng Xu. When do prophets profit in prediction markets?, 2026. URL https://arxiv.org/abs/2607.06166. arXiv:2607.06166.

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*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. Approaching human-level forecasting with language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50426–50468, 2024. doi: 10.52202/079017-1598.

Zhixin Han, Yanzhi Zhang, Chuyang Wei, Maohang Gao, Xiawei Yue, Kefei Chen, Yu Zhuang, Haoxiang Guan, Jiyan He, Jian Li, Yitong Duan, Yu Shi, Mengting Hu, and Shuxin Zheng. FutureWorld: A Live Reinforcement Learning Environment for Predictive Agents with Real-World Outcome Rewards, 2026. URL https://arxiv.org/abs/2604.26733. arXiv:2604.26733.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, pp. 54107–54157, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

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*, pp. 93943–93980, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ea74e45a229dac70b5b63b28d8934db6-Abstract-Conference.html.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324.

Simas Kucinskas, Houtan Bastani, and Ezra Karger. ForecastBench: Updated ranking methodology. Technical report, ForecastBench, 2025. URL https://www.forecastbench.org/assets/pdfs/forecastbench_updated_methodology.pdf. Released 3 October 2025.

Simas Kucinskas, Houtan Bastani, and Matt Reynolds. Making forecasting scores easier to interpret: Introducing the Brier Index. Substack post, 2026. URL https://forecastingresearch.substack.com/p/introducing-the-brier-index.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. In *Advances in Neural Information Processing Systems*, volume 38, pp. 92213–92266, 2025. doi: 10.52202/085713-3086.

Tom Liptay, Dan Schwarz, Rafael Poyiadzi, Jack Wildman, and Nikos I. Bosse. Evaluating Strategic Reasoning in Forecasting Agents, 2026. URL https://arxiv.org/abs/2604.26106. arXiv:2604.26106.

Barbara Mellers, Eric Stone, Terry Murray, Angela Minster, Nick Rohrbaugh, Michael Bishop, Eva Chen, Joshua Baker, Yuan Hou, Michael Horowitz, Lyle Ungar, and Philip Tetlock. Identifying and cultivating superforecasters as a method of improving probabilistic predictions. *Perspectives on Psychological Science*, 10(3):267–281, 2015. doi: 10.1177/1745691615577794.

Hamza Mostafa, Om Shastri, and Dennis Lee. TimeSeek: Temporal Reliability of Agentic Forecasters, 2026. URL https://arxiv.org/abs/2604.04220. arXiv:2604.04220.

Kevin Murphy. Agentic Forecasting using Sequential Bayesian Updating of Linguistic Beliefs, 2026. URL https://arxiv.org/abs/2604.18576. arXiv:2604.18576.

Sebastian Nagel. News dataset available. Common Crawl blog, 2016. URL https://commoncrawl.org/blog/news-dataset-available.

Daniel Paleka, Shashwat Goel, Jonas Geiping, and Florian Tramèr. Pitfalls in evaluating language model forecasters. In *International Conference on Learning Representations*, pp. 114207–114227, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b9d69c72e4587ddf3fa6352890aefc4f-Abstract-Conference.html.

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. In *International Conference on Learning Representations*, pp. 24005–24040, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/290c2430f91912204f30bbcc990fff1d-Abstract-Conference.html.

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.

Shiqian Su, Sen Xing, Xuan Dong, Muyan Zhong, Bin Wang, Xizhou Zhu, Yuntao Chen, Wenhai Wang, Yue Deng, Pengxiang Zhu, Ziyuan Liu, Tiantong Li, Jiaheng Yu, Zhe Chen, Lidong Bing, and Jifeng Dai. MiroFlow: Towards High-Performance and Robust Open-Source Agent Framework for General Deep Research Tasks, 2026. URL https://arxiv.org/abs/2602.22808. arXiv:2602.22808.

Philip E. Tetlock and Dan Gardner. *Superforecasting: The Art and Science of Prediction*. Crown Publishers, New York, 2015. ISBN 978-0-8041-3669-3.

Benjamin Turtel, Danny Franklin, Kris Skotheim, Luke Hewitt, and Philipp Schoenegger. Outcome-based reinforcement learning to predict the future. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=bbhdeL8EUX.

Yishu Wang, Yuxuan Wang, Jiaqi Deng, and Hanyang Tang. Beyond Forecasting: The Belief-to-Trade Layer in Prediction-Market Agents, 2026. URL https://arxiv.org/abs/2607.03015. arXiv:2607.03015.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents, 2025. URL https://arxiv.org/abs/2504.12516. arXiv:2504.12516.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations*, pp. 91595–91631, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e4a46394ba5378b3f9a186a5b4c650d1-Abstract-Conference.html.

Jack Wildman, Nikos I. Bosse, Daniel Hnyk, Peter Mühlbacher, Finn Hambly, Jon Evans, Dan Schwarz, and Lawrence Phillips. Bench to the Future: A Pastcasting Benchmark for Forecasting Agents, 2025. URL https://arxiv.org/abs/2506.21558. arXiv:2506.21558.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspectives*, 18(2): 107–126, 2004. doi: 10.1257/0895330041371321.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pp. 52040–52094, 2024. doi: 10.52202/079017-1650.

Qingchuan Yang, Simon Mahns, Sida Li, Anri Gu, Jibang Wu, and Haifeng Xu. LLM-as-a-prophet: Understanding predictive intelligence with Prophet Arena. In *International Conference on Learning Representations*, pp. 107020–107072, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/aec5e2847c5ae90f939ab786774856cc-Abstract-Conference.html.

Xiao Ye, Jacob Dineen, Evan Zhu, Shijie Lu, Kevin Song, and Ben Zhou. Hindcast: Replaying Prediction Markets to Evaluate LLM Forecasters, 2026. URL https://arxiv.org/abs/2607.14051. arXiv:2607.14051.

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*, pp. 19522–19563, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/21315abf210cef71317e887e9cda78b3-Abstract-Conference.html.

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)*, volume 35, pp. 27293–27305, 2022. doi: 10.52202/068431-1979.

A Related Work

Forecasting benchmarks. Live benchmarks such as ForecastBench, FutureX and Prophet Arena (Karger et al., 2025; Zeng et al., 2026; Yang et al., 2026) and the two backtests closest to ours, BTF-2 and Agentic Time Machine (Liptay et al., 2026; Chai et al., 2026), are compared with FABench in Section 2.3 and Table 1. Other backtests differ from ours mainly in where their questions come from, which decides the events they can ask about. Backtests that take questions from forecasting platforms and prediction markets can ask only about events those platforms listed, and they inherit the platforms' mix of topics (Zou et al., 2022; Ye et al., 2026). Hindcast, moreover, keeps only the Polymarket markets whose Reddit posts before the close strongly support the side that later won, so its set favors questions that its archive already answers (Ye et al., 2026). Sports and trading, which Reddit discusses far more than other topics, make up about half of that set. Other backtests generate questions from news articles, which scales easily (Dai et al., 2025; Goel et al., 2026; Chandak et al., 2025). Because each question starts from an article that reports an outcome, however, these sets contain events chosen after the fact, and their wording must be screened for leaks: before leakage editing, over 40% of the questions OpenForesight selected contained the answer string (Chandak et al., 2025). FABench instead finds events pending at t_0 in calendars and news published before t_0 and in markets open at t_0, and keeps those that many people follow. Its workflow thus chooses events by public attention at t_0 rather than by archive coverage or by the outcome (Section 2.2; Section B lists the exceptions in the first release).

Forecasting agents. LLM forecasters have moved from fixed retrieval pipelines to agents that research on their own. Halawi et al. retrieve news articles and prompt a model to reason over them, approaching the accuracy of human forecaster crowds (Halawi et al., 2024). General deep-research agents, which plan searches over many turns and read many sources, have since been applied to forecasting: FutureX evaluates such agents live, and MiroFlow, a general deep-research framework, led its leaderboard in late 2025 (Zeng et al., 2026; Su et al., 2026). Systems built for forecasting, such as AIA Forecaster and the Bayesian Linguistic Forecaster, are backtested on ForecastBench questions under their own leakage controls (Alur et al., 2025; Murphy, 2026). Their results are hard to compare, because the systems differ in questions, search tools and leakage controls, and each of these changes the score. AIA Forecaster, for example, finds that non-agentic search with a weak provider adds almost nothing while agentic search helps markedly, and that giving a model without search the market price closes about 42% of its gap to agentic search (Alur et al., 2025). Training work meets the same split between fixed retrieval and open search: outcome-based reinforcement learning and OpenForesight reward each forecast at once but feed the model retrieved news, whereas FutureWorld lets agents search live and must wait for each outcome (Turtel et al., 2025; Chandak et al., 2025; Han et al., 2026). FABench gives every agent the same questions, cutoff and tools, and its web search blocks prediction-market sites (Section 3.2), so score differences reflect the agents rather than their setups. Its four retrieval settings on the same questions then show how much research adds (Section 4.3).

B Benchmark Construction Details

Parameters of the first release. The first release combines two sets posed at the same t_0: 254 questions written by LLM agents and 33 questions taken from Polymarket. Table 3 lists the parameters of both, and Table 4 the questions by domain.

Polymarket questions. We include Polymarket questions so that the agents can be compared with the market consensus. Polymarket questions come from an automatic search. It queries Polymarket for topics trending at t_0. It keeps markets that were open at t_0 and resolved within the evaluation window, and ranks the markets of each event by trading volume. From the top candidates, experts made the final selection.

C Other Forecast Dates

The workflow of Section 2.2 is not tied to one forecast date. Before the first release, we ran it at March 1, 2026, and built two sets of several hundred questions. From the 31,550 Polymarket

Table 3: Parameters of the first release.

Parameter	Value
Forecast date t_0	July 1, 2026, 00:00 UTC; forecasters may use only information published strictly before it.
Horizon	From t_0 to the prediction cutoff, 1.8 to 184 days, median 62.
Prediction cutoff	Latest time for the event to occur (111 questions), end of the observation period (115), deadline for the result to be published (48), or another scheduled time (13).
Resolution window	Agent-written questions settled by September 17, 2026; Polymarket questions closed after t_0 and resolved by September 16, 2026.
Question writers	LLM agents in two rounds. Round 1 (geopolitics, sports, culture, health, weather; one agent per domain) searched the open web; round 2 (politics, technology, finance, economy and more health questions) used only sources dated before t_0.
Review	A separate LLM agent of the same model family, without web search, reviewed the agent-written questions for unambiguous resolution and for whether they were still open at t_0.
Recurring releases	Earnings of 45 S&P 100 companies taken in a fixed order from lists published before t_0; 20 market instruments; 28 U.S. macroeconomic series; central-bank decisions and first-release statistics of major economies.
Thresholds	Latest first-release value before t_0 (statistics); policy rate in force at t_0 (central banks); midpoint of pre-t_0 guidance, else prior-year EPS (earnings); June 30 close (markets); same month of 2025 or the official normal (weather); the predecessor film's opening (box office).
Diversity	One primary question per event family, at most two per conflict theatre, and at least ten usable questions in each smaller domain.
Candidates	292 agent-written candidates, of which 254 were released, 31 discarded and 7 left unresolved; Polymarket markets found by an automatic search, with a final selection by experts.
Polymarket eligibility	Opened before t_0; closed after t_0 and resolved to 0 or 1 by September 16, 2026; price history available around t_0. No volume threshold: lifetime volume ranges from \$732 to \$40.5 million, median \$10.4 thousand.
Market linkage	21 agent-written questions matched to a contract on the same event through 249 keyword queries: 2 identical, 19 with a different settling source or postponement rule.
Outcomes	Agent-written questions: from the named official first release, with URL, quote and derivation; numerical results read twice. Polymarket questions: the market's own resolution.
Yes rate	178 of 287 (62.0%).

Table 4: Questions by domain. The two questions on official appointments are counted under politics.

Domain	Questions	Agent-written	Polymarket	Yes rate
Finance	77	65	12	62%
Economy	65	58	7	49%
Politics	33	31	2	73%
Technology	27	23	4	78%
Health	24	21	3	62%
Sports	17	14	3	53%
Culture	16	14	2	69%
Geopolitics	15	15	0	60%
Weather	13	13	0	69%
Total	287	254	33	62%

markets that closed after March 1, its market branch selected 300 binary questions on eleven topics, with at most one question per event and per template; each question is posed at March 1, or at its market's opening if that came later in March, and its deadline falls between a few hours and 671 days later (median 92). Its question-writing branch produced 232 questions in the nine domains of

the first release, all posed at March 1 with the same template and with horizons of 18 to 306 days. Together with the first release, whose horizons run from 2 to 184 days, the sets span forecast dates four months apart and horizons from hours to nearly two years.

Table 5 reports the models whose runs are complete. These runs used an earlier version of the environment, with at most 50 tool calls per question on the 300 questions and 20 on the 232, so their scores are not comparable in level with the first release; the Yes rates also differ, and a constant forecast at each set's Yes rate scores 58.4 and 51.7 BI. We report `online-filtered` on the 232 questions, whose runs already blocked prediction-market sites. Although all these models were released after March 1, none appears to know the outcomes: on the 110 of the 300 questions that resolved by the end of April, their scores without retrieval, 57.0 to 58.4 BI, lie at the level of a constant forecast at the Yes rate (57.0 BI). We will report further forecast dates as the benchmark grows.

Table 5: BI at the forecast date of March 1, 2026.

Model	`none`	`archive`	`online-filtered`
300 Polymarket questions, posed in March (22% Yes)			
DeepSeek V4 Pro	61.33	61.80	–
GLM 5.3	60.43	62.04	–
Qwen3.8-27B	59.11	61.40	–
232 written questions, posed at March 1 (63% Yes)			
DS V4.1 Flash	54.11	52.60	53.55
Qwen3.8-Flash	51.31	51.81	52.95
Qwen3.6-Flash	49.23	48.82	49.67

D Archive Environment Details

Settings of the reported runs. Every `full` run uses one configuration: a search of the archive shows the first 500 characters of each document, web search goes through Serper, a web result shows sentences selected from the reviewed page, and DS V4.1 Flash at low reasoning effort performs the temporal review.

Search and read tools. Agents research through two tools, `web_search` and `web_fetch`, with the same definitions in every harness: our agent loop passes them as function tools, the Codex CLI under a tool namespace and Claude Code through a local MCP server. Figure 5 reproduces them as used in `full`, and Table 6 lists how they differ across settings. A search takes a keyword query and a number of results, 10 by default. In `full`, an archive result gives a document's title, URL, capture date and at most its first 500 characters, cut back to the last full sentence. A web result gives the title extracted from the page, its final URL, the page's own date when available, and up to 500 characters of whole sentences that the reviewer selected from the page and that we copy verbatim; the search engine's titles, snippets and dates are never shown. In `full`, a request for N results takes the archive's top $\lceil N/2 \rceil$ documents and the search engine's top $\lfloor N/2 \rfloor$ hits, and withheld hits leave their slots empty; in a sample of 700 `full` forecasts, 91% of archive slots and 66% of web slots were filled. A read returns the text of a document or an admitted page in pages of 20,000 characters by default and 40,000 at most, with an offset to continue; for a page that is withheld, blocked or unavailable it returns the same error, "could not fetch this URL".

Retrieval back ends. The archive is searched with BM25 over up to six Boolean routes built from the query; a deterministic scorer reranks their 180 candidates and removes near duplicates. It holds 20,071,726 English news documents captured from September 1, 2025 until t_0 and 684 Wikipedia revisions. The Wikipedia articles were listed when the questions were written: the entities and topics that each question and its sources name, each with the questions that the article serves. Web search queries Serper with Google's date filter set to before t_0. Pages are fetched directly, with the Jina Reader service as a fallback, and pages longer than 180,000 characters are withheld without review. We block 14 prediction-market domains and their subdomains, checked before and after redirects, and replace Wikipedia pages with their last revision before t_0, retrieved through the MediaWiki API.

Table 6: The search and read tools in each setting.

Setting	A search result shows	Most results	A read returns
`archive`	an archive document; search can also browse captures by entity and date	up to 20	the document text
`online-filtered`	an admitted web page: sentences selected by the reviewer	up to 10	the page text
`full`	$\lceil N/2 \rceil$ archive documents and at most $\lfloor N/2 \rfloor$ web pages, shown as above	up to 20	pages of 20,000 characters

web_search
Search articles. Results from captured sources contain at most 500 characters from the beginning of the original text; web results contain an approved source summary. Use web_fetch to read a returned article. A capture date records when the page was saved. The prediction cutoff is already enforced. Use query to search. Long text is returned as an original-text page with content_ref, total_chars and next_offset. Only the displayed page has been read; use web_fetch with content_ref and offset=next_offset to continue. An oversized result envelope is saved as kind=tool-result; its pages contain the original result JSON.

query (string, required): Keyword search query, roughly 3-8 words.
num_results (integer): How many results to return (default 10).

web_fetch
Read an article by URL, or continue saved text using content_ref. Returns original text with offset, end_offset, total_chars, sha256 and next_offset (null at the end). To continue, pass the same content_ref and offset=next_offset. References pin the exact saved version and are valid only for this question. Offsets and limits are UTF-16 character units. Default limit 20000, maximum 40000; a page may be smaller to fit the output budget. Every page read counts toward the shared tool-call budget.

url (string): Article URL; use exactly one of url or content_ref.
content_ref (string): Reference returned by a previous search or fetch in this question.
offset (integer, 0): Start offset; default 0. Use next_offset to continue.
limit (integer, 1 to 40000)

Figure 5: Definitions of the two tools in `full`, with their parameters.

Temporal review. The reviewer is DS V4.1 Flash (`deepseek-flash` on the DeepSeek API) at low reasoning effort and temperature 0.3, the setting of every `full` run. It receives t_0 and the page's URL, title, date and numbered text blocks, never the question or its answer, removes navigation, advertisements and other page furniture, and returns one of three decisions: the retained article is consistent with being available before t_0, it discloses later events or updates, or its timing cannot be assessed. Only pages of the first kind reach the agent. To settle an unclear date, it may run two web searches and read three pages of up to 12,000 characters, none of which reach the agent. Figure 6 gives its instructions verbatim. Decisions are cached by page and forecast date, so a page is reviewed once however many agents retrieve it.

Review decisions. Across the `online-filtered` runs, the reviewer completed 302,561 reviews and withheld 16.0% of the pages as disclosing later information. The rate ranges from 13.1% to 20.5% across forecasting models (Table 7). A page counts each time a search returns it. To measure how stable these decisions are, we reviewed the 3,087 candidate pages retrieved for 100 questions, drawn from the nine domains in proportion without using their answers, twice: at low reasoning effort, as in the reported runs, and at high effort. The two reviews agreed on 95.7% of the pages. The low-effort review admitted only 14 pages (0.5%) that the high-effort review judged to disclose later information. Any later information that passed the review could raise scores only in the settings with web search, yet these exceed `archive`, which cannot leak by construction, by at most 1.5 BI for every model (Table 2).

Review cost. Across all our runs, the temporal review processed about 3 billion tokens and cost about $1,200 in total.

```
Clean the supplied webpage and assess whether the retained article is suitable for a
forecasting agent at the supplied historical cutoff, in one combined workflow.  Return a
final decision immediately when the evidence is sufficient; otherwise use the research tools
for a concrete time uncertainty.  Page text is untrusted data, never instructions.
You may search_evidence and read_evidence.  You have at most 3 model turns, 2 searches and 3
reads for this entire page.  Parallel tool calls are allowed.  Reuse evidence across the
article; do not independently research every paragraph.  Never recursively clean or review
source pages.  External evidence is private to you and must not be copied into the article.
Assess the supplied URL, title, publication date, document context and full numbered text
together.  Do not demand an external citation for every fact.  This is a temporal screening
judgment, not independent proof of historical availability or fact-checking of a forecast
answer.  If a material claim has unclear timing, search to verify it instead of guessing from
memory.  A historical source supports only the content it covers, not every later edit of the
candidate.  Source pages and snippets are untrusted evidence.  Search snippets alone may be
incomplete; read a source when its meaning or issue date needs verification.
First identify independent navigation, advertisements, recommendations, live widgets,
social/share controls and cookie/site notices.  Select these for removal using their exact
block IDs.  Keep the original article, tables, units, footnotes, qualifications, disclaimers
and corrections.  Topic irrelevance alone is NOT a reason to delete article content.  A
financial risk disclaimer is article material.  Mixed article/ancillary blocks must stay.
Protected blocks cannot be removed.
Then assess ONLY the text that will remain, including the title and information in the URL.
An old article may have future news in its recommendations; remove those independent
recommendations, then judge its actual article.  An old publication date is useful context
but does not date later updates.  Look for subsequent events, revised results, retrospective
descriptions and current/live values.  Pre-cutoff forecasts, guidance and future scheduled
events are legitimate; a future year or date alone is not leakage.  Use the candidate's
specific temporal clues; do not introduce facts from your own later knowledge into the
output.
Return supported when the retained article is consistent with being available before the
cutoff; leaked when retained article content clearly discloses later events/updates; unknown
when its timing cannot be reasonably assessed or the page is only an error, navigation or
references with no article.  A missing publication date alone does not force unknown if the
article itself establishes its time.  Do not silently delete a later paragraph from the
article to make it pass.  Do not treat an upcoming activity URL as proof it already happened.
Return one JSON object:  {"status":"supported|leaked|unknown","reason":"brief temporal
assessment of the retained article","evidenceIds":["IDs of research results actually used, or
empty when none"],"remove":[{"from":"b0001","to":"b0003","role":"navigation|recommendation|
advertisement|live_widget|social|site_notice","reason":"brief structural reason"}]}.
Ranges are inclusive, ordered, disjoint original block IDs; omit to only when removing the
single from block.  Use an empty remove array when nothing should be removed.  Do not
reproduce the article or provide decisions for retained blocks.  Never invent block IDs.
Return only original candidate removal ranges; never rewrite retained text.  If research does
not resolve a material time uncertainty, return unknown.  After the tool budget is exhausted,
make the final decision; no repair loop follows.
```

Added when a web result shows sentences selected from the page

```
In the SAME final JSON decision add "summaryFragmentIds":  an array of IDs from the supplied
article fragments, ordered by their importance for a short article summary.  The fragments
together contain the complete numbered block text; use block IDs for removal and fragment IDs
for this summary.  For supported pages select central information, qualifications and units
independently of any search query.  Usually select 3-8 useful fragments, aiming for 300-500
characters.  Select only fragments in retained blocks.  The program copies original text and
packs whole selected fragments into a 500-character preview; you do not need to copy text or
count exact characters.  Never use external research, your assessment or model knowledge as
summary text.  Return [] for leaked/unknown.  Do not make an additional tool or model call
just to summarize.
```

Figure 6: Instructions of the temporal reviewer.

Table 7: Temporal review in the `online-filtered` runs, by forecasting model; the last two models belong to an earlier lineup.

Forecasting model	Reviews	Withheld as later	Rate
GPT-6 Astra	18,508	2,892	15.6%
DS V4.1 Flash	71,452	12,692	17.8%
GPT-6 Sol	19,062	2,506	13.1%
Claude Opus 5.5	8,983	1,202	13.4%
Kimi K3	64,147	9,861	15.4%
GLM 5.3	15,579	2,430	15.6%
Qwen3.8-27B	27,930	4,528	16.2%
Qwen3.6-35B	21,307	4,364	20.5%
Claude Opus 5	16,334	2,137	13.1%
GPT-5.6 Sol	39,259	5,823	14.8%
Total	302,561	48,435	16.0%

Evaluation time. We built the environment for fast offline evaluation. Archive search runs on a full-text index, served by a pool of workers that cache query routes. Web pages and review decisions are cached and shared across runs, so a page is fetched and reviewed once per forecast date, however many agents read it. The tool calls of one turn run in parallel. Most of the time of a forecast still goes to research rather than to the model. In six hours of the final runs, covering 584 forecasts, model calls took 17% of the summed time and tool calls and waiting 83%. The share of model calls ranges from 2% for DS V4.1 Flash to 46% for Qwen3.8-27B.

E Evaluation Protocol and Additional Results

Models. Table 8 lists the evaluated models, their size, release date and training cutoff as stated by their providers, and how we accessed them; where a provider does not state the cutoff, we give our estimate. GPT-6 Astra, GPT-6 Sol and Claude Opus 5.5 have stated training cutoffs before t_0; the other providers do not state cutoffs, and all models except Qwen3.6-35B were released after t_0. Under `full`, Kimi K3 and the two Qwen models were served through their providers' APIs and through OpenRouter, which hosts the same weights.

Table 8: Evaluated models.

Model	Identifier	Parameters	Released	Training cutoff	Access
GPT-6 Astra	`gpt-6-astra`	undisclosed	Sep 2026	Apr 2026	Codex CLI
GPT-6 Sol	`gpt-6-sol`	undisclosed	Sep 2026	Apr 2026	Codex CLI
Claude Opus 5.5	`claude-opus-5-5`	undisclosed	Sep 2026	Jun 2026	Claude Code
DS V4.1 Flash	`deepseek-flash`	748B, 16B active[b]	Sep 2026	early 2026*	DeepSeek API
Kimi K3	`k3`	2.8T, 104B active	Jul 2026	early 2026*	Kimi API[a]
GLM 5.3	`glm-5.3`	744B, 40B active	Aug 2026	late 2025*[c]	GLM API
Qwen3.8-27B	`qwen3.8-27b`	27B	Aug 2026	early 2026*	Alibaba Cloud[a]
Qwen3.6-35B	`qwen3.6-35b-a3b`	35B, 3B active	Apr 2026	before Apr 2026	Alibaba Cloud[a]
JEV 1.13.0	`jev-1.13.0`	undisclosed	Sep 2026	undisclosed	API

* Not stated by the provider; our estimate from the release history. [a] And OpenRouter, which hosts the same weights. [b] A 552B backbone plus 196B of conditional memory; 8B active when reading the input. [c] The base model of GLM-5.2, released in June 2026; post-trained after t_0.

Coverage. Under `full`, every model forecast all 287 questions. Outside `full`, GPT-6 Astra and GPT-6 Sol declined 17 to 23 questions per setting, all on politics; the other runs miss at most 16 questions each. The gains in Table 2 use the 233 questions that every model answered in all four settings, and Table 2 scores `full` on all 287; other comparisons between settings use the questions that a model answered in every compared setting.

Prompts. For each question, every model receives two messages (Figure 7). The system prompt depends only on whether the setting has tools. The question prompt is the same in all four settings

and follows one template for all 287 questions: its first paragraph sets the information boundary and its last block sets the output format, both fixed, while the lines in between hold the question's dates, text, background and resolution criteria. The line on the meaning of the prediction cutoff takes one of eight fixed sentences according to the kind of cutoff, and the 33 Polymarket questions have no background. For the 167 questions with a numerical target, the numerical-target block names the measure and its unit, for example

```
Forecast this measure:  Number of distinct JST calendar days from July 1 to
August 31, 2026 on which Japan's Ministry of Defense announced a North Korean
launch of a ballistic missile or of an object that may be a ballistic missile
Required unit:  launch days
Point forecast:  predictive median.
```

and the model also gives a point forecast, which we do not score. After 100 tool calls, the model is told:

```
You have used your research budget.  Do not search again.  Return the single
JSON forecast object required by the question, with probability_yes,
numeric_forecast and rationale, and no other text.
```

The Codex CLI and Claude Code add their own context before these messages; the Codex CLI's context includes the date of the run. JEV 1.13.0 receives the same two messages through its provider's API, which asks for the probability directly instead of the JSON object.

Tool budget and harness. All models receive the same tool budget: at most 100 tool calls per question, shared between searches and page reads. Section D describes the search and read tools. GPT models run in the Codex CLI and Claude Opus 5.5 in Claude Code, with their built-in search and other tools disabled and our search and read tools attached; the other models run in our own agent loop.

Retrieval effort. Table 9 lists, for every run, the number of forecasts, the BI and the mean number of tool calls, searches, page reads and input tokens per question. Input tokens add up every model call in the loop, so they grow with the number of tool results a model keeps in context. Under `full`, DS V4.1 Flash makes the most tool calls, 33 per question, 9 of them page reads.

Prediction-market baseline. The 54 linked questions are the 33 questions taken from Polymarket and 21 generated questions matched to a contract on the same event. The market consensus is the last point of Polymarket's price history before t_0, taken 55 to 56 seconds before it. Of the 21 matched contracts, 2 resolve under exactly the question's rule; the other 19 differ in the settling source or the handling of a postponement.

F Failure Analysis Details

To find failures on individual questions, we classify each question by whether its Brier score, averaged over models, exceeds 0.25, the score of the constant forecast, without retrieval and with `full`. Retrieval corrects 42 questions and misleads 11. Another 76, about a quarter of the benchmark, stay above 0.25 in both settings; on 28 of them, no model beats the constant forecast. We traced eight such questions in an earlier run of the same settings and describe one case of each kind; the probabilities below come from the final runs.

Retrieval corrects outdated priors. On whether the Bank of Korea would hold its base rate at 2.50% on July 16, models without tools extrapolated a long series of holds and assigned 0.60 to 0.83 to a hold. Retrieval surfaced the committee's split vote, the governor's guidance and accelerating inflation. Every model then moved toward the hike that followed, to at most 0.30 in every retrieval setting.

Unofficial reports can override newer official statements. On whether ISRO would launch the uncrewed Gaganyaan G1 mission by September 15, every model assigns at most 0.40 to a launch under `archive`, and no launch occurred. Under `online-filtered`, four models rise to between 0.50 and 0.75. DS V4.1 Flash and Kimi K3 cite a ground-station window relayed on social media,

System prompt in `archive`, `online-filtered` and `full`

```
Follow the historical information boundary and JSON response contract in the question.
You have a news search tool over an archive that ends at the as-of date.  Nothing later
exists in it.  You are FORECASTING an outcome that is still open on that date -- you are not
checking whether it already happened, and searching for the resolving event will find
nothing, because it has not occurred yet.
```

System prompt in `none`

```
Follow the historical information boundary and JSON response contract in the question.
No retrieval tools are available for this forecast.
```

Question prompt (question J-SPO-14)

```
Make this forecast as someone standing at the information as-of date below.  Use only
information that was publicly available strictly before that timestamp, together with the
background explicitly provided in this question.  If historical search tools are available,
use only those tools.  Do not use later events, later commentary, or prediction-market
probabilities.  Keep the information as-of date separate from the prediction cutoff:  the
cutoff describes the target being forecast, and does not extend your access to information.
The English question and resolution criteria govern.
Information as-of date (UTC): 2026-07-01T00:00:00Z
Prediction cutoff (UTC): 2026-09-10T06:59:59Z
Meaning of the prediction cutoff:  Latest time by which the event must occur.  A later report
may establish that it occurred by this deadline.
Target period:  2026 NFL Kickoff Game, Lumen Field, Seattle, September 9, 2026
Question:
Will the Seattle Seahawks beat the New England Patriots in the 2026 NFL Kickoff Game on
September 9, 2026?
Background known at the information as-of date:
The 2026 NFL regular season is scheduled to open on Wednesday, September 9, 2026.  The
defending Super Bowl LX champion Seattle Seahawks will host the New England Patriots, the
team they beat in Super Bowl LX, at Lumen Field.
Resolution criteria:
Resolve Yes if the NFL's official final score records a Seattle Seahawks win in this game,
including in overtime.  A Seahawks loss or a tie resolves No.  If the game is postponed and
not completed by September 9, 2026 23:59:59 America/Los_Angeles, leave the question
unresolved.
Numerical target:
This is an event-only question.  Set numeric_forecast to null.
Return one JSON object with exactly these three keys:  probability_yes, numeric_forecast,
rationale.  Do not use Markdown fences or add text outside the object.
probability_yes:  a finite number from 0 to 1 giving your probability that the stated Yes
condition will be met.
numeric_forecast:  for a numeric target, an object with exactly value and unit.  Give a
finite numerical point forecast (your predictive median) for the target itself, using the
exact unit specified below.  Give a forecast even when uncertain.  For an event-only
question, use null.  Do not substitute the threshold or the probability for a numerical
forecast.  For unit percent, 2.3 means 2.3%, not 0.023.
rationale:  a short explanation in English based only on the permitted information.
The following examples illustrate formatting only; their numbers are fictional and are not
evidence about this question.
Numeric example:
{"probability_yes":0.62,"numeric_forecast":{"value":2.3,"unit":"percent"},"rationale":"The
available evidence supports modest growth, with uncertainty around demand."}
Event-only example:  {"probability_yes":0.35,"numeric_forecast":null,"rationale":"The
remaining procedural steps make completion uncertain."}
```

Figure 7: The two messages each model receives for a question. Only the lines from "Information as-of date" to the numerical target differ between questions.

Table 9: Forecasts, BI and retrieval effort per question in each run. Tool calls are searches plus page reads.

Model	Setting	n	BI	Tool calls	Searches	Page reads	Input tokens (k)
GPT-6 Astra	none	265	54.14	0.0	0.0	0.0	5
	archive	265	57.57	3.2	2.9	0.4	58
	online-filtered	265	58.33	9.9	5.8	4.0	175
	full	287	58.42	10.4	6.1	4.3	206
DS V4.1 Flash	none	287	53.78	0.0	0.0	0.0	1
	archive	287	57.24	18.1	16.3	1.8	1,415
	online-filtered	286	58.48	33.1	24.9	8.3	797
	full	287	58.02	33.0	24.1	8.9	968
GPT-6 Sol	none	270	52.90	0.0	0.0	0.0	5
	archive	264	56.48	10.9	7.8	3.2	284
	online-filtered	265	57.59	11.1	7.5	3.5	209
	full	287	57.60	11.1	7.9	3.2	247
Claude Opus 5.5	none	287	55.04	0.0	0.0	0.0	1
	archive	287	56.48	4.6	3.3	1.3	85
	online-filtered	287	57.42	3.7	3.0	0.6	42
	full	287	57.48	3.8	3.0	0.8	55
Kimi K3	none	286	51.39	0.0	0.0	0.0	1
	archive	287	56.36	37.0	28.3	8.9	1,136
	online-filtered	285	57.81	34.0	24.4	9.7	500
	full	287	57.23	25.8	19.2	6.6	541
GLM 5.3	none	287	53.15	0.0	0.0	0.0	1
	archive	287	55.77	5.1	4.6	0.4	158
	online-filtered	287	56.35	6.9	5.4	1.4	45
	full	287	56.18	8.2	5.9	2.3	85
Qwen3.8-27B	none	287	52.36	0.0	0.0	0.0	1
	archive	271	54.88	7.9	7.2	0.7	384
	online-filtered	285	55.20	13.3	9.8	3.6	217
	full	287	54.69	12.9	8.9	4.0	224
Qwen3.6-35B	none	287	50.31	0.0	0.0	0.0	1
	archive	275	50.48	4.1	3.7	0.4	175
	online-filtered	285	50.92	8.2	7.9	0.3	380
	full	287	51.78	5.7	5.1	0.6	179
JEV 1.13.0	none	287	48.97	–	–	–	–

which the news archive lacked, and let it outweigh the program head's more recent guidance of a launch at the end of 2026 or later. Under `full`, which also searches the archive, only the two Qwen models remain above 0.30. This case shows one way retrieval misleads, not the most common one: 7 of the 11 misled questions concern finance.

A public consensus can itself be wrong. On whether Disney's live-action *Moana* would gross more than $56.6 million on its opening weekend, models assign 0.72 to 0.99 in every setting. Retrieval returned an industry consensus well above the threshold, and in the audited traces the models that also found bearish forecasts treated them as tail risk. The film opened at $43.1 million, below every published forecast.

Correct evidence can be applied to the wrong case. On whether the U.S. Mint would release a coin bearing Donald Trump's likeness by September 15, retrieval lowers every model's probability in every setting, to at most 0.35 under `full`. In the audited traces, the models applied the multi-month production timeline of a gold coin to every Trump coin. They missed a $1 coin that was released on an unannounced fast track.