AGENTIC FORECASTING WITH STRUCTURED LINGUISTIC BELIEFS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose the Bayesian Linguistic Forecaster (BLF), which is a search-based ReAct style agent for (binary) forecasting that augments its context with a linguistic "belief state", consisting of a probability estimate and a *structured* natural-language evidence summary. (The belief update is "Bayesian-inspired" rather than exact Bayesian inference, although we do include a comparison to a more explicit Bayesian approach using LLM-elicited likelihood surrogates.) On 400 historical ForecastBench question-date instances, BLF has the highest overall Brier Index among the compared external agents, and is the only compared method to significantly outperform a crowd+empirical-prior baseline in terms of overall performance. Our backtesting framework combines date-aware search tools, leakage auditing, and paired comparisons for variance control to provide a rigorous empirical framework for comparing agentic forecasting systems.

1 INTRODUCTION

Forecasting the probability of future events is a fundamental challenge with applications in geopolitics, finance, and public health (Tetlock & Gardner, 2015; Spiegelhalter, 2025). Recent work has shown that LLMs can approach human-level forecasting when given web search access (Halawi et al., 2024), and benchmarks such as ForecastBench (Karger et al., 2025) provide standardized evaluation with online leaderboards.

Most forecasting agents use some kind of ReAct (Reason-then-Act) loop (Yao et al., 2023), where the agent sequentially accumulates search results in context. However, this can cause problems with various forms of "context rot" (see e.g., (Breunig, 2025; Liu et al., 2026b)), We present `BLF` (Bayesian Linguistic Forecaster), which augments the agent's history with a *semi-structured belief state* — a probability estimate paired with natural-language evidence summaries — updated by the LLM at each step. We refer to this loosely as "Bayesian-style" updating: the process is designed to mirror the form of a sequential Bayesian update (prior + likelihood $\rightarrow$ posterior), but the actual update is just an LLM forward pass.[1]

To evaluate `BLF` (and other agents), we develop a four-layer date-leakage defense for backtesting, which tackles a key problem in the field (Paleka et al., 2026). Our post-hoc audit shows an approximate residual leakage rate of only 1.5% of retained search results (Sec. B). In addition, we adopt a statistical approach to comparing methods based on paired analysis with bootstrap confidence intervals (Sec. E.2). This controls for the confounding factor of varying question difficulty, which accounts for 62% of the performance variance (Sec. E.1). Pairing is useful when differences between methods are small, but has been overlooked by most previous empirical work in this field.

On 400 backtesting questions from ForecastBench (FB), we find that `BLF` achieves the highest Brier Index (BI) score on every question type (Overall, Market, Dataset) compared against the top methods on the FB leaderboard at the time the work was done (2026-04-20, see Fig. 5). Although LLMs (and the leaderboard) have been updated since then, we believe our frozen results still present a useful comparison of different methods that is still relevant.

[1] Because we use the LLM to predict the probability at each step, our system may not satisfy the consistency criteria required to correspond to proper Bayesian inference (Qiu et al., 2025; Falck et al., 2024). In Sec. 6, we compare to an explicit Bayesian updating procedure using LLM-estimated likelihoods (or likelihood ratios), but we show that this gives lower point-estimate accuracy than our heuristic method.

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.

2 EXPERIMENTAL SETUP

Problem definition. We focus on binary prediction problems of the following type. Let $Y(t)$ denote the random variable of interest at time t. Let d be the forecast date and $r > d$ be the resolution date. Our task is to estimate $P(Y(r) = 1 \mid \text{data}(\leq d))$. We consider two main kinds of questions: open-ended text questions (e.g., "Will person X be the president of country C at time r?"), which require "judgemental forecasting" (Lawrence et al., 2006); and more traditional time-series forecasting problems (e.g., "Will the stock price of company C at time r be greater than its current value of v?"), which require numerical reasoning, but may also leverage text data where relevant.[2]

Datasets. We focus on our results on ForecastBench (FB) (Karger et al., 2025). This consists of judgemental forecasting questions from market sources (Polymarket, Manifold, Metaculus, Rand Forecasting Initiative (RFI)), and (binarized) time series forecasting problems from dataset sources (yfinance, FRED, DBnomics, Wikipedia, ACLED). (Formally, the dataset questions require estimating $P(Y(r_i) > v_d \mid \text{data}(\leq d))$, where v_d is a reference value available at time d, and $r_i = d + h_i$ is the resolution date at up to 8 forecast horizons h_i.)

We construct two evaluation tranches from FB (A: 2025-10-26, B: 2025-11-09), each with 100 market + 100 dataset questions (400 question-date instances across A∪B; 355 distinct underlying questions). Each market question resolves at a single date, while each dataset question may resolve at multiple horizons; the 200 market questions yield 200 binary resolution events, the 200 dataset questions yield 591 events (roughly three resolved horizons per question, since the 7/30/90-day horizons resolve within our window), giving 791 binary resolution events in total. The dates for these tranches were chosen to meet 3 criteria: (1) be after the knowledge cutoff of frontier LLMs (at the time the experiments were done); (2) maximize number of resolved questions; (3) maximize overlap with results from publicly submitted forecasts, to enable paired comparisons. See Sec. F.2 for details.

Metrics. A standard metric for binary forecasters is the (mean) Brier Score, defined as $\overline{\text{BS}} = \frac{1}{N}\sum_i (p_i - o_i)^2$, where $p_i \in [0, 1]$ is the i'th prediction, $o_i \in \{0, 1\}$ is the i'th outcome, and N is the number of predictions. For BS, the best score is 0, and a chance system scores 0.25. In (Kucinskas et al., 2026), they proposed the Brier Index, whose values are slightly more intuitive. This is defined as $\text{BI} = 100 \times (1 - \sqrt{\overline{\text{BS}}})$, so the best system scores 100%, the worst 0%, and a chance system scores 50%. See Sec. A.2 for discussion of other metrics. When evaluating on FB, overall scores are the unweighted average of market and dataset means, following the FB methodology.

Backtesting validity. To minimize parametric leakage, we select LLMs whose knowledge cutoff is before ForecastBench tranche A (2025-10-26) and tranche B (2025-11-09). To minimize leakage from search and tool use, we implement a four-layer defense: (1) search engine date filtering, (2) LLM-based leak classifier on results, (3) data tool date clamping, and (4) URL blocking for resolution sources. A second-model audit flags post-cutoff information in 1.5% of retained search results. Advertised knowledge cutoffs and date filters do not guarantee absence of leakage. See Sec. B for details.

LLMs. For the experiments we run (as opposed to historical FB results we extract from the leaderboard), we use 3 LLMs: PRO =Gemini-3.1-Pro; FLASH =Gemini-3-Flash; GPT =GPT-5.4. See Tab. 5 for details on these models.

External baselines. For our FB experiments, the main external comparison uses three methods that were top of leaderboard at the time the work was done (2026-04-20, see Fig. 5): Cassi (Cassi AI, 2025), Grok 4.20, and Foresight-32B (Lightning Rod Labs, 2025). All these methods use the crowd estimate for market questions as part of their prompt, and all use search and tools (see Sec. A.3 for

[2]These binarized time series forecasting problems may seem rather artificial. However, prediction markets such as Kalshi and Polymarket do contain similar binary questions, where a continuous outcome is binned into different (mutually exclusive) buckets, collectively called a "market" (Yang et al., 2025), and users are asked to bet on individual binary events. For example, https://polymarket.com/event/what-price-will-bitcoin-hit-in-may-2026 asks "What price will Bitcoin hit in May?" with choices such as $< 30,000$, $< 35,000$, up to $> 150,000$ (in USD). This question was posted on 2026-05-01 and will be resolved on 2026-06-01.

Algorithm 1 `BLF` and variants.

Require: Fixed context c (instructions, q, cutoff d, prior), max steps T, memory mode

$m_0 \leftarrow c$
for $t = 1, \ldots, T-1$ **do**
  $(b_{t-1}, a_t) \leftarrow \text{LLM}_{\text{mode}}(m_{t-1})$ *// NoBel emits only a_t; $b_{t-1} = \emptyset$*
  if $a_t = \texttt{submit}(\hat{p})$ **then**
    return $\hat{p}$
  end if
  $o_t \leftarrow \text{Env}(a_t; q, d)$ *// execute action; search APIs apply date cutoff d*
  if a_t is web search **then**
    $o_t \leftarrow \text{LeakFilter}(o_t, d)$ *// 2nd-pass LLM filter*
  end if
  if full-BLF: $m_t \leftarrow m_{t-1} \oplus (b_{t-1}, a_t, o_t)$
  if Markov-BLF: $m_t \leftarrow m_0 \oplus (b_{t-1}, a_t, o_t)$
  if NoBel: $m_t \leftarrow m_{t-1} \oplus (a_t, o_t)$ *// ReAct Baseline*
end for
return $\text{LLM}_{\texttt{submit}}(m_{\text{last}})$ *// reserved final call; latest available memory, all modes*

details). We also include *Crowd+emp*, an LLM-free baseline using market prices for market questions and empirical priors for dataset questions. The NoBel BLF ablation uses the same search-enabled loop as BLF but without a structured belief state.

3 METHOD

Main loop. BLF is a variant of the standard ReAct (Reason-then-Act) agent design (Yao et al., 2023), augmented with structured belief tracking. Independent rollouts can then be combined into one forecast. The main loop is illustrated in Figure 6, and the pseudocode is in Algorithm 1. At each step t, the LLM gets passed the previous message history (context) m_{t-1}, performs reasoning, and then returns an updated belief state b_{t-1} (representing a structured summary of the past) and a recommended action a_t, which can be a specialized tool call (such as looking up information in a database) or a web search. Next the agent executes the action and gets results o_t, and then it updates its message history by either appending (b_{t-1}, a_t, o_t) to m_{t-1} (full-BLF) or just appending (b_{t-1}, a_t, o_t) to the initial context m_0 (Markov-BLF); Sec. 4 evaluates whether retaining older history improves accuracy, without establishing that the summary is a sufficient statistic. The loop runs for up to $T_{\max} = 10$ steps, although the agent can choose to submit its prediction early if it is confident enough.

Belief state. The linguistic belief state is a semi-structured JSON object containing: a probability estimate $p \in [0, 1]$, a confidence level, key evidence for/against, and open questions, which is helpful for passing to the search query generation prompt. (See Sec. C.1 for more details.) NoBel is a ReAct-like ablation that retains the full history and appends (a_t, o_t) to m_{t-1}, without requesting or maintaining a structured belief.

Tools. At each step, the agent selects one of several actions: `web_search` (we use the Brave search engine with date filtering; see Sec. C.3 for a discussion), `summarize_results` (read and summarize retrieved pages), `lookup_url` (fetch a specific URL), source-specific data tools (e.g., `fetch_ts_yfinance`, `fetch_wikipedia_section`), or `submit`. The submit tool clips probabilities to $[0.02, 0.98]$, while the timeout fallback clips to $[0.05, 0.95]$. A meta-controller selects the set of tools available to the agent on a per-question-type basis. See Sec. C.2–Sec. C.6 for the details.

Crowd and empirical prior. For market questions, the crowd signal (market price) is injected into the prompt as an anchor; adding it substantially improves market BI. For dataset questions, we provide an *empirical prior* π_q — the base rate for each source and question subtype. The empirical prior has negligible effect on BLF (which acquires better data via search), but helps the no-LLM baseline. See Sec. C.7 for details.

Combining independent trials. We run K independent tool-use trajectories and combine their final probabilities. Arithmetic averaging, $\hat{p} = K^{-1} \sum_k p_k$, provides a simple reference. We also evaluate logit averaging and a shrinkage variant, $\hat{p} = \sigma(\alpha_q K^{-1} \sum_k \text{logit}(p_k))$. Here $\alpha_q = \max(f, 1 - c\sigma_q)$ depends on trial-logit dispersion, with (f, c) selected by cross-validation; a hierarchical Bayesian alternative is described in Sec. C.9. When $\alpha_q = 1$, shrinkage reduces to logit averaging; smaller values pull the estimate toward 0.5. The belief and memory ablations use arithmetic averaging without calibration. The external-agent comparisons retain the historical logit-aggregation and hierarchical-calibration configuration (Tab. 11).

In preliminary results (not reported here), we experimented with the LLM-based aggregation method from the AIA Forecaster (Alur et al., 2025), which passes the raw reasoning traces (but not the probability estimates) from all K trials to a final aggregation LLM (that can also perform its own search, to resolve disagreements). However, this worked worse than simple mean aggregation on our FB benchmark.

Post-hoc calibration. We evaluated ordinary Platt scaling (Platt, 1999) and a hierarchical extension with regularized source-specific offsets (Sec. C.10). Although preliminary results (not reported here) showed small gains on Kimi K2.5[3], we found no statistically resolved overall BI gain for Pro in the historical calibration comparison. This does not rule out gains on other backbones or subsets.

4 RESULTS

In this section, we present our results; see Sec. D for more details. External comparisons use the historical aggregation and calibration configuration in Tab. 11.

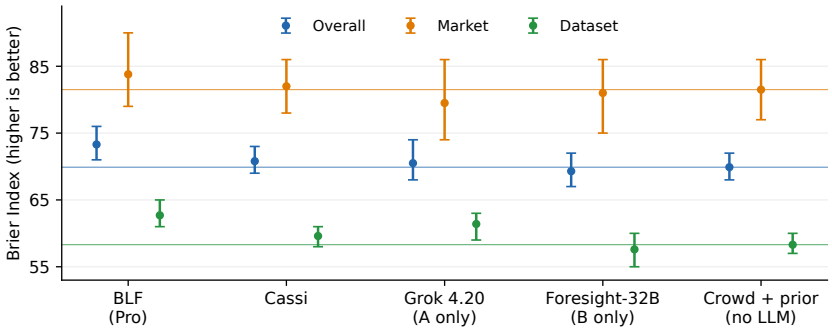

Figure 1: **Comparison to external methods and baselines on FB A∪B**. We plot Brier Index, split by Overall, Market, and Dataset. Thin horizontal lines mark the corresponding crowd+prior baseline means. Error bars: rounded 95% bootstrap intervals from Tab. 12. See Tab. 1 for pairwise comparison and significance testing.

Comparison to SOTA. Fig. 1 compares `BLF` (using PRO) with external methods and baselines on ForecastBench. BLF has the highest reported point estimate for Overall, Market, and Dataset BI. Exploratory paired bootstrap tests on common resolution events (shown in Tab. 1) find significant Overall and Dataset improvements over Cassi and Foresight. Grok is not statistically distinguishable from BLF on its tranche-A coverage ($p = 0.51$), despite its lower unpaired aggregate score. Market comparisons have less power: A∪B contains 200 market events, and Grok and Foresight each cover only one 100-event tranche. Only the market gap against Foresight reaches significance. The table reports tranche coverage, paired differences, and significance; Secs. D.1 and E.2 give evaluation details.

[3]When using Platt calibration, we saw +0.72/+1.48 BI gains on FB questions with/without the crowd baseline. When using our hierarchical calibration, we saw +0.98/+1.80 BI gains. (The gains are larger on questions without the crowd baseline, since that tends to act as a very informative prior, reducing the need for calibration.)

Table 1: **Comparison of BLF (Pro) to 3 external agentic methods.** We show Brier Index (higher = better) on FB A∪B. All methods use the crowd signal for market questions. Absolute BI per (question type, tranche) plus paired Δ BI vs BLF (Δ < 0 = BLF wins). Tranche A: 2025-10-26; B: 2025-11-09; All = A∪B. Grok submitted only A and Foresight only on B, so the unsubmitted-tranche cells are shown as ×. Sample sizes for each partition are in Tab. 13. Significance: $^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$, $^{ns}p\geq0.05$ (paired bootstrap, 5,000 resamples).

	Market				Data				All			
Method	A	B	All	Δ	A	B	All	Δ	A	B	All	Δ
BLF (PRO)	81.3	86.7	**83.8**		62.4	63.1	**62.7**		71.9	74.9	**73.3**	
Cassi	80.1	84.2	82.0	-1.7^{ns}	61.8	57.4	59.6	-3.1^{***}	71.0	70.8	70.8	-2.4^{*}
Grok	79.5	×	79.5	-1.9^{ns}	61.4	×	61.4	-1.0^{ns}	70.5	×	70.5	-1.4^{ns}
Foresight	×	81.0	81.0	-5.7^{***}	×	57.6	57.6	-5.5^{***}	×	69.3	69.3	-5.6^{***}

Table 2: **Comparison to the LLM-free crowd+empirical-prior baseline.** The first row gives absolute BI; subsequent rows give paired ΔBI (method minus baseline, so positive means better for method) on common coverage. Within each type, A and B are the tranches in Tab. 1; All pools available tranches. Overall BI equally weights Market and Data. ×: no submission. Stars: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, ns otherwise (5,000 paired resolution-event bootstrap resamples, stratified by type; unadjusted tests, fitted calibration held fixed). Only BLF significantly improves Overall BI on its full coverage.

	Market			Data			Overall		
Method	A	B	All	A	B	All	A	B	All
Crowd+prior (absolute)	81.4	81.6	81.5	56.7	59.9	58.3	69.0	70.8	69.9
BLF (Pro)	-0.1^{ns}	$+5.1^{ns}$	$+2.3^{ns}$	$+5.7^{***}$	$+3.2^{*}$	$+4.5^{***}$	$+2.8^{ns}$	$+4.1^{**}$	$+3.4^{**}$
Cassi	-1.3^{ns}	$+2.7^{ns}$	$+0.6^{ns}$	$+5.2^{***}$	-2.5^{*}	$+1.3^{ns}$	$+1.9^{ns}$	$+0.1^{ns}$	$+1.0^{ns}$
Grok	-1.9^{ns}	×	-1.9^{ns}	$+4.7^{***}$	×	$+4.7^{***}$	$+1.4^{ns}$	×	$+1.4^{ns}$
Foresight	×	-0.6^{ns}	-0.6^{ns}	×	-2.3^{ns}	-2.3^{ns}	×	-1.5^{ns}	-1.5^{ns}

Comparison to LLM-free baseline. Tab. 2 shows that BLF is the only method (in our comparison set) to significantly beat the LLM-free crowd+empirical-prior baseline in terms of overall BI. Grok and Cassi trend positive but are not significantly above it.

Benefits of the belief state. Tab. 3 isolates the benefits of the belief state. The first row for each LLM backbone reports absolute NoBel BI; the next reports the paired effect of adding the structured belief state. PRO and FLASH improve overall, driven mainly by market questions; GPT has no resolved overall gain and a small negative dataset effect.[4] We do not know exactly why the belief state helps with PRO and FLASH, but not with GPT. However, we did notice that the mean within-question cross-trial standard deviation on market forecasts falls for PRO (0.094 to 0.030) and FLASH (0.105 to 0.032), but rises slightly for GPT, suggesting some modest stabilizing effect of a structured context. However, it is also possible that the structured representation helps formulate better search queries; unfortunately it is hard to test for this in our current experimental setup (for engineering reasons).

Benefits of averaging multiple trials. By default we use arithmetic averaging of the probabilities from $K = 5$ trials per question. In Fig. 2(a) we show the effect of varying K on pre-computed BLF(PRO) trajectories collected without crowd forecasts or empirical-prior anchors on ForecastBench. We see gains with increasing K when using the arithmetic mean, and a small boost when using the logit mean. When using leave-one-out cross-validation, the shrinkage method selects a shrinkage coefficient of $\alpha_q = 1$, which reduces to standard logit averaging.

[4] Earlier experiments showed large gains on Kimi-K2.5, but its market submission rate was only 6.2% without belief tracking versus 82.4% with it. Failed NoBel runs defaulted to 0.5, while belief agents retained an intermediate forecast. We omit these results because they confound forecast quality with recovery from submission failures.

Table 3: **Belief state ablation.** We report NoBel baseline BI, and full-BLF minus NoBel ΔBI (higher is better), crowd prior, five trials, arithmetic averaging, no shrinkage or calibration. Pro/GPT: 400 question–date instances; Flash: 399. Brackets: exploratory 95% question-cluster intervals, stratified by type; bold differences exclude zero. All equally weights Market and Data. See Sec. 4; intervals use 5,000 paired resamples.

Model	Reference / comparison	Market	Data	All
PRO	NoBel (absolute BI)	78.87	61.45	70.16
	+ Belief	$+\mathbf{5.34}$ [3.32, 7.72]	$+\mathbf{0.67}$ [0.06, 1.31]	$+\mathbf{3.00}$ [1.95, 4.25]
FLASH	NoBel (absolute BI)	79.07	61.44	70.26
	+ Belief	$+\mathbf{4.06}$ [2.38, 6.02]	-0.07 [-0.65, 0.50]	$+\mathbf{1.99}$ [1.12, 3.02]
GPT	NoBel (absolute BI)	80.41	61.96	71.18
	+ Belief	-0.62 [-1.49, 0.21]	$-\mathbf{0.22}$ [-0.39, -0.05]	-0.42 [-0.86, 0.002]

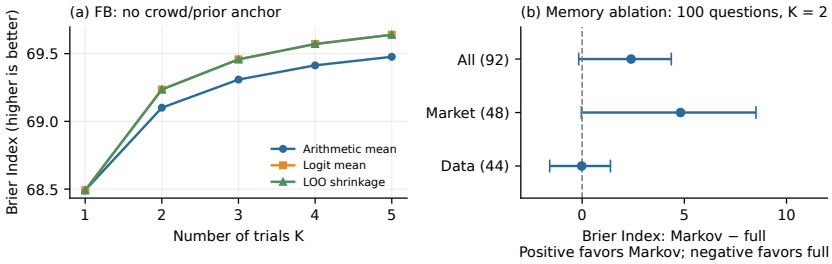

Figure 2: **Effect of multi-trial averaging and Markovian memory.** (a) Trial averaging without crowd/prior anchors: 400 ForecastBench questions, 791 events. Expected loss over trial subsets is converted to BI; All equally weights Market/Data. Shrinkage uses leave-one-question-out tuning and coincides with logit averaging. (b) Separate 100-question memory study, $K = 2$. We show the difference in Brier Index for BLF-Markov minus BLF-Full, together with 95% paired question-cluster intervals. Counts show complete pairs; positive favors Markov. All equally weights Market/Data; every interval includes zero.

Full-history versus Markov memory. Fig. 2(b) compares full BLF, $m_t \leftarrow m_{t-1} \oplus (b_{t-1}, a_t, o_t)$, with Markov-BLF, $m_t \leftarrow m_0 \oplus (b_{t-1}, a_t, o_t)$, on 100 fresh questions (50 Market, 50 Data), selected independently of outcomes. Both arms use Pro, $K = 2$, at most ten calls including reserved final submission, and identical tools, including date-filtered search and URL lookup, but no lookup by index into previously stored documents (which could violate the Markov property). Full BLF submits in 191/200 rollouts versus 199/200 for Markov; 92 questions (48 Market, 44 Data) have both trials complete in both arms. After averaging the two forecasts, overall BI is 72.72 for full BLF and 75.12 for Markov: $\Delta\text{BI} = +2.40$ (95% CI $[-0.17, 4.36]$; 10,000 paired question-cluster resamples, stratified by type). The point estimate favors Markov, mainly on markets, but every interval includes zero.

5 EXAMPLE OF A BELIEF STATE TRACE

Belief evolution. In this section, we perform a qualitative analysis of the behavior of the BLF agent using PRO. Figure 3 shows how the agent's probability estimate evolves across steps for 5 independent trials on a single question.[5] All trials start at $p_0 = 0.5$ (no information) but quickly diverge as different search queries lead to different evidence. The high inter-trial variance ($\sigma = 0.20$)

[5]The question is from the Metaculus AI Benchmark Tournament (Q2 2025), available at https://www.metaculus.com/tournament/aibq2/. The question is as follows: "Will WorldAtlas.com display the Gulf of America before July 1, 2025?" The context is that President Trump had ordered that the "Gulf of Mexico" be renamed to "Gulf of America" on January 20, 2025. See https://en.wikipedia.org/wiki/Executive_Order_14172 for context.

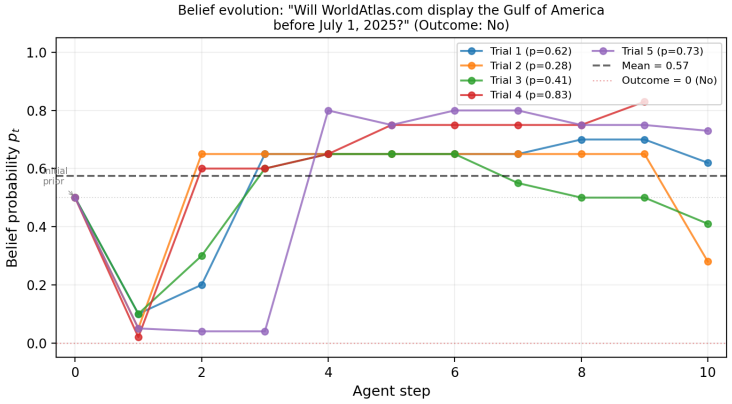

Figure 3: **Belief evolution across 5 trials.** The question is "Will WorldAtlas.com display the Gulf of America before July 1, 2025?" (outcome: No). All trials start at $p_0 = 0.5$ and dip low at steps 1–2 (initial search finds skepticism), then diverge as different search paths lead to different conclusions. High inter-trial variance (σ=0.20) motivates aggregation.

Question: Will WorldAtlas.com display the Gulf of America before July 1, 2025?
Trial 2 (forecast: 0.28) — the "skeptic" among 5 trials **Outcome:** No

Step	p_t	**Action & key evidence**
0	0.50	Prior (no information)
1	0.05	`search`("Gulf of America Gulf of Mexico")
		↓ "Renaming attempts usually fail; base rate <1%"
2	0.65	`lookup_url`(worldatlas.com/.../na.htm)
		↑ Executive Order signed; Google Maps changed label
3–8	0.65	Multiple searches — inconclusive
9	0.65	`search`("worldatlas.com maps API")
		↓ WorldAtlas uses *static* map images
10	**0.28**	`submit` — "Static sites take 1–3 years to update"

Figure 4: **Agent trace for Trial 2 (Figure 3).** The agent initially drops to $p = 0.05$ (skeptical prior), jumps to 0.65 after finding the executive order (from President Trump), then stays at 0.65 for 7 steps while searching inconclusively. The static-map concern is incorporated into the final forecast at the step-10 submission, when the probability drops to $p = 0.28$. This trial correctly identifies the key uncertainty — whether the site has updated its static maps — that the other four trials miss.

motivates multi-trial aggregation as a way to reduce reliance on any one trajectory; the mean forecast is 0.57. The trace from one of the rollouts (Figure 4) shows how the belief state captures key evidence: Trial 2 correctly identifies that WorldAtlas uses static maps (not dynamic Google Maps), leading this agent to become skeptical that the change will happen before the deadline. (And indeed the true outcome is "No".)

The Trial 2 "aha" moment. The drop $0.65 \rightarrow 0.28$ at Trial 2's submit step (Fig. 4) is not a new search hit — it is the LLM *finally synthesizing* two pre-existing latent concerns (the `webimage/countrys/` URL structure looks like a legacy static page, and WorldAtlas relies on proprietary static map images that require manual editing) into a new `evidence_against` entry and corresponding `key_uncertainty`. This crystallization happens inside the final reasoning pass, after the agent decides it has enough material to commit; it overrides the optimistic priors driven by Trump's EO and WorldAtlas's other article-title update, and the belief drops by 0.37 in a single step. This is precisely the regime that Trial 2 captures correctly and the other four trials miss; the mean does not beat the best individual trial on this example.

6 Explicit Bayesian Updating

The approach described in Sec. 3 is heuristic, and uses an LLM to compute the probability at each step. In this section, we describe a more "orthodox" approach, where we update the numerical

beliefs with Bayes' rule, using an LLM to estimate the likelihood or likelihood ratio, as suggested in (Papamarkou et al., 2026; Amin, 2026).

The explicit update. We will use this notation: $s \in \{0, 1\}$ is the latent (unknown) outcome to be predicted, a_t is the search query (action), o_t the resulting text response, i.e., the observations (concatenated together), c is the context (prompt containing the question), and $b_t = p(s = 1|a_{1:t}, o_{1:t}, c)$ is the belief state at step t. The sequential update for the posterior in log-odds form is

$$\operatorname{logit} b_T \;=\; \operatorname{logit} b_0 \;+\; \alpha \sum_{t=1}^{T} \lambda_t, \qquad \lambda_t \;=\; \log \frac{q_t(1)}{q_t(0)} \tag{1}$$

where each log likelihood term λ_t depends on $q_t(s)$, which we discuss below. Here α is a tempering coefficient we discuss below.

Likelihoods. In the simplest case, we have $q_t(s) = p(o_t \mid s)$, so λ_t is a standard log-likelihood ratio, but we can also condition on other quantities. In Tab. 4 we list many possibilities. Here $h_{t-1} = (a_{1:t-1}, o_{1:t-1})$ denotes the raw history, and $b_{t-1}.h$ denotes the structured belief's evidence summary with its probability removed, to avoid feeding the running posterior into the likelihood estimate. Conditioning on a_t accounts for the issued query; conditioning on h_{t-1} accounts for all the past evidence; conditioning on $b_{t-1}.h$ aims to just condition on evidence in the structured belief state. An exact chain-rule update would require full-history conditioning or a sufficient summary and correctly specified likelihoods; neither is guaranteed here.

Eliciting the likelihood. To estimate the quantities $q_t(s)$, we follow the procedure in Amin (2026, Eq. 61): for each hypothesis $s \in \{0, 1\}$ we ask the LLM (FLASH) to rate, on an integer scale 0–10, how *typical* the observations o_t look given the conditioning context s plus relevant history. (See Sec. I.7 for the prompt.) A typicality judgment is more natural for an LLM than a probability, and querying each state in a separate call avoids the implicit discriminative normalization of a single classification prompt. (A more direct alternative that asks the model for the log-ratio $\log p(o_t \mid s{=}1)/p(o_t \mid s{=}0)$ in a single call produces inflated, frequently wrong-signed weights and was uniformly worse.) Writing $r_t(s) \in [0, 10]$ for the two scores, we define

$$\lambda_t \;=\; \log \frac{r_t(1) + \epsilon}{r_t(0) + \epsilon}, \qquad \epsilon = 0.5, \tag{2}$$

where the floor ϵ keeps a zero score from producing an infinite weight. The bounded integer scale also caps the magnitude of each λ_t. These typicality scores are likelihood surrogates, not normalized text likelihoods.

Summarizing text observations. The raw observation o_t — the concatenated text of one search step — averages 7,571 characters. So we also test a compressed, *task-aware* observation: a single auxiliary LLM call, given the question c and the issued query a_t, rewrites o_t into a short summary, $\sigma_t \;=\; \sigma(o_t \mid a_t, c)$, of mean length 798 characters (11% of o_t); conditioning the summary on a_t and c keeps what is relevant to the question. The per-state likelihood is then evaluated on σ_t in place of o_t.

Tempering. The α term is a *tempering* coefficient, the same as the composite-likelihood exponent of Papamarkou et al. (2026, Eq. 1). This can be useful because with conditionally dependent observations, the naive product over-counts evidence. If we set $\alpha < 1$, it down-weights each likelihood factor to restore approximately calibrated precision (a Gibbs / power-posterior correction). We sweep α on the replay sample and report its best value; these exploratory results are not held-out validation estimates.

Results. At the shared $\alpha = 0.35$, removing the belief summary increases BS by 0.0057 (95% paired question-bootstrap CI $[-0.0122, 0.0290]$); its incremental benefit is unresolved. Alpha was selected on this sample, and the old and new likelihood scores were elicited on different dates.

Some results, using $n = 80$ tranche-A questions (154 events), are shown in Tab. 4. Of all the likelihood-based approaches, the best one (in terms of BI) uses $q_t(s) = p(\sigma_t \mid s, a_t, b_{t-1}.h, \sigma_{1:t-1})$,

Table 4: **Explicit-Bayes belief update vs the linguistic update (`BLF`)**. We replay all methods on the same evidence sequence. BS first averages horizons within each question, then averages the 80 questions; BI transforms this mean, while ECE weights resolution events equally. Higher BI $= 100(1 - \sqrt{\text{BS}})$ and lower ECE are better; $\alpha^\star$ is the best tempering coefficient; both metrics are evaluated at $\alpha^\star$. The top block uses the raw observation o_t, the bottom the task-aware summary σ_t. All quantities are implicitly conditioned on the question context c (omitted for brevity).

Method	$q_t(s)$	$\alpha^\star$	BI	ECE
`BLF` (linguistic)	—	—	**70.38**	**0.083**
Prior (market / 0.5)	—	0.00	63.08	0.117
s	$p(o_t \mid s)$	0.05	63.11	0.114
sa	$p(o_t \mid s, a_t)$	0.05	63.13	0.112
sab	$p(o_t \mid s, a_t, b_{t-1}.h)$	0.35	64.93	0.097
sah	$p(o_t \mid s, a_t, h_{t-1})$	0.00	63.08	0.117
sabh	$p(o_t \mid s, a_t, b_{t-1}.h, h_{t-1})$	0.20	64.29	0.093
σ-sab	$p(\sigma_t \mid s, a_t, b_{t-1}.h)$	0.35	65.13	0.107
σ-saσ	$p(\sigma_t \mid s, a_t, \sigma_{1:t-1})$	0.20	65.18	0.126
σ-sabσ	$p(\sigma_t \mid s, a_t, b_{t-1}.h, \sigma_{1:t-1})$	0.35	65.73	0.103

which conditions on the belief state plus the past (summarized) evidence. This is analogous to the non-Markovian version of BLF, and gets BI=65.73. Interestingly, this is slightly better than just conditioning on the past without the belief state, $q_t(s) = p(\sigma_t \mid s, a_t, \sigma_{1:t-1})$, which gets BI=65.18, or just conditioning on the belief state alone, $q_t(s) = p(\sigma_t \mid s, a_t, b_{t-1}.h)$, which gets BI=65.13. However, all of these methods are worse than BLF, which gets BI=70.38, and ECE=0.083.

7 Related work[6]

Retrieval-augmented forecasting systems combine search, reasoning, aggregation, and calibration (Halawi et al., 2024; Hsieh et al., 2024; Alur et al., 2025). BLF follows the sequential reasoning-and-action pattern of ReAct (Yao et al., 2023); its belief representation is a structured running summary with a probability, not a new exact Bayesian inference algorithm. Our contribution is a careful empirical study of this representation. BayesAgent (Huang et al., 2026) instead constructs explicit graphical models; likelihood-based proposals (Papamarkou et al., 2026; Amin, 2026) motivate our negative results in Sec. 6. Model training for forecasting (Turtel et al., 2026; Chandak et al., 2026; Jeen et al., 2026b) is complementary to these changes to the agent harness.

8 Conclusion

We studied structured linguistic beliefs as a form of context engineering for search-enabled forecasting agents. BLF has the highest overall BI point estimate among the compared external agents on 400 ForecastBench question-date instances.

Limitations and future work. Our evaluation is restricted to binary outcomes and backtesting; it does not establish live forecasting performance or complete cost-normalized superiority. Cutoff filters reduce, but cannot eliminate, leakage risk. The memory comparison is exploratory and excludes incomplete paired submissions. Future work should test non-binary outcomes, improve numerical tools, train models for calibration and evidence integration, and validate the system in live forecasting.

[6]For more related work, see Sec. H.

AI USE STATEMENT

In this work, we used generative AI tools to help write the code, launch and analyse experiments, and help with writing and figure preparation.

ETHICS

Forecasting systems can have sensitive uses, including geopolitical forecasting, so we recommend caution and human oversight.

REPRODUCIBILITY STATEMENT

Code and data will be released on GitHub upon publication.

REFERENCES

Yogesh Agrawal, Aniruddha Dutta, Md Mahadi Hasan, Santu Karmaker, and Aritra Dutta. FinTradeBench: A financial reasoning benchmark for LLMs. *arxiv*, 2026. URL https://arxiv.org/abs/2603.19225.

Md Atik Ahamed, Mihir Parmar, Palash Goyal, Yiwen Song, Long T. Le, Qiang Cheng, Chun-Liang Li, Hamid Palangi, Jinsung Yoon, and Tomas Pfister. TFRBench: A reasoning benchmark for evaluating forecasting systems. *arxiv*, 2026. URL https://arxiv.org/abs/2604.05364.

Matthew Aitchison, Scott Jeen, Toby Shevlane, and Ben Day. Diversity is the strength of the AI crowd. In *ICML Workshop on Forecasting*, June 2026. URL http://dx.doi.org/10.48550/arXiv.2606.29661.

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. *arxiv*, 2025. URL https://arxiv.org/abs/2511.07678.

Danial Amin. Bayesian orchestration of multi-LLM agents for cost-aware sequential decision-making. *arXiv [cs.AI]*, January 2026. URL http://dx.doi.org/10.48550/arXiv.2601.01522.

Anthropic. Web search tool. https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/web-search-tool, 2025.

Ilze Amanda Auzina, Joschka Strüber, Sergio Hernández-Gutiérrez, Shashwat Goel, Ameya Prabhu, and Matthias Bethge. Intrinsic credit assignment for long horizon interaction. *arXiv [cs.LG]*, February 2026. URL http://dx.doi.org/10.48550/arXiv.2602.12342.

Houtan Bastani, Simas Kucinskas, and Ezra Karger. How well can large language models predict the future?, 2025. URL https://forecastingresearch.substack.com/p/ai-llm-forecasting-model-forecastbench-benchmark.

Drew Breunig. How long contexts fail, June 2025. URL https://www.dbreunig.com/2025/06/22/how-contexts-fail-and-how-to-fix-them.html.

Cassi AI. Cassi: AI-powered forecasting. https://cassi-ai.com/, 2025.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Scaling open-ended reasoning to predict the future. *arxiv*, 2026. URL https://arxiv.org/abs/2512.25070.

Pu Cheng, Juncheng Liu, and Yunshen Long. PolyBench: Benchmarking LLM forecasting and trading capabilities on live prediction market data. *arxiv*, 2026. URL https://arxiv.org/abs/2604.14199.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *ICLR*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

Bradley Efron and Carl Morris. Stein's estimation rule and its competitors—an empirical Bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

Fabian Falck, Ziyu Wang, and Chris Holmes. Is in-context learning in large language models bayesian? a martingale perspective. In *ICML*, June 2024. URL https://arxiv.org/abs/2406.00793.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N. Angelopoulos, and Ion Stoica. Prompt-to-leaderboard: Prompt-adaptive LLM evaluations, 2025. URL https://arxiv.org/abs/2502.14855.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *J. Amer. Statist. Assoc*, 102:359–378, 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. *arXiv [cs.LG]*, May 2026. URL http://dx.doi.org/10.48550/arXiv.2605.15188.

Google. Grounding with Google search. https://ai.google.dev/gemini-api/docs/grounding, 2025.

Yong Guan, Hao Peng, Xiaozhi Wang, Lei Hou, and Juanzi Li. OpenEP: Open-ended future event prediction. *arxiv*, 2024. URL https://arxiv.org/abs/2408.06578.

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

Elvis Hsieh, Preston Fu, and Jonathan Chen. Reasoning and tools for human-level forecasting. *arxiv*, 2024. URL https://arxiv.org/abs/2408.12036.

Hengguan Huang, Xing Shen, Songtao Wang, Lingfa Meng, Dianbo Liu, David Alejandro Duchene, Hao Wang, and Samir Bhatt. BayesAgent: Bayesian agentic reasoning under uncertainty via verbalized probabilistic graphical modeling. In *AAAI*, 2026. URL https://arxiv.org/abs/2406.05516.

Scott Jeen, Matthew Aitchison, Maximilian Anthony Hugh Clark, Toby Shevlane, and Ben Day. Reaching the frontier of AI forecasting with reinforcement learning. In *ICML Workshop on Forecasting as a New Frontier of Intelligence*, June 2026a. URL https://openreview.net/forum?id=lbpDR9pj5F.

Scott Jeen, Matthew Aitchison, and Mantic. Training LLMs to predict world events. *Thinking Machines Lab: News*, 2026b. URL https://thinkingmachines.ai/news/training-llms-to-predict-world-events/.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip Tetlock. ForecastBench: A dynamic benchmark of AI forecasting capabilities. In *ICLR*, 2025.

Chinmay Karkar and Paras Chopra. Future is unevenly distributed: Forecasting ability of LLMs depends on what we're asking. *arxiv*, 2025. URL https://arxiv.org/abs/2511.18394.

Simas Kucinskas, Houtan Bastani, and Ezra Karger. ForecastBench: An updated ranking methodology, 2025. URL https://forecastbench.org/assets/pdfs/forecastbench_updated_methodology.pdf.

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

Michael Lawrence, Paul Goodwin, Marcus O'Connor, and Dilek Önkal. Judgmental forecasting: A review of progress over the last 25 years. *International Journal of Forecasting*, 22(3):493–518, 2006.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-End Optimization of Model Harnesses. *arxiv*, 2026. URL https://arxiv.org/abs/2603.28052.

Bojie Li. Incompressible Knowledge Probes: Estimating Black-Box LLM Parameter Counts via Factual Capacity. *arXiv preprint arXiv:2604.24827*, 2026.

Zehan Li, Yuxuan Wang, Ali El Lahib, Ying-Jieh Xia, and Xinyu Pi. Simulated ignorance fails: A systematic study of LLM behaviors on forecasting problems before model knowledge cutoff. *arxiv*, 2026. URL https://arxiv.org/abs/2601.13717.

Lightning Rod Labs. Foresight-32b: An LLM-based forecasting system, 2025. URL https://blog.lightningrod.ai/p/using-the-future-to-train-prediction-models.

Tom Liptay, Dan Schwarz, Rafael Poyiadzi, Jack Wildman, and Nikos I Bosse. Evaluating strategic reasoning in forecasting agents. *arXiv [cs.AI]*, April 2026. URL http://dx.doi.org/10.48550/arXiv.2604.26106.

Jiashuo Liu, Siyuan Chen, Zaiyuan Wang, Zhiyuan Zeng, Jiacheng Guo, Liang Hu, Lingyue Yin, Suozhi Huang, Wenxin Hao, Yang Yang, Zerui Cheng, Zixin Yao, Lingyue Yin, Haoxin Liu, Jiayi Cheng, Yuzhen Li, Zezhong Ma, Bingjie Wang, Bingsen Qiu, Xiao Liu, Zeyang Zhang, Zijian Liu, Jinpeng Wang, Mingren Yin, Tianci He, Yali Liao, Yixiao Tian, Zhenwei Zhu, Anqi Dai, Ge Zhang, Jingkai Liu, Kaiyuan Zhang, Wenlong Wu, Xiang Gao, Xinjie Chen, Zhixin Yao, Zhoufutu Wen, B Aditya Prakash, Jose Blanchet, Mengdi Wang, Nian Si, and Wenhao Huang. FutureX-pro: Extending future prediction to high-value vertical domains. *arXiv [cs.AI]*, January 2026a. URL http://dx.doi.org/10.48550/arXiv.2601.12259.

Mingyang Liu, Qingcan Kang, Yuke Wang, Shixiong Kai, Kaichao Liang, Hui-Ling Zhen, Tao Zhong, Mingxuan Yuan, and Linqi Song. From long news to accurate forecast: Importance-aware fusion and PRM-guided reflection for time series forecasting. *arXiv [cs.AI]*, June 2026b. URL http://dx.doi.org/10.48550/arXiv.2606.03097.

Zijia Liu, Peixuan Han, Haofei Yu, Haoru Li, and Jiaxuan You. Time-R1: Towards comprehensive temporal reasoning in LLMs. *arxiv*, 2025. URL https://arxiv.org/abs/2505.13508.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. AutoHarness: improving LLM agents by automatically synthesizing a code harness. *arxiv*, 2026. URL https://arxiv.org/abs/2603.03329.

Metaculus. FutureEval: Continuously updated AI forecasting benchmark, 2026. URL https://www.metaculus.com/futureeval/.

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

Theodore Papamarkou, Pierre Alquier, Matthias Bauer, Wray Buntine, Andrew Davison, Gintare Karolina Dziugaite, Maurizio Filippone, Andrew Y K Foong, Vincent Fortuin, Dimitris Fouskakis, Jes Frellsen, Eyke Hüllermeier, Theofanis Karaletsos, Mohammad Emtiyaz Khan, Nikita Kotelevskii, Salem Lahlou, Yingzhen Li, Fang Liu, Clare Lyle, Thomas Möllenhoff, Konstantina Palla, Maxim Panov, Yusuf Sale, Kajetan Schweighofer, Artem Shelmanov, Siddharth Swaroop, Martin Trapp, Willem Waegeman, Andrew Gordon Wilson, and Alexey Zaytsev. Position: agentic AI orchestration should be bayes-consistent. In *ICML*, May 2026. URL http://dx.doi.org/10.48550/arXiv.2605.00742.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.

Sarah Pratt, Seth Blumberg, Pietro Kreitlon Carolino, and Meredith Ringel Morris. Can language models use forecasting strategies? *arxiv*, 2024. URL https://arxiv.org/abs/2406.04446.

Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Bayesian teaching enables probabilistic reasoning in large language models. *Nat. Commun.*, March 2025. URL http://arxiv.org/abs/2503.17523.

Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, and Philip E. Tetlock. Wisdom of the silicon crowd: LLM ensemble prediction capabilities rival human crowd accuracy. *arxiv*, 2024. URL https://arxiv.org/abs/2402.19379.

Philipp Schoenegger, Cameron R. Jones, Philip E. Tetlock, and Barbara Mellers. Prompt engineering large language models' forecasting capabilities. *arxiv*, 2025. URL https://arxiv.org/abs/2506.01578.

David Spiegelhalter. *The Art of Uncertainty: How to Navigate Chance, Ignorance, Risk and Luck*. W.W. Norton, 2025.

Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pp. 197–206, 1956.

Philip E. Tetlock and Dan Gardner. *Superforecasting: The Art and Science of Prediction*. Crown, 2015.

Benjamin Turtel, Danny Franklin, Kris Skotheim, Luke Hewitt, and Philipp Schoenegger. Outcome-based reinforcement learning to predict the future. *arxiv*, 2025. URL https://arxiv.org/abs/2505.17989.

Benjamin Turtel, Paul Wilczewski, Danny Franklin, and Kris Skothiem. Future-as-label: Scalable supervision from real-world outcomes. *arxiv*, 2026. URL https://arxiv.org/abs/2601.06336.

Kartik Waghmare and Johanna Ziegel. Proper scoring rules for estimation and forecast evaluation. *arxiv*, 2025. URL https://arxiv.org/abs/2504.01781.

Yishu Wang, Yuxuan Wang, Jiaqi Deng, and Hanyang Tang. Beyond forecasting: The belief-to-trade layer in prediction-market agents. *arXiv [cs.AI]*, July 2026. URL http://dx.doi.org/10.48550/arXiv.2607.03015.

Zhen Wang, Xi Zhou, Yating Yang, Bo Ma, Lei Wang, Rui Dong, and Azmat Anwar. Beyond inherent cognition biases in LLM-based event forecasting: A multi-cognition agentic framework. In *Findings of EMNLP*, 2025. URL https://aclanthology.org/2025.findings-emnlp.258/.

Qingchuan Yang, Simon Mahns, Sida Li, Anri Gu, Jibang Wu, and Haifeng Xu. LLM-as-a-prophet: Understanding predictive intelligence with prophet arena. *arxiv*, 2025. URL https://arxiv.org/abs/2510.17638.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023. URL http://dx.doi.org/10.48550/arXiv.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, Jose Blanchet, Xipeng Qiu, Mengdi Wang, and Wenhao Huang. FutureX: An advanced live benchmark for LLM agents in future prediction. *arxiv*, 2025. URL https://arxiv.org/abs/2508.11987.

Jaden Zhang, Gardenia Liu, Oliver Johansson, Hileamlak Yitayew, Kamryn Ohly, and Grace Li. Prediction arena: Benchmarking AI models on real-world prediction markets. *arxiv*, 2026. URL https://arxiv.org/abs/2604.07355.

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 *NeurIPS (Datasets and Benchmarks)*, 2022. URL https://arxiv.org/abs/2206.15474.

A Experimental Setup Details

A.1 Datasets

We use two binary-question benchmarks: AIBQ2 (113 questions from the Metaculus AI Benchmark Tournament, Q2 2025) and ForecastBench (FB, ∼500 questions per fortnightly round). Full descriptions, exam construction, base rates, and a representativeness analysis are in Sec. F.

A.2 Metrics

We use multiple metrics to evaluate performance, as defined below.

- **Metaculus Baseline Score**: $\text{MS} = 100 \log_2(p_o/\pi_0) = 100(\log_2 p_o + 1)$ where p is the predicted probability of a YES outcome, $p_o = p$ if $o = 1$, $p_o = 1-p$ if $o = 0$, and $\pi_0 = 0.5$ is the uniform baseline.[7] A perfect prediction ($p_o = 1$) gives MS $= 100$, while the baseline ($\pi_0 = 0.5$) gives MS $= 0$. MS punishes incorrect predictions in the tails more severely than Brier score. Higher is better.
- **Brier Score**: $\text{BS} = (p - o)^2$, where p is the forecast and $o \in \{0, 1\}$ the outcome. Lower is better (despite the use of the word "score"), and predicting the uniform distribution of $p = 0.5$ scores 0.25. BS is unscaled unless a table explicitly labels BS×100 (so 0.25 becomes 25).
- **Brier Index**: $\text{BI} = 100 \times (1 - \sqrt{\overline{\text{BS}}})$, where $\overline{\text{BS}} = \frac{1}{N}\sum_i (p_i - o_i)^2$ is the mean Brier score across N predictions. Higher is better; always-0.5 scores 50%. Note the square root is applied *after* averaging, not per question. See (Kucinskas et al., 2026) for details. Within a common evaluation split, BI and mean BS give equivalent rankings. Averaging Market and Data BI need not preserve the ranking of the equally weighted mean of their BS values.
- **Adjusted Brier Score**: $\text{ABS}_i = \text{BS}_i - \gamma_i + \bar{\gamma}$ is the difficulty adjusted Brier score for question i, where γ_i is an estimate of question i's difficulty, and $\bar{\gamma}$ is the reference mean difficulty used to center the adjustment (Bastani et al., 2025; Kucinskas et al., 2025). For market questions, $\gamma_i = (m_i - o_i)^2$ where m_i is the market price (crowd estimate). For dataset questions, γ_i is a fixed effect estimated across all methods on the ForecastBench.[8] ABS controls for question difficulty, so a method that does well on hard questions scores better than one that only does well on easy questions. ABS was used as the official metric on the FB leaderboard until March 2026, when they switched to adjusted Brier index (see below).
- **Adjusted Brier Index**: $\text{ABI} = 100 \times \left(1 - \sqrt{\max(0, \overline{\text{ABS}})}\right)$, where $\overline{\text{ABS}} = \frac{1}{N}\sum_i \text{ABS}_i$ is the mean adjusted Brier score across N predictions. As of March 2026, this is the official metric used by the FB leaderboard (Kucinskas et al., 2026).[9]

MS and BS metrics are proper scoring rules, which reward well-calibrated forecasts (see e.g., (Gneiting & Raftery, 2007; Waghmare & Ziegel, 2025)). Technically speaking BI is not proper, but it converges to properness as the sample size increases (see (Kucinskas et al., 2026) for details). (Adjusted versions do not affect whether a metric is proper or not.)

[7] See https://www.metaculus.com/help/scores-faq/#baseline-score for the full derivation. Metaculus also defines a **Metaculus Peer Score (MPS)** that measures performance relative to other forecasters rather than a uniform baseline: $\text{MPS} = 100 \times (\ln(p) - \ln(\text{GM}(q_i)))$, where $\text{GM}(q_i)$ is the geometric mean of all other predictions for the same question. (See https://www.metaculus.com/help/scores-faq/#peer-score.) We do not use MPS in this paper, but it could be approximated by using the crowd estimate as the reference prediction (cf. the adjusted Brier score below).

[8] The γ_i values are available at https://www.forecastbench.org/datasets/. However, relying on γ_i means this metric cannot be used for novel datasets. In Sec. E.2, we use a paired analysis to achieve a similar effect in a more general way.

[9] Per Eq. (2) of Kucinskas et al. (2025), ABS is rescaled so that the always-0.5 forecaster scores 0.25 (matching raw BS), but $\text{ABS}_i = b_i - \hat{\gamma}_j + \bar{\hat{\gamma}}$ can still be negative for individual questions (whenever the forecaster out-performs the difficulty estimate $\hat{\gamma}_j$ on question j), and the mean $\overline{\text{ABS}}$ can in principle dip below zero for a forecaster who consistently beats the difficulty estimate across its evaluation subset. The $\max(0, \cdot)$ clamp keeps the index well-defined; in practice all top FB-leaderboard methods (and all methods reported in this paper) have positive $\overline{\text{ABS}}$, so the clamp is inert.

A.3 External methods we compare to

The historical leaderboard snapshot of https://www.forecastbench.org/datasets/ from 2026-04-20 is shown in Fig. 5. The term 'zero-shot" denotes a single probability response without search or tools, it does not mean they do not use the crowd prior; we omit all such zero-shot methods, and just retain the following search-based agents:

- **Cassi** (ensemble+crowd-adj): ABI=67.9; both tranches A and B.
- **Grok 4.20**: ABI=67.8; tranche A only.
- **Foresight-32B**: ABI=67.2; tranche B only.

Rank	Team	Org	Model	Dataset (N)	Dataset 95% CI	Market (N)	Market 95% CI	Overall (N)	Overall 95% CI	Supers > Forecaster?	Forecaster > Public?
1			Superforecaster median forecast	63.9 (521)	[62.6, 65.2]	80.3 (56)	[76.3, 85.8]	70.9 (577)	[69.2, 72.7]	—	Yes
2			Cassi ensemble_2_crowdadj	61.5 (1,421)	[60.8, 62.2]	75.8 (200)	[73.5, 78.7]	67.9 (1,621)	[66.9, 69.0]	Yes	Yes
3			Gemini-3-Pro-Preview (zero shot with crowd forecast)	61.5 (2,163)	[60.9, 62.1]	75.6 (281)	[74.0, 77.2]	67.8 (2,444)	[67.0, 68.4]	Yes	Yes
3			Grok 4.20 (Preview)	61.8 (727)	[61.0, 62.6]	75.2 (100)	[71.2, 80.1]	67.8 (827)	[66.1, 69.5]	Yes	Likely
5			GPT-5-2025-08-07 (zero shot with crowd forecast)	61.3 (2,548)	[60.8, 61.8]	74.5 (282)	[72.3, 76.8]	67.2 (2,830)	[66.3, 68.1]	Yes	Likely
5			Foresight-32B	61.1 (694)	[60.0, 62.2]	74.9 (100)	[70.3, 79.8]	67.2 (794)	[65.3, 69.2]	Yes	Likely

Figure 5: Screenshot of the ForecastBench tournament leaderboard (https://www.forecastbench.org/leaderboards/#tournament) as of 2026-04-20, showing the top 5 methods. The metric is Adjusted Brier Index. Note that each method's ABI is computed on a different set of resolved questions, so direct comparison of leaderboard scores is approximate; our paired analysis (Table 1) provides rigorous comparisons on shared questions.

A.4 LLMs used

Tab. 5 lists every LLM used in this paper, with provider, version, knowledge-cutoff date, maximum context, an order-of-magnitude effective parameter-count estimate (closed proprietary models from Li, 2026; open models from official model cards), and per-token cost (USD per million tokens, OpenRouter pricing as of September 25, 2026). Knowledge-cutoff and context-window values come from the respective providers; cost figures are subject to change.

Table 5: LLMs used in this paper. Parameter estimates for closed models are from Li (2026). Cost estimates: USD per million input/output tokens via OpenRouter or direct provider. Input tokens can be cheaper when there is a cache hit. Cutoff dates are provider-reported knowledge cutoffs.

LLM (alias)	Full version	Cutoff	Context	Params	Cost ($/M in / $/M out)
Main agent					
Pro	gemini-3.1-pro-preview	Jan 2025	1M	?	$2 / $12
Flash	gemini-3-flash-preview	Jan 2025	1M	?	$0.50 / $3
GPT-5	gpt-5.4	Aug 2025	1M	∼2T	$2.50 / $15
Auxiliary roles (search filtering, summarization, leakage audit)					
Flash	gemini-3-flash-preview	Jan 2025	1M	?	$0.50 / $3
Grok-4.1-fast	x-ai/grok-4.1-fast	May 2025	2M	?	$0.20 / $0.50

A.5 Compute requirements

Our system uses 3 LLMs: for the main agent, for search results date filtering, and for search results summarization. (We optionally use a 4th LLM, Grok-4.1-fast, to perform post-hoc leakage auditing, as described in Sec. B.3.) We use Gemini-3.1-Pro for most of our agent experiments (although we also experiment with other models), and use Gemini-3-Flash for the search filtering and summarization because it is much cheaper. (An agent that takes 10 steps and returns 10 search results per step requires 100 summarization LLM calls.)

We access all LLMs through OpenRouter (https://openrouter.ai/), using the LiteLLM interface (https://github.com/BerriAI/litellm), which provides a uniform way to use multiple LLM providers (e.g., Google, OpenAI, Anthropic, Moonshot). All experiments are launched from a laptop. A single question takes 1–8 minutes depending on API latency and the number of steps taken by the agent. (We limit the maximum number of steps to 10, and the maximum time to 10 minutes; if the agent has not submitted by then, we take its last estimate of $\hat{p}$.) Using 50 parallel workers, a full evaluation (5 trials on 400 questions) completes in 1–3 hours wall clock.

The agentic methods consume 50–100M tokens across 5 trials, the large majority of which are input tokens (system prompt, tool schemas, accumulated conversation). At Gemini-3.1-Pro pricing ($2/M input, $12/M output), evaluating one method on the full $A \cup B$ dataset costs approximately $250. Two factors keep this well below the naive uncached estimate ($\sim$$1000): (i) Gemini's automatic context caching reuses the stable prefix (system + tools + initial user message) across the steps of an agent loop, billing the cached portion at $\sim$25% of the base input rate, and (ii) output tokens, which carry the higher per-token rate, are a small fraction of total tokens because most steps emit short tool calls or short final answers rather than long free-text responses.

B BACKTESTING VALIDITY

A key concern with backtesting is information leakage: the LLM's parametric knowledge or web search results may contain post-cutoff information that would not be available in a live setting. (In our backtesting context, the cutoff date is set equal to the forecast due date.)

B.1 MITIGATING SEARCH LEAKAGE.

As mentioned in the main text, we adopt a four-layer defense: (1) search engine date filtering, (2) LLM-based leak classifier on results, (3) date clamping for tools, and (4) URL blocking for resolution sources. We give more details below.

Stage 1. The first layer is to add date range filtering to web search, so that we do not retrieve pages that are date stamped after the cutoff date. Specifically, we use Brave Search, which supports a `freshness` parameter to restrict results to before the cutoff date. We evaluated other engines (Perplexity, AskNews, Exa, Serper) but they either did not support reliable date filtering, or returned only snippets (not full page text), which harmed downstream performance.

Note that algorithmic date filtering at the search engine level is critical: without it, we would need to retrieve many more results knowing that the LLM-based classifier (layer 2) will discard most of them, which would be very inefficient.

Also, note that web search is controlled entirely through our own tool schemas passed via the function-calling API, so we can enforce date filtering. The core LLM could choose to "spontaneously" do search on its own without us asking it to; however, both Gemini and Anthropic models require explicit opt-in to enable built-in search (Google, 2025; Anthropic, 2025), which we avoid using.

Stage 2. The second layer is to use an LLM classifier to decide if any results that pass the first stage are after the cutoff date based on their content (not meta data). We use Gemini-3-Flash for this, since it is fast and cheap and reasonably reliable. This stage is necessary because a document might have a time stamp that is before the cutoff date, and hence pass the stage 1 filter, but it may have been updated and contain information from after the cutoff.

Stage 3. The tools we use, such as the time series tools discussed in Sec. C.4, all support date filtering which enforces the requested observation-date cutoff. This alone does not guarantee historical data-vintage correctness for sources that revise past values. (For example, we can retrieve yfinance history up to the cutoff date, or retrieve a Wikipedia snapshotted page from before the cutoff.)

Stage 4. Finally, since sometimes the question or its resolution criteria contain a URL which might have the resolved outcome (e.g. `https://manifold.markets/AlexLiesman/will-ding-liren-repeat-as-world-che`), we block access to such URLs for both web search and the lookup-url tool.

B.2 MITIGATING PARAMETRIC LEAKAGE.

Besides leakage from web search, the LLM will contain certain forms of knowledge (including potentially the true outcome of a given forecasting question) up until its knowledge cutoff date; the amount of memorized world knowledge scales roughly log-linearly with effective parameter count (Li, 2026), so larger frontier models are at higher parametric-leakage risk. To avoid this risk, we only use models whose cutoff date is before the date range of the questions we test on.[10]

B.3 POST-HOC LEAKAGE AUDIT.

To assess the effectiveness of our leakage mitigation techniques, we run a separate "leak detective" that re-classifies all saved search results and URL lookups conducted by the agent, as well as

[10] An extreme example would be the "Talkie" model from `https://talkie-lm.com/introducing-talkie`, which was only trained on data up until 1930. However, this is a very small model that is not good at reasoning or tool use, so is not capabable of reliable forecasting.

examining the reasoning traces. For the detective, we use Grok-4.1-Fast. This is fast and cheap, but somewhat uncorrelated with the runtime filter (Gemini-3-Flash).

Search leakage analysis After classifying all the search results from our `BLF` applied to tranche A of FB, we produced the confusion matrix shown in Table 6. We see that the runtime filter has 93.8% recall (catches 320 of 341 leaks) but also drops 577 clean results (29.9% false positive rate). Crucially, of the 1,377 results the agent actually sees (the "kept" column), only 21 are leaks — an undetected flag rate of 1.5% among retained results. These are second-LLM verdicts, not human-verified ground truth. At the question level, at least one result was flagged for 14 of 76 web-search questions (18.4%), or 14 of all 200 tranche-A questions (7.0%). These exposure rates do not directly measure how much the flagged evidence affected forecasts. However, if this is a problem, we could use a stronger classifier to reduce FPs even further, while maintaining the high leak detection rate.

Table 6: **Leakage audit confusion matrix** ($n = 2{,}272$ search results across 9 sources from Tranche A). Rows: post-hoc verdict from a second LLM classifier (Grok-4.1). Columns: runtime filter decision (Gemini-3-Flash).

	Filter: Kept	Filter: Dropped
Post-hoc: Clean	1,354	577
Post-hoc: Leak	21	320

Parametric leakage analysis To assess parametric leakage, we scan the LLM's reasoning traces (including chain-of-thought) for mentions of post-cutoff events, dates, or outcomes that could not have been known at the forecast date. The retained backbones' advertised knowledge cutoffs precede the ForecastBench forecast dates. This is a selection criterion, not proof that their post-training data contain no later information. An absence of explicit future facts in a reasoning trace likewise cannot rule out parametric leakage.

Error analysis The runtime filter (Gemini-3-Flash) and post-hoc detective (Grok-4.1) are intentionally different models to reduce correlated errors. However, both make mistakes, as we now discuss.

One kind of error is when Flash is too aggressive. For example, a search result about historical weather patterns in Dijon (from https://www.weather2visit.com/europe/france/dijon-june.htm) mentioned "June 2025" in a generic seasonal context (e.g. "What's the weather like in Dijon in June 2025?"). Flash wrongly classified this as a leak ("describes the weather in June 2025 which is after the March 2025 cutoff"), but Grok correctly recognized it as generic climatological information, not a post-cutoff observation.

Another kind of error is when Grok is too aggressive. For example, a search result about the Ethiopia conflict (from https://acleddata.com/update/ethiopia-weekly-update-13-august-2024) contained a link titled "Ethiopia situation update (30 April 2025)" while the cutoff was April 27, 2025. Flash kept the result because the *content* of the page did not include post-cutoff information — only the link title referenced a future date. However, Grok wrongly flagged this as a leak.

These examples illustrate the fundamental difficulty of date-leakage classification: it requires understanding whether a date reference describes a *specific observed event* (leak) or a *generic/scheduled reference* (not a leak). We could get better results if we use a more powerful classifier, but this would also be more expensive, so we choose not to do so.

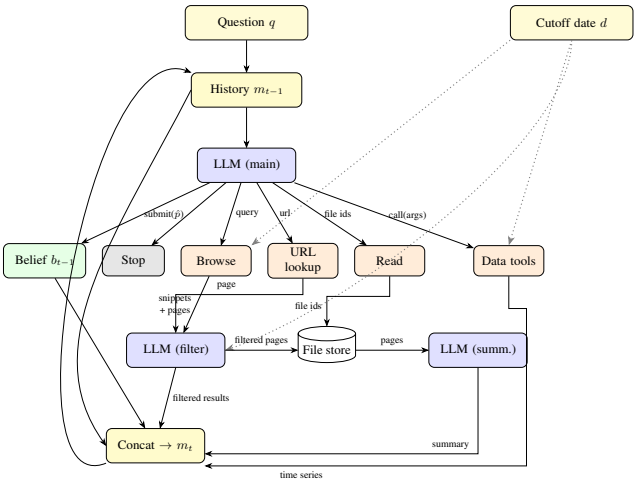

Figure 6: **BLF agent loop**. At each step, the LLM reads the message history m_{t-1} and produces an action a_t and belief b_{t-1} summarizing evidence through step $t-1$. The action is executed in the environment (with cutoff-date restrictions), producing an observation o_t. The loop terminates at submit or `max_steps`.

C Method Details

C.1 Linguistic Belief State

The core innovation of our system is the *Linguistic belief state*: a semi-structured representation that the LLM maintains and updates at each step of the agent loop. The belief state consists of:

- A **probability estimate** $p \in [0, 1]$ for the binary outcome;
- A **confidence level** (low/medium/high);
- **Key evidence** for and against the outcome (natural-language summaries);
- **Open questions** the agent plans to investigate next.

At each step, the LLM receives the full message history (including the current belief state) alongside new evidence (search results, tool outputs), and produces an updated belief state, which is appended to the history for the next step. See Sec. C.2 for details.

This approach contrasts with two common alternatives: (1) *text accumulation*, where all retrieved evidence is appended to an ever-growing context[11], which may exceed the model's effective attention span; and (2) *batch-search then reason*, where multiple queries are issued in parallel, and then the model reasons about all of them all at once, without iterative (sequential) refinement of the search queries.

C.2 Agent Loop

The agent operates as an iterative tool-use loop, as shown in Figure 6 and Algorithm 1. To explain in more detail, let us define some notation: q is the question; t is the time step; a_t is the action chosen by the agent; o_t is the resulting observation from the environment; b_{t-1} is the belief state; and m_t is the (non-Markovian) message history, $m_0 = c$, $m_{t-1} = c \oplus \bigoplus_{j=1}^{t-1}(a_j, o_j, b_{j-1})$ in full-BLF, where c contains fixed instructions, question, cutoff, and prior. The belief state b_{t-1} is a semi-structured object containing the agent's probability estimate $b_{t-1}.p \approx P(\text{outcome} = 1 \mid m_{t-1})$, as well as natural-language evidence summaries, as we discussed in Sec. C.1. Consequently, we call our agent the "Bayesian Linguistic Forecaster" or **BLF**.

[11]For examples of this kind of method, See e.g., https://github.com/ai-prophet/mini-prophet/ (public) and (Alur et al., 2025) (private).

At each step, the LLM produces both an action a_t and an updated belief state b_{t-1} in a single generation: $(a_t,\, b_{t-1}) = \text{LLM}(m_{t-1})$. The belief update is embedded as a structured JSON field (`updated_belief`) within the tool call arguments, so both outputs come from one LLM call. This field includes the updated probability $b_{t-1}.p$, as well as an `update_reasoning` string in which the LLM explains why its belief changed (e.g., "page contradicts initial assessment" — see Algorithm 2 for an example trace). We have found that requiring the LLM to articulate its reasoning for each belief update encourages coherent probabilistic updating.

The action a_t is executed in the environment, returning an observation, $o_t = \text{Env}(a_t; q, d)$, where d is the cutoff date restricting data access. Finally the message history is updated deterministically by concatenation: $m_t = m_{t-1} \oplus (b_{t-1}, a_t, o_t)$.[12] The loop returns the submitted probability when the agent calls `submit`. At the step limit T, the historical timeout fallback returns $b_{T-1}.p$, which has not incorporated o_T. The controlled memory study instead reserves a final submission call within the budget (Sec. 4). Markov-BLF replaces the history update by $m_t = m_0 \oplus (b_{t-1}, a_t, o_t)$, retaining the newest observation but discarding older raw observations and reasoning. Remaining budget is supplied separately to the model in both memory conditions.

The process is structured to *resemble* sequential Bayesian updating and decision making for a POMDP, but we are not actually doing Bayesian inference in any technical sense: the update rule $(a_t, b_{t-1}) = \text{LLM}(m_{t-1})$ is an LLM forward pass with no explicit likelihood, no marginalisation, and no formal posterior — the LLM does whatever it does, and we treat the structured belief slots as a working-memory scaffold that constrains the form of the update rather than its semantics. A textbook POMDP loop would first perform a *belief update*, $b_{t-1} = f(m_{t-1})$, integrating all available information into a Markov-sufficient belief state, and then an *action selection*, $a_t = \pi(b_{t-1})$, choosing which tool to call next or whether to submit. We do not compute a Markov-sufficient belief: actions are chosen with access to the entire history m_{t-1}, and the structured belief b_{t-1} is an *additional* field of the message — a structured scratchpad derived from m_{t-1} — that we hope helps the model maintain coherent probabilistic reasoning across steps.

Algorithm 2 Simplified agentic trace for $T = 4$ steps. (See Fig. 4 for a real trace.) The web-search action returns a set of 10 snippets, as well as the corresponding page content; the full content is stored in a filing system (not shown), and the snippets are concatenated and returned to the agent. The summarize-results tool call will read the relevant files generated by a previous web search, and then summarize their contents and return the result to the agent.

$m_0 \leftarrow c$ // fixed instructions, question, cutoff, and prior
$(a_1, b_0) \leftarrow \text{LLM}(m_0)$ // a_1=(lookup-url, url); $b_0.p$=0.65 *"Seems likely based on question"*
$o_1 \leftarrow \text{Env}(a_1)$ // page text
$m_1 \leftarrow m_0 \oplus (a_1, o_1, b_0)$
$(a_2, b_1) \leftarrow \text{LLM}(m_1)$ // a_2=(search, query); $b_1.p$=0.40 *"Page contradicts initial assessment"*
$o_2 \leftarrow \text{Env}(a_2)$ // $(\text{snippet}_1, \ldots, \text{snippet}_{10})$
$m_2 \leftarrow m_1 \oplus (a_2, o_2, b_1)$
$(a_3, b_2) \leftarrow \text{LLM}(m_2)$ // a_3=(summarize, $[3, 6, 9]$); $b_2.p$=0.35 *"Sources confirm opposition"*
$o_3 \leftarrow \text{Env}(a_3)$ // $(\text{text}_3, \text{text}_6, \text{text}_9)$
$m_3 \leftarrow m_2 \oplus (a_3, o_3, b_2)$
$(a_4, b_3) \leftarrow \text{LLM}(m_3)$ // a_4=(submit, 0.30); $b_3.p$=0.30 *"Confident: evidence is clear"*

C.3 Web search tools

The agent can issue web search queries via an API. We call this tool `browse-web` (query). For backtesting, it is crucial that search support date filtering, to avoid leakage. We have found that the Brave search engine, which supports a `freshness` parameter to restrict results to before the cutoff date, to be quite reliable, although we still need the second-stage content-based date filter discussed in Sec. 2. We briefly evaluated other search engines (Perplexity, AskNews, Exa, Serper, Tavily) but they all had one or more of the following problems: unreliable (or non-existent) date filtering; only returned snippets, not full page text; expensive. It might be interesting to explore these other search

[12] This process is thus amenable to KV-caching, which can significantly reduce costs. Gemini and OpenAI cache stable prefixes automatically, with no client-side action required.

Table 7: List of all the tools and their arguments. Here q denotes the question, which contains information about its source s_q, forecast date f_q and resolution date(s) r_q, which are needed by certain tools. The first letter is bolded, to represent the short-hand we use to describe each tool.

Type	Tools
Web search	**b**: `browse-web` (query), **r**: `read-files` (ids)
Time series	**h**: `history-fetch` (q), **m**: `model-fetch` (q), **c**: `combo-fetch` (q)
Other	**u**: `url-lookup` (url), **w**: `wikipedia-fetch` (url, [section]), **x**: `submit`

engines more carefully in future work. However, it is important to emphasize that `BLF` can work with any search engine.

For each search, Brave returns up to 10 short snippets by default, but we also request extra snippets to get more informative results (which we find generally helps performance). The snippets are immediately added to the context, but the full results are not, to avoid overwhelming the attention mechanism. Instead we save search results to a set of local files, `search_{i}_result_{j}.md`. We give the agent an additional tool, called `read-files` $(\text{id}_1, \ldots, \text{id}_k)$, which can load selected files into the context of a sub-LLM (we use Gemini Flash), which returns a summary to the main LLM. (The summarizer is given a prompt generated by the main LLM, so it extracts question-specific facts.) The main agent can choose which files it wants to read based on their snippets. This progressive disclosure approach — similar to how Anthropic uses skills — avoids flooding the context with noise from irrelevant results while letting the agent drill into the most promising leads.

C.4 Time-series tools

As we briefly discussed in Sec. 2, all the dataset questions in FB require the ability to do time series forecasting, i.e., predicting a value of interest at multiple fixed time points into the future and then comparing to a reference value, to create a binary prediction problem of the form

$$\hat{p}_q^r = P(Y_q(r) > v_q \mid \text{data}(\leq f)) \tag{3}$$

where q is the question, r is the resolution date, and f is the forecast date. Most of these dataset sources have publicly available APIs that allow you to retrieve historical values of the quantity of interest, which is very useful for tackling this problem. (The exception is ACLED, whose API requires special permission to use.) More precisely, these database retrieval tools return the history of values for a given question of interest q up to a given date t in the form of a CSV list:

$$\mathcal{H}(q, t; W) = \{(t_j, y_s(t_j)) : t - W \leq t_j \leq t\} \tag{4}$$

where W is the window size (which we set by hand, depending on the source).[13] For brevity, we define $\mathcal{H}_q = \mathcal{H}(q, f_q; W_{s_q})$, where f_q is the forecast date of question q and s_q is the source of question q (e.g. a FRED question). We denote the tool for fetching this history by `history-fetch`(q).

Note that $\mathcal{H}_q$ set contains $v_q = y_q(f_q)$, allowing us to use the correct reference value in our comparisons, as opposed to the potentially stale "freeze value" given in the original question. This is needed because the freeze value is of the form $v'_q = y_q(t')$, from the time $t' \leq f_q$ when the question was created. In some cases, v'_q may be quite different to v_q (see e.g., Fig. 22), so forecasters will be comparing to the wrong reference.

Although providing $\mathcal{H}_q$ to the LLM is often helpful, sometimes we find that the LLM cannot make sense of this raw numerical data. In such cases, we can optionally compute an estimate of $\hat{p}_q = \{\hat{p}_q^r : r \in \text{res-dates}(q)\}$ directly from $\mathcal{H}_q$ by using simple statistical models (see Appendix G for details). We denote the tool for computing this model-based estimate by `model-fetch`(q). We can either use this estimate directly (bypassing the LLM), or pass it back to the LLM, just like any other tool call. Currently we use $\hat{p}_q$ directly for DBnomics, since we found the LLM struggled with directly interpreting the raw data (see Tab. 17). For yfinance and FRED, we use a combined tool, `combo-fetch`(q), that returns both $\mathcal{H}_q$ and $\hat{p}_q$ in a single call (this is for efficiency reasons).

[13] We also consider more sophisticated versions of the history, that sample past data at different strides, allowing us to look back further without creating very long traces, but we omit this for simplicity. Furthermore, we always enforce $t \leq f$, so the agent cannot cheat by looking into the future.

Table 8: Elementary actions (tools) available to each question, depending on its source type. Abbreviations are defined in Tab. 7. We define $\mathcal{T}_{\text{search}} = \{\text{b}, \text{r}, \text{u}, \text{x}\}$; this is available to all sources except DBnomics (which uses $\mathcal{T}_{\text{basic}} = \{\text{u}, \text{x}\}$, without search tools). Currently, time-series sources use the combo tool c (returning both history and model estimate); the separate h and m tools are a planned decomposition for future work.

Source s	Tools $\mathcal{T}_s$
FRED, yfinance	$\{\text{c}\} \cup \mathcal{T}_{\text{search}}$
DBnomics	$\{\text{m}\} \cup \mathcal{T}_{\text{basic}}$
Polymarket, Manifold	$\{\text{h}\} \cup \mathcal{T}_{\text{search}}$
Wikipedia	$\{\text{w}\} \cup \mathcal{T}_{\text{search}}$
RFI, Metaculus, ACLED	$\mathcal{T}_{\text{search}}$

C.5 Other tools

We now describe a few other tools we make available to the agent.

For polymarket and manifold questions, we let the agent query the history of prices for the relevant question (we limit to last 60 days, and apply a date cutoff to stop it seeing into the future). The hope is that this will let the agent decide how much to trust the market estimate. However, the results so far seem to indicate these tools are not useful beyond just using web search, perhaps because the LLM does not know how to interpret this data. We leave improving this to future work.

We also provide the `url-lookup`(url) to retrieve the contents of a specific URL. This is useful because the question text, or resolution criteria, often contain specific URLs with relevant information. However, we have to be careful that the agent does not just lookup the answer (since date filtering cannot be applied to url lookup). To combat this, we create a blacklist of sites that might contain the true outcome for the question, such as `polymarket.com`. If the agent tries to perform a `url-lookup` call with such an argument, we return an empty string and an error message (in text form). This way the agent can learn not to try this invalid action again (c.f., (Lou et al., 2026)). (Such invalid calls should not happen, since the tool descriptions specifies the constraints, but smaller models sometimes struggle to obey these instructions).

Finally, we discuss a special tool for Wikipedia questions. FB considers these dataset questions because they require the user to predict the value at multiple future resolution dates. For example, consider the question "According to Wikipedia, will Alexandra Kosteniuk have an Elo rating on [resolution dates] that's at least 1% higher than on [forecast date]?". It is potentially possible to predict this answer just using information from web search, but it is easier to do if you can retrieve the values from the data table specified in the resolution criteria (for the case of Elo ratings of chess players, the relevant data is at `https://en.wikipedia.org/wiki/FIDE_rankings`). See Fig. 24. Of course, we must avoid leakage. Fortunately the wikipedia API allows retrieval of a page at or before a given timestamp, which we set to the forecast date f. We denote this tool by `wikipedia-fetch`(url, f_q).

C.6 Policies

Table 9: Typical action sequences (traces) for different question types. Superscript $*$ denotes repetition (0 to $T-1$ times). Note that Bypass and Zero-shot have fixed traces.

Policy	Sources	Typical trace
TS + search	yfinance, FRED	$\text{c} \to \text{b}^* \to \text{x}$
Market + search	Polymarket, Manifold	$\text{h} \to \text{b}^* \to \text{x}$
Wiki + search	Wikipedia	$\text{w} \to \text{b}^* \to \text{x}$
Search-only	Metaculus, RFI, ACLED	$\text{b}^* \to \text{x}$
Bypass	DBnomics	$\text{m} \to \text{x}$
Zero-shot	(ablation)	x

The set of tools (actions) available to the agent depends on the type of question it is tackling, as shown in Tab. 8. Denote this set by $\mathcal{A}_q$. At each step t, the agent's policy picks the action

$a_t = \pi(m_{t-1}; \mathcal{A}_q, \mathcal{L})$, where m_{t-1} is the incoming message history. In our case, the policy is implemented by the LLM $\mathcal{L}$, as discussed in Sec. C.2. In addition, the LLM generates the next belief state, $(a_t, b_{t-1}) = \pi(m_{t-1}; \mathcal{A}_q, \mathcal{L})$; we can think of b_{t-1} as an internal "mental" action, as opposed to a_t which is sent to the external environment. We call the sequence of actions generated by this process a policy trace, $\tau = (a_1, \ldots, a_T)$, where $a_t \in \mathcal{A}_q$, $|\tau| \leq T$, and $a_T = x$ (the submit/stop action). See Tab. 9 for some typical traces. (Note that the agent can choose to stop at any time, but we also impose an upper bound of $T_{\max} = 10$ steps.)

Some traces do not make much sense. For example, calling `read-files` before `browse-web` has no effect, since no files have been retrieved by the search tool. In addition, one or more URLs are often provided in the resolution criteria, so it makes sense to initially perform `url-lookup`, before following up with optional `browse-web` calls, since we know that these URLs are relevant to the question, but web search may just add noise. Similarly, when a source-specific tool, such as `wikipedia-fetch` or `combo-fetch`, is available, it is usually best to call this first. We currently encourage this behavior by using source-specific prompt instructions (e.g., "call the FRED tool as your first action"), but we could enforce it more rigorously by using a meta-controller, which determines which set of actions are available at each step. We denote this by $\mathcal{A}_{q,t} = \pi_2(s_q, t)$, where π_2 is the second-level policy, indexed by source type s_q and step t. This can be viewed as a form of agentic harness (c.f., (Lou et al., 2026; Lee et al., 2026)).

Currently we use a very primitive form of meta-controller. For DBnomics questions, we set $\mathcal{A}_{q,1} = \{m\}$ and $\mathcal{A}_{q,2} = \{x\}$; this forces the agent to first call the domain-specific modeling tool and then submit its (model-based) estimate, skipping all further steps. For all other question types, we use $\mathcal{A}_{q,t} = \mathcal{A}_q = \mathcal{T}_{s_q}$, so the base LLM can choose any action it wants in any order, limited only by the set of tools applicable to that source. However, since π_2 depends only on (s_q, t), it can be represented as a simple lookup table, which could be easy to learn, e.g., via offline cross-validation on backtesting data, or online via a bandit algorithm. For questions that do not come from a well-defined source (i.e., for anything beyond the FB dataset), we can replace s_q with c_q, which is class label that we estimate for the question. We leave learning the meta-controller to future work.

C.7 Crowd estimate and empirical priors

We optionally provide the LLM with an informative starting estimate π_q, controlled by two independent flags:

Crowd signal (`crowd=0/1`). For market questions, ForecastBench provides a crowd estimate $\pi_q = m_q$ known as the "freeze value": this is the market price (Polymarket, Manifold, RFI) or Community Prediction (Metaculus). This is the most powerful single input for market questions: simply predicting $\hat{p}_q = m_q$ achieves BI ≈ 81.5 (Table 1). The crowd flag has no effect on dataset questions (no market price exists).

Empirical prior (`emp=0/1`). For dataset questions, no crowd estimate exists. Instead, we use the *empirical base rate* for each source and question subtype as π_q. This is computed using all data prior to the question being asked, approximating what an online learner would quickly converge to. Table 10 shows the priors. The emp flag has no effect on market questions (which use the crowd signal instead).

Table 10: Empirical priors π_q for dataset questions, computed from all ForecastBench questions. The different question subtypes are explained in Sec. F.

Source	Question subtype	Prior π_q
ACLED	10× spike	0.00
ACLED	Any increase	0.23
Wikipedia	Vaccine	0.00
Wikipedia	FIDE Elo ≥1%	0.01
Wikipedia	FIDE rank	0.68
Wikipedia	Swimming WR	0.99
FRED	(all)	0.42
yfinance	(all)	0.58
DBnomics	(all)	0.56

These priors are strong baselines for sources with extreme base rates: for example, predicting $\pi_q = 0$ for ACLED 10× spike questions (Sec. F.6) or Wikipedia vaccine questions (Sec. F.7) is nearly optimal. On dataset questions, simply submitting these priors achieves BI of 58.3; our full system (`BLF`) improves this to 62.7 (see Table 1). Note that the crowd and emp signals are not always available (e.g., AIBQ2 has neither market prices nor empirical priors), so we also evaluate without them (`crowd=0, emp=0`).

C.8 Multi-Trial Aggregation

We run $K = 5$ independent trials per question, each producing a forecast $p_k \in (0, 1)$. Sometimes these can differ quite a lot, based on the particular search queries performed during each trial (see Fig. 3).

The simplest aggregation method is the arithmetic mean of the probabilites:

$$\hat{p} = \frac{1}{K} \sum_{k=1}^{K} p_k \tag{5}$$

We now prove that this improves all three metrics (BS, BI, and the log-based MS) in expectation as K increases. Consider K forecasts $p_1, \ldots, p_K$ for a question with outcome o. By Jensen's inequality applied to the convex function $f(e) = e^2$:

$$\mathrm{BS}(\bar{p}) = (\bar{p} - o)^2 \leq \frac{1}{K} \sum_{k} (p_k - o)^2 = \overline{\mathrm{BS}(\{p_k\})}, \tag{6}$$

i.e. scoring the arithmetic-mean forecast is no worse than the average per-trial Brier score; the same statement holds for the log-based MS because $-\log p$ is also convex in p, so Jensen gives $\mathrm{MS}(\bar{p}) \geq \overline{\mathrm{MS}(p_k)}$ in the same fashion. Note this does *not* say that averaging beats every individual trial — a single trial can outperform $\bar{p}$ by chance — only that it beats the *mean* of the per-trial scores in expectation, which is the relevant comparison when we view the trials as exchangeable draws. Since $\mathrm{BI} = 100(1 - \sqrt{\overline{\mathrm{BS}}})$ is a monotone decreasing function of $\overline{\mathrm{BS}}$, arithmetic averaging also improves BI. Concretely, if averaging across K trials reduces $\overline{\mathrm{BS}}$ from B_1 to $B_K \leq B_1$, then

$$\mathrm{BI}_K = 100(1 - \sqrt{B_K}) \geq 100(1 - \sqrt{B_1}) = \mathrm{BI}_1. \tag{7}$$

The BI improvement with more trials is confirmed empirically (see Fig. 2).

Logit averaging is another option, but does not consistently outperform arithmetic averaging in Fig. 2. This is defined as follows:

$$\hat{p} = \mathrm{sigmoid}\left(\frac{1}{K} \sum_{k=1}^{K} \mathrm{logit}(p_k) \right) \tag{8}$$

(Note that this is a special case of the shrinkage estimator in Sec. C.9.) Unfortunately, the Jensen based proof that avergaging is better does not apply to logit-space averaging. In particular, the convexity used above is convexity of $f(e) = e^2$ as a function of the *probability-space* error $e = p - o$, which justifies the arithmetic mean $\bar{p} = \frac{1}{K} \sum_k p_k$. The Brier score viewed as a function of the logit $y = \mathrm{logit}(p)$, namely $\mathrm{BS}_y(y) = (\sigma(y) - o)^2$ where σ is the sigmoid, is *not* globally convex in y: it has an inflection where the sigmoid passes through its steepest slope. Jensen's inequality therefore does *not* directly imply that the logit-space mean reduces BS or BI relative to the arithmetic mean. Our preference for logit-space averaging on FB is therefore empirical, not formal.

We attribute the empirical benefits of logit space averaging to two practical effects. First, the logit mean preserves extremity better, producing predictions further from 0.5 when individual trials are concordant. Second, logit-space averaging degrades gracefully when an individual trial fails (e.g., due to network timeout). Such failures fall back to $\hat{p}_k = 0.5$, with $\mathrm{logit}(0.5) = 0$, which is the prior on the logit scale — so a dead trial pulls the logit mean only by $-\bar{y}/K$ where $\bar{y}$ is the surviving-trial mean logit, rather than by a fixed prior-distance proportional to $(p - 0.5)/K$ in the arithmetic case. For a confident prediction $p \approx 0.95$, one dead trial in five costs the arithmetic mean $\sim$9 percentage points but the logit mean only $\sim$4.

C.9 SHRINKAGE AGGREGATION

We also explored *hierarchical shrinkage* as a way to aggregate results. This is inspired by James–Stein estimation (Stein, 1956; Efron & Morris, 1973), which is a form of empirical Bayes estimation for a certain hierarchical model, as we explain below.

Model. We model the K trial forecasts for a question q in logit space. Let $y_{qk} = \text{logit}(p_{qk})$ and let θ_q denote the "true" logit-forecast for the question. The likelihood is

$$y_{qk} \mid \theta \sim \mathcal{N}(\theta_q, \sigma_q^2), \quad k = 1, \ldots, K, \tag{9}$$

where σ_q^2 is estimated as the sample variance of the logit-transformed forecasts. We place a Gaussian prior on θ_q centered at the *question-specific* logit-prior $\mu_q = \text{logit}(\pi_q)$,

$$\theta_q \sim \mathcal{N}(\mu_q,\ \tau^2), \tag{10}$$

where π_q is the crowd estimate (for market questions where it is available) or the empirical/base-rate prior (for dataset questions; Sec. C.7). When neither signal is available we fall back to the uninformative prior $\pi_q = 0.5$, equivalently $\mu_q = 0$. By standard calculations for linear Gaussian models, the posterior is $\theta_q \mid y_{q,1:K} \sim \mathcal{N}(m_q, v_q)$ with

$$m_q = \alpha_q \bar{y}_q + (1 - \alpha_q)\mu_q, \qquad \alpha_q = \frac{K\tau^2}{K\tau^2 + \sigma_q^2}, \tag{11}$$

where $\bar{y}_q = \frac{1}{K}\sum_k y_{qk}$.

Prediction. Given the posterior mean $\mathbb{E}[\theta_q \mid y_{q,1:K}] = m_q$, we use the following plugin approximation to the posterior predictive distribution:

$$\hat{p}_q \ = \ \text{sigmoid}\big(\alpha_q\, \bar{y}_q + (1 - \alpha_q)\, \mu_q\big) \tag{12}$$

where $\text{sigmoid}(x) = 1/(1 + e^{-x})$. When trials agree ($\sigma_q^2$ small) $\alpha_q \approx 1$ and $\hat{p}_q$ collapses to logit-space averaging of the trials; when trials disagree $\alpha_q \to 0$ and $\hat{p}_q$ is pulled toward π_q (the crowd/empirical prior, or 0.5 when neither is available).

Practical implementation. Equation (12) requires a per-question $\alpha_q = K\tau^2/(K\tau^2 + \sigma_q^2)$, which depends on the unknown prior variance τ^2 and the (estimable) per-question logit-trial variance σ_q^2. The standard empirical Bayesian approach is to maximize the marginal likelihood wrt the hyperparemeter τ. We adopt the simpler, and more robust, approach of using LOO-CV to estimate τ^2.

We also evaluated a heuristic shrinkage rule (Sec. D.4) based on the same principle: that we should shrink toward the prior more aggresibely (i.e. smaller α_q) when there is a lot of cross-trial variation (i.e.,)σ_q^2 is large). More precisely, we adopt the following estimator:

$$\alpha_q \ = \ \max\big(f,\ 1 - c \cdot s_q\big), \qquad s_q = \text{std}\big(y_{q1}, \ldots, y_{qK}\big), \tag{13}$$

where s_q is the per-question sample standard deviation of trial logits, $f \in [0, 1]$ is a floor on α_q, and $c \geq 0$ controls how fast α_q contracts toward f as trials disagree. The two hyperparameters (f, c) are global (shared across questions). Operationally, f=1 disables shrinkage entirely ($\alpha_q \equiv 1$, pure logit averaging), while f=0 allows full shrinkage to the prior; intermediate f caps how aggressively a single high-disagreement question can be shrunk. The interaction is mildly degenerate in the f>0 regime: whenever $c \cdot s_q \leq 1 - f$, the slope c has no effect (the floor f is binding), so LOO effectively selects between "always at the floor" (c large) and "mostly at the ceiling" (c→0).

Connection to James–Stein. This method is analogous to James–Stein shrinkage (Stein, 1956; Efron & Morris, 1973), which shrinks individual estimates toward a common mean to reduce total MSE. Two differences are worth highlighting. First, classical James–Stein shrinks toward a single global mean; we instead shrink toward a *per-question* center μ_q (the crowd or empirical prior in logit space) when one is available. Second, our setting can optimize (f, c) wrt log-scoring rules where overconfident wrong predictions are catastrophically penalized, rather than being restricted to squared error.

C.10 CALIBRATION

We apply Platt scaling (Platt, 1999) as a post-processing calibration step:

$$\hat{p}_{\text{cal}} = \sigma\big(a \cdot \text{logit}(\hat{p}) + b\big), \tag{14}$$

where a and b minimize log loss on the training records. Historical calibration results use leave-one-record-out cross-validation: each forecast entry is calibrated using a model fitted to the other entries. Other horizons or forecast dates from the same underlying question can remain in training, so these estimates may be optimistic and do not establish performance on unseen questions. Intervals are conditional on the fitted predictions. For deployment, a calibrator can be trained on historical labeled data and applied to new forecasts. This record-level protocol differs from the question-held-out aggregation analysis in Sec. D.4.

For ForecastBench, which spans 9 heterogeneous sources (Table 16), we also evaluate *hierarchical Platt scaling*: a shared slope a and intercept b, plus per-source offsets δ_s:

$$\hat{p}_{\text{cal}} = \sigma\big(a \cdot \text{logit}(\hat{p}) + b + \delta_s\big). \tag{15}$$

When fitting, we add L2 regularization, $\lambda \sum_s \delta_s^2$, to the NLL. We use $\lambda = 1.0$ as a fixed default rather than tuning it formally; the metric is fairly insensitive to λ within an order of magnitude on a benchmark of our size, so we did not run an outer LOO loop over λ values. The per-source offsets account for systematic miscalibration that varies across question types (e.g., the model tends to be overconfident on polymarket questions but underconfident on dbnomics).

D FB: FURTHER RESULTS

This section presents more ForecastBench results.

Table 11: Historical external-comparison configuration on each benchmark; Table 3 and the memory study instead use arithmetic averaging without calibration. AIBQ2 has no crowd signal, empirical prior, or source-specific tools (all questions are Metaculus, search-only). DBnomics bypasses the LLM entirely and uses a custom KNN tool (see Sec. G.1).

Component	AIBQ2	ForecastBench
LLM	Gemini-3.1-Pro	Gemini-3.1-Pro
Thinking level	high	high
Search engine	Brave	Brave
Source tools	(none available)	yfinance, FRED, Wikipedia, DBnomics; polymarket, manifold
Belief state	yes	yes
Agent mode	sequential ($T_{\max} = 10$)	sequential ($T_{\max} = 10$)
Crowd signal	off (not available)	market price for market questions
Empirical prior	off (not available)	base rate for dataset questions
Trials	$K = 5$	$K = 5$
Aggregation	logit-space LOO shrinkage	logit-space LOO shrinkage*
Calibration	Platt (record-LOO)	hier. Platt (per-source, record-LOO)
DBnomics	n/a	KNN bypass (no LLM)

*On FB at c=1 the LOO grid selects f=1, c=0 (no shrinkage), so the aggregator collapses to plain logit-space mean (Eq. (8)); shrinkage is still *applied* but is a no-op at the chosen hyperparameters. These historical summaries differ from the regenerated common-pool trial-count analysis: on FB without anchors, question-held-out tuning selects no shrinkage (Fig. 2a).

D.1 FB: COMPARISON TO SOTA

Table 12: ForecastBench A∪B comparisons under three metrics. Higher BI and ABI and lower BS are better. Brackets give 95% intervals: bootstrap intervals for BI; ABI intervals are approximate. BS points and split-wise intervals are obtained by transforming the rounded BI summaries, so are approximate; no overall BS interval is reported. ABI adjusts for question difficulty (Sec. A.2). * denotes an external leaderboard submission; † denotes partial coverage (Grok: A only; Foresight: B only). For BI and BS, All averages Market and Data with equal weight. See Fig. 1 and Tab. 1 for visualization and paired BI tests.

	BI ↑			ABI ↑			BS×100 ↓		
Method	Market	Data	All	Market	Data	All	Market	Data	All
BLF (Pro)	**83.8** [79,90]	**62.7** [61,65]	**73.3** [71,76]	**84.7** [82,88]	**62.0** [59,65]	**71.0** [68,74]	2.6 [1.0,4.4]	13.9 [12.2,15.2]	8.3
Cassi*	82.0 [78,86]	59.6 [58,61]	70.8 [69,73]	82.0 [79,85]	59.6 [57,63]	68.7 [66,72]	3.2 [2.0,4.8]	16.3 [15.2,17.6]	9.8
Grok 4.20*†	79.5 [74,86]	61.4 [59,63]	70.5 [68,74]	77.6 [75,81]	61.4 [58,65]	68.4 [65,72]	4.2 [2.0,6.8]	14.9 [13.7,16.8]	9.6
Foresight-32B*†	81.0 [75,86]	57.6 [55,60]	69.3 [67,72]	82.3 [79,85]	57.6 [54,61]	67.5 [64,71]	3.6 [2.0,6.2]	18.0 [16.0,20.2]	10.8
Crowd+emp (no LLM)	81.5 [77,86]	58.3 [57,60]	69.9 [68,72]	81.5 [79,84]	58.3 [55,61]	67.7 [65,71]	3.4 [2.0,5.3]	17.4 [16.0,18.5]	10.4

Tab. 12 compares BLF with search-enabled external agents and the LLM-free crowd+empirical-prior baseline using BI, ABI, and BS. The BLF configuration is in Tab. 11. BLF has the highest point estimates for BI and ABI and the lowest BS. On market BI, BLF scores 83.8, Cassi 82.0, and the crowd baseline 81.5; paired comparisons are needed because method coverage differs.

Table 1 reports paired-bootstrap comparisons: BLF vs each external method; Tab. 2 compares each LLM-based method with the crowd+emp baseline. The headline finding is that *BLF is the only LLM-based method that significantly beats the crowd+emp baseline on overall BI* ($+3.4^{**}$); on the dataset subset alone the gap is $+4.5^{***}$. On the market subset the paired test for BLF vs. baseline is

Table 13: Sample sizes (number of binary resolution events) per method, split (Market / Data), and tranche, for the comparisons in Tab. 1. BLF and Cassi cover both tranches. Grok submitted only on 2025-10-26 (tranche A) and Foresight only on 2025-11-09 (tranche B); for the paired comparisons involving them, we use only the events from their submitted tranche to avoid scoring a forecast against a resolution event that falls outside the method's actual ask date.

	Market n			Data n			All n		
Method	A	B	All	A	B	All	A	B	All
BLF (Pro)	100	100	200	298	293	591	398	393	791
Cassi	100	100	200	298	293	591	398	393	791
Grok†	100	0	100	298	0	298	398	0	398
Foresight†	0	100	100	0	293	293	0	393	393
Baseline (no LLM)	100	100	200	298	293	591	398	393	791

$+2.3^{\text{ns}}$ (the difference is in the right direction but is within noise on n_{mkt}=200 events); none of the three external methods beats the baseline on overall BI either, although Grok is significantly better on the dataset subset alone.

D.2 FB: Per-source analysis

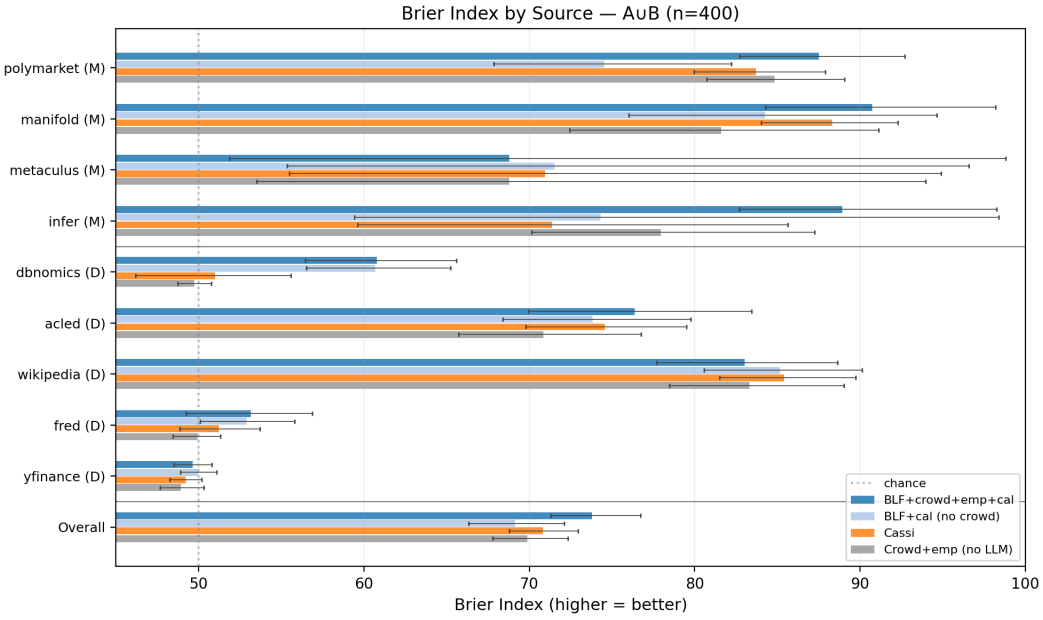

Figure 7: Brier Index by source on FB A∪B, comparing BLF (with and without the crowd anchor on market questions), the LLM-free crowd+empirical baseline, and Cassi, the retained external agent covering both tranches.

Figure 7 breaks down BI by source, comparing BLF against Cassi and two baselines (LLM-free crowd+empirical, and BLF without the crowd anchor). We see that all methods are at chance on yfinance, due to the random walk nature of stock prices (see Sec. F.3). All methods struggle with FRED (Federal Reserve Economic Data) for similar reasons (see Sec. F.4), although our method has an edge, likely due to the simple statistical model we use (Eq. (25)), whose estimate is made available to the agent to supplement the raw data when the fetch-time-series tool is called. Our method has an edge on DBnomics for similar reasons: the agent uses a simple KNN-based probability estimator, bypassing the LLM entirely, since we found the LLM struggles with seasonal data such as temperature (see the ablation study in Tab. 17). All methods do quite well on ACLED and Wikipedia, because the questions are quite easy, since the base rates are so extreme (see Sec. F.6 and Sec. F.7).

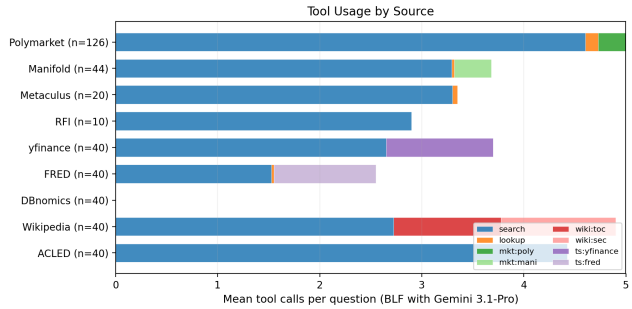

Figure 8: Mean tool calls per question by source for BLF (trial 1, A∪B). Source-specific tools are used selectively. DBnomics bypasses the LLM (KNN model).

On market questions, we outperform other methods by a large margin, which struggle to beat the prior. The sources of these gains are studied below.

D.3 Tool Usage

Tool usage. Figure 8 shows the mean number of tool calls per question, broken down by source. Not surprisingly, web search dominates across all sources. Source-specific data tools (market info, Wikipedia sections, time-series fetchers) are used selectively — only for their respective sources. DBnomics questions are handled entirely by the KNN model (Sec. G.1) with no LLM tool calls. Polymarket questions use the most tools (∼5.5 per question), reflecting the additional market-info fetch on top of web search.

E ANOVA AND PAIRED ANALYSIS

E.1 VARIANCE DECOMPOSITION

We perform a two-way ANOVA (method × question) on trial-level BI scores pooled across 12 historical configurations. This is a descriptive analysis of the original experiment pool, separate from the matched belief and memory contrasts in Tab. 3. Each observation is BI_{ijt} for method i, question j, trial t. The total sum of squares is decomposed as:

$$\text{SS}_{\text{total}} = \text{SS}_{\text{method}} + \text{SS}_{\text{question}} + \text{SS}_{\text{residual}}$$

where $\text{SS}_{\text{method}} = \sum_i n_i(\bar{y}_{i\cdot} - \bar{y})^2$ sums over the 12 methods, $\text{SS}_{\text{question}} = \sum_j n_j(\bar{y}_{\cdot j} - \bar{y})^2$ sums over the 400 questions, and the residual includes inter-trial variability and any method–question interaction not captured by the additive model.

The F-statistic tests whether the group means differ significantly: $F = \text{MS}_{\text{effect}}/\text{MS}_{\text{residual}}$, where $\text{MS} = \text{SS}/\text{df}$ is the mean square. A large F (relative to the F-distribution with appropriate degrees of freedom) indicates the effect is significant.

Table 14: Two-way ANOVA on trial-level BI scores (FB A∪B, 12 methods × 400 questions × 5 trials). A perfectly balanced design would yield $12 \times 400 \times 5 = 24{,}000$ observations; the residual df of 23,442 implies N=23,853 actual trials, i.e. 147 (0.6%) were dropped because the agent hit the per-question timeout or returned no parseable probability (see Sec. C.8 for how trial-level dead/missing values are handled in the aggregation step).

Source	SS	% total	df	F
Method (M=12)	38.1	2.7%	11	164.5
Question (Q=400)	877.6	62.2%	399	104.3
Residual	494.3	35.1%	23,442	—
Total	1,410.1	100%	23,852	

From Tab. 14, we see that question difficulty dominates (62.2% of total variance), consistent with the observation that some questions are inherently easy (e.g., "Will a vaccine for Sepsis exist by next month?") while others are genuinely hard (e.g., stock price movements), cf. (Frick et al., 2025). See also Fig. 9, which shows that on AIBQ2, most questions are very easy, so performance differences are dominated by a subset of hard questions.

Method effects account for only 2.7% of total variance, but the omnibus effect is significant ($F = 164.5$, $p \ll 0.001$). This does not establish significance for every pairwise contrast. The residual (35.1%) includes inter-trial stochasticity and unmodeled method–question interactions; it is not a pure estimate of LLM sampling variability.

E.2 PAIRED ANALYSIS

For interpreting paired comparisons, consider an additive two-way linear model

$$\text{BI}_{ij} = \mu + \alpha_i + \gamma_j + \epsilon_{ij} \tag{16}$$

where μ is the grand mean, α_i is the effect of method i (with $\sum_i \alpha_i = 0$), γ_j is the effect of question j (with $\sum_j \gamma_j = 0$), and ϵ_{ij} is the residual.

We consider both a fixed-effects (FE) estimation method, that computes point estimates (MLEs) of all parameters, and a mixed-effects (ME) estimation method, that treats the question effects as random, $\gamma_j \sim \mathcal{N}(0, \sigma_q^2)$. The ME method is statistically preferable, since it accounts for uncertainty in γ_j and enables inference on new questions, but requires iterative algorithms such as REML. Under exact balance the fixed-effects and mixed-effects estimates for α_i coincide (a standard result on orthogonal designs); under approximate balance they differ by an amount on the order of the imbalance fraction, which in our case is very small, as we showed in Tab. 14.

FE estimation via alternating projections. We estimate $(\mu, \{\alpha_i\}, \{\gamma_j\})$ using an alternating least-squares (ALS) algorithm, which iterates:

$$\gamma_j \leftarrow \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} (\mathrm{BI}_{ij} - \mu - \alpha_i) \tag{17}$$

$$\alpha_i \leftarrow \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} (\mathrm{BI}_{ij} - \mu - \gamma_j) \tag{18}$$

where $\mathcal{I}_j$ is the set of methods that answered question j, and $\mathcal{J}_i$ is the set of questions answered by method i. This is equivalent to coordinate descent on the least-squares objective $\sum_{ij}(\mathrm{BI}_{ij} - \mu - \alpha_i - \gamma_j)^2$, which is convex and separable in the two sets of parameters, so iteration converges to a global minimiser of the fit.

The minimiser is not unique in (α, γ) alone — adding any constant c to every α_i and subtracting c from every γ_j leaves the fit unchanged — so we impose the identifying sum-to-zero constraints $\sum_i \alpha_i = 0$ and $\sum_j \gamma_j = 0$ already stated in Eq. (16) by recentering after each sweep ($\alpha_i \leftarrow \alpha_i - \bar{\alpha}$, $\gamma_j \leftarrow \gamma_j - \bar{\gamma}$, $\mu \leftarrow \mu + \bar{\alpha} + \bar{\gamma}$); this fixes a unique parameter configuration.

In our setting, every method attempts every question and the 0.6% dropped trials are spread thinly across methods, so the design is balanced for practical purposes. Under exact balance, the effects reduce to centered marginal means. With missing trials, we use the iterative fit rather than assuming that one sweep yields the exact solution.

Bootstrap analysis of pairwise differences. For the reported BI contrasts we pair methods on common resolution events, recompute each method's BI from its mean squared error, and take the difference. On ForecastBench, Overall BI is the equal-weight mean of Market and Data BI. This nonlinear statistic is not the mean of per-event BI differences. The resampling unit is stated with each analysis: the external comparisons use paired events, while belief, memory, and trial-count analyses resample questions and retain their resolution horizons. Pairing controls differences in the scored sample, but does not by itself remove confounding from model, tools, or compute budgets. Event resampling does not capture within-question dependence across horizons; resampling question–date instances also leaves cross-date dependence unmodeled.

Percentile intervals are obtained from the paired bootstrap distribution. Where stars are reported, the two-sided bootstrap tail estimate is twice the smaller fraction of differences at or below zero and at or above zero, capped at one. Tests are exploratory and not adjusted for multiple comparisons. Intervals for fitted post-processing are conditional on the fits unless otherwise stated; they do not include a full nested refitting analysis.

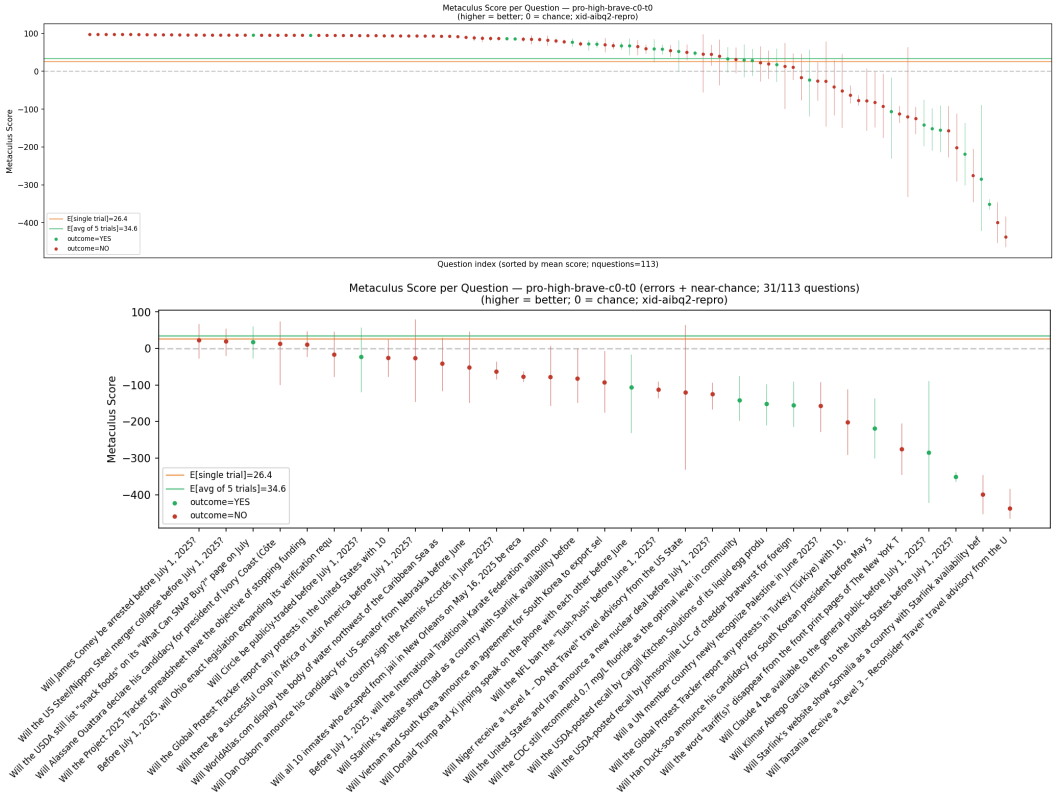

Figure 9: Top: Metaculus Baseline Score for all $n = 113$ AIBQ2 questions (most are easy). Bottom: Zoom into questions at or below chance. Error bars: bootstrap 95% CIs across 5 trials.

F DATASETS AND THEIR ANALYSIS

F.1 AIBQ2

The AIBQ2 benchmark consists of 113 binary questions from the Metaculus AI Benchmark Tournament (Q2 2025). [14] Questions were asked between 2025-04-22 and 2025-06-15, with resolution dates up to 2025-07-01. The base rate is 22% (25 True, 88 False). Questions do not have a market or crowd base rate. Figure 10 shows the forecast horizon distribution: most questions have short horizons (1–8 weeks), with many resolving on 2025-07-01 (the tournament deadline). See also Fig. 11 for another visualization of this data. Table 15 shows some example questions, which we have clustered into 10 manually chosen topics. (The clustering is done by applying an LLM classifier to each question, and then grouping questions with the same class label or tag into the same cluster). We see that the most common topics are Domestic Politics (35/113) and Geopolitics & Conflict (29/113). By contrast, the FB topic distribution is quite different (see Fig. 13).

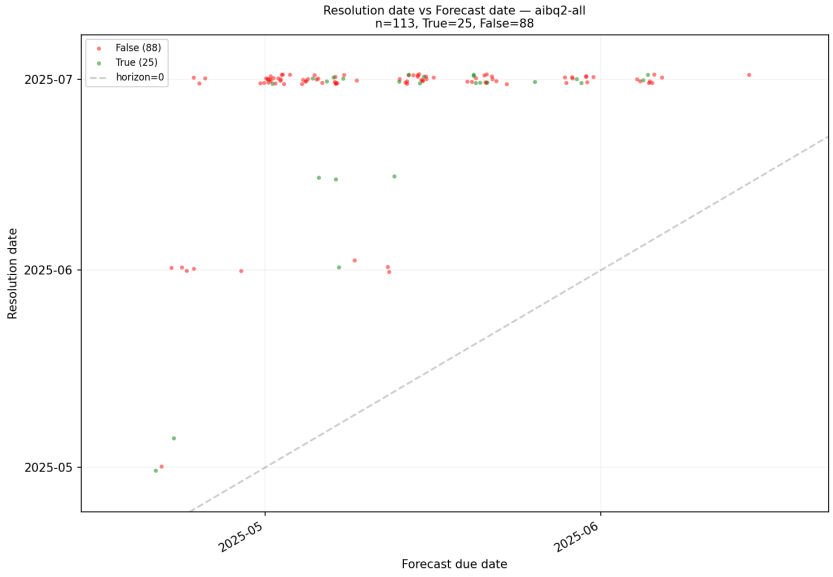

Figure 10: AIBQ2 date distribution of questions and answers. Green = resolved True (25), Red = resolved False (88).

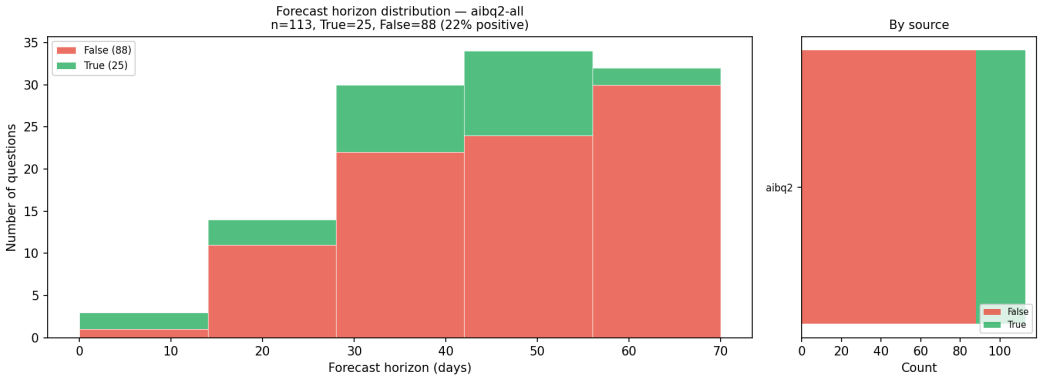

Figure 11: AIBQ2 forecast horizon distribution ($n = 113$). Right panel shows outcome balance.

[14] See https://www.metaculus.com/tournament/aibq2/. The full list of questions and outcomes is available at https://gist.github.com/enjeeneer/86e24a52e6041a3d78e333bcab16984d, taken from (Jeen et al., 2026b).

Table 15: AIBQ2 topic distribution with example questions across 10 categories ($n = 113$). Domestic Politics dominates (35 questions).

Category	n	Example question
Domestic Politics	35	Will a state of emergency be in effect in Samoa on April 30, 2025?
Geopolitics & Conflict	29	Will there be a successful coup in Africa or Latin America before Jul...
Business & Industry	16	Will Uber be available in the Turks and Caicos Islands on July 1, 2025?
Society & Law	9	Will Harvard University lose its tax exempt status before July 1, 2025?
Macroeconomics	7	Will South Sudan ratify AfCFTA (the pan-African free trade agreement)...
Science & Technology	6	Will Starlink's website show Lesotho as a country with Starlink avail...
Health & Biology	5	Will the CDC raise the Travel Health Notice status of Colombia to abo...
Financial Markets	2	Will any of these companies get dropped from the Dow Jones Industrial...
Other	2	Will WorldAtlas.com display the body of water northwest of the Caribb...
Sports & Entertainment	2	Before July 1, 2025, will the International Traditional Karate Federa...
Total	113	

F.2 FORECASTBENCH

In this section, we discuss the ForecastBench dataset from (Karger et al., 2025). This consists of binary prediction questions from market sources (Polymarket, Manifold, Metaculus, Rand Forecasting Initiative (RFI)), which assess the ability to do "judgemental forecasting" (Lawrence et al., 2006) on various topics for various forecast horizons. We can formalize this as follows. Let $Y(t)$ denote the random variable of interest (e.g., "Will A be president of country B at time t?"). Let f be the forecast date and r the resolution date. Our task is to estimate $P(Y(r) = 1 \mid \text{data}(\leq f))$.

FB also contains questions derived from dataset sources (yfinance, FRED, DBnomics, Wikipedia, ACLED), which assess the ability to do (univariate) time series forecasting. These are converted to binary prediction problems by asking the forecaster what the probability will be that the value of interest at multiple future resolution dates, out to a horizon of $h \in \{7, 30, 90, 180, 365, 1095, 1825, 3650\}$ days, will be larger or smaller (possibly by some factor) compared to its value at the forecast date. More formally we are required to estimate $P(Y(r_i) > v \mid \text{data}(\leq f))$, where $r_i = f + h_i$ is the resolution date for the i'th forecast, for $i = 1 : 8$, $Y(r_i)$ is the unknown future value, and $v = y(f)$ is the known threshold or reference value. For example, $Y(r_i) > v$ might represent the event "the temperature at location X on date r_i will be higher than its current value v". (Note that the value v is not always given in the question, a subtlety we discuss in Sec. C.4.)

500 new FB questions are released every two weeks (250 market and 250 dataset), and these resolve over time at different rates.

The combined tranches contain 400 question-date instances but 355 distinct underlying (source, question) IDs. Date-specific instances can therefore be dependent even when treated as separate questions in an analysis.

F.2.1 TRANCHES

We construct two evaluation datasets (which we call "tranches") from ForecastBench, selecting dates that satisfied several criteria: after the knowledge cutoff of all models of interest (2025-08-31), has many resolved values (especially important for market questions, which resolve less often), and maximizes overlap with the submissions from existing leading methods (see Fig. 12). (See Sec. A.3 for details on these existing methods which we compare to.) Each tranche uses questions from a single `forecast_due_date` and selects those which are resolved by 2026-04-10 (ensuring ground truth is available). Of the ∼500 questions per forecast date, approximately 100 market and 200 dataset questions had resolved by 2026-04-10; we select all 100 market questions and 100 dataset questions (20 per source, via stratified sampling with a fixed seed) for each tranche, to ensure neither dominates the performance metrics. (Note that each dataset question has multiple resolution dates, so the total number of binary questions with answers is n=791.) More precisely, we use these tranches:

- **Tranche A** (forecast_due_date = 2025-10-26): 100 market + 100 dataset questions (200 total). Comparisons available: Cassi, xAI Grok 4.20, GPT-5.

- **Tranche B** (forecast_due_date = 2025-11-09): 100 market + 100 dataset questions (200 total). Comparisons available: Cassi, Foresight-32B, GPT-5.

The combined dataset (**Tranche A∪B**, $n = 400$) is used for the main comparison. Paired statistical tests are conducted within each tranche (where all methods forecast the same questions). Note that for market questions (but not dataset/timeseries questions), ForecastBench includes a crowd estimate, based on the market price for Polymarket, Manifold, and RFI, and the Community Prediction score for Metaculus. This is a very strong baseline, and most methods on the leaderboard rely on it quite heavily.

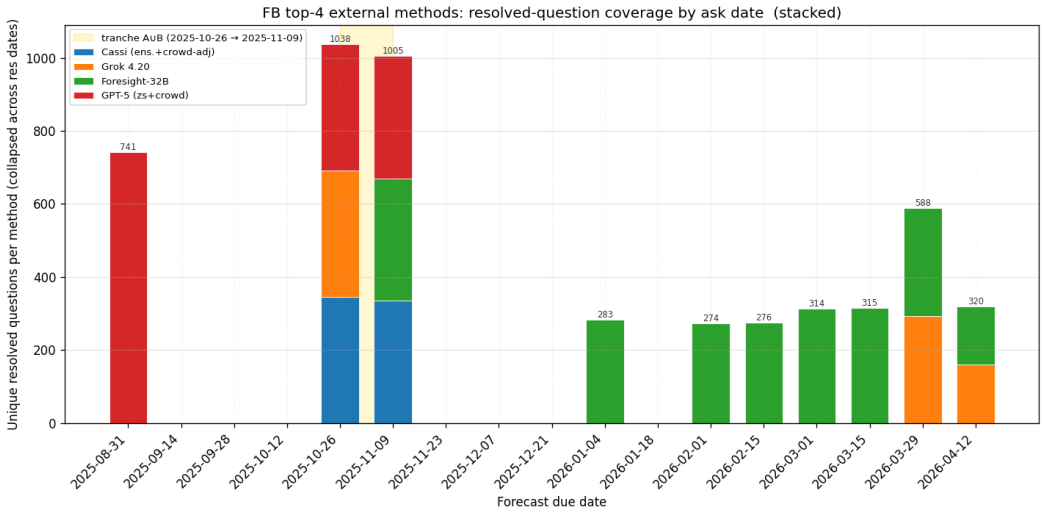

Figure 12: Number of resolved questions by the top external methods across time. The golden region corresponds to tranches A and B that we use.

F.2.2 Statistical properties of the tranches

Table 16: Question composition by source for FB Tranches A and B. "BR" = base rate (fraction resolving True). Market sources provide judgemental forecasting questions; dataset sources provide univariate time-series questions with multiple resolution dates per question. ACLED = Armed Conflict Location and Event Data. FRED = Federal Reserve Economic Data. DBnomics = database of international economic statistics.

Source	URL	n_A	n_B	n	BR(A)	BR(B)
RFI	randforecastinginitiative.org	5	5	10	0.00	0.00
Manifold	manifold.markets	18	26	44	0.22	0.15
Metaculus	metaculus.com	10	10	20	0.10	0.20
Polymarket	polymarket.com	67	59	126	0.13	0.08
Market total		100	100	200	0.14	0.11
ACLED	acleddata.com	20	20	40	0.25	0.00
DBnomics	db.nomics.world	20	20	40	0.95	0.80
FRED	fred.stlouisfed.org	20	20	40	0.60	0.55
Wikipedia	wikipedia.org	20	20	40	0.20	0.25
Yahoo Finance	finance.yahoo.com	20	20	40	0.20	0.45
Dataset total		100	100	200	0.44	0.41
Total		200	200	400	0.29	0.26

In this section, we discuss some properties of the FB dataset. Tab. 16 shows some key statistics. Fig. 13 shows the topic distribution: Financial Markets (76) leads, followed by Sports & Entertainment (69) and Geopolitics & Conflict (66), with the remaining 7 categories more evenly distributed.

Figure 7 shows per-source BI for four methods. Performance of our best system (BLF+crowd+emp+cal) varies dramatically across sources: Wikipedia (83.0) and Manifold (90.7) are near-perfect, while yfinance (49.6) is essentially chance. FRED (53.2) is the second-weakest source.

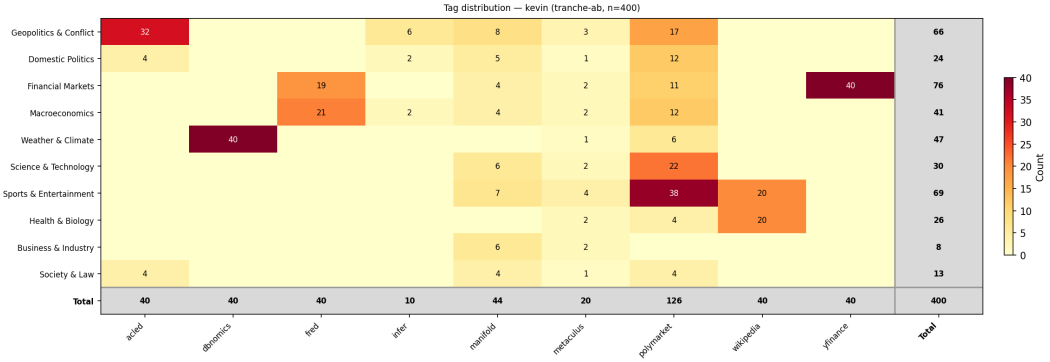

Figure 13: FB A∪B topic distribution across 9 sources.

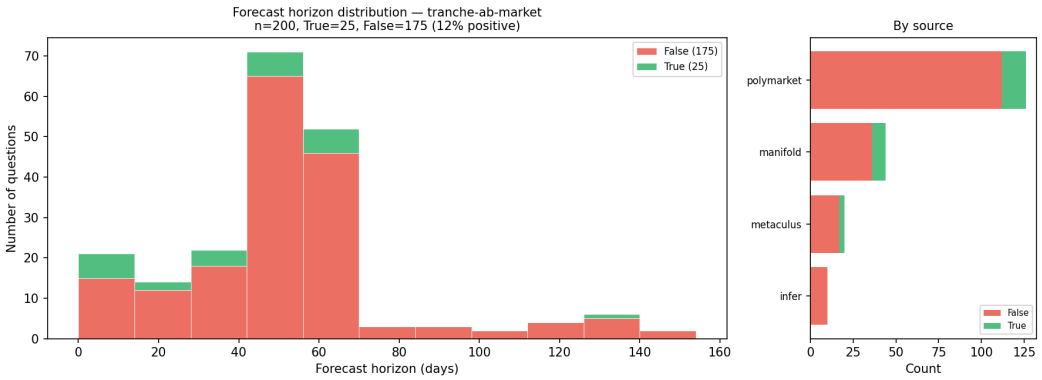

Figure 14: FB market questions: forecast horizon distribution ($n = 200$). Most horizons are 1–5 months. Base rate is 12% (heavily skewed toward False).

Market questions (Polymarket, Manifold, Metaculus, RFI) are judgemental forecasting questions similar in nature to AIBQ2 (see Table 15 for example topics), but spanning a wider range of sources and with a different topic distribution (Figure 13). Forecast horizons range from 1 week to 5 months (Figure 14).

The base rates are very low (Table 16): Polymarket 11%, Manifold 18%, Metaculus 15%, RFI 0%. This means most questions resolve "No,". Furthermore, these questions come with a market estimate; using this as the predicted probability achieves high market BI — a strong baseline that is very hard to beat. The belief-only comparison in Tab. 3 isolates market gains for Pro and Flash without calibration.

Dataset questions are univariate time series forecasting problems, which are converted to binary form by asking whether (some function of) the value of a quantity of interest at a future resolution date will be higher or lower than the value at the forecast date. Topics are very narrow, and depend on the source, as shown in Figure 13. Forecast horizons are fixed at 7, 30, and 90 days, as shown in Figure 15. We give more details on individual dataset sources below.

F.2.3 Representativeness of the tranches

After completing the main experiments, we extended our evaluation by running our SOTA model on every FB question that the agent could ever have forecasted since after Gemini knowledge cutoff (n=23,150 resolved (source, question, resolution-date) entries spanning 14 ask dates from 2025-03 through 2026-04). This allowed us to estimate how representative the A,B tranches are.

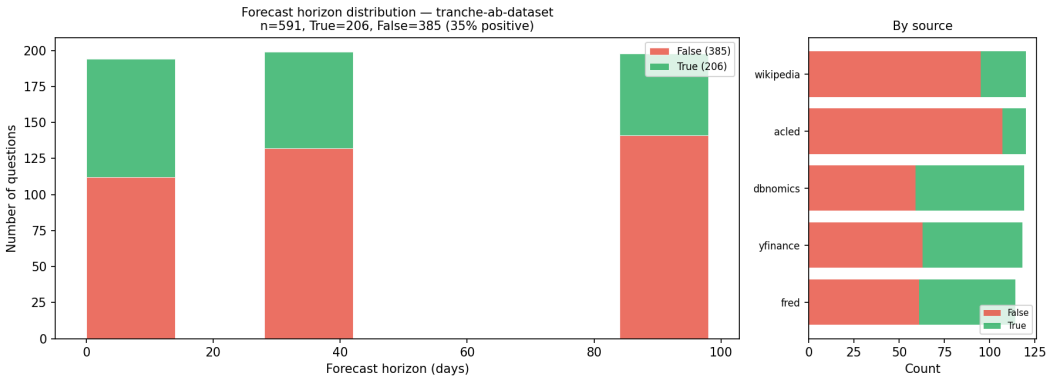

Figure 15: FB dataset questions: forecast horizon distribution ($n = 200$ questions, 591 resolution dates). Horizons are fixed at 7, 30, and 90 days (only the first 3 of 8 standard resolution dates have resolved by our evaluation cutoff). Base rate is 35%.

As we see from Fig. 16, the tranche window sits in the middle of a fairly stable region of the BI timeline, although the overall BI level (∼72) is slightly higher than the wider corpus mean (∼69). The difference is mostly due to the market questions in the tranches being easier (+6.5 BI), driven mostly by differences in the manifold/infer/metaculus sources. However, the dataset questions in the tranches were slightly harder (-1.2 BI), and the net effect (after averaging over all sources) is that the BI only differs by about 3. This suggests our tranche BI numbers may be higher than what one would expect on the overall FB leaderboard. However, we expect this bias to affect other methods in the same way.

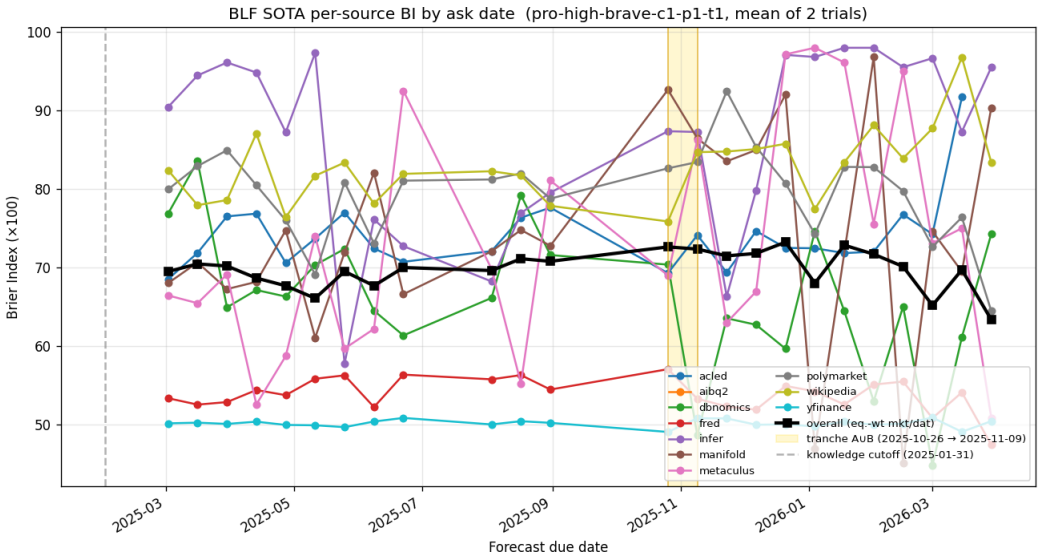

Figure 16: Brier Index of our method across all resolved questions since after the Gemini knowledge cutoff. Gold band represents tranches A and B.

F.3 FB: Yahoo Finance (stock prices)

Yahoo Finance questions ask whether a stock's closing price on a future resolution date will be higher than its closing price on the forecast due date. Each question has 8 resolution dates ranging from 1 week to 10 years out. Figure 17 shows an example.

The question text uses placeholders `{resolution_date}` and `{forecast_due_date}` that are substituted at preprocessing time. Crucially, the question does *not* specify the reference price on the forecast due date — it only provides a `freeze_datetime_value` from an earlier date (when the question was created, typically 10 days before the forecast due date). For example, the CPB question below has `freeze_datetime_value` = 30.15 from 2025-10-16, but the actual closing price on the forecast due date (2025-10-24, since 10-26 is a Sunday) was 30.79. Our system retrieves the correct reference value via the `fetch_ts_yfinance` tool, which downloads the price history and returns the closing price on or before the forecast due date.

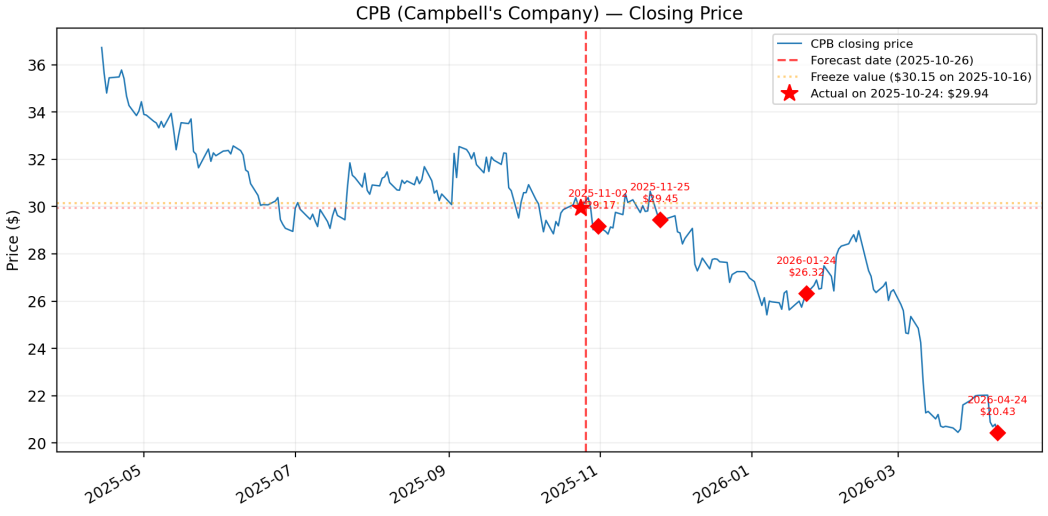

Figure 17: CPB (Campbell's Company) closing price. Red dashed line: forecast date (2025-10-26). Star: actual price on 2025-10-24 ($29.94, since 10-26 was Sunday). Orange: freeze value ($30.15 from 10-16). Red diamonds: resolution date prices (all below reference, all resolved False). The decline to ∼$20 was unpredictable at forecast time.

Figure 17 illustrates why yfinance questions are inherently difficult: stock prices follow an approximate random walk, making it nearly impossible to predict whether the price will be higher or lower than the reference value on any given future date. The optimal strategy is to predict $p \approx 0.5$, which our system correctly learns (yfinance BI ≈ 0.5 across both tranches; see Table 16).

Example question (verbatim, abbreviated).

```
{
  "forecast_due_date": "2025-10-26",  // global
  "id": "CPB",
  "source": "yfinance",
  "question": "Will CPB's market close price on
    {resolution_date} be higher than its market
    close price on {forecast_due_date}?",
  "resolution_criteria": "Resolves to the market
    close price at https://finance.yahoo.com/quote/CPB.
    If the resolution date coincides with a day the
    market is closed, the previous close is used.",
  "freeze_datetime": "2025-10-16T00:00:00+00:00",
  "freeze_datetime_value": "30.15",
  "resolution_dates": ["2025-11-02", "2025-11-25",
    "2026-01-24", ...],
  "resolved_to": [0, 0, 0, ...]  // all False
}
```

F.4 FB: FRED (ECONOMIC INDICATORS)

FRED (Federal Reserve Economic Data) questions ask whether an economic indicator will have increased by each resolution date compared to the forecast due date. An example question is shown below:

> **Question (DTB6):** "Will the Federal Reserve's 6-month secondary market treasury bill rate have increased by `{resolution_date}` compared to its value on `{forecast_due_date}`?"
> **Freeze value:** 3.69% (as of 2025-10-16)
> **Resolution dates:** 2025-11-02, 2025-11-25, 2026-01-24, 2026-04-24, 2026-10-26, 2028-10-25, 2030-10-25, 2035-10-24

The 40 FRED questions in our tranches (Tab. 16) ask about 36 distinct economic series; the series span interest rates (DTB6, DGS30), corporate bond spreads (BAMLC0A0CM), exchange rates (DEXUSUK), stock indices (DJIA), and banking data (DPSACBW027SBOG).

Unlike yfinance (approximate random walk), many FRED series exhibit persistent trends. For example, Figure 18 shows the 6-month T-bill rate declining steadily from ∼5.25% to ∼3.5% over 2024–2026, reflecting Fed rate cuts. The first resolution date (2025-11-02) caught a brief uptick (resolved True), but subsequent dates resolved False as the rate continued falling. This trend is partially predictable from recent history, giving our `fetch_ts_fred` tool useful context.

Figure 19 shows the USD/GBP exchange rate, which is harder to predict — more similar to a random walk, with a mix of True and False resolutions.

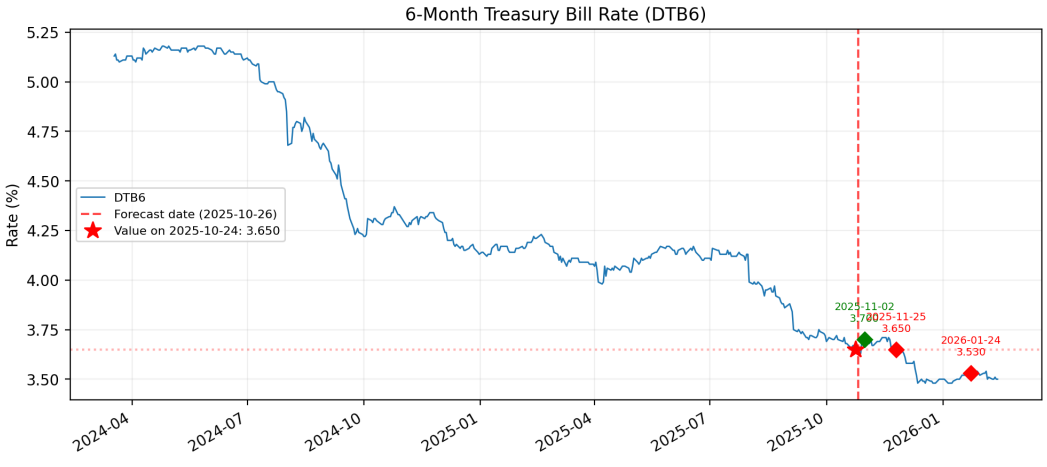

Figure 18: 6-Month Treasury Bill Rate (DTB6). Forecast date is 2025-10-26 (Sunday), so the reference value is from 2025-10-24 (red star). Clear downward trend from Fed rate cuts. Green diamond = resolved True (rate increased); red = resolved False.

Overall, FRED has mixed predictability: of our 36 distinct series, roughly 8 exhibit strong autocorrelation (trending), 18 are effectively random walks, and the rest are intermediate. Trend-following series (interest rates, banking aggregates) reward tools that detect recent direction, while noisy series (exchange rates, bond yields) are closer to chance. The combined base rate for FRED is 57% (Table 16), between DBnomics (88%) and yfinance (33%).

A natural improvement would be to classify each FRED series into a small number of patterns (random walk, trending, mean-reverting) and apply the appropriate statistical model — e.g., $p = 0.5$ for random walks, a trend-extrapolation model for persistent series. ForecastBench's full question bank contains 165 unique FRED series, so this per-series approach would require a lightweight classifier, which we leave to future work.

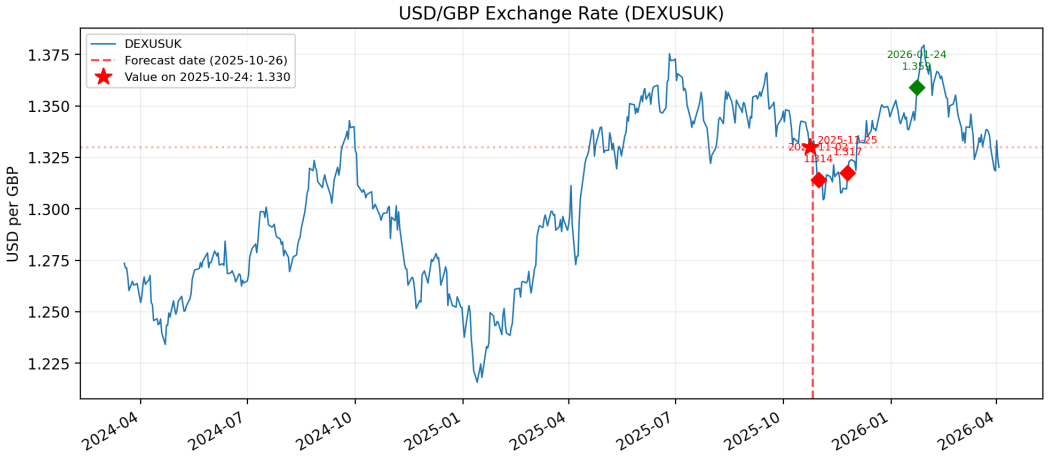

Figure 19: USD/GBP exchange rate (DEXUSUK). Reference value from 2025-10-24 (red star, since forecast date 2025-10-26 is a Sunday). More erratic than interest rates — harder to predict.

F.5 FB: DBnomics (temperature)

All DBnomics questions in ForecastBench ask whether the daily average temperature at a French weather station will be higher on a future resolution date than on the forecast due date. Our tranches contain 29 stations (24 metropolitan France, 5 overseas), as shown in Figure 20; each generates one question per forecast date with 8 resolution dates. Figure 21 shows ∼14 years of daily temperature data for one station (Mont-de-Marsan), exhibiting strong annual seasonality.

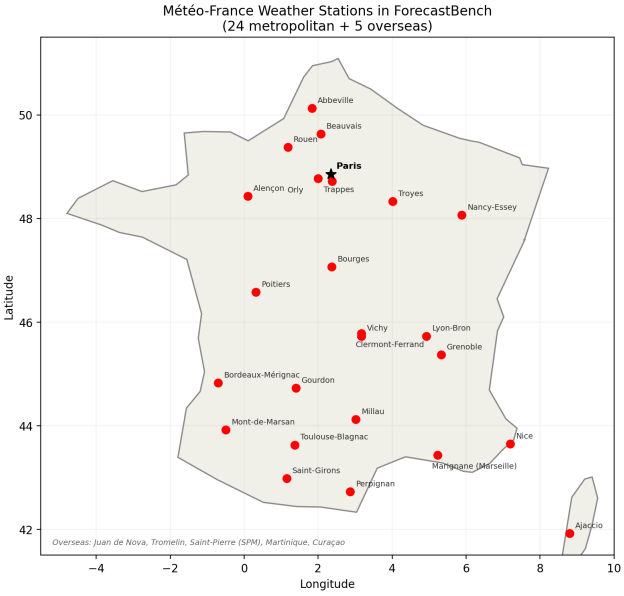

Figure 20: Météo-France weather stations used in ForecastBench. Stations are highly correlated (mean pairwise $r = 0.93$).

Although the stations are spatially distributed, we found that pooling bias-corrected observations from nearby stations does not improve forecasts (negligible BI gain). This is because each station already has ∼200 historical observations per resolution date, so the per-station empirical exceedance estimate is already well-determined. The main bottleneck for dbnomics is near-horizon weather forecast data (e.g., 7-day forecasts), not statistical estimation from historical data.

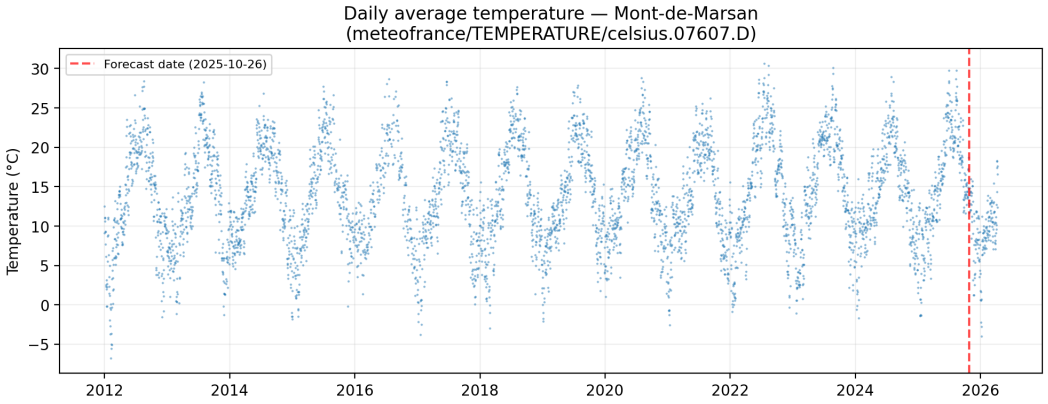

Figure 21: Daily average temperature at Mont-de-Marsan, 2012–2025. Strong annual seasonality makes statistical forecasting effective.

Figure 22 zooms into the period around the forecast date. The key challenge is the same as yfinance: the question provides a `freeze_datetime_value` of 14.4°C from 2025-10-16, but the actual value on the forecast date (2025-10-26) was 11.7°C — the threshold the agent must predict against. Our `fetch_ts_dbnomics` tool retrieves the correct value and computes an empirical exceedance probability (see Appendix G).

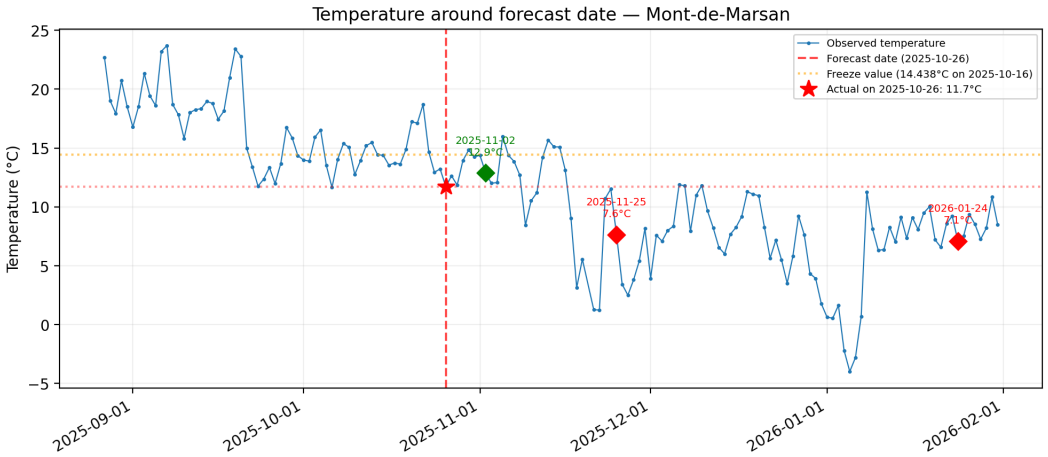

Figure 22: Temperature around the forecast date for Mont-de-Marsan. Star: actual value on forecast date (11.7°C). Orange: stale freeze value (14.4°C). Green diamond: resolution date warmer than reference; red: colder. The seasonal decline from October to January is predictable from historical data.

Unlike yfinance (random walk), temperature has strong seasonal structure: the October-to-January decline is predictable from ∼14 years of historical data. Our empirical exceedance model achieves good BI by computing $P(\text{temp} > \text{threshold})$ from same-calendar-date observations in prior years (see Sec. G for model comparison). We bypass the LLM entirely for DBnomics, since LLM reasoning does not improve on the statistical model. Our system outperforms Cassi on this source (BI 60.8 vs Cassi 51.0). Further improvement likely requires short-term weather forecast data for near-horizon resolution dates.

Example question (verbatim, abbreviated).

```
{
  "forecast_due_date": "2025-10-26",  // global
```

```
  "id": "meteofrance_TEMPERATURE_celsius.07607.D",
  "source": "dbnomics",
  "question": "What is the probability that the
    daily average temperature at Mont-de-Marsan
    will be higher on {resolution_date} than on
    {forecast_due_date}?",
  "url": "https://db.nomics.world/meteofrance/
    TEMPERATURE/celsius.07607.D",
  "freeze_datetime": "2025-10-16T00:00:00+00:00",
  "freeze_datetime_value": "14.438",
  "resolution_dates": ["2025-11-02", "2025-11-25",
    "2026-01-24", ...],
  "resolved_to": [1, 0, 0, ...]  // True, False, False
}
```

F.6 FB: ACLED (ARMED CONFLICT)

ACLED (Armed Conflict Location & Event Data) questions ask whether the number of conflict events of a given type in a given country will exceed a threshold over a future 30-day window. There are two question templates (Figure 23):

- **Any increase** (20 questions): will events exceed the baseline level? These have a 25% base rate (5/20). Example:

  > **Any increase:** "Will there be more 'Protests' in Sri Lanka for the 30 days before `{resolution_date}` compared to the 30-day average of 'Protests' over the 360 days preceding 2025-11-09?"
  > Resolution dates: 2025-11-16, 2025-12-09, 2026-02-07.
  > Resolved: False, False, True (protest activity surged later).

- **10x spike** (20 questions): will events in the next 30 days exceed *ten times* the 360-day rolling average? These have a 0% base rate — no country in our tranches experienced such an extreme spike. Example:

  > **10x spike:** "Will there be more than ten times as many fatalities in Finland for the 30 days before `{resolution_date}` compared to *one plus* the 30-day average of fatalities over the 360 days preceding 2025-11-09?"
  > The "one plus" ensures the threshold is non-zero even when the baseline average is 0.
  > Resolved False on all 3 dates.

Questions span 30 countries (from Antarctica to Venezuela) and 6 event types (Battles, Strategic developments, Riots, Violence against civilians, Explosions/Remote violence, Protests). No source-specific data tool is available for ACLED (the API requires special access), so the agent relies entirely on web search.

The very low overall base rate (12.5%, 5 True out of 40) means that predicting "No" for all questions is a strong baseline. The 10x-spike questions are particularly easy: since none resolved True, any well-calibrated system should predict near 0 for these. The agent's main challenge is the "any increase" questions, which require understanding recent conflict dynamics via web search.

F.7 FB: WIKIPEDIA

Wikipedia questions ask whether specific facts on a Wikipedia page will change by the resolution date. Our 40 questions come from three articles:

- **List of infectious diseases** (20 questions, BR=0%): "Will a vaccine have been developed for [disease] by [resolution date]?" Since vaccine development takes years, none of these resolved True over our 1–3 month horizons.
- **FIDE rankings** (15 questions, BR=27%, 4/15): either "Will [player]'s FIDE ranking be as high or higher?" (5 rank questions) or "Will [player]'s Elo rating be at least 1% higher?" (10 Elo questions). The rank questions are easier (4/5 always True), while gaining 1% Elo in 1–3 months is very rare (0/10 True). See Figure 24.

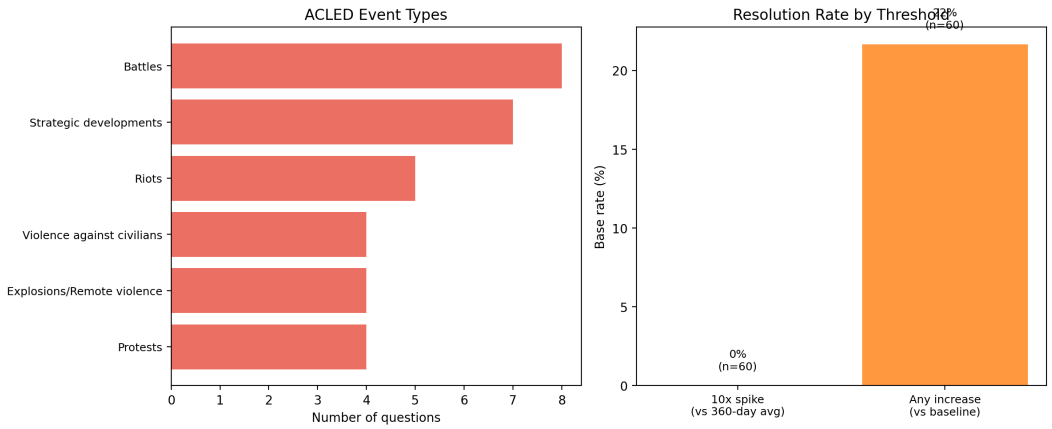

Figure 23: ACLED questions in A∪B. Left: event type distribution. Right: base rate by threshold type. The "10x spike" questions never resolved True.

- **List of world records in swimming** (5 questions, BR=100%): "Will [swimmer] still hold the world record for [event]?" All resolved True (world records are rarely broken in a few months).

The overall base rate is 22.5% (9 True out of 40), but it varies dramatically by article (0%, 27%, 100%). Our system achieves very high BI on Wikipedia (83.0; see Figure 7) because the questions are often answerable from the structure of the question itself: vaccine questions are almost certainly No, world record questions are almost certainly Yes. The `fetch_wikipedia_toc/section` tools allow the agent to verify these priors by checking the actual Wikipedia page content.

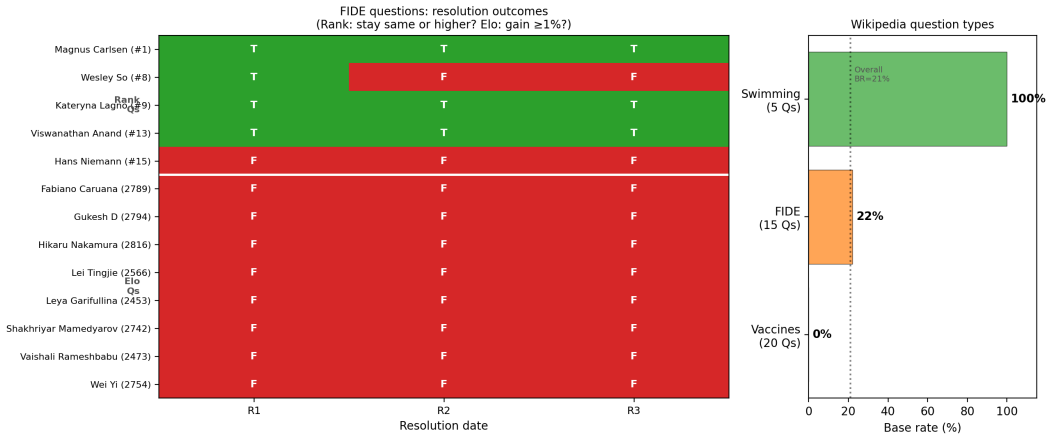

Figure 24: Left: FIDE question outcomes. Rank questions (top 5) mostly resolve True; Elo ≥1% questions (bottom 10) all resolve False. Wesley So (#8) is the only mixed case (True, False, False — his ranking briefly improved then dropped). Right: base rates for the three Wikipedia question types.

G TIME-SERIES FORECASTING MODELS

In this section we discuss specialized statistical modeling tools which we optionally make available to the agent for certain data sources. These all return an estimate of $\hat{p}_q^r$ defined in Eq. (3). These are all simple heuristic estimators that we developed to see if they helped the LLM with its task. Clearly many more sophisticated methods could be used, but that is not the focus of this paper.

G.1 KNN MODEL FOR DBNOMICS

Table 17: Time-series model comparison on DBnomics questions (A∪B, n=40 questions, 119 resolution dates). All models above the line are evaluated standalone (no LLM). KNN is our default for DBnomics and outperforms Cassi.

Model	BI
KNN (±10 days, 10+ years)	**59.1**
Always 0.5 (baseline)	50.0
Cassi (reference)	51.0

We tried several simple models for estimating $\hat{p}_q^r$ for DBnomics data. The best performing method, which we call KNN, is a non-parametric estimator which is analogous to a binary k-nearest-neighbor classifier, where "neighbors" are selected by day-of-year proximity. More precisely, define the neighbor set as follows:

$$\mathcal{N}_{a,w}(q,r) = \{y_q(t_i) : d_{\text{cyc}}(\text{doy}(t_i), \text{doy}(r)) \le w,\ f_q - a < t_i < f_q\}, \tag{19}$$

where $\text{doy}(\cdot) \in \{1, \dots, 365\}$ extracts the day-of-year and

$$d_{\text{cyc}}(a,b) = \min(|a-b|,\ 365 - |a-b|) \tag{20}$$

is the cyclic (year-wrap-aware) distance — so e.g. Dec 31 and Jan 2 are at distance 2, not 363. a is the maximum age (to encourage focusing on data close to q's forecast date), w is the window size around $\text{doy}(r)$, and i indexes across all available data. (Leap days are folded onto day 365 by Pandas' `dayofyear`; the resulting ±1 slack is negligible compared to our w=10 window.) The forecast is the Laplace-smoothed empirical exceedance frequency:

$$\hat{p}_q^{\text{knn}}(a,w) = \frac{|\{y_q(t_i) \in \mathcal{N}_{a,w}(q,r) : y_q(t_i) > v_q\}| + 1}{|\mathcal{N}_{a,w}(q,r)| + 2}. \tag{21}$$

This directly estimates Eq. (3) without distributional assumptions. For DBnomics, we use a window of $w = 10$ days, and set $a = \infty$ to remove age restrictions, reflecting an assumption the periodic distribution is stationary. With ∼14 years of daily data, each resolution date typically has $|\mathcal{N}_{a,w}(q,r)| \approx 270$ neighbors.[15]

Table 17 shows results on A∪B. Our KNN method achieves BI 59.1, outperforming Cassi (51.0) on this source. We also tried feeding the time series directly to the LLM, as well as linear and harmonic models, but KNN was the best standalone approach. Further improvement might come from short-term weather forecast data for near-horizon resolution dates.

G.2 LINEAR TREND MODELS FOR YFINANCE AND FRED

For non-seasonal time series, such as yfinance and FRED, we use a parametric approach, which combines a local linear model with the baseline estimate of 0.5:

$$p_q^{\text{safe-linear}}(\alpha) = \alpha \cdot p_q^{\text{linear}} + (1-\alpha) \cdot 0.5 \tag{22}$$

where α is a source-specific shrinkage factor controlling how much to trust the linear extrapolation. We set $\alpha = 0.1$ for yfinance, which reflects that stock prices follow an approximate random walk,

[15] $2w + 1 = 21$ days/yr times 14 years ≈ 294. But some locations drop about 5% of their data, bringing the typical realised count down to about 270.

so the linear extrapolation is unreliable. We set $\alpha = 0.5$ for FRED. (All α values were hand-tuned based on informal experimentation; systematic cross-validation is left to future work.)

The linear probability estimate is given by

$$p_q^{\text{linear}} = \Phi^c\left(\frac{v_q - \hat{y}(r)}{\hat{\sigma}}\right) \tag{23}$$

where Φ^c is the standard normal survival function, and $\hat{y}(t) = at + b$ is the linear model fit by OLS on the most recent W observations, $\mathcal{H}(s_q, f_q; W)$. (We use $W = 30$ for FRED and $W = 60$ for yfinance.) $\hat{\sigma}$ is the standard deviation of the residuals for this data source s_q. (We could make this a function of forecast horizon, $r - f_q$, as well as source type, s_q, but we leave that to future work.)

G.3 Hybrid KNN and linear trend model for yfinance and FRED

For FRED and yfinance, we also consider a hybrid parametric and non-parametric estimate of the form

$$\hat{p}_q^{\text{hybrid}}(\alpha, \beta) \quad = \beta \cdot p_q^{\text{safe-linear}}(\alpha) + (1 - \beta) \cdot p_q^{\text{prior-year}} \tag{24}$$

where $p_q^{\text{prior-year}} = p_q^{\text{knn}}(a, w)$ uses the same nonparametric exceedance estimate as Eq. (21), but with a narrower window ($w = 7$ days) and limited to the 5 most recent years, by setting $a = 5 \times 365$ days. This captures annual periodicity: for a given resolution date, we check whether the value exceeded the threshold at the same time of year in prior years. Expanding out, this estimate is a convex combination of three terms, the linear forecast, the prior-year (KNN) forecast, and the uniform 0.5 fallback:

$$\hat{p}_q^{\text{hybrid}}(\alpha, \beta) \quad = \alpha\beta\, p_q^{\text{linear}} + (1 - \beta)\, p_q^{\text{prior-year}} + (1 - \alpha)\beta \cdot 0.5, \tag{25}$$

where the three weights sum to one (since $\alpha\beta + (1-\beta) + (1-\alpha)\beta = \beta + (1-\beta) = 1$).

We set $\beta = 0.5$ for both FRED and yfinance when prior-year data is available ($\beta = 1$ otherwise, falling back to $p^{\text{safe-linear}}$ alone). For DBnomics, the hybrid model is not used — we apply the KNN model (Eq. (21)) directly.

H Extended Related Work

We organize the growing literature on LLM-based forecasting into eight themes.

Benchmarks. Zou et al. (2022) introduced Autocast, the first large-scale dataset for neural forecasting of real-world events, and found that LM performance was far below human experts but improved with model scale and retrieval. Halawi et al. (2024) scaled this up with 5,000+ binary questions from five platforms, showing that retrieval-augmented GPT-4 approaches the human crowd's Brier score. ForecastBench (Karger et al., 2025) provides a rolling benchmark with market and dataset questions, using difficulty-adjusted Brier scores (Kucinskas et al., 2025) to compare methods on a common leaderboard. TFRBench (Ahamed et al., 2026) evaluates forecasting *reasoning* (not just accuracy). FutureX (Zeng et al., 2025) and FutureX-Pro (Liu et al., 2026a) introduces a live benchmark with daily-updated questions and an automated pipeline to eliminate data contamination, but focuses on 0-1 accuracy rather than probabilistic forecasting. Metaculus FutureEval (Metaculus, 2026) provides a continuously updated live benchmark that resists contamination (since answers are not yet known), with $175K in annual prizes for bot tournaments. OpenEP (Guan et al., 2024) moves beyond binary questions to open-ended outcome prediction. (Liptay et al., 2026) present "Bench to the Future 2", a benchmark of 1417 binary judgemental forecasting questions, asked between 2025-10-18 and 2025-10-28, with a resolution window in 2025-10 to 2025-12; BTF2 also ships with a frozen pre-retrieved corpus of documents. (Goel et al., 2026) present FutureSim, which is a benchmark for evaluating online forecasting (continual learning) using a fixed text corpus (to avoid leakage). ProphetArena (Yang et al., 2025) is a benchmark that bundles multiple related binary forecasting questions into a "market" (e.g., "will A,B or C win US election" becomes 3 binary questions), but at the time of writing, they do not release historical data, precluding backtesting.

Evaluation. Paleka et al. (2026) provide a critical analysis of offline evaluation pitfalls, including temporal leakage and retrieval leakage. Li et al. (2026) show that LLMs fail to "simulate ignorance" of events before their knowledge cutoff, further complicating backtesting validity. Our four-layer leakage defense (Sec. B) addresses several of these concerns. Prophet Arena (Yang et al., 2025) argues that economic utility (profit from bets) may be more relevant than Brier scores; they find that LLMs show impressive calibration but suffer from inaccurate event recall.

Financial forecasting and trading. Several benchmarks evaluate LLMs as financial predictors and traders. Prediction Arena (Zhang et al., 2026) deploys six frontier models with real capital on Kalshi and Polymarket, finding that most models lose money (-16% to -31% on Kalshi). PolyBench (Cheng et al., 2026) evaluates seven LLMs on $\sim$38K Polymarket questions with order-book data, finding that only two models achieve positive returns. FinTradeBench (Agrawal et al., 2026) evaluates financial reasoning combining fundamentals and trading signals, finding that retrieval helps fundamental analysis but not time-series reasoning — consistent with our finding that LLMs struggle with raw numerical data (Sec. C.4). (Wang et al., 2026) proposes a new trading benchmark.

Zero-shot and prompt-based forecasting. Karkar & Chopra (2025) show that LLM forecasting ability is "unevenly distributed" across domains and question types (c.f., Fig. 9). Pratt et al. (2024) found that superforecasting prompting strategies (decomposition, base rates, retrieval) failed to improve PaLM 2 over a basic prompt, attributing the model's apparent accuracy to a negativity bias that happens to align with low base rates. Schoenegger et al. (2025) tested 38 prompt variants across GPT-4o, Claude 3.5, and Llama 3.1, finding that most modifications yield negligible gains and some (e.g., explicit Bayesian reasoning prompts) actively hurt. These negative results motivate our agentic approach: rather than engineering better prompts, we provide tools (search, data access) and structured belief tracking.

Agentic and tool-augmented systems. Hsieh et al. (2024) proposed RTF (Reasoning and Tools for Forecasting), a zero-shot framework using hierarchical agents with Python REPL and Google Search, achieving Brier scores competitive with human crowds on Manifold Markets without any weight updates. The AIA Forecaster (Alur et al., 2025) employs adaptive iterative search and statistical calibration (Platt scaling), similar to us, but uses simple text aggregation rather than a belief state. Also, they use an LLM-based calibration system, which we found to be inferior to simple mean

aggregation. CogForecast (Wang et al., 2025) uses multi-agent debate with diverse cognitive profiles to mitigate inherited biases.

Bayesian agents. BayesAgent (Huang et al., 2026) also targets Bayesian reasoning under uncertainty in LLM agents, but via *verbalized probabilistic graphical models* (vPGM): the LLM is prompted to identify latent variables and dependencies, then they use standard Bayesian inference algorithms applied to this model. This is complementary to our approach: BayesAgent constructs an explicit PGM per query, while `BLF` maintains a non-Markovian natural-language belief state updated sequentially from tool-call evidence, avoiding the need for an explicit graph structure. In the position paper, (Papamarkou et al., 2026; Amin, 2026) argue that LLM agents should model uncertainty using a more classical sequential Bayesian paradigm, in which the likelihood of observed text data o_t (e.g. the results of a search tool call a_t) is modeled using $p(o_t|s, a_t)$, where s is the unknown outcome we are trying to forecast. In Sec. 6 we present some preliminary results evaluating this kind of method, and find lower accuracy than linguistic BLF for the tested likelihood elicitation schemes on fixed replayed evidence.

Reinforcement learning for forecasting. Several recent works apply RL to improve forecasting. Time-R1 (Liu et al., 2025) uses a two-stage curriculum (temporal comprehension then prediction) to build "temporal logic" into the model's representations. Turtel et al. (2025) apply RLVR (RL with Verifiable Rewards) to train a 14B model on historical Polymarket questions, achieving frontier-level Brier scores (0.190) and demonstrating economic utility (10%+ ROI in simulated trading). OpenForecaster (Chandak et al., 2026) synthesizes 50K+ training questions from historical news and uses GRPO with a composite accuracy+Brier reward to combat hedging bias, showing that specialized 8B models can match 120B+ generalist models. Jeen et al. (2026b;a) use RLFT on open source models for the Metaculus AI Benchmark Tournament, achieving the current AIBQ2 SOTA (MS=45.8). (Damani et al., 2026) use RLFT to improve calibration. Turtel et al. (2026) introduce "Foresight Learning" (the training method behind the *Foresight-32B* model from Lightning Rod Labs that we compare against in Tab. 1), using resolved outcomes as free supervision via proper scoring rule rewards, showing that a trained Qwen3-32B outperforms the untrained Qwen3-235B (27% better Brier score). (Auzina et al., 2026) uses RL where the reward function is related to the change in the belief state, rather than just the final outcome. These RL approaches are complementary to our work: they improve the *base model*, while we improve the *agent harness* (tools, belief tracking, calibration).

Ensemble methods. Schoenegger et al. (2024) showed that aggregating 12 diverse LLMs (a "silicon crowd") matches human crowd accuracy, and that frontier models can update beliefs when shown human median forecasts. Ensemble gains depend on diversity among the component forecasts (see also Aitchison et al., 2026).

I PROMPTS

This appendix lists the prompts used in our system and the zero-shot baseline.

I.1 SYSTEM PROMPT

The following is the system prompt used for atyes ll `BLF` configurations (abbreviated for space; source-specific tool descriptions omitted).

```
You are an expert superforecaster.  Your task is to predict
the probability that a binary question will resolve to YES
given information up to a certain date.
You work in a tool-use loop:
1.  Read the question, its resolution criteria, and
background.
2.  Form a base rate estimate (outside view / reference
class reasoning).
3.  Perform a loop:
3a.  Choose ONE tool to call.
3b.  After each tool call, your belief state is updated.
4.  When you have gathered enough evidence, call
submit(probability, reasoning).
Belief state rules:
- Evidence lists should ACCUMULATE across steps.
- Each evidence item MUST cite its source
(search_X_result_Y).
- Include update_reasoning explaining WHY evidence changed
your probability.
- Consider RECENCY and AUTHORITATIVENESS of sources.
Rules:
- You MUST call submit before step {max_steps}.
- Call submit once probability has stabilized.
- Probabilities must be between 0.05 and 0.95.
```

I.2 NOBEL SYSTEM PROMPT

For the NoBel ablation we use a separate, simplified system prompt that does *not* mention belief state, evidence-accumulation rules, or an `updated_belief` field — the tool schemas also have the `updated_belief` parameter stripped (see Sec. C.4). The prompt is reproduced below (abbreviated the same way as Sec. I.1). Crucially, neither the prompt nor the tool schema asks the agent to maintain a structured belief state, so the agent operates by accumulating raw search results in its message history.

```
You are an expert superforecaster.  Your task is to predict
the probability that a binary question will resolve to YES
given information up to a certain date.
You work in a tool-use loop:
1.  Read the question, its resolution criteria, and
background.
2.  Perform a loop:
2a.  Choose ONE tool to call.
2b.  The tool result will be added to the conversation.
3.  When you have gathered enough evidence, call
submit(probability, reasoning).
Suggested strategy:
- Use web_search to find specific evidence.
- Use summarize_results to read the full content of
promising results.
- Consider all the evidence you have gathered when making
your final probability estimate.
Rules:
- You MUST call submit before step {max_steps}.  If
```

```
you reach the last step without submitting, a default
probability of 0.5 will be used.
- Call submit once your sense of the probability has
stabilized.
- Probabilities must be between 0.05 and 0.95.
```

The same template-substitution machinery that fills `{max_steps}`, the source-specific tool list, and the backtest-vs-live mode paragraph in the BLF prompt also runs on this template; the only difference is the loop-and-rules text shown above.

I.3 Question Prompt

Each question is formatted as follows (with crowd/prior sections included only when `crowd=1` or `emp=1`):

```
# Question
{question text}
## Background and resolution criteria
{background}
{resolution_criteria}
## Resolution dates
{dates}
You must submit {n} probabilities (one per resolution date).
Your uncertainty should INCREASE with forecast horizon.
## Knowledge cutoff
{cutoff_date}
You must not use any information from after this date.
## Market estimate (if crowd=1, market questions only)
The market estimate on {date} was {value}.
## Prior estimate (if emp=1, dataset questions only)
{explanation}: {prior}.
Use this as your starting point, but adjust based on
question-specific evidence from search and tools.
```

I.4 Zero-Shot Prompt

For the zero-shot baseline (`search=none`, `tools=0`) we use the same standard system prompt as `BLF` (Sec. I.1); the only difference at runtime is that the tool schema contains only `submit`, so the agent reads the question and immediately submits a probability without performing search or any other tool call. The loop-rules and belief-state portions of the standard prompt are effectively dead text in this mode. Interestingly, we found that switching the entire prompt to the simplified zero-shot template from Halawi et al. (2024) (also used by ForecastBench for its zero-shot leaderboard entries) gave worse performance (65.3 BI on FB A∪B for ZS+crowd+emp, vs 70.6 BI using our standard prompt). The Halawi template is shown below for reference; the experiments reported in this paper use our standard prompt throughout.

```
You are an expert superforecaster, familiar with the work of
Tetlock and others. Make a prediction of the probability
that the question will be resolved as true. You MUST
give a probability estimate between 0 and 1 UNDER ALL
CIRCUMSTANCES. If for some reason you can't answer, pick
the base rate, but return a number between 0 and 1.
Question: {question}
Question Background: {background}
Resolution Criteria: {resolution_criteria}
Question close date: {close_date}
[If freeze value available:] The freeze value is {value}.
{explanation}
Output your answer (a number between 0 and 1) with an
asterisk at the beginning and end of the decimal. Do not
output anything else.
```

```
Answer:
{{ Insert answer here }}
```

I.5 Search-Result Date-Filter Prompt

Web search results are often metadata-stale: a page's `page_age` reflects its original publish date, but the page body may have been updated with post-cutoff content (liveblogs, "last updated" patches, automatically-appended related links, etc.). Our second pass strips the metadata age flags and asks an LLM to re-judge each result purely on its *textual content*, dropping anything that describes events or outcomes after the knowledge cutoff. This is the layer audited in Sec. B (the leak detective). The exact prompt (taken from `src/search/search_lib.py:filter_results`; the LLM is Gemini-3-Flash by default):

```
You are a strict information-cutoff filter.  The knowledge
cutoff date is {cutoff_date}.
For each numbered search result, decide KEEP or DROP.

Focus on the CONTENT of each result -- the highlights, extra
snippets, and any dates mentioned in the text body.  Do NOT
trust the 'Published on' date alone; pages are often updated
after their original publish date.

Rules:
  • If ANY part of the content describes events, outcomes,
    or updates from AFTER {cutoff_date} → DROP the entire
    result
  • Look for dates in the text body (e.g. 'January 25, 2026',
    'February 6, 2026').  If any mentioned date is after
    {cutoff_date} and the text describes what happened on
    that date → DROP
  • Past tense describing events scheduled after cutoff (e.g.
    'the climb was postponed', 'he completed the ascent') →
    DROP
  • Liveblog/timeline pages with entries after cutoff → DROP
  • Content only about events before or on the cutoff date →
    KEEP
Reply with one line per result:  the number, a colon, KEEP
or DROP, then a brief reason.
Example:
1:  KEEP - article from Nov 2025 about the announcement
2:  DROP - contains Jan 25 update describing postponement

Results:
{numbered_results_with_age_flags_stripped}
```

The `KEEP/DROP` decisions are parsed by regex; any result the LLM does not explicitly mark as `DROP` is retained (conservative AND with the client-side `page_age` filter that already runs inside the search-engine wrapper).

I.6 Search-Result Summarization Prompt

After the agent calls `web_search` and selects a subset of the returned snippets via `summarize_results`, the selected results are concatenated and passed to a separate "summarizer" LLM (Gemini-3-Flash by default; see Sec. A.4) for compression. We use a question-aware prompt that extracts only facts relevant to the resolution criteria, rather than producing a generic summary — this both shortens the text the agent must reason over and reduces the risk of injecting irrelevant evidence into the belief state. The exact system prompt (taken from `src/search/search_lib.py:summarize_results`):

```
You are an assistant to a superforecaster.  Extract the
facts and data from the search results below that are most
```

```
relevant to predicting the outcome of this question.

Question: {question}

Resolution criteria: {resolution_criteria}

Instructions:
  • Extract concrete facts, statistics, dates, and
    named-source expert opinions.
  • Note any quantitative data (prices, percentages, counts,
    trends).
  • Distinguish hard facts from speculation or editorial
    opinion.
  • Omit information that is not relevant to the resolution
    criteria.
  • Do NOT add your own analysis or forecast -- only extract
    what the sources say.
Search results:
{filtered_search_results}
```

When `resolution_criteria` is unavailable (short-form summarization fallback), the prompt collapses to "Summarize the following search results clearly and concisely, highlighting the most relevant facts for answering the question: {question}", followed by the same `{filtered_search_results}` block.

I.7 Likelihood estimation prompts

The likelihood ratio prompt. Each weight uses a fixed system prompt and a user prompt instantiated once per state $s \in \{\text{YES}, \text{NO}\}$; the action, prior-evidence, and summary lines appear only for the conditioning variants that use them.

```
[SYSTEM] You estimate how typical an observation is under
a hypothesized outcome of a binary forecasting question.
Use forward simulation: assume the stated outcome is the
truth, then judge how typical/representative the evidence
would be of a world in which that outcome holds. Output a
single integer 0-10: 10 = highly typical, 5 = neutral, 0 =
completely atypical. Output ONLY the integer.
[USER] Binary question: {question}
The agent issued this search query: {a_t}.
Evidence already gathered before this search: {b_{t-1}.h}
NEW evidence retrieved by this search: {o_t}
Assume the TRUE outcome is: {state}. How
typical/representative is the evidence above of a world
in which the answer is {state}? Judge only what the NEW
evidence adds beyond what is already known. Answer with a
single integer 0-10. Output only the integer.
```

The model is queried twice ({state}=YES and NO), giving the two typicality scores in Eq. (2).

As an example, consider the NextEra Energy (NEE) stock question. The belief summary $b_{t-1}.h$ (the value substituted for the prior-evidence line) taken mid-trajectory is as follows:

```
Evidence FOR: Stocks generally go up over time (base rate
> 0.5); Linear trend from Aug to Oct is positive (slope =
+0.16/day)

Evidence AGAINST: Trend analysis gives combined estimate of
0.41; Price has been slightly trending down/flat over the
last 10 days of October
```

```
        Open questions:  Earnings dates and ex-dividend dates for
        NEE in Q4 2025; Any recent major news affecting NEE stock
```