From Narrative to Auditable Forecasts: A Structured Scaffold for Agentic Forecasting

Yuanpu Cao [1] **Yongkang Du** [1] **Yurui Chang** [1] **Lu Lin** [1] **Jinghui Chen** [1]

Abstract

LLM agents are increasingly used for live forecasting, where they retrieve up-to-date information and produce estimates for unresolved future events. However, current agentic forecasting often relies on implicit narrative aggregation: agents collect evidence, discuss it in prose, and often assign a probability without an explicit update path from evidence to forecast. This limits both calibration and auditability. We propose AUDITFORECAST, an agentic scaffold for structured probabilistic forecasting. AUDITFORECAST decomposes each forecast into a model-derived base probability and a set of evidence-grounded factor updates, which are then combined mechanically in odds space. This yields a traceable derivation from evidence to forecast and enables per-component attribution and post-hoc auditing. Across multiple live forecasting benchmarks, AUDITFORECAST improves forecasting accuracy and calibration relative to strong agentic baselines while also making the forecasting process more interpretable and auditable.

1. Introduction

Forecasting is central to real-world decision-making, from financial markets (Rapach & Zhou, 2013) to business and societal planning (Hyndman & Athanasopoulos, 2018). Yet probabilistic forecasting is not merely another reasoning task. Unlike mathematical problem solving (Hendrycks et al.) or code generation (Jain et al., 2025), which operate over information already available in well-defined form, forecasting concerns uncertain future outcomes under incomplete and evolving information. Its goal is not to recover a determinate solution from fixed premises, but to produce a calibrated probabilistic judgment under uncertainty.

[1]College of Information Sciences and Technology, The Pennsylvania State University. Correspondence to: Yuanpu Cao <ymc5533@psu.edu>, Jinghui Chen <jzc5917@psu.edu>.

Forecasting as a New Frontier of Intelligence Workshop at ICML 2026, Seoul, South Korea.

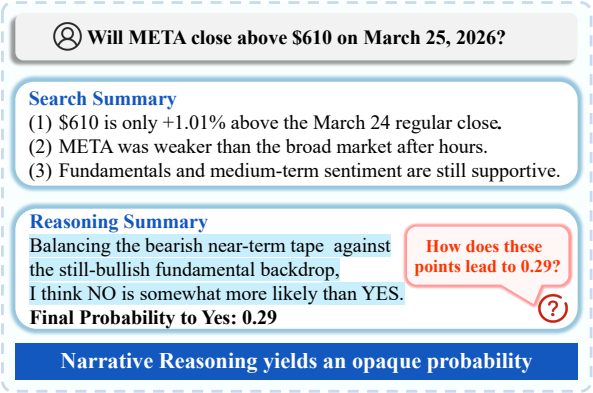

Figure 1. A narrative forecast from GPT-5.4 under an unstructured scaffold. The agent retrieves data, lists unstructured pros and cons, and outputs an untraceable probability.

Large language model (LLM) agents (Luo et al., 2025) augment foundation models with web search, API access, and other tools for acquiring up-to-date information (Liu et al., 2025; Yang et al., 2024; OpenAI, 2025a), making live forecasting increasingly plausible. Recent benchmarks (Yang et al., 2026; Zeng et al., 2026) have shown meaningful gains from such search-augmented agents. However, existing agentic scaffolds are not explicitly designed for forecasting as a probabilistic reasoning problem. In practice, this gap manifests as implicit *narrative aggregation*: as illustrated in Figure 1, they list various pieces of information, weigh them qualitatively in prose, and then attach a probability estimate without an explicit computational pathway from evidence to forecast. This creates two problems: qualitative balancing can yield suboptimally calibrated probabilities, and the absence of auditable intermediate components makes forecasting errors difficult to localize and interpret.

To address these issues, we propose AUDITFORECAST, a structured agentic forecasting scaffold motivated by a bayesian perspective. AUDITFORECAST decomposes each forecast into two explicit components: (1) a base probability capturing the default trajectory implied by a quantitative model, and (2) a set of evidence-grounded factor updates capturing situational developments outside the model's standing assumptions. These components are combined mechanically in odds space, yielding a traceable derivation from evidence to forecast. By enforcing this decomposition as a schema-level constraint, AUDITFORECAST enables per-

component attribution, explicit probabilistic updating, and post-hoc auditing. Across three live forecasting benchmarks, AUDITFORECAST improves forecasting accuracy and calibration relative to strong agentic scaffolds, enables frontier LLMs to surpass market-implied baselines, and transfers effectively to smaller backbones. Beyond these empirical gains, it improves interpretability by exposing the intermediate probabilistic structure of each forecast and making the forecasting process auditable.

2. Method

2.1. Overview

A broad class of real-world forecasting questions admits a natural quantitative anchor. A stock threshold event can be modeled from a return distribution, while a macroeconomic target can be projected from time-series dynamics. Such models capture the *default trajectory* of the world and provide a natural probabilistic baseline. Yet no model captures every case-specific development relevant to the forecast. We refer to these model-external developments as *situational surprises*. Forecasting therefore requires combining a model-derived baseline with evidence-grounded updates outside its scope. Existing unstructured agents combine these sources narratively, without explicit accounting for how the final probability is produced. We propose AUDITFORECAST, an agentic scaffold that decomposes each forecast into a base probability p_0 derived from the default trajectory and a set of factor updates outside the scope of p_0. These components are combined mechanically in odds space. By enforcing this decomposition as a schema-level constraint, AUDITFORECAST makes the forecasting process traceable, auditable, and interpretable. Appendix B illustrates the AUDITFORECAST workflow.

2.2. Bayesian Decomposition

AUDITFORECAST is motivated by a bayesian odds-space formulation of forecasting. The default trajectory induces a prior belief over the outcome, while each validated factor contributes an evidence-supported update. We express each update as an odds multiplier, equivalently a likelihood ratio (LR), so that factor contributions combine multiplicatively in odds space. We focus on binary event forecasting in this work. Let $p_0 = P(\mathrm{YES})$ denote the base probability under the default trajectory, with corresponding odds $o_0 = \frac{p_0}{1-p_0}$. For each validated factor i, the agent assigns an odds multiplier $\mathrm{LR}_i = \frac{P(E_i|\mathrm{YES})}{P(E_i|\mathrm{No})}$, where E_i denotes the evidence supporting that factor. Operationally, LR_i is the multiplicative term by which factor i updates the odds. Assuming conditional independence across factors, the final odds and probability are $o = o_0 \prod_{i=1}^{n} \mathrm{LR}_i$ and $p = \frac{o}{1+o}$. This decomposition underlies AUDITFORECAST's auditability: p_0 is explicit, each factor contributes a single quantitative update term, and the final probability p is obtained through a mechanical aggregation step. A detailed derivation of this formulation is provided in Appendix C.

2.3. Base Probability Derivation

The first stage derives the base probability p_0, defined as the probability of YES under the default trajectory before any case-specific factor updates are applied. The agent first performs lightweight retrieval to establish the forecasting context, including the current state of the question, the relevant entities, and the resolution horizon. It then applies *model-first anchoring*: before conducting open-ended evidence search, the agent determines whether the question admits a suitable quantitative model for baseline estimation. If so, the agent commits to that model and records the model choice, required inputs, and assumptions. This commitment induces *model-guided search*: subsequent retrieval targets the specific inputs required by the chosen model rather than proceeding in an unrestricted manner. These inputs are passed to code execution, which computes p_0 mechanically rather than through qualitative estimation. If no suitable quantitative model is identified, the agent falls back to a qualitative baseline while still recording its assumptions and sources. This stage therefore outputs both a base probability p_0 and an auditable record of how it was obtained.

2.4. Factor Identification

With p_0 established, the second stage identifies *factors*: evidence-grounded situational developments that lie outside the scope of p_0 and shift the odds of the outcome. Each candidate factor must satisfy three constraints. **(1) Out-of-Scope.** The factor must not already be encoded in the derivation of p_0. For each accepted factor, the agent records an explicit `why_not_in_p0` justification. This prevents double-counting between the baseline and factor updates. **(2) Independence.** Listed factors must not share the same underlying causal mechanism. When two candidates trace to the same root cause, they are merged into a single factor rather than updated separately. This preserves the conditional-independence assumption underlying multiplicative odds updates. **(3) Evidence-Supported.** Every factor must be grounded in concrete, current external evidence retrieved by the agent. The supporting evidence is recorded explicitly, creating a traceable link from each factor to its source material. Any candidate that fails one of these checks is revised or discarded. The output of this stage is therefore a set of validated factors that are both mathematically well-formed and empirically grounded.

2.5. Factor Assessment and Aggregation

The third stage converts the validated factors into the final forecast. For each factor i, the agent assigns an odds multiplier LR_i that summarizes the direction and magnitude of

Table 1. Brier score (↓) on Polymarket and Kalshi. Average is the overall Brier across all questions. The MARKET-IMPLIED reference for Gemini-3-flash differs from that of the GPT backbones because Gemini is evaluated on a different live time range.

Method	Polymarket				Kalshi			
	Stocks	Crypto	Sports	Average	Finance	Entertain.	Sports	Average
GPT-5.4								
Think&Search	0.1063	0.1599	0.2538	0.1342	0.0588	0.1276	0.2258	0.1028
Market-Implied	0.1265	0.1582	0.2540	0.1494	0.0759	0.1073	0.2263	0.1060
AuditForecast	**0.0950**	**0.1257**	**0.2517**	**0.1225**	**0.0524**	**0.0943**	**0.2119**	**0.0870**
GPT-5.4-mini								
Think&Search	0.1116	0.1531	0.2600	0.1387	0.0704	0.1862	0.2296	0.1280
Market-Implied	0.1265	0.1582	0.2540	0.1494	0.0759	**0.1073**	0.2263	**0.1060**
AuditForecast	**0.1021**	**0.1240**	**0.2465**	**0.1270**	**0.0567**	0.1555	**0.2041**	0.1074
Gemini-3-flash								
Think&Search	0.1197	0.2252	0.2477	0.1603	0.0972	0.2458	0.2549	0.1699
Market-Implied	0.1251	0.2071	0.2452	0.1617	0.0697	**0.1143**	0.2588	**0.1166**
AuditForecast	**0.0982**	**0.2021**	**0.2398**	**0.1420**	**0.0651**	0.1695	**0.2441**	0.1284

its evidential impact: $\mathrm{LR}_i > 1$ favors YES, $\mathrm{LR}_i < 1$ favors NO, and $\mathrm{LR}_i = 1$ is neutral. The agent records both the supporting assessment and the resulting multiplier. Each factor is thus represented by a single scalar update term, which is the minimal quantity required for auditable odds-space aggregation. Once p_0 and $\{\mathrm{LR}_i\}_{i=1}^{n}$ are available, the final probability is computed mechanically using the equations in Section 2.2. This arithmetic is carried out via code execution. By the end of stage 3, every component of the forecast is explicit: the p_0 model and its inputs, the validated factors and their evidence, the per-factor odds multipliers, and the final aggregation step. The resulting forecast can therefore be decomposed post hoc into its intermediate contributions.

3. Experiments

3.1. Experimental Settings

Agent Backbones and Implementation. Across all experiments, we consider three frontier LLM backbones: GPT-5.4, GPT-5.4-mini, and Gemini-3-flash. The same set of backbones is used for both AUDITFORECAST and our baselines, so any performance difference can be attributed to the agentic scaffold rather than the underlying model. We build the agentic scaffold using the OpenAI Agents SDK (OpenAI, 2025b) for GPT-5.4 and GPT-5.4-mini, and Google's Agent Development Kit (Google, 2025) for Gemini-3-flash.

Benchmarks and Evaluation. We evaluate on three live forecasting benchmarks: two public prediction markets, Polymarket (Polymarket, 2026) and Kalshi (Kalshi, 2026), and one academic multi-domain benchmark, FutureX (Zeng et al., 2026). Our evaluation covers questions resolving in April–May 2026, restricted to those with verified binary resolutions: 831 questions for GPT-5.4 and GPT-5.4-mini, and 709 questions for Gemini-3-flash. Following standard practice in probabilistic forecasting (Halawi et al., 2024b; Schoenegger et al., 2024; Yang et al., 2026), we report the Brier score $\mathrm{Brier} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$, where $p_i \in [0,1]$ is the forecast for question i and $y_i \in \{0,1\}$ is the realized outcome. Smaller scores indicate better forecasts.

Baselines. We compare against two baselines. THINK & SEARCH is the default agentic scaffold used by both the ProphetArena Leaderboard (ProphetArena, 2026) and FutureX as the strongest frontier-LLM configuration for forecasting tasks: it interleaves reasoning with web search, but does not impose explicit structure on forecasting reasoning. MARKET-IMPLIED denotes the probability derived from prediction-market prices at each question's *run-time* (i.e., when the agent produced its forecast), available for the Polymarket and Kalshi subsets. It captures the aggregate forecast of market participants and serves as a reference point.

3.2. Main Results

AUDITFORECAST consistently outperforms baseline scaffolds with the same backbone, enables frontier LLMs to surpass market performance, and generalizes to smaller LLMs, allowing them to exceed the forecasting performance originally achieved by larger models. Table 1 and Table 2 (Appendix D) summarize the live evaluation results on public prediction platforms (i.e., Polymarket and Kalshi) and on FutureX, respectively. We observe consistent and substantial gains over the baseline scaffold across all categories. On Polymarket, every backbone surpasses the market baseline, while on Kalshi, GPT-5.4 also exceeds the market. Furthermore, GPT-5.4-mini under AuditForecast even outperforms GPT-5.4 under the baseline scaffold on Polymarket, suggesting that our framework unlocks additional forecasting capability from underlying base models.

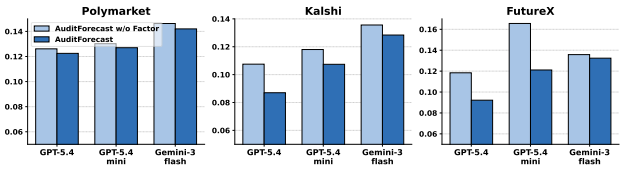

Figure 2. Brier score (↓) of AUDITFORECAST with and without factor updates, on Polymarket, Kalshi, and FutureX. Factor updates consistently improve forecasting performance.

In AUDITFORECAST, factor updates further calibrate the base probability and thereby improve forecasting accuracy. Figure 2 reports the Brier scores of AUDITFORECAST with and without factor updates across three backbones on Polymarket, Kalshi, and FutureX. In all settings, factor updates consistently improve performance, demonstrating the value of the bayesian decomposition in refining base probabilities with evidence-grounded adjustments.

3.3. Auditing the Base Probability

In Section 3.2, we show that AUDITFORECAST yields calibrated gains across three live forecasting benchmarks. Here, we use its structured decomposition to audit intermediate reasoning components and expose forecasting-specific behaviors that remain hidden under unstructured scaffolds. We focus on the base probability p_0 and ask three questions: **Q1:** When appropriate, do agents estimate p_0 through quantitative modeling? **Q2:** For questions of the same type, do agents converge on the same modeling family? **Q3:** Even when the modeling family is shared, do agents agree on the data source used to instantiate it? Our analysis focuses primarily on Polymarket stock price range questions.

Base Probability Hierarchy Extraction. To analyze how agents construct p_0, we induce a two-level taxonomy of p_0 computations with an LLM-as-judge. Our scaffold logs each p_0 in a structured form recording the modeling method, data sources, and explicit assumptions. We feed these records to a judge LLM in incremental batches from an empty taxonomy; at each stage, the judge may add, refine, or split categories while keeping earlier codes stable for backward compatibility. The resulting hierarchy has two levels: Level 1 captures the modeling family, and Level 2 captures the data source supplying the key input. A second judge LLM then independently assigns each record to a Level 2 leaf under the finalized hierarchy. This bottom-up procedure yields a method-by-data-source taxonomy aligned with how agents actually reason, rather than one imposed top-down.

Results. Across 368 stock forecasts, GPT-5.4 agents under AUDITFORECAST arrive at p_0 through a highly consistent set of approaches. As many as 91.8% of forecasts fall into the *Parametric CDF* family, in which the asset price at resolution is assumed to follow a parametric distribution (typically a zero-drift lognormal), and p_0 is computed by evaluating the cumulative distribution function (CDF) at the question threshold. The remaining forecasts are distributed across direct option-derived proxies (4.6%), empirical or hybrid methods (2.2%), and only 1.4% based on purely qualitative judgment. These findings answer **Q1** and **Q2**:

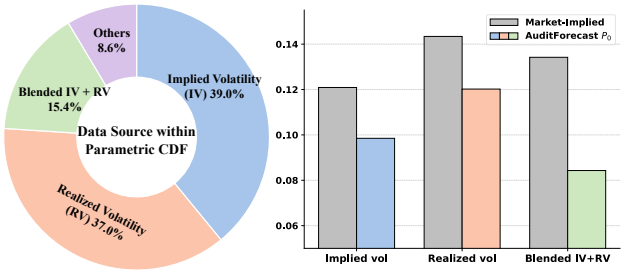

Figure 3. GPT-5.4 under AUDITFORECAST on stock forecasting questions. **Left:** Distribution of data-source choices within the *Parametric CDF* family for P_0. **Right:** Brier score of the dominant data-source subtypes versus the MARKET-IMPLIED baseline.

Findings for Q1–Q2

When forecasting questions admit a clear quantitative anchor, AUDITFORECAST induces agents to use quantitative models in nearly all cases and to converge on a shared modeling family.

While AUDITFORECAST stabilizes agents' choice of modeling family for questions of the same type, we find that agents still diverge in their choice of data source for instantiating the model. As shown in Figure 3 (left), within the *Parametric CDF* family, agents primarily split among implied volatility, realized volatility, and deliberate combinations of the two when selecting the volatility input. Furthermore, Figure 3 (right) reports the Brier score of each dominant data-source subtype against the MARKET-IMPLIED baseline on the matched question subset. All three subtypes' p_0 achieves lower Brier than MARKET-IMPLIED, indicating that the Parametric CDF template, when instantiated with any reasonable volatility input, already yields a competitive base probability. These findings answer **Q3**:

Findings for Q3

Under AUDITFORECAST, even when agents converge on the same modeling family, they may still diverge in how that model is instantiated, particularly in the choice of data source.

4. Conclusion

We present AUDITFORECAST, an agentic scaffold for probabilistic forecasting that replaces implicit narrative aggregation with an explicit decomposition into a base probability and evidence-grounded factor updates. Across three live benchmarks, it improves forecasting accuracy while making the forecasting process more interpretable. A natural next step is factor auditing: understanding how agents identify model-external developments, assign odds multipliers, and propagate these updates to the final forecast.

Impact Statement

This work aims to make LLM-agent forecasting systems more accurate, calibrated, and interpretable. By introducing an explicit and auditable update process, AUDITFORECAST may improve transparency in applications where probabilistic forecasts inform real-world decisions. At the same time, more capable forecasting agents may be used in high-stakes settings where overreliance on model outputs could be harmful. Forecasts from such agents should therefore be treated as decision-support tools rather than authoritative judgments, and should remain subject to human oversight.

References

Chandak, N., Goel, S., Prabhu, A., Hardt, M., and Geiping, J. Scaling open-ended reasoning to predict the future. *arXiv preprint arXiv:2512.25070*, 2025.

Google. Agent Development Kit (ADK), 2025. URL https://google.github.io/adk-docs/.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. *Advances in Neural Information Processing Systems*, 37: 50426–50468, 2024a.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=FlcdW7NPRY.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hyndman, R. J. and Athanasopoulos, G. *Forecasting: principles and practice*. OTexts, 2018.

Jain, N., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Kalshi. Kalshi, 2026. URL https://kalshi.com/.

Liu, J., Li, Y., Zhang, C., Li, J., Chen, A., Ji, K., Cheng, W., Wu, Z., Du, C., Xu, Q., et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025.

Luo, J., Zhang, W., Yuan, Y., Zhao, Y., Yang, J., Gu, Y., Wu, B., Chen, B., Qiao, Z., Long, Q., et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

OpenAI. Introducing deep research, 2025a. URL https://openai.com/index/introducing-deep-research/.

OpenAI. OpenAI Agents SDK, 2025b. URL https://openai.github.io/openai-agents-python/.

Polymarket. Polymarket, 2026. URL https://polymarket.com/.

ProphetArena. ProphetArena Leaderboard: Agent Forecasting, 2026. URL https://prophetarena.co/leaderboard/forecast?tab=agent.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741, 2023.

Rapach, D. E. and Zhou, G. Forecasting stock returns. handbook of economic forecasting, 2 (part a), graham elliott and allan timmermann, 2013.

Schoenegger, P. and Park, P. S. Large language model prediction capabilities: Evidence from a real-world forecasting tournament. *arXiv preprint arXiv:2310.13014*, 2023.

Schoenegger, P., Tuminauskaite, I., Park, P. S., Bastos, R. V. S., and Tetlock, P. E. Wisdom of the silicon crowd: Llm ensemble prediction capabilities rival human crowd accuracy. *Science Advances*, 10(45):eadp1528, 2024.

Schoenegger, P., Jones, C. R., Tetlock, P. E., and Mellers, B. Prompt engineering large language models' forecasting capabilities. *arXiv preprint arXiv:2506.01578*, 2025.

Turtel, B., Franklin, D., and Schoenegger, P. Llms can teach themselves to better predict the future. *arXiv preprint arXiv:2502.05253*, 2025a.

Turtel, B., Franklin, D., Skotheim, K., Hewitt, L., and Schoenegger, P. Outcome-based reinforcement learning to predict the future. *arXiv preprint arXiv:2505.17989*, 2025b.

Wang, X., Feng, M., Qiu, J., Gu, J., and Zhao, J. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Yang, Q., Mahns, S., Li, S., Gu, A., Wu, J., and Xu, H. LLM-as-a-prophet: Understanding predictive intelligence with prophet arena. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VpiHkMSPqI.

Zeng, Z., Liu, J., Chen, S., He, T., Liao, Y., Tian, Y., wangjinpeng.levi, Wang, Z., YangYang, Yin, L., Yin, M., Zhenwei, Z., Cai, T., Chen, X., Chen, Z., Chen, J., Du, Y., Gao, X., Guo, J., HU, L., Jiao, J., Li, X., Liu, J., nishuang, Wen, Z., Zhang, G., Zhang, K., Zhou, X., Blanchet, J., Qiu, X., Wang, M., and Huang, W. Futurex: An advanced live benchmark for LLM agents in future prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z28PLIEj6l.

A. Related Work

LLM forecasting. Early work on LLM forecasting primarily studies the forecasting ability of standalone language-model systems, rather than explicitly agentic scaffolds. Schoenegger & Park (2023) evaluate GPT-4 in a real-world forecasting tournament and show that it underperforms the human crowd. Schoenegger et al. (2024) find that aggregating forecasts across multiple LLMs substantially improves performance, revealing a "wisdom of the silicon crowd" effect. Related work also explores retrieval-augmented pipelines and prompt design, showing that better information access can improve forecasting performance (Halawi et al., 2024a), while prompting alone appears to yield only limited gains (Schoenegger et al., 2025). Together, these works establish the promise of LLM-based forecasting, but do not explicitly structure how evidence is translated into a probabilistic forecast.

Training-based forecasting improvement. Recent work has also begun to improve forecasting through training-based methods. Halawi et al. (2024a) introduce a self-supervised fine-tuning procedure within a retrieval-augmented system. More recent approaches move further toward outcome-based learning, including self-play with DPO (Turtel et al., 2025a; Rafailov et al., 2023), outcome-based reinforcement learning on prediction-market questions (Turtel et al., 2025b), and large-scale reinforcement learning on synthetic open-ended forecasting questions derived from news (Chandak et al., 2025). Adjacent work has also explored LLM-based time-series forecasting with external textual context, such as integrating news and reflection into numerical forecasting pipelines (Wang et al., 2024). These works improve forecasting ability through stronger supervision, whereas our focus is on imposing an explicit probabilistic scaffold on the forecasting process itself.

Live forecasting benchmarks and agent evaluation. Recent work has also introduced live forecasting benchmarks in dynamically updating settings. Yang et al. (2026) propose ProphetArena, a live benchmark for studying LLM predictive intelligence through probabilistic forecasting, calibration, and market-based evaluation. Zeng et al. (2026) extend this direction to LLM agents, introducing a large-scale live benchmark that evaluates future prediction together with search, planning, and tool use in dynamic environments. These benchmarks provide an important evaluation substrate for forecasting systems and agents. By contrast, our work focuses on the forecasting scaffold itself, with the goal of making the mapping from evidence to probability explicit, auditable, and better calibrated.

B. Example AuditForecast Workflow

AUDITFORECAST organizes the agent's behavior into three explicit stages: deriving a base probability p_0, identifying model-external factors, and assessing and aggregating factor updates. While the underlying agent may iterate within a stage through search, reasoning, and computation, the scaffold constrains the final output to a structured forecasting report so that the path from evidence to probability remains explicit and auditable. Figure 4 (left) illustrates the corresponding agent workflow under the scaffold. The right side of Figure 4 shows the forecasting report produced by AUDITFORECAST. By making the intermediate objects explicit, the scaffold supports direct inspection of how the forecast was constructed and how each retained factor contributed to the final prediction. It also enables clearer post hoc analysis once the event is resolved, providing a more transparent view of LLM-agent forecasting behavior.

C. Bayesian Derivation

Let $Y \in \{\text{YES}, \text{NO}\}$ denote the binary event outcome, and let $E = (E_1, \ldots, E_n)$ denote the evidence associated with a set of validated factors. Applying Bayes' rule gives

$$P(Y = \text{YES} \mid E) = \frac{P(E|Y=\text{YES})P(Y=\text{YES})}{P(E)}, \tag{1}$$

$$P(Y = \text{NO} \mid E) = \frac{P(E|Y=\text{NO})P(Y=\text{NO})}{P(E)}. \tag{2}$$

Taking the ratio of Equations (1) and (2) yields

$$\frac{P(Y=\text{YES}|E)}{P(Y=\text{NO}|E)} = \frac{P(E|Y=\text{YES})}{P(E|Y=\text{NO})} \cdot \frac{P(Y=\text{YES})}{P(Y=\text{NO})}. \tag{3}$$

Define the odds of the YES outcome as

$$\text{odds}(Y = \text{YES}) = \frac{P(Y=\text{YES})}{P(Y=\text{NO})}. \tag{4}$$

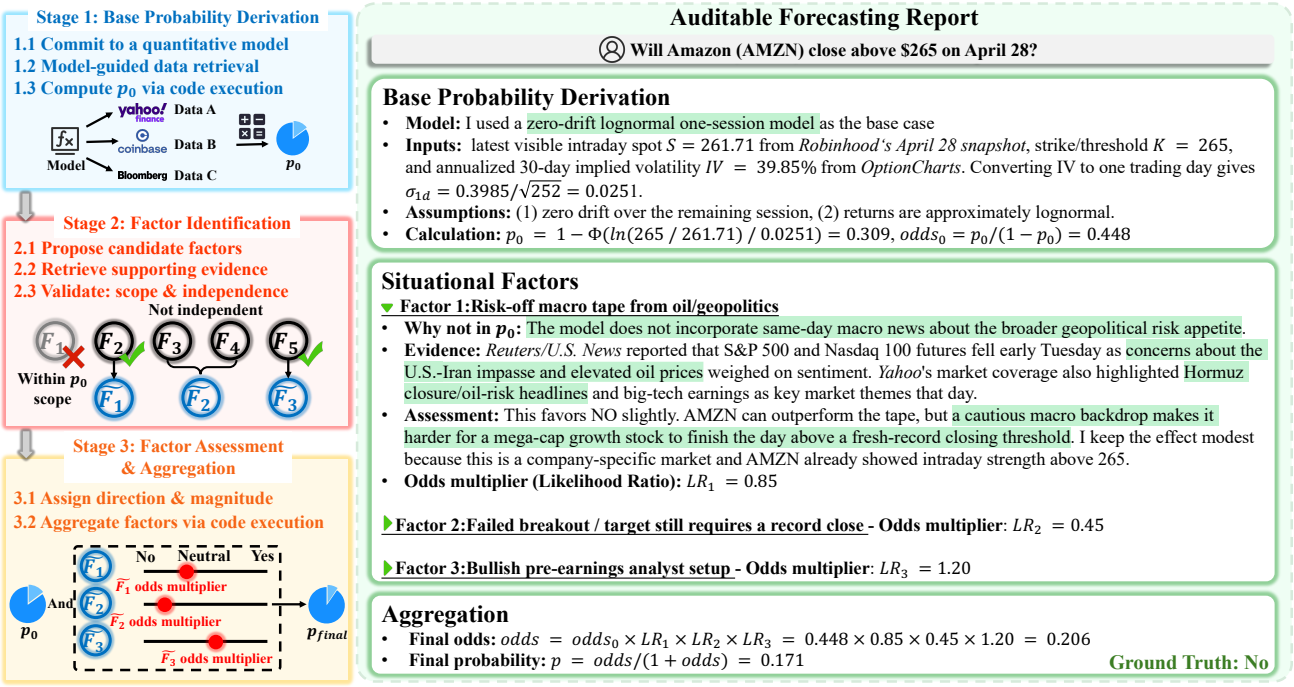

Figure 4. **Left:** the agent workflow under AUDITFORECAST. **Right:** the corresponding auditable forecasting report. Highlighted text emphasizes key information and steps in the forecasting process.

Then Equation (3) can be written as

$$\text{odds}(Y = \text{YES} \mid E) = \text{odds}(Y = \text{YES}) \cdot \frac{P(E \mid Y=\text{YES})}{P(E \mid Y=\text{NO})}. \tag{5}$$

showing that posterior odds equal prior odds multiplied by a likelihood-ratio term. If the evidence variables are conditionally independent given the event outcome, then

$$P(E \mid Y = \text{YES}) = \prod_{i=1}^{n} P(E_i \mid Y = \text{YES}). \tag{6}$$

and similarly,

$$P(E \mid Y = \text{NO}) = \prod_{i=1}^{n} P(E_i \mid Y = \text{NO}). \tag{7}$$

Substituting Equations (6) and (7) into the likelihood-ratio term in Equation (5) gives

$$\begin{aligned} \frac{P(E \mid Y = \text{YES})}{P(E \mid Y = \text{NO})} &= \frac{\prod_{i=1}^{n} P(E_i \mid Y = \text{YES})}{\prod_{i=1}^{n} P(E_i \mid Y = \text{NO})} \\ &= \prod_{i=1}^{n} \frac{P(E_i \mid Y = \text{YES})}{P(E_i \mid Y = \text{NO})}. \end{aligned} \tag{8}$$

Define the factor-level likelihood-ratio term

$$\text{LR}_i = \frac{P(E_i \mid Y=\text{YES})}{P(E_i \mid Y=\text{NO})}. \tag{9}$$

Then the posterior odds become

$$\text{odds}(Y = \text{YES} \mid E) = \text{odds}(Y = \text{YES}) \prod_{i=1}^{n} \text{LR}_i. \tag{10}$$

In AUDITFORECAST, the agent instantiates the prior odds through a quantitative-model-derived base probability p_0,

$$o_0 = \frac{p_0}{1-p_0}. \tag{11}$$

and the final odds and probability are therefore written as

$$o = o_0 \prod_{i=1}^{n} \text{LR}_i, \quad p = \frac{o}{1+o}. \tag{12}$$

D. Results on FutureX

We report the detailed live evaluation results on FutureX in Table 2. These results complement the main results in Section 3.2 and provide the full breakdown by backbone and benchmark level.

Table 2. Brier score (↓) on FutureX. Average is the overall Brier across all questions.

Method	Level 1	Level 2	Average
GPT-5.4			
Think&Search	0.2879	0.1091	0.1274
AuditForecast	**0.0932**	**0.0921**	**0.0922**
GPT-5.4-mini			
Think&Search	0.0970	0.1251	0.1223
AuditForecast	**0.0965**	**0.1239**	**0.1211**
Gemini-3-flash			
Think&Search	0.0728	0.1837	0.1671
AuditForecast	**0.0675**	**0.1438**	**0.1324**