HINDCAST: Replaying Prediction Markets to Evaluate LLM Forecasters

Anonymous ACL submission

Abstract

Forecasters are evaluated by backtesting, replaying resolved questions to grade the probability the system would have assigned before the outcome was known. For LLMs, two channels leak the answer into this test. Web retrieval can surface reports written after the event, turning forecasting into a lookup task. And each new model is trained on more recent data, so a question that lay in the future for last year's models lies inside this year's training data, where the answer is already memorized. Either way, the test rewards recall rather than foresight. We introduce HINDCAST, which closes both leaks by grading a model as if it stood at a chosen past date t_0, before the outcome existed in either channel. HINDCAST replays resolved Polymarket prediction markets against a frozen snapshot of public Reddit, lets the model read only posts written before t_0, and scores each forecast against both what actually happened and the market's own price at t_0, which is itself a human forecast made from the same past information. Because the cutoff is set per market and the snapshot never changes, the evaluation re-runs on new markets as models improve, without going stale. With this leakage removed, we find that retrieval helps most models forecast, but only when Reddit discussed the event beforehand. Otherwise it feeds on speculation and backfires.

1 Introduction

Calibrated probabilistic forecasting is becoming a standard test of whether a language model can reason under uncertainty, and increasingly a way to decide which agents to trust with open-ended tasks. Writing forecasting questions by hand is slow, so the common shortcut is to grade models on *resolved* questions whose answers are now known and to score the probability the model assigns to the true outcome (Halawi et al., 2024; Ye et al., 2024). The supervision is automatic and cheap, but it can reward memory instead of foresight. A model asked today for the 2022 World Cup champion (Figure 1) names Argentina with near-certainty not because it forecasts well but because it has read the post-tournament write-ups. At inference, a model that retrieves can pull in content created after the event, and across releases each new model is trained on data closer to the event, so a question safely out-of-cutoff for this year's models leaks into next year's (Yan et al., 2026; Levine et al., 2026). Either way, an evaluation that looks like forecasting today is quietly ineffective.

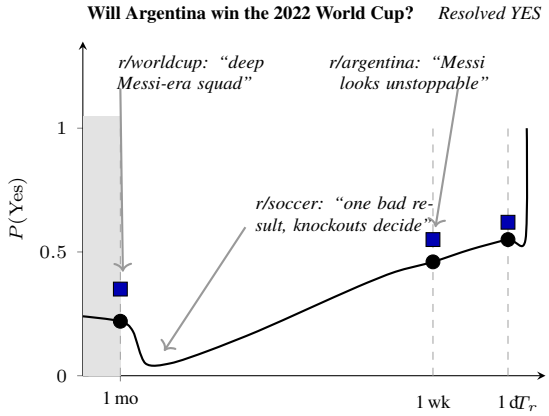

Figure 1: **HINDCAST on a resolved market (illustrative).** At a simulated query time t_0, set here to a one-month lookback for illustration (in our experiments t_0 is pinned per market, see §4.2), the agent sees only Reddit content created before t_0 and emits a forecast (■). We score it against the market's implied probability at the same t_0 (●). Argentina's group-stage loss to Saudi Arabia drops the market sharply, but Reddit's discussion across the tournament stays directional, and the agent's forecast tracks the longer-horizon narrative rather than the short-term price move.

What we actually want to measure is the counterfactual probability the model *would* have assigned at a past time t_0, before the outcome was known. This is also the only forecast that was ever useful, since it is the one a trader or analyst could have

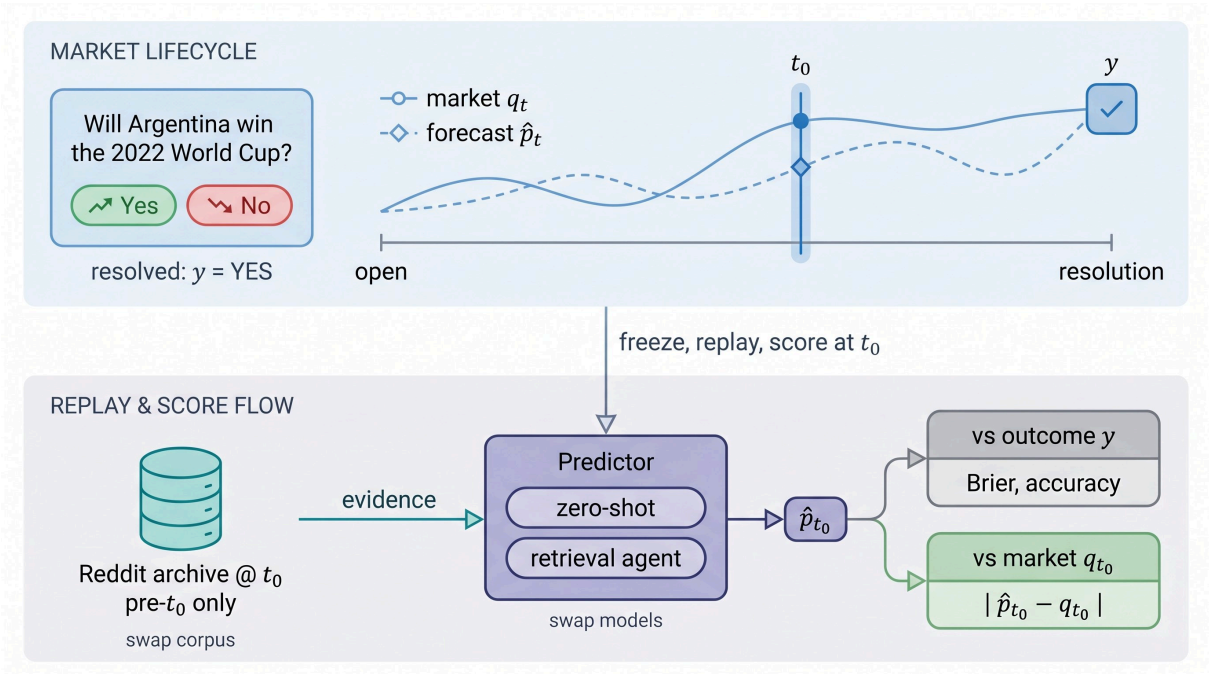

Figure 2: **Overview of HINDCAST.** Any resolved Polymarket market can be frozen at any past time t_0 along its open-to-resolution lifecycle (top). At the chosen t_0, the Reddit archive is restricted to documents created before t_0, and a predictor (zero-shot baseline or retrieval-grounded agent) reads that frozen evidence and emits a probability $\hat{p}_{t_0}$. The forecast is scored against both the resolved outcome y (Brier, accuracy) and the contemporaneous market-implied probability q_{t_0} ($|\hat{p}_{t_0} - q_{t_0}|$). Both the corpus and the predictor can be swapped without changing the protocol, and sweeping t_0 along the lifecycle yields a calibration trajectory for any benchmarked model.

acted on at t_0 (Tetlock and Gardner, 2015).But it is hard to recover after the fact, because the live web has moved on. Pages are rewritten and re-indexed, so retrieval can no longer return what was actually visible at t_0. Live benchmarks (Karger et al., 2025; Zeng et al., 2025) avoid this by posing questions that are still open and waiting for them to resolve, but waiting forfeits the two things resolved questions provide together, namely a known outcome to grade against and, for a prediction market, a human forecast at t_0 to measure against. We therefore ask the following.

> *On already-resolved questions, can we evaluate the forecast a model would have made at a past t_0, holding fixed the information it could retrieve then?*

We answer with HINDCAST, which makes this counterfactual measurable through *hindcasting*, by replaying each resolved market against a frozen archive of the past and pinning every retrieval call to that market's t_0. Two design choices follow directly. First, the archive must not change after the fact, so we build it from public Reddit, whose monthly snapshots are immutable (unlike web pages or Wikipedia, a captured month stays captured) and which discusses unfolding events early and broadly, giving a forecaster real pre-event signal to find. Second, the questions need a yardstick, so we draw on resolved Polymarket markets, each of which supplies the graded outcome and a matched-time price at t_0 that is itself a human forecast distilled by traders from the same public information. Because the past is fixed per market, t_0 becomes a free variable and the evaluation re-runs on markets that resolve after a new model ships, so it never decays into recall. The concurrent FutureSim (Goel et al., 2026) shares the replay idea but differs on two axes: it streams a news corpus forward over a single shared window to measure adaptation, while HINDCAST pins an immutable Reddit archive to a per-market t_0 and scores against the matched-time market price.

Predictors reach the archive through one probability-emitting interface, instantiated twice as a *zero-shot baseline* that forecasts from the model alone and isolates parametric memory, and as a *Retrieval agent* grounded in evidence pulled from the archive at t_0. Figure 2 sketches the pipeline.

With a leakage-free evaluation in hand, we

revisit whether retrieval-grounded reasoning improves forecasting, a claim prior work makes under possible contamination. If those reported gains came partly from reading the future, they should shrink once the future is sealed off. The gain does not vanish, since retrieval still lowers Brier on eight of the 9 open-weight models we test. But it concentrates on markets the corpus genuinely covered in advance and reverses where the agent over-reads speculative chatter (§7). Retrieval helps, then, but only when the pre-t_0 corpus already carried outcome-relevant discussion of the event; where it carried only speculation, retrieval backfires.

Contributions. (i) *Hindcasting*, an evaluation protocol that measures a model's forecast at a past t_0 on resolved questions, holding its retrievable information fixed, and that re-runs as models advance (§3, §4). (ii) An immutable, temporally pinned public-Reddit archive serving leakage-free retrieval at any past t_0 (§3). (iii) Two reference predictors behind one probability-emitting interface, a zero-shot baseline and a Retrieval agent (§5). (iv) An empirical study locating where retrieval-grounded forecasting helps and where it backfires (§6).

Taken together, what limits a retrieval-grounded forecaster is less its parametric reasoning than the evidence environment: where the corpus carried fact-anchored, outcome-relevant discussion the models read the answer off it, but where it carried only speculative chatter they could not separate it from signal, and retrieval backfired.

2 Related Work

LLM forecasters and benchmarks. Early benchmarks frame future-event prediction as temporally restricted QA over news or tournament questions (Jin et al., 2021; Zou et al., 2022). A newer wave of retrieval-augmented and agentic systems extends this into open-ended forecasting against eventual outcomes (Halawi et al., 2024; Ye et al., 2024; Das et al., 2026). Concurrent with our work, FutureSim (Goel et al., 2026) streams news forward chronologically over a recent window to measure adaptation. HINDCAST instead holds the corpus fixed at a chosen t_0 per market and measures improvement over the matched-time market price. Live benchmarks counter static-test contamination through continual collection (Karger et al., 2025; Zeng et al., 2025; Yang et al., 2025), and a parallel line moves closer to prediction-market deployment by formulating the task as trading over live market states (Yu et al., 2025; Cheng et al., 2026). HINDCAST is the inverse move, replacing online evaluation with replay against a fixed past information environment, so any predictor can be re-run against any past t_0 without waiting for new questions to resolve.

Temporal controls and contamination. Existing responses to contamination either refresh evaluation items (Vu et al., 2024; White et al., 2025; Jain et al., 2025; Xu et al., 2024), retrain models from scratch on temporally-restricted corpora that are either annually partitioned (Yan et al., 2026) or pinned to a single pre-cutoff vintage (Levine et al., 2026), or fix the retrieval context to a static news snapshot (Chandak et al., 2025; Nagel, 2016; Cheng et al., 2024). The retraining strategies attack the parametric-memory channel by removing post-cutoff text from the weights, whereas HINDCAST attacks the retrieval channel by pinning the archive at t_0, so the two approaches are complementary. We instead treat retrieval itself as the experimental object. Archived Reddit at the simulated query time gives the agent a denser, broader-topic information substrate than a single news snapshot can provide.

Prediction markets and calibration. Prediction markets supply resolvable questions, crowd probabilities, and tradable payoffs. Prior work has used them as evaluation targets (Zou et al., 2022; Halawi et al., 2024; Karger et al., 2025), as delayed supervision under proper scoring rules (Turtel et al., 2026), and as trading environments measured by market returns (Yang et al., 2025; Yu et al., 2025; Cheng et al., 2026). We use Polymarket as a dual benchmark. Resolved binary markets supply ground truth, and historical market-implied probabilities supply a matched-time baseline. Together they let us separately measure outcome recovery, improvement over the market's information state at t_0, and calibration as resolution approaches. Calibration is therefore a first-class target, scored under proper rules such as the Brier score (BRIER, 1950; Gneiting and Raftery, 2007). Prior calibration work spans LLM ensembling (Schoenegger et al., 2024), outcome-based training (Turtel et al., 2026; Chandak et al., 2025), and Bayesian factor decomposition (Feng et al., 2025). HINDCAST is a controlled testbed under which such retrieval and calibration strategies can be replayed at matched timepoints and compared against both binary outcomes and contemporaneous market probabilities.

3 Retrieval Backend

The information side of HINDCAST is served by a temporally pinned archive of public Reddit. The backend is deliberately thin and keeps one immutable copy of each document, indexes it for retrieval, and answers every call under a single cutoff predicate, with no document re-ranking, no learned post-filter, and no model-side opinion about which results are relevant. Any difference between predictors therefore stems from how the agent queries and reads the archive rather than from a hidden inductive bias inside the backend.

3.1 Corpus

We build the corpus from the Pushshift archive of public Reddit submissions and comments (Baumgartner et al., 2020), which provides static monthly dumps captured at posting time. The archive spans 25 consecutive monthly snapshots, January 2024 through January 2026, drawn from 20 subreddits that together cover the topics our markets touch.

We retain authored and engaged content and discard the rest, conditioning only on authorship and engagement signals rather than on what a document says. A record is dropped if it lacks a stable id, if its author or body is deleted or removed, if it matches an optional bot list, or if it falls below specific engagement and length floors, namely a voting score below 5, submissions with fewer than 3 comments, comments shorter than 5 non-URL characters or consisting mostly of a link, and any document with fewer than 20 whitespace-delimited words. Since none of these filters condition on subreddit, topic, or stance, the corpus stays uncurated by content. We keep it broad on purpose, since the eval set spans markets across politics, sports, crypto, macroeconomics, and entertainment, and a topic-narrow corpus would advantage predictors on the topics it covers and penalize them everywhere else. After filtering, the archive holds 220,943 submissions and 18,094,365 comments ($\approx 18.3M$ documents). Each document is stored once with its text and metadata (subreddit, author, a coarse month key, and an exact creation timestamp), and the snapshot does not change between t_0 and evaluation even when the post is later edited or removed on the live site.

3.2 Retrieval

The corpus is loaded into PostgreSQL with a `pgvector` embedding column and is reached two ways, namely dense semantic search and direct by-id lookup. For dense search the query is embedded with the Qwen3-Embedding-0.6B encoder of Zhang et al. (2025) and documents are ranked by cosine distance over per-month HNSW indexes (Malkov and Yashunin, 2020). The reference agent restricts this search to submissions through a `doc_type` filter, so every hit is a whole post rather than a fragment of a comment thread. This keeps the retrieved set focused, avoids burying the signal under noisy, near-duplicate comment chatter, and returns self-contained units the agent can reason over. Submissions also carry the title and self-text that best summarise a thread's topic.

The dense search excludes comments, but the archive still holds all 18,094,365 of them and exposes them through by-id lookup tools that resolve a post or comment, walk a submission's comment thread, and list an author's prior posts and comments. A predictor can therefore start from a retrieved submission and pull the surrounding discussion for detail the post text alone does not carry. Every call, whether dense search or a by-id lookup, carries the same filter block, namely the cutoff t_0, the document type, and optional subreddit, author, month, and start-time restrictions, so the agent can narrow a query without ever weakening the cutoff guarantee. The full tool surface and signatures are deferred to Appendix A.1.

3.3 Cutoff enforcement

To make the cutoff guarantee robust to routing errors, we enforce it at two levels. The primary guarantee is a *row-level predicate* under which every document a tool returns is checked against t_0 with the strict comparison $\texttt{created_at} < t_0$, evaluated in the SQL `WHERE` clause of dense search and of all four by-id lookup tools (post, comment, thread, and author). On top of this, a *partition selector* guards and accelerates dense search by deriving from t_0 the monthly partitions no later than t_0's month (and no earlier than any start-time bound), running a separate top-k search against each partition's month-scoped HNSW index, and merging the per-partition results into an exact global top-k. A leak through dense search would thus require both layers to fail on the same document, and a leak through a direct lookup would require the row-level predicate to fail. The boundary partition (the cutoff month itself) is searched, but its post-t_0 rows are removed by the row-level predicate. More details are included in Appendix A.2.

4 Evaluation Setup

4.1 Eval-set construction

A resolved market tests a retrieval-grounded predictor only if the Reddit corpus discussed its event before it resolved, so resolution alone is not a sufficient inclusion criterion. The eval set must also avoid two skews that would otherwise let aggregate accuracy be driven by a constant guess or a single domain, since most Polymarket markets resolve NO and their topics are long-tailed. We therefore construct the eval set in two stages, a coverage test that keeps only well-supported markets followed by outcome and topic balancing.

Coverage test. The candidate pool is the 7,913 binary markets in our Polymarket window (across 1,916 events), of which 3,427 resolve to a Yes/No outcome. For each, an automated probe queries the archive at the market's close date and scores how strongly the pre-event corpus supports the resolved side. An LLM judge (Qwen3-32B) issues 5 queries per side and labels each returned block on a four-level scheme. The label STRONG (weight $+2$) marks explicit, specific evidence for the resolved side, WEAK ($+1$) marks vague, indirect, or correlational evidence, IRREL (0) marks blocks that are on-topic but do not bear on the hypothesis, and CONTRA (-1) marks evidence against the resolved side. Per side this yields a score $s_{\text{side}} = 2\,\text{strong} + \text{weak} - \text{contra}$, and a contrastive statistic $c \in [-1, 1]$ compares the resolved side against its counterfactual. A market the corpus barely mentions scores near zero on both sides and cannot exercise the retrieval pipeline, so we keep only the markets with STRONG coverage.

Outcome and topic balancing. The strongly-covered markets inherit the skews of the raw pool, since 83% of resolved markets are NO (2,851 of 3,427) and coverage is uneven across domains because Reddit discusses sports and trading far more than, say, corporate or weather events. We therefore subsample the strongly-covered markets to equalise the Yes/No outcomes and to spread topics as evenly as the available evidence allows, prioritising outcome balance over topic uniformity. This leaves 216 markets (106 YES and 110 NO), and Figure 3 shows the resulting topic distribution. The probe and this selection run as a single automated pipeline that we release, so the eval set can be regenerated as new markets resolve.

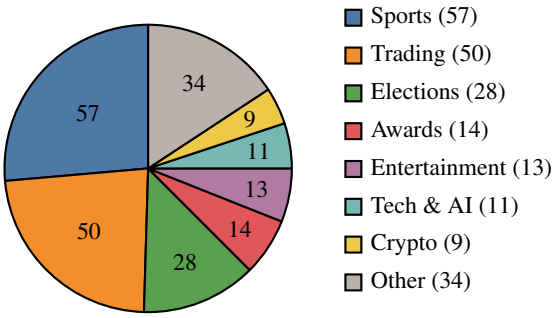

Figure 3: Topic distribution of the 216-market eval set, with slices sized by market count (legend gives counts, slice labels give percent of the set). *Other* aggregates the smaller topics (Social, Geopolitics, Economic, Health, Corporate, and Weather). Topics are unconstrained and follow the markets that carry STRONG pre-event coverage, while outcomes are balanced by construction (106 Yes / 110 No). By market duration the split is 36 short, 129 medium, and 51 long.

4.2 Metrics and cutoff

Each predictor returns a Yes-side probability $\hat{p} \in [0, 1]$ and a chosen option $\hat{y} = \mathbb{1}[\hat{p} > 0.5]$. We score against the resolved binary outcome $y \in \{0, 1\}$ with accuracy and the Brier score,

$$\text{Acc} = \mathbb{1}[\hat{y} = y], \qquad \text{Brier} = (\hat{p} - y)^2. \quad (1)$$

Acc rewards a correct decision and Brier $\in [0, 1]$ jointly penalises miscalibration and miscalls, each averaged over five consistency runs per (model, pipeline) cell. We pin the cutoff t_0 to the day before each market closes, capped at the archive's final day (2026-01-31).

5 Agentic Prediction Pipelines

Both predictor families share an interface, which takes a market, its resolution criteria, an option set, and a cutoff t_0. The interface returns a chosen option, per-option probabilities, a self-reported confidence, and a brief rationale. Figure 4 sketches the two pipelines next to each other.

5.1 Zero-shot baseline

The zero-shot baseline forecasts from a single LLM call with no retrieval, instructed to use only knowledge that could be known at t_0. It serves as the parametric-memory floor against which the Retrieval agent is measured.

5.2 Retrieval agent

The Retrieval agent runs a bounded research loop over the archive at the t_0 cutoff. It opens with seed

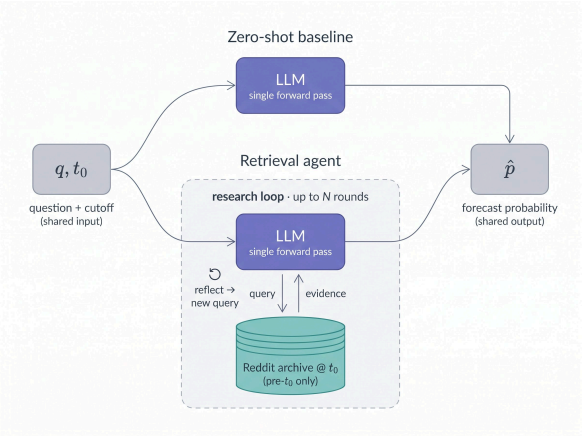

Figure 4: **The two predictor pipelines, side by side.** The zero-shot baseline is a single LLM call. The Retrieval agent runs a bounded loop over the archive at t_0, alternating queries to the archive with a short reflection that proposes the next query, then emits a final probability after up to N rounds. Inputs (q, t_0) and the probability-emitting interface are shared, and only the body of the call differs.

queries targeting the market's claim and the factors that plausibly drive it, then expands promising hits along thread and author dimensions. In subsequent rounds, the model reads the accumulated evidence, writes a short reflection that estimates its confidence and names the evidence it still needs, and proposes new queries. The loop terminates on a confidence signal, on diminishing query yield, or on a round budget. A final forecast pass reads the labeled evidence under the same four-level scheme used by the coverage probe (§4.1) and emits the same probability format as the zero-shot baseline. The two families therefore differ only in whether the archive is consulted. Cutoff enforcement comes from §3.3 rather than from the agent.

Both pipelines forecast at temperature 0.1 with top-p 1.0, and the agent raises the temperature to 0.2–0.3 only for its intermediate query-generation steps and runs a default budget of 3 rounds. Per-stage decoding, retrieval, and forecast-parsing settings are listed in Appendix B.3. A typical trace opens with a parallel batch of seed queries against the archive, expands the highest-ranked submissions with their full text and the author's recent history, and closes with the labeling and final-forecast passes. The exact tool surface (search, post lookup, comment lookup, author lookup) is documented in Appendix A.1, and the prompts driving both pipelines in Appendix B.

6 Experiments

We benchmark 9 open-weight LLMs under both pipelines on the eval set of §4.1 at the cutoff t_0 of §4.2, spanning the Qwen3 family at 4B, 8B, and 32B parameters (Team, 2025b), the Llama 3 family with Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct (Team, 2024), Gemma 3 at 4B and 27B (Team, 2025a), Ministral-8B-Instruct (Mistral AI, 2024), and the DeepSeek-R1 distillation onto Qwen-7B (DeepSeek-AI, 2025). All models share the prompt template, the decoding configuration of §5.2, and the retrieval defaults. All models are released before the starting date of the eval set. Any cross-family or cross-model variation therefore isolates either pretraining differences or the contribution of retrieval. Each (model, pipeline) cell is averaged over five consistency runs.

Table 1 reports the main results. The Retrieval agent lowers Brier on eight of the nine models, with the largest relative reduction on Qwen3-32B (.234 → .179, a 23% drop) and the largest accuracy lift on Qwen3-8B (55.8% → 64.7%). Conditioning on the archive helps most on markets the coverage probe rates *strong*, where retrieval lifts accuracy on most backbones (median +14 points), while *None*-bin accuracy moves little. R1-Distill-Qwen-7B is the only model whose overall Brier rises under retrieval, and §7.3 unpacks this case.

Lookback fraction. The cutoff above is a fixed date, so its effective lookback shifts from market to market because lifetimes differ by orders of magnitude. As a finer probe of how much pre-resolution discussion the agent actually needs, we re-run a two-model subset at four per-market cutoff fractions, placing t_0 at 25%, 50%, 75%, and 100% of each market's open-to-resolution span.

The zero-shot baseline barely moves across the four fractions for either probed model (Ministral-8B and Qwen3-8B), so the parametric-memory floor is not what drives any cross-fraction differences. The Retrieval agent's accuracy and Brier also shift only slightly, with Qwen3-8B rising from 61.0% accuracy at the 25% fraction to 63.4% at the 100% fraction while its Brier hovers near 0.21, and Ministral-8B's numbers essentially unchanged across the sweep. Figure 5 plots Brier against the cutoff fraction for both models and pipelines (per-cell numbers in Appendix C.3), and the gap between the agent and its zero-shot baseline is roughly constant across fractions for each model. The agent's gain therefore saturates well below the

	Zero-shot (Vanilla)					Retrieval (Agent)				
		Acc by coverage					*Acc by coverage*			
Model	Acc ↑	S	W	N	Brier ↓	Acc ↑	S	W	N	Brier ↓
Qwen3-4B	$60.1_{\pm 1.0}$	$45.9_{\pm 1.9}$	$64.9_{\pm 3.3}$	$63.2_{\pm 0.9}$	$.245_{\pm .004}$	$\mathbf{65.9}_{\pm 2.1}$	$73.5_{\pm 6.5}$	$70.3_{\pm 4.6}$	$58.3_{\pm 2.0}$	$\mathbf{.218}_{\pm .006}$
Qwen3-8B	$55.8_{\pm 0.4}$	$61.7_{\pm 2.8}$	$54.7_{\pm 3.2}$	$50.3_{\pm 2.5}$	$.261_{\pm .004}$	$\mathbf{64.7}_{\pm 2.2}$	$80.9_{\pm 4.4}$	$63.7_{\pm 6.2}$	$48.9_{\pm 5.2}$	$\mathbf{.213}_{\pm .008}$
Qwen3-32B	$64.6_{\pm 1.0}$	$65.6_{\pm 3.2}$	$67.9_{\pm 6.2}$	$51.3_{\pm 9.1}$	$.234_{\pm .007}$	$\mathbf{72.8}_{\pm 3.0}$	$79.3_{\pm 5.4}$	$70.5_{\pm 3.6}$	$57.8_{\pm 6.0}$	$\mathbf{.179}_{\pm .008}$
R1-Distill-Qwen-7B	$\mathbf{58.7}_{\pm 1.2}$	$54.2_{\pm 3.6}$	$57.9_{\pm 2.2}$	$52.2_{\pm 3.8}$	$\mathbf{.256}_{\pm .012}$	$51.2_{\pm 5.1}$	$50.9_{\pm 6.8}$	$52.2_{\pm 6.6}$	$48.5_{\pm 7.6}$	$.318_{\pm .015}$
Llama-3.2-3B	$56.1_{\pm 2.1}$	$53.7_{\pm 9.4}$	$43.7_{\pm 14.2}$	$53.0_{\pm 1.4}$	$.375_{\pm .013}$	$\mathbf{56.5}_{\pm 1.6}$	$51.8_{\pm 13.3}$	$68.9_{\pm 26.8}$	$56.7_{\pm 1.3}$	$\mathbf{.292}_{\pm .006}$
Llama-3.1-8B	$\mathbf{66.1}_{\pm 1.0}$	$64.7_{\pm 6.2}$	$67.7_{\pm 2.1}$	$57.5_{\pm 9.1}$	$.250_{\pm .006}$	$60.0_{\pm 1.9}$	$78.3_{\pm 10.1}$	$60.8_{\pm 1.4}$	$40.5_{\pm 7.9}$	$\mathbf{.230}_{\pm .010}$
Gemma-3-4B	$54.5_{\pm 0.2}$	$53.8_{\pm 6.0}$	$54.1_{\pm 3.7}$	$47.7_{\pm 41.2}$	$.297_{\pm .002}$	$\mathbf{56.2}_{\pm 2.5}$	$56.6_{\pm 3.2}$	$56.1_{\pm 2.4}$	$60.3_{\pm 27.2}$	$\mathbf{.262}_{\pm .008}$
Gemma-3-27B	$70.5_{\pm 0.2}$	$70.6_{\pm 2.2}$	$72.1_{\pm 1.5}$	$63.6_{\pm 3.7}$	$.214_{\pm .001}$	$\mathbf{71.1}_{\pm 3.2}$	$87.2_{\pm 4.7}$	$68.7_{\pm 4.5}$	$48.3_{\pm 5.2}$	$\mathbf{.184}_{\pm .008}$
Ministral-8B	$52.9_{\pm 1.2}$	$42.1_{\pm 12.1}$	$60.0_{\pm 3.0}$	$48.2_{\pm 4.7}$	$.281_{\pm .004}$	$\mathbf{56.8}_{\pm 2.9}$	$66.3_{\pm 4.9}$	$59.0_{\pm 5.2}$	$48.3_{\pm 3.8}$	$\mathbf{.242}_{\pm .010}$

Table 1: **Main results on the 216-market eval set at the cutoff t_0 of §4.2.** Overall accuracy (↑, in %), accuracy stratified by the coverage probe's three labels (*S*/*W*/*N*, shaded), and Brier (↓, in points). Model rows are grouped by family via the rules between blocks (Qwen, Llama 3, Gemma 3, Mistral). *Zero-shot* is the bare model and *Retrieval* is the same model with archive access at t_0. Subscripts are run-to-run standard deviation across five consistency runs, and the S/W/N bin assignment is per-run from the agent's own retrieval tally, so per-bin std also reflects bin-membership churn. Best Acc and best Brier per model are bolded.

full available lookback, which suggests the wall-clock offset choice is not picking up an artifact of how late in each market's life the cutoff lands. Full per-run and per-bin numbers are deferred to Appendix C.

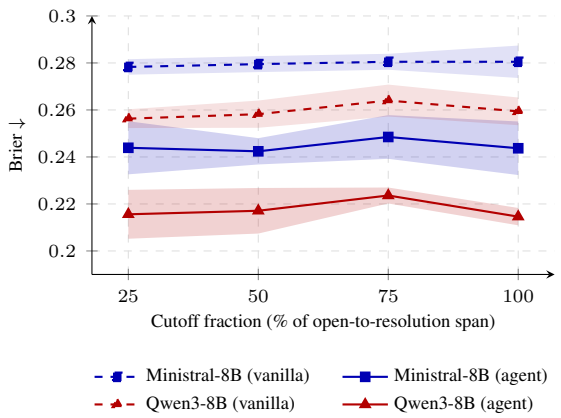

Figure 5: **Brier vs. per-market cutoff fraction for the duration sweep.** Each fraction places t_0 at 25%, 50%, 75%, or 100% of an individual market's open-to-resolution span, so the per-market wall-clock cutoff scales with how long the market was actually open. The Retrieval agent (solid) lowers Brier over the zero-shot baseline (dashed) for both probed models, and the gap between the agent and its zero-shot baseline stays roughly constant across fractions. The flat profile is consistent with retrieval gain saturating well below the full available lookback. Points are means and shaded bands are ±1 standard deviation across five consistency runs per cell.

7 Analysis

The main results in §6 mask a topic-level split. Pairing every Agent and Vanilla run by market id and aggregating across the eight models with per-run logs available, we count per topic how often the agent *recovers* (the agent is right while the zero-shot baseline is wrong) and how often it *regresses* (the reverse). Figure 6 plots the resulting net effect of retrieval per topic, and the split is sharp and runs in both directions.

7.1 Where retrieval helps

Sports (+16), Awards (+12), and Trading/commodities (+7) concentrate the positive net. These topics share three properties that the archive matches well. They carry continuous, fact-anchored Reddit discussion in the lead-up to events, so the agent's seed queries land on posts that contain decision-relevant signal rather than opinion. Their resolution criteria reduce to observable, widely reported outcomes (which team won, who took home an award, whether a price crossed a threshold), so an agent that has seen the supporting discussion can read off the answer directly. And the coverage probe rates many of these markets *strong*, so the path from query to evidence to answer is short. Most of the strong-coverage accuracy lift in Table 1 is contributed by the four topics on the right of Figure 6, and the agent's confidence-driven early stop fires more often on recovered markets (52

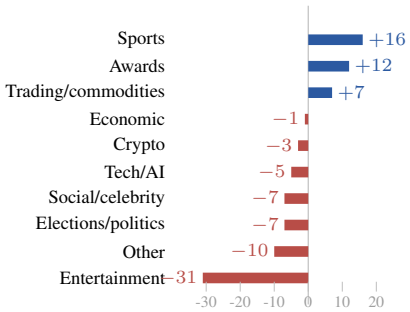

Figure 6: **Per-topic net effect of the Retrieval agent.** For each topic with at least four markets in the eval set, the bar shows the difference between the number of markets the agent *recovers* (the agent is right while the zero-shot baseline is wrong) and the number it *regresses* (the reverse), summed across the eight models with per-run logs available. Blue bars to the right of zero are topics where retrieval helps on net, and red bars to the left are topics where it hurts.

across models) than on regressed ones (34), so on these topics the agent finds enough evidence to commit before exhausting its round budget.

7.2 Where retrieval hurts

Entertainment is the standout regression with a net of -31, and Other (-10), Elections/politics (-7), Social/celebrity (-7), Tech/AI (-5), and Crypto (-3) all sit on the negative side. The Entertainment regression has a recognizable shape. The eval set's Entertainment markets are dominated by the Billboard "will X be the #1 song this week" family, whose resolution is overwhelmingly *No* because most candidate songs do not become Hot 100 chart-toppers. A model with no retrieval anchors near this base rate. The Retrieval agent, in contrast, queries for the candidate artist and pulls back speculative pre-release hype from music subreddits, which surfaces as *evidence* the agent then weighs toward Yes. The tally and reflection scaffolds were designed to favour signal asymmetry but cannot tell speculative engagement from outcome-relevant discussion, so retrieval converts a correct default into an over-confident wrong answer. The same mechanism plausibly drives the smaller regressions on Social/celebrity, Tech/AI, and Crypto, all topics where the corpus carries opinion-heavy chatter that does not reliably track outcomes. By lifecycle duration, long-lifetime markets regress more often than medium-lifetime ones (144 markets vs. 106 recovery, compared with 329 vs. 381 for medium), consistent with retrieval over-fitting on chatter as a market accumulates discussion.

7.3 Per-model effects, and accuracy versus calibration

The topic-level pattern is consistent across models, but its magnitude is not. Qwen3-8B has the cleanest accuracy split ($+43$ markets), Qwen3-4B is also clearly net positive ($+19$), and Gemma-3-4B sits at zero. Llama-3.2-3B and Ministral-8B-Instruct are slightly net negative (-3 and -6), and the Llama-3.1-8B-Instruct, Gemma-3-27B, and DeepSeek-R1-Distill-Qwen-7B backbones regress more often than they recover (-27, -10, and -19 net respectively). The Qwen3 family carries an unusual amount of headroom, since of every backbone it is the most receptive to the same evidence stream, which is consistent with its strong-bin lift in Table 1.

The per-question regression count measures decision flips, whereas Brier scores the probability mass placed on the true side. Llama-3.1-8B-Instruct regresses on 91 markets against 64 recoveries (a net of -27), yet its Brier still falls under retrieval ($.250 \rightarrow .230$). The agent flips more decisions to the wrong side on this model than the baseline does, but those wrong decisions come with less confident probabilities, and Brier rewards a 0.55-confidence wrong answer over a 0.95-confidence one. The reflection scaffold appears to pull probabilities toward the middle on uncertain markets even when it flips the answer, and calibration improves on most models even where decision-level accuracy does not. R1-Distill-Qwen-7B is the sole model on which both metrics move in the wrong direction. We attribute this to its reasoning chain consuming the model's effective context budget rather than to retrieval failing per se, since the same retrieval surface produces a clean win on Qwen3-8B at the same scale.

8 Conclusion

HINDCAST grades LLM forecasters at a past time t_0, pinning retrieval to an immutable pre-t_0 Reddit snapshot and scoring against both the resolved Polymarket outcome and the matched-time price. Retrieval lowers Brier on eight of nine models (up to 23%), but the average hides a topic split the cutoff reveals: it recovers Sports, Awards, and Trading, where Reddit carries fact-anchored pre-event discussion, and regresses on Entertainment, where speculative hype reads as evidence. A retrieval-grounded forecaster is only as good as whether the corpus carried the signal before resolution.

Limitations

The archive is built from a fixed set of 20 public subreddits, so the corpus skews English-language and community-driven and under-represents topics Reddit discusses thinly (e.g., regional politics, niche corporate events). Polymarket inherits a similar skew, and we restrict to binary Yes/No markets, leaving multi-outcome and continuous markets out of scope.

Because the archive ends on 2026-01-31, most markets in our eval set have t_0 pinned at the archive boundary rather than at a long pre-resolution lookback (median gap close–t_0 is 23 days), which limits what we can say about long-horizon forecasting. We also benchmark only 9 open-weight models behind two reference pipelines, and closed-weight models and alternative agent scaffolds are left to future work.

HINDCAST is built from public Reddit content and historical Polymarket data, and three risks deserve mention. First, although we use only already-public posts and never attempt to re-identify users, the corpus retains pseudonymous author handles, so artifacts derived from it should not be used to profile or de-anonymize individuals. Second, the predictors we study are forecasting agents; deployed without calibration audits they could lend unwarranted authority to automated bets or decisions, and our analysis shows that retrieval can make a model confidently wrong on speculative topics. We therefore frame the system as an evaluation testbed, not a deployable trading tool. Third, Reddit discussion can carry biased or offensive language that an evaluated model may echo; our filters condition only on engagement and length and do not screen for such content. All released artifacts are intended for research use only.

References

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

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

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Scaling open-ended reasoning to predict the future. *CoRR*, abs/2512.25070.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn J. Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. *CoRR*, abs/2403.12958.

Pu Cheng, Juncheng Liu, and Yunshen Long. 2026. Polybench: Benchmarking LLM forecasting and trading capabilities on live prediction market data. *CoRR*, abs/2604.14199.

Sarkar Snigdha Sarathi Das, Palash Goyal, Mihir Parmar, Nanyun Peng, Vishy Tirumalashetty, Chun-Liang Li, Rui Zhang, Jinsung Yoon, and Tomas Pfister. 2026. Nexus : An agentic framework for time series forecasting. *Preprint*, arXiv:2605.14389.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Yu Feng, Ben Zhou, Weidong Lin, and Dan Roth. 2025. BIRD: A trustworthy bayesian inference framework for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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

Shashwat Goel, Nikhil Chandak, Arvindh Arun, Ameya Prabhu, Steffen Staab, Moritz Hardt, Maksym Andriushchenko, and Jonas Geiping. 2026. Futuresim: Replaying world events to evaluate adaptive agents. *Preprint*, arXiv:2605.15188.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. 2024. Approaching human-level forecasting with language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. 2021. Forecastqa: A question answering challenge for event forecasting with temporal text data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4636–4650. Association for Computational Linguistics.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip Tetlock. 2025. Forecastbench: A dynamic benchmark of AI forecasting capabilities. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Nick Levine, David Duvenaud, and Alec Radford. 2026. Introducing Talkie: A 13B vintage language model from 1930. Blog post, https://talkie-lm.com/introducing-talkie. Accessed 2026-05-25.

Yury A. Malkov and Dmitry A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836.

Mistral AI. 2024. Un ministral, des ministraux: Introducing the world's best edge models. https://mistral.ai/news/ministraux/. Ministral-8B-Instruct-2410; released as a model with an announcement post rather than a formal paper.

Sebastian Nagel. 2016. News Dataset Available. https://commoncrawl.org/2016/10/news-dataset-available/. Common Crawl Foundation.

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

Gemma Team. 2025a. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Llama Team. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Qwen Team. 2025b. Qwen3 technical report. *CoRR*, abs/2505.09388.

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

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

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. 2024. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 13697–13720. Association for Computational Linguistics.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *CoRR*, abs/2406.04244.

Yutong Yan, Raphael Tang, Zhenyu Gao, Wenxi Jiang, and Yao Lu. 2026. Datedgpt: Preventing lookahead bias in large language models with time-aware pretraining. *CoRR*, abs/2603.11838.

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

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. 2024. MIRAI: evaluating LLM agents for event forecasting. *CoRR*, abs/2407.01231.

Haofei Yu, Fenghai Li, and Jiaxuan You. 2025. Livetradebench: Seeking real-world alpha with large language models. *CoRR*, abs/2511.03628.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, 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, and 11 others. 2025. Futurex: An advanced live benchmark for LLM agents in future prediction. *CoRR*, abs/2508.11987.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. 2022. Forecasting future world events with neural networks. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Appendix Contents

A Backend Details
A.1 Tool Surface
A.2 Retrieval Latency
B Pipeline Details

Tool	Purpose
`search_database`	Dense (vector) retrieval over submissions.
`getpostcoreinfo`	Fetch a post by id.
`getcommentcoreinfo`	Fetch a comment by id.
`getpostcommentslist`	List comments under a post, with optional ancestor / descendant thread traversal.
`getauthorhistorylist`	List recent posts and comments by an author.

Table 2: The five tools exposed to the prediction agent. All accept the same filter block and inherit the cutoff guarantee (§3.3).

B.1 Zero-shot (Vanilla) Prompt
B.2 Retrieval-Agent Prompts
B.3 Decoding and Retrieval Configuration
C Full Results
C.1 Per-Run Numbers
C.2 Coverage Bin Sizes
C.3 Lookback-Fraction Sweep

A Backend Details

A.1 Tool Surface

Table 2 lists the five tools exposed to the Retrieval agent. All five accept the same filter block (cutoff t_0, document type, optional subreddit, author, month, and start-time bounds) and inherit the row-level cutoff guarantee of §3.3.

A.2 Retrieval Latency

The partition selector of §3.3 restricts each dense query to the monthly partitions admitted by the cutoff t_0, so retrieval cost scales with the number of partitions before t_0 rather than with the full corpus. Table 3 measures this against the live index. Per-query latency rises monotonically with the admitted partition count, from a median of 0.14 s at three partitions to 0.84 s at all 25. The cutoff used in our experiments, the archive's final day (2026-01-31), sits at the top of this range, since it admits essentially every partition. Latencies are end-to-end per query, comprising query embedding, per-partition HNSW search, and merge. They are measured over the running retrieval server on a single GPU and averaged over 32 queries per cutoff, so absolute values will vary with hardware and load.

B Pipeline Details

This appendix gives the prompts and per-call configuration that instantiate the two pipelines of §5.

Partitions $\leq t_0$	Median (ms)	p95 (ms)
3	144	155
9	392	410
15	516	555
21	710	786
25	836	944

Table 3: Per-query dense-retrieval latency against the live index as a function of the monthly partitions admitted by the cutoff t_0. The end-to-end measurement covers query embedding, per-partition HNSW search, and merge on a single GPU, averaged over 32 queries per row. Cost scales with the partitions before t_0 rather than the full corpus, so earlier cutoffs are cheaper. Most markets in our eval set pin t_0 at the archive boundary (2026-01-31), so retrieval admits essentially all 25 partitions (bottom row).

The tool surface they call through is documented separately in Appendix A.1.

B.1 Zero-shot (Vanilla) Prompt

The vanilla pipeline has no retrieval. The model receives the system line below, followed by the per-market user template.

System.

```
You are a forecaster. Predict the probability
of the event.
```

User template.

```
Question: {question}

Resolution criteria:
{description}

Forecast cutoff date: {cutoff_date}

Output a single JSON object only, matching:
{
  "choice": "Yes" | "No",
  "prob_yes": 0.0-1.0,
  "prob_no": 0.0-1.0,
  "confidence": 0.0-1.0,
  "reasoning": "brief"
}
```

B.2 Retrieval-Agent Prompts

The agent pipeline runs in two stages. A research stage drives the ReAct tool loop and writes an evidence report. A forecasting stage turns that report into a calibrated forecast under the same schema as the vanilla pipeline. Both stages share a cutoff date set to the simulated query time t_0.

Research system.

```
You are preparing evidence for a forecasting
task.

Use ONLY the local Reddit tools. Do not use
web search, browser tools, or outside
```

```
knowledge as evidence. Every search must
respect the provided cutoff date. Gather
broad evidence first, then follow up on
concrete names, dates, organizations, IDs,
and claims found in the first pass.
```

Research user template (abridged).

```
Cutoff date for all tool calls: {cutoff_date}

Task:
{prompt}

Options:
{options}

Research requirements:
- Generate multiple diverse research
  questions instead of searching the
  question verbatim.
- Use many local Reddit tool calls: broad
  vector searches first, then targeted
  follow-ups.
- Retrieve full post/comment/thread/author
  context when search hits look relevant.
- Produce an evidence report with sections:
  Key facts, Evidence for each option,
  Evidence against each option, Uncertainty,
  and Suggested probability direction.
- Do not use information after the cutoff
  date.

Local Reddit tool rules:
- Available subreddits are exactly:
  {AVAILABLE_SUBREDDITS}
- search_database must use
  doc_type="submission" only.
- Valid month format is "YYYY-MM" only.
```

Forecast system.

```
You are a calibrated forecasting judge.

You receive a Polymarket-style question plus
a local Reddit evidence report. Your job is
to produce one forced-choice forecast with
calibrated probabilities.

Forecasting approach:
1. Form your own base-rate prior using world
   knowledge, the resolution criteria, and
   the timing/cutoff.
2. Compare the strength and asymmetry of
   evidence on each side. Move away from the
   prior in proportion to that asymmetry;
   confident moves require asymmetric
   evidence, not absence of evidence.
3. Treat indirect/contextual signals as
   actionable when they consistently lean
   one way.

Return ONLY a single JSON object matching
the requested schema.
```

Forecast user template (abridged). Step 1 sets a prior anchor based on a five-category classification of the question. Step 2 walks the model through tally-based updates from the evidence report. Step 3 emits the same JSON schema as the vanilla pipeline.

```
Question: {question}
Resolution criteria: {description}
Allowed options: {options}
Forecast cutoff date: {cutoff_date}

Step 1 - PRIOR (ignore Reddit evidence for
this step). Pick a specific anchor from one
of the five categories below:
  1. Narrow numeric range       -> .05-.20
  2. Underdog / against-trend   -> .10-.30
  3. Structurally uncertain     -> .40-.60
  4. Favorite / with-trend      -> .55-.85
  5. Broadly likely outcomes    -> .70-.95
State this prior P(Yes) as the first
sentence of "reasoning".

Step 2 - ADJUST using the Reddit evidence:
{report}

- Clear asymmetry: move the prior toward
  the stronger side.
- evidence_quality is "none" or "weak" with
  no asymmetry: keep the Step-1 prior,
  unless it was extreme and anchor-driven.
- Indirect/contextual signals ARE actionable
  when they consistently lean one way.

Step 3 - Output the final forecast as a
single JSON object matching the schema in
\ref{sec:appendix-prompts-vanilla}.
```

LLM call	Temp.	Max tokens
Query generation	0.2	4096
Follow-up / reflection	0.3	4096
Evidence summarisation	0.1	3500
Evidence tally	0.1	2200
Final forecast	0.1	3200
Forecast retry (repair/compact/JSON)	0.1	1400/2400/900
Zero-shot baseline	0.1	4096

Table 4: Per-call decoding settings for the two pipelines. Top-p is left at the vLLM default (1.0) throughout. The agent forecasts at temperature 0.1 and raises the temperature only for the intermediate query-generation and reflection steps that benefit from diversity.

B.3 Decoding and Retrieval Configuration

Table 4 lists the per-call decoding settings. Top-p is left at the vLLM default (1.0) throughout. Thinking is left at each model's default (on for the Qwen3 family), and any <think> span is stripped from a reply before it is parsed.

Retrieval. The agent runs a default budget of 3 rounds. Round one issues six seed queries, and each subsequent round adds four to six follow-up queries. All searches are dense (vector) over submissions (§3.2). The backend returns the top 20 hits per query, and the agent expands the 20 highest-ranked submissions with their full post text and the author's recent history (up to five posts and five comments each). The loop stops early when the running confidence reaches 0.7 or the round budget

is spent.

Forecast parsing. The final forecast is read from a JSON object through a four-step fallback, namely the main forecast, then a JSON-repair pass, a compact re-prompt, and a strict JSON-only re-prompt. Across all 9 models this leaves at most three of 216 markets without a parsed prediction in a run, so truncation of the forecast pass is not a material source of missing data.

C Full Results

C.1 Per-Run Numbers

Table 5 reports accuracy and Brier for all five consistency runs behind each cell of Table 1.

C.2 Coverage Bin Sizes

Table 6 reports the mean coverage bin sizes per model (S/W/N), averaged across the five Agent runs that produced them.

C.3 Lookback-Fraction Sweep

Table 7 reports the per-cell accuracy and Brier for the lookback-fraction sweep of §6, i.e. the data plotted in Figure 5.

C.4 Use of AI Assistants

We used a large language model solely as a writing aid, to polish wording and improve the clarity and fluency of the prose. It was not used for research ideation, experimental design, code generation, data analysis, or interpreting results, and all technical content and claims are the authors' own. The authors reviewed and verified all text and take full responsibility for the final manuscript.

C.5 Ethics

Our corpus is derived from the public Pushshift Reddit dataset and our questions from historical Polymarket data; both are public resources used strictly for non-commercial research, and the posts remain subject to Reddit's User Agreement. The benchmarked checkpoints are used within their licenses (Apache-2.0 for the Qwen3 models and the Qwen3-Embedding encoder; the Llama 3.1/3.2 Community Licenses; the Gemma Terms of Use; MIT for DeepSeek-R1-Distill-Qwen-7B; and the research-only Mistral Research License for Ministral-8B-Instruct-2410), and all use here is research-only and consistent with these terms. The corpus contains only public posts. We retain pseudonymous author handles for thread- and author-level lookups but make no attempt at re-identification, drop documents whose author or body is deleted or removed and accounts on a bot list, and release document and market identifiers with regeneration code rather than raw text. We do not screen for offensive content, since our filters condition only on engagement and length, so some posts may contain sensitive material. Finally, the predictors we study are forecasting agents: deployed without calibration audits they could lend false authority to automated decisions, and retrieval can make a model confidently wrong on speculative topics. We therefore frame HINDCAST as an evaluation testbed rather than a deployable trading tool.

Model	Pipe	Run 1		Run 2		Run 3		Run 4		Run 5	
		Acc	Brier	Acc	Brier	Acc	Brier	Acc	Brier	Acc	Brier
Qwen3-4B	Vanilla	59.8	.238	60.5	.247	59.0	.247	59.7	.247	61.6	.245
	Agent	68.8	.212	66.1	.226	66.7	.212	64.2	.220	63.7	.223
Llama-3.2-3B	Vanilla	53.0	.396	57.9	.367	54.9	.377	57.4	.362	57.3	.374
	Agent	56.5	.285	54.9	.296	55.1	.296	58.6	.296	57.4	.286
Qwen3-8B	Vanilla	56.1	.260	55.2	.263	55.6	.265	55.9	.254	56.3	.263
	Agent	66.1	.199	62.5	.215	62.0	.221	66.7	.216	66.1	.213
Ministral-8B	Vanilla	53.7	.279	54.6	.275	51.9	.281	51.9	.283	52.3	.287
	Agent	60.7	.225	56.9	.244	58.3	.241	53.7	.252	54.2	.245
Llama-3.1-8B	Vanilla	66.5	.253	66.2	.248	66.1	.245	64.4	.259	67.1	.245
	Agent	56.9	.246	59.3	.224	61.1	.233	61.6	.224	61.1	.222
R1-Distill-Qwen-7B	Vanilla	58.7	.267	57.8	.236	60.5	.258	59.2	.257	57.5	.261
	Agent	54.2	.321	58.3	.299	50.0	.309	45.4	.336	48.2	.327
Gemma-3-4B	Vanilla	54.2	.301	54.6	.296	54.6	.296	54.6	.297	54.6	.296
	Agent	60.2	.248	54.6	.269	53.7	.263	56.0	.264	56.5	.263
Gemma-3-27B	Vanilla	70.4	.215	70.4	.215	70.8	.213	70.4	.214	70.4	.213
	Agent	69.9	.180	75.5	.176	66.7	.196	72.2	.180	71.3	.186
Qwen3-32B	Vanilla	63.0	.239	64.8	.222	64.8	.236	65.3	.236	65.3	.236
	Agent	72.7	.176	75.5	.170	69.0	.190	70.8	.183	75.9	.174

Table 5: **Per-run accuracy and Brier for all five consistency runs.** Same models and pipelines as Table 1. This is the data the run-to-run subscripts there are computed over.

Model	S	W	N	Total
Qwen3-4B	55	63	98	216
Llama-3.2-3B	25	6	185	216
Qwen3-8B	65	86	65	216
Ministral-8B	32	116	68	216
Llama-3.1-8B	20	170	26	216
R1-Distill-Qwen-7B	60	113	40	213
Gemma-3-4B	75	137	4	216
Gemma-3-27B	62	121	33	216
Qwen3-32B	100	87	29	216

Table 6: **Coverage bin sizes (mean over 5 Agent runs).** The S/W/N label is assigned per run from that run's own retrieval tally, so a single market can occupy different bins across runs. Bin sizes also vary widely across models. Llama-3.2-3B reaches *strong* on only 25 markets, while Gemma-3-4B reaches *none* on only 4, which is why Gemma-3-4B's *N* cell in Table 1 has very high std. R1-Distill-Qwen-7B totals 213 because three markets had no parseable retrieval tally in any of the five runs.

Model	Frac.	Vanilla		Agent	
		Acc	Brier	Acc	Brier
Ministral-8B	25%	53.6	.278	56.9	.244
	50%	53.3	.280	57.0	.242
	75%	53.2	.281	55.3	.249
	100%	53.4	.281	56.5	.244
Qwen3-8B	25%	56.4	.256	61.0	.216
	50%	56.3	.258	63.1	.217
	75%	54.8	.264	62.6	.224
	100%	55.8	.259	63.4	.215

Table 7: **Lookback-fraction sweep on the two-model subset.** Mean accuracy (↑, in %) and Brier (↓, in points) over five consistency runs per cell, for the per-market cutoff fractions of §6 (placing t_0 at 25%, 50%, 75%, or 100% of each market's open-to-resolution span). These are the numbers underlying Figure 5.