# Approximate Recall, Approximate Forecasts: Recall as a Diagnostic for LLM Forecasting Errors

**Prashanth Bhaskara** [*1] **Shubhaankar Gupta** [*1] **Seojoon Yeon** [*1]

## Abstract

Large Language Models (LLMs) exhibit approximate recall: they retain coarse associations about past events while losing fidelity on specific details. Past research on LLMs has quantified different models' forecasting ability in real-world events using prediction markets, but has not studied how losing said fidelity impacts their forecasting abilities. For example: while an LLM correctly recalls that the answer to: *"Did Anna Sawai win the Best Drama Actress in the 76th Emmy Awards?"* is *"Yes,"* it incorrectly states the event date as September 22 rather than September 15. **In this paper, we introduce a novel metric called the Recall Fidelity Score to study how imprecision in internalized knowledge propagates to forecasting capability on future events. On resolved historical markets, we find that higher local recall fidelity around semantically similar past events predicts lower forecasting error on resolved markets, and the same signal predicts smaller distance from open markets prices**.

## 1. Introduction & Related Work

The emergence of reasoning capabilities in Large Language Models (LLMs) has made predictive reasoning a promising application domain for LLMs. Prophet Arena (Yang et al., 2025) introduced a Brier score-based benchmark for evaluating LLM forecasting accuracy on unresolved prediction market events. They discovered that frontier LLMs exhibit meaningful forecasting capability, but remain less calibrated than market-implied probabilities on Polymarket and Kalshi (Yang et al., 2025).

Meanwhile, prior research such as (Machlab & Battle, 2024) and (Gemini Team, 2024) illustrated that LLMs often exhibit "approximate rather than precise recall" of past events, particularly regarding exact dates and fine-grained details. The research indicated that relevant information is buried deep within context windows, leading to a "needle in a haystack" retrieval problem that limits reliable recall (Liu et al., 2024). These findings raise the question of whether failures in recalling past events systematically degrade forecasting quality, and whether inaccurate recall leads models to produce overconfident but poorly calibrated predictions.

A key challenge is that forecasting metrics such as the Brier score conflate calibration errors with failures of underlying knowledge recall (Brier, 1950). Two models may obtain similar Brier scores despite behaving very differently: one may remain cautious and weakly informative, while another may produce highly confident predictions grounded in inaccurate recollections of past events, analogous to the "imitative falsehoods" identified in prior work on LLMs (Lin et al., 2022). In particular, confidently incorrect forecasts may suggest that faulty internalized knowledge distorts predictive reasoning. Motivated by these observations, we investigate how inaccuracies in LLM internalization and recall of past events affect forecasting performance on future events.

## 2. Data and Methodology

We study how a model's ability to recall past events within a domain $D \in \{$*Climate, Commodities, Companies, Economics, Elections, Entertainment, Financials, Politics, Science & Technology*$\}$ predicts its ability to forecast future events within and/or across domains. To construct the dataset, we collected binary prediction markets from Polymarket. Markets were passed through a multi-stage filtering pipeline described in A.4, then split along a knowledge-cutoff axis into two datasets: a recall set of resolved markets from *2023-01-01* to *2025-01-01*, and a forecast set of unresolved markets from *2026-09-01* to *2030-01-01*. The final corpus contains 1,300 recall markets and 1,350 forecast markets, for 2,650 unique markets.

We evaluate Claude Opus 4.7 (Anthropic, 2026a), Claude Sonnet 4.6 (Anthropic, 2026b), GPT-5.5 (OpenAI, 2026), and Gemini 3.1 Pro (Google, 2026), representing frontier forecasting systems across distinct model families. This

[1]University of Chicago. Correspondence to: Prashanth Bhaskara <pbhaskara@uchicago.edu>, Shubhaankar Gupta <shubhaankar@uchicago.edu>, Seojoon Yoon <seojoony@uchicago.edu>.

*Preprint. June 11, 2026.*

gives us a small but diverse set of high-performing models, allowing us to test whether recall–forecast relationships are consistent across architectures. We note that our design is observational – recall and forecast quality may share common causes (e.g., training data coverage), and we cannot rule out this confound. We treat recall as a diagnostic signal: a model's pattern of recall errors on historically analogous events is a useful predictor of where its forecast errors will concentrate regardless of the underlying mechanism. Per-model coverage statistics and knowledge cutoff dates are reported in A.6. Each (model, market) pair was queried exactly once with a shared system prompt and hyperparameters across all providers given in A.1 and A.2.

### 2.1. Metrics

We define metrics to assess model performance across different temporal contexts: recall before market resolution (pre-cutoff) and forecast accuracy after market resolution (post-cutoff).

#### 2.1.1. Recall Brier Score

To evaluate accuracy in the pre-cutoff period, we compute the recall Brier Score

$$\text{Brier}_{\text{recall}} = \overline{(p_t - y)^2}, \qquad y \in \{0, 1\}. \tag{1}$$

where $p_t$ is the model's probability estimate and $y$ is the binary outcome of market $m$ on domain $D$. Lower values indicate better pre-cutoff forecasting accuracy.

#### 2.1.2. Recall Fidelity Score (RFS)

We introduce the Recall Fidelity score, a new metric to study consistency in LLM recall accuracy by examining prediction fidelity during the pre-cutoff period:

$$\begin{aligned} \text{RecallFidelity}_{m,d}(R_{m,d}) = \overline{s}_{m,d}, \\ s \in \{0, 0.5, 1\} \end{aligned} \tag{2}$$

We used a two-stage grading pipeline. First, each model's recalled outcome was compared against the true resolved market outcome using deterministic string matching. If the model could not recall, a value of `NULL` was assigned while incorrect recalls received $s = 0.0$. If the recalled outcome was correct, the accompanying reasoning and evidence were fact-checked by with agentic web-search. The grader used a system prompt and parameters included in A.3.

$s = 0.0$: The recalled outcome does not match the true resolved outcome.

$s = 0.5$: The recalled outcome matches, but the reasoning or evidence contains clear factual errors.

$s = 1.0$: The recalled outcome matches, and the reasoning and evidence are factually consistent.

#### 2.1.3. Forecast Squared Loss & Log Loss (Brier Proxies)

For unresolved forecast markets, true outcomes are unavailable, so we cannot compute post-resolution Brier score. As a proxy, we measure disagreement between the model forecast and the contemporaneous market-implied probability:

$$Loss_{fc} = \overline{(p_t - p_m)^2} \tag{3}$$

$$LogLoss_{fc} = -\overline{p_m \log(p_t) + (1 - p_m) \log(1 - p_t)} \tag{4}$$

where $p_t$ is the model's final probability estimate and $p_m$ is the consensus market probability.

#### 2.1.4. Local Recall Metric

In order to measure recall more closely within specific domains, we construct a market-specific local recall score using Nearest Neighbors. Each market is represented as a TF–IDF (Salton & Buckley, 1988), (Spärck Jones, 1972) vector over its question text and metadata (i.e category, domain, market slug, etc), and market similarity is computed through cosine similarity (Lloyd, 1982). For each forecast market $f$, we identify the $k$ nearest resolved recall set markets $\mathcal{N}_k(f)$, excluding any sharing $f$'s `event_slug` to prevent leakage from sibling markets in distribution and perpetual market families, and compute the model's average recall score on those neighbors:

$$R^{\text{local}}_{m,f,k} = \frac{1}{k} \sum_{r \in \mathcal{N}_k(f)} R_{m,r}, \tag{5}$$

where $R_{m,r} \in \{0, 0.5, 1\}$ is the graded recall score for model $m$ on resolved market $r$. The recall score measures how well a model's ability to recall nearby historical facts predicts its forecast quality on a new market, rather than relying only on coarse domain averages.

## 3. Results

### 3.1. Recall Ability

We compute domain-wise Recall Fidelity Score (Eq.2), $R_{m,d}$, and Brier score (Eq.1) for each model (A.9). The RFS rewards clean reasoning regardless of how the model expressed its uncertainty in probability terms while the recall-task Brier rewards calibrated probabilistic judgement. Domain-wise RFS and recall Brier score for each model are shown in Fig. 1 and A.9.

Table 1 validates RFS as a proxy for recall Brier: $\beta_{R_{m,d}} = -0.22^{***}$ is stable across bivariate (M1),

*Table 1.* Domain-level regression of recall Brier scores on Recall Fidelity Score $R_{m,d}$ across $N = 36$ (model, domain) cells. Entropy defined in A.8

| Specification | Intercept | $\beta_{R_{m,d}}$ | $R^2$ |
|---|---|---|---|
| M1: $R_{m,d}$ | $0.226^{***}$ | $-0.204^{***}$ | 0.398 |
| M2: $+D_{\text{entropy}}$ | $0.286^{***}$ | $-0.222^{***}$ | 0.429 |
| M3: +Model FE | $0.285^{***}$ | $-0.223^{***}$ | 0.691 |

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$; $\beta \approx -0.22^{***}$

difficulty-controlled (M2), and fixed-effect (M3) specifications. A shift from the lowest- to highest-recall domain-model cell (0.40 to 0.87 RFS) predicts $\approx 0.10$ Brier reduction, the spread between best and worst model in our panel.


*Figure 1.* Recall Fidelity Scores (null = 0) and Log Loss for all models grouped by domain. Elections and Economics domains seem to produce strongest recall across models. Exact figures reported in A.9

## 3.2. Recall Decomposition & Forecast Across Domains

### 3.2.1. Confidence Score

Since the Recall Fidelity Score $R_{m,d} = \overline{s}, s \in \{0, 0.5, 1\}$ where $s$ is the grade assigned to an individual recall from the LLM, we can study the impact of a confidently wrong recall, where the model reports a remembered outcome that does not match the true resolution. Using the LLM generated probability $p$ for recall event, we define $confidence = |p - 0.5|$ and plot it against the confidence levels for all models in Fig. 2. We can see the visual asymmetry directly: confident-wrong cells (red) cluster at high Brier with intermediate confidence, while correct-clean and partial cells (green, yellow) cluster at near-zero Brier with high confidence.

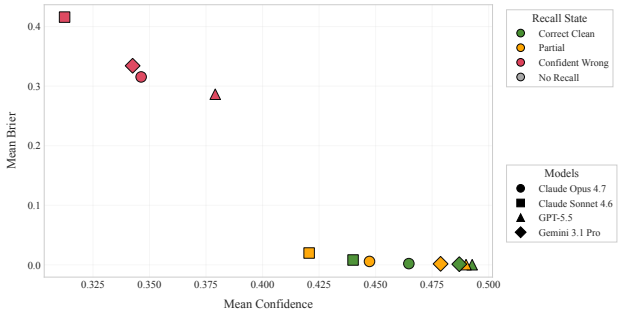


*Figure 2.* Confident-wrong recall produces high Brier scores despite reduced confidence, while correct and partial recall yield near-zero Brier at high confidence. Confident-wrong cells are separated from all other states by an order of magnitude in Brier.

### 3.2.2. Local Recall

We estimate the relationship between local recall and forecast performance using market level OLS regressions with model fixed effects and forecast market fixed effects, clustering standard errors by condition ID:

$$L_{m,f} = \beta R^{\text{local}}_{m,f,k} + \alpha_m + \gamma_f + \varepsilon_{m,f}, \tag{6}$$

where $L_{m,f}$ is the model's forecast loss on the post-cutoff market. The main specification ($k = 10$) yields $\beta = -0.0652$ ($SE = 0.0231$, $p = 0.0048$, $N = 4{,}897$, 1,264 clusters), indicating that models with stronger recall on nearby historical markets have lower forecast loss on similar forecast markets. These results remain robust across neighborhood sizes: the coefficient remains negative for $k \in \{3, 5, 10, 20, 50\}$. Negative control variants using shuffled, farthest neighbor, and cross-domain neighbors do not reproduce the same relationship, suggesting that the signal comes from local historical similarity.

We also report a domain-level robustness specification with model fixed effects and a domain-difficulty control, where difficulty is defined as the mean binary entropy of Polymarket prices within each domain; full specifications and estimates are provided in A.7.

### 3.2.3. Recall Fidelity to Forecast

We provide model-wise, domain-wise Recall Fidelity Score, recall Brier, and Forecast losses in A.9. Across the four models, we observe positive correlations between recall and forecast abilities, the magnitude of which varies by model and domain, as can be ascertained by Fig.1 and Fig.3. Upon directly regressing forecast log-loss on RFS, we find $\beta_{RFS} = -0.78$ with $p = 0.012^{*}$, pointing to an inverse relationship between the two.

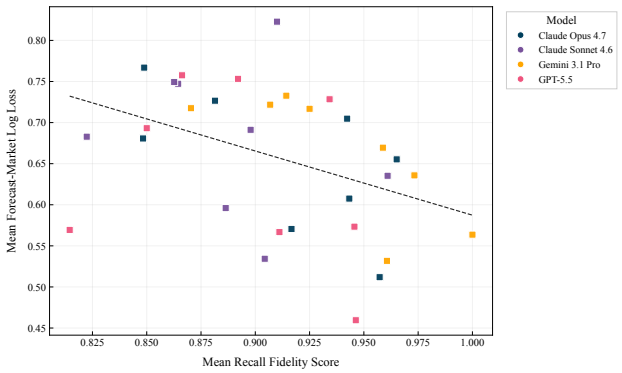


*Figure 3.* Mean forecast log-loss (from unresolved markets) versus mean Recall Fidelity color coded by model across nine domains.

### 3.3. Local Recall Error-Type Decomposition.

We further decompose local recall to distinguish whether forecast degradation is driven by missing knowledge, partially correct recall, or confidently incorrect recall. For each forecast market $f$, we compute the model's ability to recall historical facts of $k = 10$ nearest resolved recall markets (Eq.6). We regress forecast loss on these local error-type shares with model and forecast-market fixed effects, using the partial-recall share as the omitted reference category.

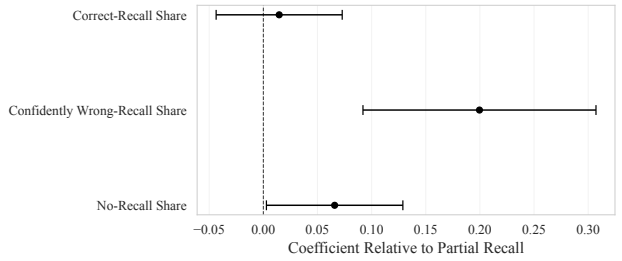


*Figure 4.* Local recall error-type decomposition. Coefficients and 95% confidence intervals come from a market-level OLS regression of forecast Brier loss on the shares of the ten nearest historical recall markets falling into each recall category, with model and forecast-market fixed effects and standard errors clustered by condition ID.

Fig.4 shows that the strongest predictor of worse forecasting is local confident-wrong recall. The coefficient on the local wrong-recall share $\beta = 0.1996$ ($SE = 0.0549$, $p = 0.0003$), meaning that forecasts are substantially worse when a model's nearest historical analogues are cases where it recalled the wrong outcome. The local no-recall share is also associated with higher forecast loss, though more weakly ($\beta = 0.0659$, $SE = 0.0321$, $p = 0.0405$). By contrast, the local correct-recall share is not statistically distinguishable from the partial-recall baseline ($\beta = 0.0146$, $SE = 0.0297$, $p = 0.6223$). These results suggest an association between forecast degradation and locally relevant yet incorrectly remembered outcomes, rather than gaps in memory.

### 3.4. Open-Market Forecast Proxy: Squared Distance and Soft-Target Log Loss

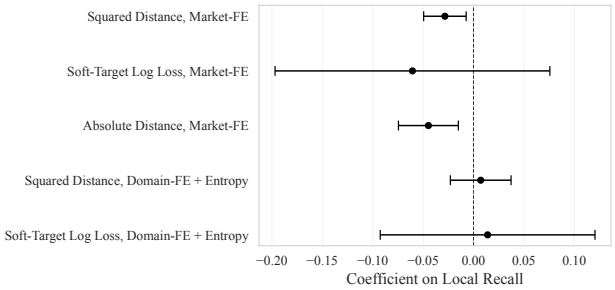


*Figure 5.* Regression coefficients for local historical recall across open-market Brier-proxy specifications. Coefficient on $R^{\text{local}}_{m,i,10}$ with 95% confidence intervals using standard errors clustered by condition ID shown. Higher local recall significantly predicts lower squared forecast-market disagreement and lower absolute forecast-market distance.

We estimate whether local historical recall predicts open-market forecast disagreement by regressing our aforementioned Brier proxies (Eq. 3, 4) on $R^{\text{local}}_{m,i,10}$, the model's average recall score on the ten nearest historical resolved markets. For squared forecast-market disagreement, higher local recall is associated with lower disagreement: $\beta = -0.0284$ ($SE = 0.0108$, $p = 0.0084$, $N = 5{,}724$, 1,897 clusters). The result is similar using absolute forecast-market distance ($\beta = -0.0448$, $SE = 0.0152$, $p = 0.0032$). Full historical and open-market regression specifications are provided in A.7. We also estimate domain-level specifications with model fixed effects, domain fixed effects, and a market-entropy control (A.8). Overall, the open-market proxy results suggest that local recall is most informative at the fine-grained market level, where historically analogous recall failures map onto specific forecast tasks.

## 4. Conclusion

We find that LLM forecast quality is degraded by failures in knowledge internalization, with confidently wrong recall, not absent recall, emerging as the dominant driver of forecast-market disagreement across domains. Forecasting errors are systematically associated to the model's ability to faithfully recall nearby historical facts, and confident false recall remains the most damaging form of error. These results suggest that efforts to improve LLM forecasting should prioritize reducing false memory over expanding knowledge coverage, as a model that knows it doesn't know outperforms one that confidently misremembers.

## References


Anthropic. What's new in claude opus 4.7. https://platform.claude.com/docs/en/about-claude/models/whats-new-claude-4-7, 2026a. Accessed 2026-05-10.

Anthropic. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, 2026b. Accessed 2026-05-10.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2.

Gemini Team, Petko Georgiev, V. I. L. e. a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Google. Gemini 3.1 pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026. Accessed 2026-05-10.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2022.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Lloyd, S. Least squares quantization in pcm. volume 28, pp. 129–137. IEEE, 1982.

Machlab, D. and Battle, R. Llm in-context recall is prompt dependent. *arXiv preprint arXiv:2404.08865*, 2024.

OpenAI. Introducing gpt-5.5. https://openai.com/index/introducing-gpt-5-5/, 2026. Accessed 2026-05-10.

Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

Spärck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

Yang, Q., Mahns, S., Li, S., Gu, A., Wu, J., and Xu, H. Llm-as-a-prophet: Understanding predictive intelligence with prophet arena, 2025. URL https://arxiv.org/abs/2510.17638.

# A. Appendix

## A.1. Recall Prompt

We describe the exact prompts used to elicit recall judgments and probabilistic forecasts from each model.

The recall prompt is presented to each model for every pre-cutoff resolved market.

```
SYSTEM PROMPT:
You are an AI assistant specialized in forecasting AND in recalling previously seen
real-world events.  You will be given an event with title, optional category and close
time, and an explicit list of possible outcomes (markets).

Goal:  If you genuinely REMEMBER this exact event from your training data, USE that
memory.

Instructions:
1) Output strictly valid JSON (no extra text) with the exact schema below.
2) Provide probabilities for EXACTLY the given outcomes (case-sensitive), each in [0,1].
3) If you remember the event, include concrete verifying details (e.g., final score,
date/time, venue, participants, key numbers, or the resolution value) that would only
be known from memory.
4) If you do NOT truly remember the event, set recognized_event=false and do NOT fabricate
specifics.  Give best-effort probabilities from reasoning only.

Output JSON shape (exact keys):

{
  ''rationale'':  ''One or two short sentences (max 50 words).'',
  ''probabilities'':  { ''<outcome1>'':  <float>, ''<outcome2>'':  <float>, ...},
  ''recall_assessment'':  {
    ''recognized_event'':  <true|false>,
    ''evidence_facts'':  [
      ''Concrete verifying details you recall (dates/scores/metrics/participants/
       venue/etc.)'',
      ''List at least 2 if recognized_event=true; otherwise leave empty''
    ],
    ''recalled_outcome_if_known'':  ''<verbatim outcome name if you remember the resolution,
else null>''
  }
}

Hard constraints:
-- JSON only.  No text before/after.
-- Use only the provided outcome names.
-- Do not invent specifics unless you genuinely remember them.

USER PROMPT:
This is the event:  {title}
Category:  {category}
Close Time (UTC): {close_time}

Example market meaning (rules):
-- {title}:  {rule_text}

Possible outcomes (provide probabilities for exactly these):
{outcomes_block}

Your JSON must look like:
{
  ''rationale'':  ''<short 2--3 sentence rationale>'',
  ''probabilities'':  {
```

```
    {probs_template}
  },
  ''recall_assessment'': {
    ''recognized_event'': <true|false>,
    ''evidence_facts'': [
      ''<verifying detail 1>'',
      ''<verifying detail 2>''
    ],
    ''recalled_outcome_if_known'': ''<outcome name if you remember the resolution, else
null>''
  }
}
```

### A.2. Forecast Prompt

The forecast prompt is presented to each model for every post-cutoff market. The model has no access to the true outcome and must reason from its internalized prior knowledge alone. We elicit a calibrated probability rather than a binary prediction.

```
SYSTEM PROMPT:
You are an AI assistant specialized in analyzing and predicting real-world events.

Event: <event title>
Close Time (UTC): <close_time>

Example market rule:
-- <market_name>: <rule text>

Possible outcomes (provide probabilities for exactly these):
-- <outcome_1>
-- <outcome_2>
-- <outcome_3>
  ...

Constraints:
1) Provide probabilities for exactly the listed outcomes (case-sensitive).
2) Do not invent additional outcomes.
3) Each probability must be a float in [0, 1].
4) Return JSON only; no extra text.

Output JSON:

{
  ''rationale'': ''<concise 2--3 sentence rationale>'',
  ''probabilities'': {
    ''<outcome_1>'': <float>,
    ''<outcome_2>'': <float>,
    ...
  }
}

USER PROMPT:
Here is the given event:
Event title: {title}
Category: {category}
Close time (UTC): {close_time}

Possible outcomes:
{outcomes_block}

Example rule excerpt: {rule_text}
```

### A.3. Grader Prompt

The grader prompt is presented to Claude Haiku 4.5 (with web search, max tokens 2000) for every recall item whose recalled outcome matches the actual resolution under deterministic string matching. Items with mismatched recalled outcomes are scored 0.0 without invoking the grader.

```
SYSTEM PROMPT:
You are a meticulous fact-checker.  You will be given:
1.  A prediction market question and its actual resolution
2.  An LLM's reasoning explaining why it predicted the (correct) outcome

Your job:  identify factual errors in the reasoning.  Check concrete claims --- dates,
names, years, ceremony numbers, chart positions, box-office figures, who beat whom,
song/album/film titles.

CRITICAL RULE ON FLAGGING: Only flag a claim as wrong if you are highly confident it is
wrong.  A confidently asserted ``correction'' that turns out to be wrong is worse than
missing an error.  If you are pattern-matching from memory and not certain, do NOT flag
it.  When in doubt, default to CORRECT. The actual_resolution is your strongest anchor
--- claims consistent with it should not be flagged unless clearly contradicted by other
widely known facts.

Do NOT penalize:
-- Hedging language (``I recall'', ``I think'', ``uncertain'')
-- Probabilistic estimates
-- Missing information (only flag stated facts that are wrong)
-- Minor rounding or approximations within ~10% (e.g.  ``$80M'' vs ``$80.5M'', ``$35M'' vs
``$35.4M'')
-- Slight phrasing differences in titles, names, or roles

Flag as errors only when clearly wrong:
-- Wrong dates or years (off by months/years, not days)
-- Wrong winners or who-beat-whom claims
-- Box-office or chart figures off by more than ~10%
-- Misattributed songs, albums, or films to the wrong artist
-- Internally inconsistent claims within the reasoning itself

Respond in this exact format:

VERDICT: CORRECT
or
VERDICT: ERRORS_FOUND

ERRORS:
-- <error 1, one line, with the correct fact>
-- <error 2, one line, with the correct fact>
(omit ERRORS section if VERDICT is CORRECT)

USER PROMPT:
Question:  {title}
Actual resolution:  {actual_resolution}
Close time:  {close_time}

LLM's reasoning:
"""
{rationale}
"""

Evidence facts the LLM cited:
{evidence_facts}

Fact-check the reasoning and evidence.  Are there any factual errors?
```

### A.4. Dataset Construction and Filtering Pipeline

All markets were collected from Polymarket's public Gamma API (https://gamma-api.polymarket.com) using per-domain wrapper scripts that delegate to a shared downloader. Two collection passes produce two disjoint datasets along a knowledge-cutoff axis: the recall set (markets.csv, status=closed, close window 2023-01-01–2025-01-01) and the forecast set (newmarkets.csv, active markets, close window 2026-09-01–2030-01-01).

**Eight-step filter chain.** Each candidate market returned by the API must pass, in order, every check below; the first failure discards the market. Counts in this section refer to the recall pull unless noted.

**Filter 1 --- Status.**
Recall pass: status = "closed". Forecast pass: active markets (closed = false, accepting outcome-pending markets).

**Filter 2 --- Resolved (recall only).**
actual_resolution $\in$ {Yes, No}. Markets with null, ambiguous, or non-binary resolutions are dropped.

**Filter 3 --- Close-time window.**
close_time must lie strictly inside the requested [start_date, end_date] interval (Unix-second comparison after parsing the ISO-8601 timestamp).

**Filter 4 --- Volume.**
If --min-volume is set, retain markets whose USD trade volume is $\geq$ the threshold. The headline pull used the API default (min_volume = None); a published threshold will be added in v2.

**Filter 5 --- Title-keyword exclusion.**
Drop any market whose title or slug matches any term in exclude_keywords. The forecast pass uses a fixed global list:
-- "luigi mangione"
-- "manifesto"
-- "brian thompson"
-- "brian-thompson"

**Filter 6 --- Within-domain dedup.**
condition_id (or market_slug fallback) must not already be in seen_ids for this domain.

**Filter 7 --- Cross-domain dedup.**
condition_id must not appear in any previously written markets.csv of another domain. This guarantees that no market is counted in more than one domain.

**Filter 8 --- First-N cap.**
Iterate the filtered stream until len(markets) $\geq$ N, where $N = 150$ for both the recall pass and the forecast pass (the forecast script uses an upstream cap of 300 and is then truncated to 150). Markets are accepted in API-default pagination order, not randomly sampled.

**Sampling procedure.** Sampling is non-random: the API returns paginated results (page_limit = 100 per request, max_pages = 200) in its own internal ordering, and the downloader accepts the first $N$ markets that pass Filters 1–7. There is no stratification within a domain, no balancing across resolution outcome, and no weighting by volume. The Climate domain produces only 100 resolved markets inside the 2023–2025 window that survive the filter chain, so Climate's recall slice undershoots the $N = 150$ target.

**Deduplication rules.**

- *Within-domain:* a market is identified by its condition_id (a 32-byte hex string assigned by Polymarket); the

market_slug is used as a fallback only when condition_id is missing. Identical IDs from a later page are ignored.

- *Cross-domain:* before each new domain pull, the downloader loads every markets.csv already on disk and unions their condition_id columns into an exclusion set. Any candidate matching this set is skipped, regardless of which tag returned it.
- *Same-event awareness for the local-recall analysis (downstream):* markets sharing an event_slug are not deduplicated at the collection step but are excluded from each other's nearest-neighbor sets in the market-level regression to prevent leakage.

**API collection details.**

- Endpoint: GET /markets on https://gamma-api.polymarket.com.
- Query parameters: limit=100, status, start_time / end_time (Unix seconds), one tags=<slug> parameter per domain tag, optional min_volume, optional free-text search.
- Pagination: cursor-based, capped at 200 pages per (domain, tag) pair.
- Rate-limit: client-side throttle of 0.1 s minimum between requests; HTTP retries on 429/5xx with exponential backoff.
- Recorded fields per market: condition_id, market_slug, event_slug, title, category, open_time, close_time, settlement_ts, volume, status, actual_resolution (recall) or market_price (forecast).
- Code: scripts/polymarket_market_downloader.py (recall pass) and scripts/download_polymarket_future_markets.py (forecast pass).

### A.5. Domain Definitions and Market Taxonomy

The corpus is partitioned into nine disjoint domains. Domain membership is determined entirely by Polymarket's own tag system; we do not re-label markets manually. Each domain script supplies a fixed category display name and a tuple of one or more Polymarket tag slugs. A market is admitted to a domain if and only if the API returns it for at least one of that domain's tag-keyed queries *and* it is not already in another domain's set (cross-domain dedup, Filter 7 above).

**Tag slugs per domain.** The tag tuple sent verbatim to the Gamma API for each domain is:

```
Climate.
Weather, Climate, Global Temp, global warming, Hurricane, Hurricanes
(supplemental tag for under-fill:  science, with keyword filter)

Commodities.
crypto, commodities

Companies.
business, companies

Economics.
economics
(supplemental tags for under-fill:  business, finance, news, current-events, with keyword
filter)

Elections.
elections

Entertainment.
pop-culture, entertainment
```

**Financials.**
finance, financials

**Politics.**
politics, us-politics

**Science/Technology.**
science, technology

**Domain mapping methodology.**

1. For each domain $d$, the API is queried with one `tags=<slug>` parameter per primary tag in the tuple, paginated as described in A.4.
2. Returned markets are passed through Filters 1–7. Cross-domain dedup (Filter 7) operates in the order in which the per-domain scripts are run; in practice this is alphabetical by output directory, so a market that satisfies multiple tags is assigned to the first domain that claims it.
3. If a domain's resolved-market pool falls short of $N = 150$ after the primary tags, a small *supplemental* pull broadens the tag set and applies a domain-specific keyword filter (`supplement_kws` in `fetch_markets.py`) to keep the supplement topically aligned. This affected only Climate (kept at 100, no supplement applied) and Economics in the recall pass.
4. No manual relabeling is performed. The `category` field stored on each market is purely a display label; the operative grouping is the directory the market is written to.

**Example markets.** Two illustrative recall-pass titles per domain (drawn from `Research/<Domain>Markets/recall/ claude-opus-4-7_1000_recall.json`):

**Climate.**
-- Iceland volcanic eruption by Nov 15?
-- Will the chopsticks catch SpaceX's Super Heavy?

**Commodities.**
-- Will BTC or ETH reach all-time high first?
-- Will crude oil hit $100 in 2023?

**Companies.**
-- OpenSea acquired before March?
-- Will Brazil unban X before October?

**Economics.**
-- Russian Ruble ¥110 to $1 USD by December 20?
-- U.S. Recession in 2024?

**Elections.**
-- Will Stancil win Minnesota House Democratic Primary?
-- House control after 2024 election?

**Entertainment.**
-- Will ``Oppenheimer'' win the Oscar for Best Picture?
-- Taylor Swift cancels more tour dates in August?

**Financials.**
-- European Central Bank cuts rates in Oct meeting?
-- Trump vs. Biden:  First to 1B?

**Politics.**
-- Another Iran strike on Israel in 2024?
-- Saudi and Israel peace deal by March 2024?

**Science/Technology.**
-- GPT-5 released in 2024?
-- Will Gabriel Haines win Crypto: The Game?

**Realized counts per domain.** Recall set: Climate 100; Commodities, Companies, Economics, Elections, Entertainment, Financials, Politics, Science/Technology each 150. Total recall: **1,300**. Forecast set: 150 per domain; total forecast: **1,350**.

### A.6. Model Coverage and Knowledge Cutoffs

We evaluate five frontier LLMs across two independent passes (recall and forecast). Four of the five have graded recall scores and enter the recall–forecast regression; the fifth (Gemini 3 Flash) appears only in the forecast set and serves as a coverage cross-check. The forecast window's earliest `close_time` is 2026-09-01, which sits 19–35 months past every model's stated knowledge cutoff, substantially reducing direct contamination of the forecast pass.

**Model identities and stated cutoffs.**

**Claude Opus 4.7**
Provider: Anthropic. API ID: claude-opus-4-7.
Stated knowledge cutoff: early 2025.
Months from cutoff to earliest forecast close: ≈20.

**Claude Sonnet 4.6**
Provider: Anthropic. API ID: claude-sonnet-4-6.
Stated knowledge cutoff: early 2025.
Months from cutoff to earliest forecast close: ≈20.

**GPT-5.5**
Provider: OpenAI. API ID: gpt-5.5.
Stated knowledge cutoff: late 2024.
Months from cutoff to earliest forecast close: ≈22--24.

**Gemini 3.1 Pro (preview)**
Provider: Google DeepMind. API ID: gemini-3.1-pro-preview.
Stated knowledge cutoff: early 2025.
Months from cutoff to earliest forecast close: ≈20.

**Gemini 3 Flash (preview)**
Provider: Google DeepMind. API ID: gemini-3-flash-preview.
Stated knowledge cutoff: mid 2024.
Months from cutoff to earliest forecast close: ≈26.

**Per-model graded-item counts.** Each model is queried on every market in its assigned dataset; counts below are post-hoc diagnostics rather than design choices. *Recall items* = number of recall-pass JSON entries the model produced. *Recall graded* = number with a non-null `recall_score` after the two-stage grading pipeline ( A.3). *Brier-able* = number of recall items with both a numeric `p_yes` and a binary `actual_resolution` (used as the dependent variable in the headline regression). *Forecast items* = number of forecast-pass JSON entries.

**Claude Opus 4.7.**
recall items 1300 recall graded 891 Brier-able 1264 forecast items 1313

```
Claude Sonnet 4.6.
recall items 1300  recall graded 888  Brier-able 1264  forecast items 1313

GPT-5.5.
recall items 1300  recall graded 889  Brier-able 1105  forecast items 1104

Gemini 3.1 Pro.
recall items 1300  recall graded 957  Brier-able 1264  forecast items 1308

Gemini 3 Flash.
recall items 0  recall graded 0  Brier-able 0  forecast items 1229
```

**Inclusion in the headline recall regression.** The recall–forecast regression is fit over the four models with non-empty graded-recall coverage: Claude Opus 4.7, Claude Sonnet 4.6, GPT-5.5, and Gemini 3.1 Pro. This yields a panel of $4 \times 9 = 36$ (model, domain) cells at the aggregate level and $3 \times 1264 + 1 \times 1105 = 4{,}897$ (model, market) cells at the market level (reduced to $4{,}897$ in the headline market-level spec after restricting to rows with both a graded recall score and a Brier-able target).

### A.7. Local Regression Specifications

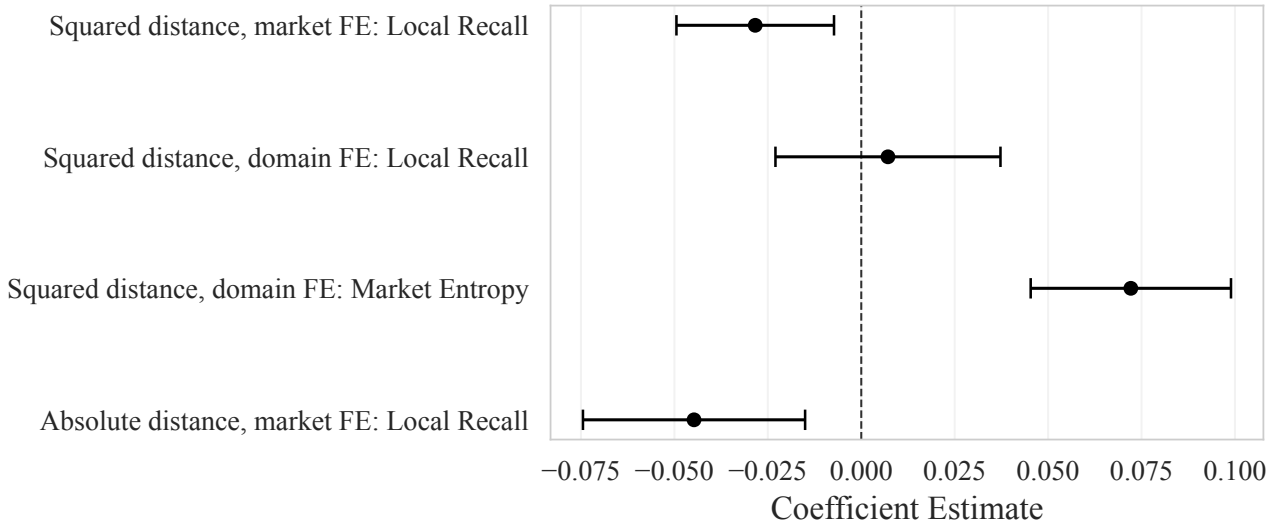


*Figure 6.* Open-market proxy regression coefficients with 95% confidence intervals by condition ID. Local historical recall is negatively associated with squared and absolute forecast-market disagreement in market fixed-effect specifications, while the domain-level specification shows that market entropy is strongly positively associated with forecast-market disagreement.

Let $m$ index models, $i$ index resolved target markets, $j$ index unresolved open markets, and $d(\cdot)$ denote domain membership. For resolved markets, the dependent variable is the Brier loss

$$B_{m,i} = (p_{m,i} - y_i)^2, \tag{7}$$

where $p_{m,i}$ is model $m$'s probability assigned to the "Yes" outcome and $y_i \in \{0, 1\}$ is the realized resolution. The primary market-level specification is

$$B_{m,i} = \alpha_m + \lambda_i + \beta R^{\text{local}}_{m,i,10} + \varepsilon_{m,i}, \tag{8}$$

where $\alpha_m$ are model fixed effects, $\lambda_i$ are market fixed effects, and $R^{\text{local}}_{m,i,10}$ is the mean recall-fidelity score over the ten nearest resolved historical neighbors of market $i$. The weighted-neighbor robustness check uses the same specification but

| Panel | Outcome | Main regressor | Controls / FE | $\hat{\beta}$ (SE) | $p$ | $N$ / clusters |
|---|---|---|---|---|---|---|
| Historical local | $B_{m,i}$ | $R^{\text{local}}_{m,i,10}$ | Model + market FE | $-0.0652$ $(0.0231)$ | $0.0048$ | $4{,}897/1{,}264$ |
| Historical weighted | $B_{m,i}$ | $R^{\text{local},w}_{m,i,10}$ | Model + market FE | $-0.0710$ $(0.0225)$ | $0.0016$ | $4{,}897/1{,}264$ |
| Domain M1 | $\bar{B}_{m,d}$ | $R_{m,d}$ | None | $-0.2038$ $(0.0430)$ | $< 0.001$ | 36 / – |
| Domain M2 | $\bar{B}_{m,d}$ | $R_{m,d}$ | $D_d$ | $-0.2216$ $(0.0445)$ | $< 0.001$ | 36 / – |
| Domain M3 | $\bar{B}_{m,d}$ | $R_{m,d}$ | Model FE + $D_d$ | $-0.2227$ $(0.0361)$ | $< 0.001$ | 36 / – |

*Table 2.* Historical resolved-market and aggregate domain-level recall regressions. The reported coefficient is always the coefficient on the recall-fidelity regressor. In Domain M2, $\hat{\gamma}_D = -0.1185$ ($SE = 0.0891$, $p = 0.193$); in Domain M3, $\hat{\gamma}_D = -0.1192$ ($SE = 0.0691$, $p = 0.0948$).

replaces $R^{\text{local}}_{m,i,10}$ with the cosine-similarity weighted score $R^{\text{local},w}_{m,i,10}$. Standard errors are clustered by condition ID in all market-level regressions.

For aggregate domain-level analyses, let

$$\bar{B}_{m,d} = \frac{1}{|\mathcal{I}_d|} \sum_{i \in \mathcal{I}_d} B_{m,i}, \qquad R_{m,d} = \frac{1}{|\mathcal{I}_d|} \sum_{i \in \mathcal{I}_d} R_{m,i},$$

where $\mathcal{I}_d$ is the set of resolved markets in domain $d$. No-recall/null recall observations are treated as $0.0$ when computing $R_{m,d}$. Domain difficulty is the mean binary entropy of Polymarket prices over open markets in the same domain,

$$D_d = \frac{1}{|\mathcal{O}_d|} \sum_{j \in \mathcal{O}_d} \left[ -\pi_j \log(\pi_j) - (1 - \pi_j) \log(1 - \pi_j) \right], \tag{9}$$

where $\pi_j$ is the contemporaneous market-implied probability of "Yes" for open market $j$. The full domain-level fixed-effect specification is

$$\bar{B}_{m,d} = \alpha_m + \beta R_{m,d} + \gamma D_d + \varepsilon_{m,d}. \tag{10}$$

The bivariate and difficulty-controlled variants omit $\alpha_m$ and/or $D_d$ as indicated in Table 2.

For unresolved open markets, true resolutions are unavailable. We therefore use market-price proxy outcomes:

$$\widetilde{B}_{m,j} = (p_{m,j} - \pi_j)^2, \qquad \widetilde{A}_{m,j} = |p_{m,j} - \pi_j|,$$

and the soft-target log loss

$$\widetilde{L}_{m,j} = -\pi_j \log(p_{m,j}) - (1 - \pi_j) \log(1 - p_{m,j}).$$

The market fixed-effect open-market specification is

$$Y_{m,j} = \alpha_m + \lambda_j + \beta R^{\text{local}}_{m,j,10} + \varepsilon_{m,j}, \tag{11}$$

where $Y_{m,j} \in \{\widetilde{B}_{m,j}, \widetilde{A}_{m,j}, \widetilde{L}_{m,j}\}$. The domain-level open-market comparison is

$$Y_{m,j} = \alpha_m + \delta_{d(j)} + \beta R^{\text{local}}_{m,j,10} + \gamma H(\pi_j) + \varepsilon_{m,j}, \tag{12}$$

where $H(\pi_j)$ is the binary entropy of the market price. Standard errors are again clustered by condition ID.

### A.8. Market Entropy

For a binary market with market probability $p_m$, we define:

$$H(p_m) = -p_m \log(p_m) - (1 - p_m) \log(1 - p_m). \tag{13}$$

If $p_m \approx 0.5$, entropy is high because the market is uncertain. If $p_m \approx 0$ or $p_m \approx 1$, entropy is low because the market is confident.

| Panel | Outcome | Term | Controls / FE | $\hat{\beta}$ (SE) | $p$ | $N$ / clusters |
|---|---|---|---|---|---|---|
| Open market FE | $\widetilde{B}_{m,j}$ | $R^{\text{local}}_{m,j,10}$ | Model + market FE | $-0.0284$ $(0.0108)$ | $0.0084$ | $5{,}724/1{,}897$ |
| Open market FE | $\widetilde{A}_{m,j}$ | $R^{\text{local}}_{m,j,10}$ | Model + market FE | $-0.0448$ $(0.0152)$ | $0.0032$ | $5{,}724/1{,}897$ |
| Open market FE | $\widetilde{L}_{m,j}$ | $R^{\text{local}}_{m,j,10}$ | Model + market FE | $-0.0606$ $(0.0696)$ | $0.384$ | $5{,}724/1{,}897$ |
| Open domain FE | $\widetilde{B}_{m,j}$ | $R^{\text{local}}_{m,j,10}$ | Model + domain FE + $H(\pi_j)$ | $0.0071$ $(0.0154)$ | $0.642$ | $5{,}724/1{,}897$ |
| Open domain FE | $\widetilde{B}_{m,j}$ | $H(\pi_j)$ | Model + domain FE | $0.0721$ $(0.0137)$ | $< 0.001$ | $5{,}724/1{,}897$ |
| Open domain FE | $\widetilde{L}_{m,j}$ | $R^{\text{local}}_{m,j,10}$ | Model + domain FE + $H(\pi_j)$ | $0.0140$ $(0.0544)$ | $0.797$ | $5{,}724/1{,}897$ |
| Open domain FE | $\widetilde{L}_{m,j}$ | $H(\pi_j)$ | Model + domain FE | $1.2601$ $(0.0469)$ | $< 0.001$ | $5{,}724/1{,}897$ |

*Table 3.* Open-market proxy regressions. Market fixed effects compare models on the same unresolved market. Domain fixed effects with entropy controls test whether local recall explains broader domain-level variation after accounting for market-price uncertainty.

### A.9. Modelwise, Domainwise Metrics

| Domain | Metric | Opus 4.7 | Sonnet 4.6 | Gemini 3.1 Pro | GPT-5.5 | Overall |
|---|---|---|---|---|---|---|
| Climate | FC Loss | 0.0534 | 0.0598 | 0.0540 | 0.0486 | 0.0540 |
| | FC Log Loss | 0.5714 | 0.5971 | 0.5645 | 0.5714 | 0.5761 |
| | RC Brier | 0.1271 | 0.1452 | 0.1139 | 0.1425 | 0.1322 |
| | RC Fidelity | 0.9167 | 0.8864 | 1.0000 | 0.8145 | 0.9044 |
| Commodities | FC Loss | 0.0911 | 0.0858 | 0.0802 | 0.0666 | 0.0809 |
| | FC Log Loss | 0.7709 | 0.7604 | 0.7225 | 0.6932 | 0.7367 |
| | RC Brier | 0.0978 | 0.1236 | 0.0843 | 0.0747 | 0.0951 |
| | RC Fidelity | 0.8488 | 0.8645 | 0.8704 | 0.8380 | 0.8554 |
| Companies | FC Loss | 0.1139 | 0.1148 | 0.1156 | 0.1135 | 0.1144 |
| | FC Log Loss | 0.7413 | 0.7547 | 0.7366 | 0.7560 | 0.7472 |
| | RC Brier | 0.1092 | 0.1507 | 0.1120 | 0.0941 | 0.1165 |
| | RC Fidelity | 0.8814 | 0.8626 | 0.9143 | 0.8820 | 0.8851 |
| Economics | FC Loss | 0.0708 | 0.0639 | 0.0643 | 0.0531 | 0.0631 |
| | FC Log Loss | 0.6553 | 0.6353 | 0.6358 | 0.5734 | 0.6249 |
| | RC Brier | 0.0666 | 0.0855 | 0.0578 | 0.0516 | 0.0654 |
| | RC Fidelity | 0.9651 | 0.9609 | 0.9733 | 0.9457 | 0.9612 |
| Elections | FC Loss | 0.0457 | 0.0581 | 0.0612 | 0.0393 | 0.0511 |
| | FC Log Loss | 0.5119 | 0.5343 | 0.5317 | 0.4595 | 0.5094 |
| | RC Brier | 0.0777 | 0.1063 | 0.0551 | 0.0567 | 0.0740 |
| | RC Fidelity | 0.9573 | 0.9043 | 0.9606 | 0.8740 | 0.9241 |
| Entertainment | FC Loss | 0.0600 | 0.0906 | 0.0680 | 0.0651 | 0.0709 |
| | FC Log Loss | 0.7020 | 0.8191 | 0.7138 | 0.7244 | 0.7399 |
| | RC Brier | 0.0867 | 0.1338 | 0.0584 | 0.0535 | 0.0831 |
| | RC Fidelity | 0.9423 | 0.9100 | 0.9250 | 0.9181 | 0.9239 |
| Financials | FC Loss | 0.0776 | 0.0760 | 0.0957 | 0.0780 | 0.0818 |
| | FC Log Loss | 0.6817 | 0.6825 | 0.7221 | 0.7566 | 0.7107 |
| | RC Brier | 0.1355 | 0.1925 | 0.0947 | 0.0984 | 0.1303 |
| | RC Fidelity | 0.8483 | 0.8224 | 0.9069 | 0.8663 | 0.8610 |
| Science/Technology | FC Loss | 0.0642 | 0.0871 | 0.0934 | 0.0621 | 0.0767 |
| | FC Log Loss | 0.6075 | 0.6912 | 0.6695 | 0.5667 | 0.6337 |
| | RC Brier | 0.1223 | 0.1737 | 0.1321 | 0.1096 | 0.1344 |
| | RC Fidelity | 0.9433 | 0.8978 | 0.9588 | 0.9111 | 0.9278 |
| Overall | FC Loss | 0.0721 | 0.0795 | 0.0791 | 0.0658 | 0.0741 |
| | FC Log Loss | 0.6552 | 0.6843 | 0.6621 | 0.6377 | 0.6598 |
| | RC Brier | 0.1029 | 0.1389 | 0.0885 | 0.0851 | 0.1039 |
| | RC Fidelity | 0.9129 | 0.8886 | 0.9386 | 0.8812 | 0.9053 |

### A.10. Murphy Breakdown

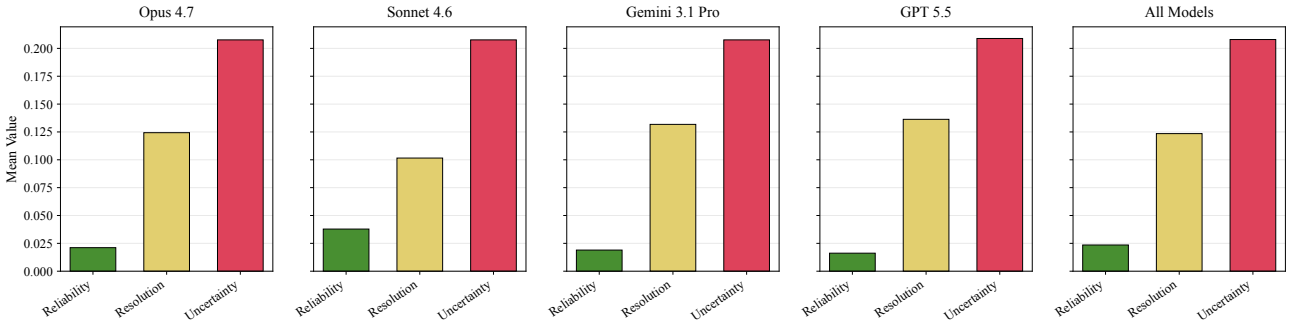


*Figure 7.* Murphy Breakdown of recall Brier for each model. A lower reliability score is more favorable while a higher resolution score is more favorable.

| Model | $n$ | Brier | Uncertainty | Reliability | Resolution | Null Rate |
|---|---|---|---|---|---|---|
| Claude Opus 4.7 | 1150 | 0.1017 | 0.2110 | 0.0102 | 0.0031 | 0.3174 |
| Claude Sonnet 4.6 | 1150 | 0.1406 | 0.2110 | 0.0239 | 0.0052 | 0.3252 |
| Gemini 3.1 Pro | 1150 | 0.0877 | 0.2110 | 0.0080 | 0.0042 | 0.2591 |
| GPT-5.5 | 993 | 0.0816 | 0.2132 | 0.0075 | 0.0022 | 0.2115 |

*Table 4.* Murphy decomposition components and null prediction rates by model. Lower values are better for Brier, Reliability, and Null Rate, while higher values are better for Resolution.