# MacroBench: Measuring Frontier LLM Macroeconomic Forecasting Ability

**Anonymous Author(s)** [1]

## Abstract

Macroeconomic forecasting is an important task, yet no existing benchmark both anonymizes the underlying data and validates that evaluations are uncontaminated, a critical gap given that historical price series are heavily represented in pretraining corpora. We introduce MacroBench, a benchmark that measures macroeconomic forecasting in LLMs via US 10-year Treasury yield forecasts across 35 years of macro regimes, with anonymized historical and temporal context and a contamination adjustment mechanism. Each task presents a 36-month window of twelve z-scored macro indicators, gives the model a fixed statistical toolset in Python, and elicits a distributional 10Y yield forecast at 1-, 3-, and 6-month horizons. After contamination adjustment, no frontier LLM significantly outperforms walk-forward AR(1) on anonymized Treasury yield forecasting except GPT-5.5, which narrowly beats it. More broadly, we establish a recipe for contamination-free benchmarks on any historically rich macro series, enabling rigorous evaluation in central-bank scenario analysis, monetary policy research, and debt markets.

## 1. Introduction

Macroeconomic forecasting is a hard, economically important task with mature statistical baselines that remain difficult to beat in out-of-sample tests (Stock & Watson, 2003; Faust & Wright, 2013). A natural question is whether frontier large language models (LLMs), with broad macroeconomic context and quantitative tools, can match those baselines and serve as useful probes of economic reasoning. LLMs are known to memorize substantial fractions of their training data (Carlini et al., 2023) and to recall numerical sequences (Gruver et al., 2023). In LLM-based financial forecasting specifically, Lopez-Lira and colleagues argue that strong predictive performance on historical series is largely explained by memorization rather than reasoning (Lopez-Lira et al., 2024).

This creates a conflict as evaluating LLM forecasting ability requires testing the model across many macro regimes, but without relying on the LLM's recall from its pretraining corpora. Existing forecasting benchmarks resolve the bind by relying exclusively on future predictions, including ForecastBench (Karger et al., 2024), Halawi et al. (Halawi et al., 2024), and the silicon-crowd study (Schölkopf et al., 2024); this avoids contamination by construction, but it also makes it impossible to stress-test models on a variety of macro regimes. Many financial time-series benchmarks for LLMs, by contrast, evaluate on historical data but typically do not measure or adjust for contamination (Gruver et al., 2023).

Our goal with **MacroBench** is to build a benchmark that measures LLM forecasting ability across a variety of macro regimes by anonymizing time series data and adjusting for recall-based contamination. We specifically measure distributional forecasts of one-, three-, and six-month changes in the US 10-year Treasury yield across 35 years of data. We cover the anonymization and contamination adjustment protocol in detail in section 2.

**Contributions.** (i) An anonymization protocol that strips dates, levels, country labels, and z-scores time series data within a 36-month window, (ii) A deterministic contamination adjustment pipeline that flags and adjusts Brier scores for contamination in evals. (iii) An empirical demonstration on 2,351 result-bearing instances across four frontier LLMs, showing that frontier models struggle to beat walk-forward AR(1) baselines, and Gemini-3.1-Pro's perceived lead in raw prediction accuracy collapses after our contamination adjustment.

## 2. Methodology

### 2.1. Data and Task Overview

We pull 12 monthly end-of-month series from FRED (Federal Reserve Bank of St. Louis, 2026) spanning 1989–2026: the 2Y, 5Y, and 10Y Treasury constant-maturity yields; the effective federal funds rate; civilian unemployment (UN-

[1]Anonymous Institution. Correspondence to: Anonymous <anon@example.com>.

*Preprint. May 13, 2026.*

RATE); industrial production and nonfarm payrolls (YoY); headline and core CPI (YoY); the S&P 500 trailing return; the VIX; the broad trade-weighted USD index (YoY); and the BAA-10Y credit spread. The future 10Y rate is the target, and the 12 36-month trailing series are anonymized as described below. We have $N = 393$ for the 1-month prediction task, $N = 130$ for the 3-month prediction, and $N = 65$ for the 6-month prediction per model.

For each anchor month $T$, the model receives: (a) a $36 \times 12$ matrix of *within-window z-scored* monthly indicators with rows labeled $t-35, \ldots, t-0$ and no dates; (b) four bucket boundaries $[q_{20}, q_{40}, q_{60}, q_{80}]$ in local-$\sigma$ units, where $\sigma$ represents the units of standard deviation for that time series. (c) a Python sandbox with `df` pre-loaded, Python libraries including `Pandas` and `numpy`, and a budget of 40 tool calls. The sandbox restricts all network requests and additional Python imports.

The model returns a JSON object with two fields: `probs`, a length-5 vector $[p_1, \ldots, p_5]$ summing to 1, where $p_b$ is the predicted probability that the realised $H$-month change in the 10Y yield falls in bucket $B_b$; and `rationale_short`, a short natural-language explanation of the forecast. The bucket boundaries themselves are calibrated to recent history: at each anchor $T$ we take all $H$-month changes of the 10Y yield observed inside the 36-month input window, divide them by their standard deviation $\sigma_{\text{local}}$, and set $[q_{20}, q_{40}, q_{60}, q_{80}]$ to their 20th, 40th, 60th, and 80th percentiles, so each of the five buckets holds exactly 20% of the in-window changes. Eliciting a full probability distribution over five equal-mass buckets lets us measure whether the model can identify shifts away from the recent 36-month historical distribution, since any deviation from the uniform 20%-per-bucket baseline reflects a genuine directional or magnitude signal the model has extracted from the state pack rather than from the target's recent history. We discuss extensions of this approach to other distributions in the appendix.

### 2.2. Data Anonymization

The state pack contains only the z-scored matrix and informative variable names. It does not contain any year or date, any absolute level, the in-window mean or std of any series, or any reference to country, regime, or central bank. Normalization removes regime markers, which prevents LLMs from being able to recall the time period.

For each anchor $T$ and horizon $H \in \{1, 3, 6\}$ we compute the standard deviation $\sigma_{\text{local}}$ of past $H$-period changes inside the 36-month window, and place bucket boundaries at the local quintiles of those changes. The uniform distribution $[0.2]^5$ scores Brier $= 0.800$ on the historical in-window distribution, so any model output must extract out-of-window signal to beat it.

### 2.3. Contamination-Adjusted Metrics

We report three families of metrics on every $(\text{model}, \text{anchor}, \text{horizon})$ instance. We first calculate *Accuracy*, which is multinomial Brier

$$\text{Br} = \sum_{b=1}^{5} (p_b - \mathbf{1}[\text{realized} = b])^2$$

range $[0, 2]$, lower is better (Brier, 1950; Gneiting & Raftery, 2007); we also report the Brier Index

$$(1 - \sqrt{\text{Br}/2}) \cdot 100\%$$

(higher is better) to match ForecastBench's reporting convention.

Even with window z-scoring and no other temporal information, a sufficiently determined model can recover the anchor date from cross-series correlations and general patterns of the time series data. We therefore run three deterministic checks (no LLM-as-judge calls): *(i) direct lexical evidence* – the trace contains a year, named-event, or explicit month-year string inside the hidden anchor window (`anchor_identified`); *(ii) output-shape anomalies* – entropy, peakedness, and the famous-vs-non-famous Brier delta defined in Section 2.3; *(iii) runtime trace evidence* – attempted imports of `requests`, `urllib`, `pandas_datareader`, `yfinance`, `fredapi`, HTTP calls, subprocess/socket calls, and FRED-CSV signatures (benign documentation URLs excluded).

We then report a *contamination-adjusted Brier score* alongside the raw score. For every row flagged `anchor_identified` we replace the model's distribution with the uniform $[0.2]^5$ and recompute Brier, which by construction scores $0.800$ on those rows; unflagged rows are left alone. The adjustment is deliberately conservative: it neither penalises low-entropy forecasts without anchor evidence nor zeroes out a model's score for borderline lexical mentions. Entropy, peakedness, and the famous-vs-non-famous delta are reported as corroborating signals but do not drive the adjustment.

*Distribution shape* captures how confident the model is, independent of whether it is right: Shannon entropy $H(p) = -\sum_b p_b \log p_b$ (uniform reference $\log 5 \approx 1.609$) and peakedness $\text{peak}(p) = \max_b p_b$ (uniform reference $0.2$). *Stress-period stratification* reports Brier separately on famous and non-famous anchors, where famous windows are the GFC (2007-08 to 2009-12), the Taper Tantrum, COVID (2020-02 to 2020-06), the 2022–23 hiking cycle, and LTCM (1998-08 to 1998-10); the famous-minus-non-famous Brier delta is reported as an unadjusted signal alongside the audit, since a model that improves substantially on stress periods on real macro data is showing the failure mode contamination would produce. Anecdotally, we found that the models

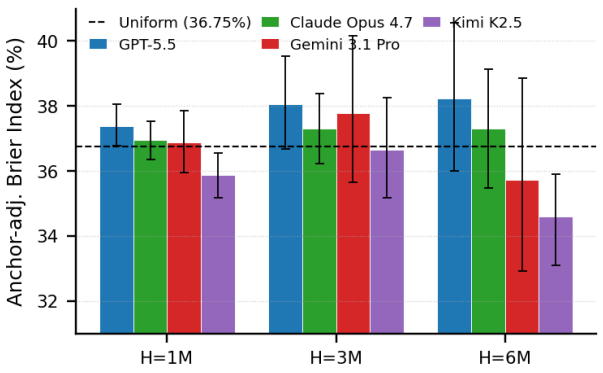



*Figure 1.* Anchor-adjusted Brier Index $(1 - \sqrt{\mathrm{Br}/2}) \cdot 100\%$ by model and horizon (higher is better). Thin black lines show 95% block-bootstrap CIs (block length 12 at 1M, 4 at 3M, 2 at 6M; 2,000 replicates). The dashed line marks the uniform $[0.2]^5$ baseline (36.75%).

would output a very low-entropy, high-peaked distribution when they were able to reward-hack the data and find the exact time period of the input data (see appendix C for more).

## 3. Models and Baselines

**Scored models.** We evaluate four frontier LLMs at default sampling, each invoking a Python sandbox (E2B) up to 40 times per instance: Anthropic `claude-opus-4.7`, OpenAI `gpt-5.5`, Google `gemini-3.1-pro-preview`, and Moonshot AI `kimi-k2.5`. Mean tool calls range from 6 (Opus, GPT-5.5) to 34 (Gemini); Gemini also exhibits the largest variance in tool use and is the only model that frequently hits the budget. Parse failures and timeouts are handled by re-prompting once before recording a fallback $[0.2]^5$. Kimi exhibits a notably higher parse-failure rate at $H = 1$ (73 out of 392) which we discuss as a limitation in Section 5.

**Baselines.** We compare models to two baselines: (i) *Uniform climatology*, fixed at $[0.2]^5$, scoring exactly 0.800 by construction; and (ii) *Walk-forward AR(1)*, where for each anchor $T$, AR(1) is fit independently using only data inside $T$'s 36-month window, represented by $z_t = \Delta y_t / \sigma_{\text{monthly}}$. For $H = 1$ we use the one-step forecast; for $H = 3$ we use the closed-form cumulative 3-step AR(1) distribution, rescaled into $\sigma_{\text{local}}$ units and integrated over each bucket. We avoid fitting on overlapping $H$-period changes as it would inflate $\hat{\phi}$ through sample overlap.

## 4. Results

### 4.1. Frontier LLMs barely beat the uniform baseline

Figure 1 and Table 1 report the raw and anchor-adjusted mean Brier across models and horizons. The substantive finding is that frontier LLMs barely clear a uniform $[0.2]^5$ baseline once contamination is accounted for. After adjustment, only GPT-5.5 has a CI that clears uniform at every horizon, and it does so narrowly: $0.784/0.767/0.763$ versus the uniform 0.800. Claude Opus 4.7 is statistically indistinguishable from a two-parameter walk-forward AR(1) (0.795 vs 0.792 at 1M, 0.786 vs 0.794 at 3M); Kimi K2.5 trails the uniform baseline at every horizon. Gemini's large raw advantage ($0.739/0.698/0.688$) is overwhelmingly contributed by anchor-identified rows: replacing those rows with the uniform distribution moves Gemini to $0.797/0.774/0.826$, surrendering essentially all of its raw lead; the other three models barely move under the same adjustment because their anchor-detection rates are far lower. The combined picture is that an audit-clean LLM is, at best, a narrow improvement on classical baselines on this task.

*Table 1.* Raw and anchor-adjusted mean Brier across four frontier LLMs and three horizons. The *Adj.* column replaces anchor-identified rows (lexical audit found a year, named-event, or month-year string from inside the hidden 36-month window in the trace) with the uniform $[0.2]^5$ before averaging. **Bold** marks the lowest adjusted Brier per horizon. AR(1) is not audited, so its adjusted entry equals raw; AR(1) at $H$=6 is left to future runs.

| | 1M | | 3M | | 6M | |
|---|---|---|---|---|---|---|
| Entity | Raw | Adj. | Raw | Adj. | Raw | Adj. |
| gpt-5.5 | 0.783 | **0.784** | 0.767 | **0.767** | 0.763 | **0.763** |
| claude-opus-4.7 | 0.795 | 0.795 | 0.785 | 0.786 | 0.794 | 0.786 |
| gemini-3.1-pro | 0.739 | 0.797 | 0.698 | 0.774 | 0.688 | 0.826 |
| kimi-k2.5 | 0.822 | 0.823 | 0.809 | 0.802 | 0.853 | 0.856 |
| AR(1) walk-fwd | 0.792 | 0.792 | 0.794 | 0.794 | — | — |
| Uniform $[0.2]^5$ | 0.800 | 0.800 | 0.800 | 0.800 | 0.800 | 0.800 |

### 4.2. Audit findings

The three audit signals defined in Section 2.3 concentrate on a single model. Figure 2 summarises all three across horizons.

*Anchor identification.* Gemini's traces are flagged anchor-identified in $9.7\%/12.3\%/23.1\%$ of rows at $H$=1/3/6; GPT-5.5's rates are $0.25\%/0\%/0\%$, with Opus and Kimi in between (3–15%). Conditional on anchor detection, Gemini's entropy collapses from $1.48 \to 0.69$ at 1M, $1.40 \to 0.67$ at 3M, $1.27 \to 0.64$ at 6M, and peak rises to 0.75–0.77 – a sharply different distribution from the no-anchor rows. Anchor-row Brier on those traces is 0.20 (versus 0.79 on no-anchor rows), so the model is both confident and correct when it has named the date.

*Output shape.* Across all rows, Gemini's mean peakedness is $0.38/0.42/0.53$ at 1M/3M/6M with a $p90$ above 0.6 and a maximum near 0.99; the other three models stay below 0.40 mean peak and 0.55 at $p90$, with maxima below 0.75. Figure 2(b) shows the right-tail mass directly.

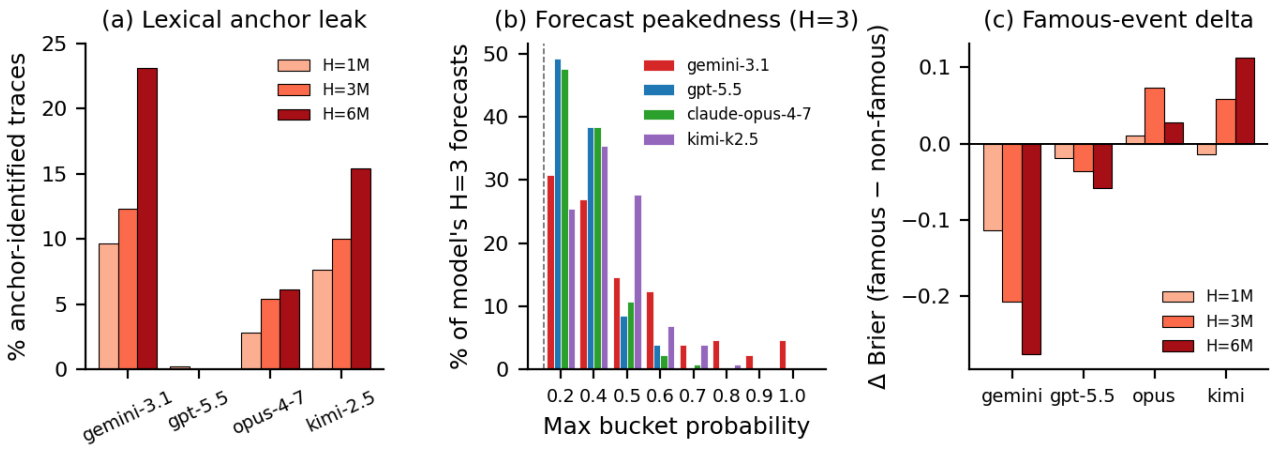



*Figure 2.* Audit signals across horizons. **(a)** Fraction of each model's traces flagged anchor-identified by the lexical check. **(b)** Distribution of per-instance peakedness (max bucket probability) at $H{=}3$. **(c)** Brier shift on famous-event anchors. The three signals concentrate on a single model.

*Famous-vs-non-famous delta (unadjusted).* A clean forecaster should be flat or worse on stress periods. The unadjusted Brier delta tells the same story as the other two signals: Gemini improves by $-0.113/-0.207/-0.276$ at 1M/3M/6M, monotone in horizon and an order of magnitude larger than any other model. GPT-5.5 has small negative deltas consistent with mild absorbed pattern-matching ($-0.018/-0.036/-0.058$); Opus and Kimi mostly degrade on famous events, the prior-consistent direction.

*Runtime trace evidence.* Gemini is the only scored model with non-zero high-severity sandbox flags (13 attempted imports of `urllib`/`requests`/`pandas_datareader` across 588 traces); after excluding benign documentation URLs, the other three models have zero.

## 5. Discussion and Conclusion

Both Gemini and Kimi show significant evidence of leakage. Figure 2(b) shows a small but distinct mode in Gemini's peakedness distribution at 0.9–1.0 that none of the other three models produce, consistent with the diagnosis that Gemini is reverse-engineering the anchor and emitting a near-degenerate forecast on those rows. Combining the audit signals with our own trace review, the picture is that Gemini is genuinely good at recovering the time period and escaping the anonymization, so its high-peakedness predictions land correctly more often than chance. Kimi K2.5 also attempts to reverse-engineer the anchor but does not succeed at the same rate, which is why its peakedness is elevated relative to GPT-5.5 and Opus while its raw Brier is the worst of the four.

Figure 2(c) shows that Gemini has the largest famous-vs-non-famous Brier gap, which fits the reward-hacking story: famous regimes (the GFC, COVID, the 2022–23 hiking cycle) are the easiest anchors to recognise from in-window structure alone, so a model that recovers the date will pick up the biggest gain there. Opus and GPT-5.5 stay close to their non-famous mean. Across horizons, most models score marginally better at $H{=}6$ than at $H{=}1$ in raw Brier, plausibly because the 6-month change averages over short-horizon noise and is easier to pattern-match from the in-window dynamics, whereas 1-month signals are more noise-dominated.

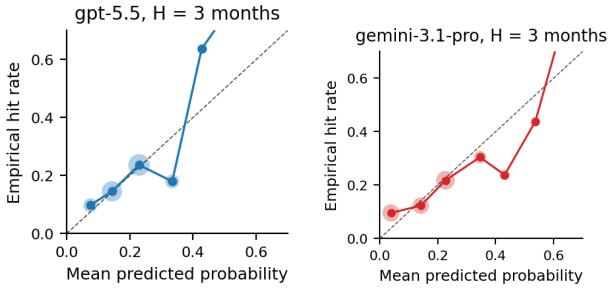
**Limitations and next steps.** Our anonymization protocol and contamination adjustments are imperfect. The models we tested were still able to determine the time period despite z-scoring the data to account for regime change. Additionally, while we were able to catch many explicit instances of contamination, we do not yet have a way of identifying implicit contamination. We primarily seek to extend this work in three ways: (1) run the evaluations on a rolling basis after pretraining cutoffs (January to May 2026); (2) extend the protocol to WTI, broad USD, EUR/USD, USD/JPY, S&P 500, and VIX; (3) source and score ground-truth annotation of macro reasoning to develop a better reward mechanism to score reasoning and prediction quality.

## Impact Statement

We release a measurement, not a trading system. There are potential societal consequences of work in this area, including over-reliance on LLM forecasts in financial decision-making, none of which we feel must be specifically highlighted here.

*Table 2.* Anchor-conditional output statistics (Appendix B). For each (model, horizon) cell, rows are split by whether the lexical audit flagged them as anchor-identified; we then report the mean entropy, mean peakedness (max bucket probability), and mean Brier on the flagged and unflagged subsets separately. $N_a$ is the count of flagged rows. Lower entropy and higher peak indicate a more collapsed/confident distribution. Dashes indicate that there were no anchor-flagged rows in that cell.

| | | | Mean entropy | | Mean peakedness | | Mean Brier | |
|---|---|---|---|---|---|---|---|---|
| Model | $H$ | $N_a$ | Anchor | No-anchor | Anchor | No-anchor | Anchor | No-anchor |
| GPT-5.5 | 1M | 1 | 1.398 | 1.553 | 0.450 | 0.293 | 0.393 | 0.784 |
| Claude Opus 4.7 | 1M | 11 | 1.544 | 1.542 | 0.300 | 0.309 | 0.799 | 0.795 |
| Gemini 3.1 Pro | 1M | 38 | 0.690 | 1.478 | 0.748 | 0.345 | 0.199 | 0.797 |
| Kimi K2.5 | 1M | 32 | 1.515 | 1.474 | 0.298 | 0.331 | 0.804 | 0.823 |
| GPT-5.5 | 3M | 0 | — | 1.528 | — | 0.310 | — | 0.767 |
| Claude Opus 4.7 | 3M | 7 | 1.493 | 1.504 | 0.326 | 0.323 | 0.791 | 0.785 |
| Gemini 3.1 Pro | 3M | 16 | 0.666 | 1.401 | 0.748 | 0.377 | 0.183 | 0.771 |
| Kimi K2.5 | 3M | 13 | 1.341 | 1.407 | 0.391 | 0.377 | 0.871 | 0.803 |
| GPT-5.5 | 6M | 1 | 1.441 | 1.470 | 0.420 | 0.351 | 1.032 | 0.759 |
| Claude Opus 4.7 | 6M | 4 | 1.415 | 1.446 | 0.375 | 0.368 | 0.927 | 0.785 |
| Gemini 3.1 Pro | 6M | 15 | 0.640 | 1.271 | 0.773 | 0.458 | 0.204 | 0.834 |
| Kimi K2.5 | 6M | 11 | 1.251 | 1.478 | 0.448 | 0.311 | 0.817 | 0.860 |




*(a)* GPT-5.5 *(b)* Gemini 3.1 Pro

*Figure 3.* Reliability at $H$=3. Bubble size encodes the number of (instance, bucket) pairs in each predicted-probability bin.

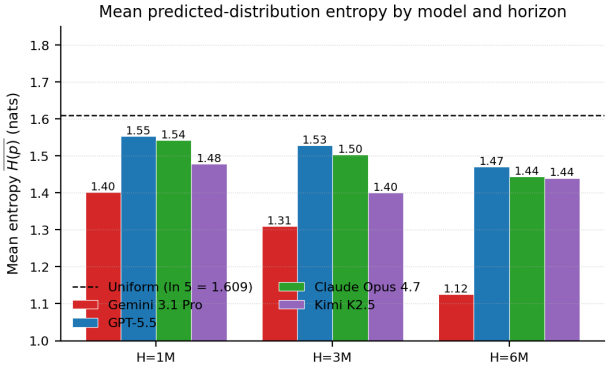



*Figure 4.* Mean predicted-distribution entropy by model and horizon. Uniform reference $\ln 5 \approx 1.609$ shown as a dashed line; lower entropy means a more confident forecast.

## A. Calibration

Figure 3 shows reliability at $H$=3 for GPT-5.5 and Gemini 3.1 Pro. GPT-5.5's mass lives in the 0.1–0.3 range and tracks the diagonal with mild over-confidence in the highest bin; Gemini's mass extends past 0.4 and the upper bins are where the audit signal lives: high-confidence forecasts that are usually right.

## B. Anchor-conditional Output Statistics

To check whether anchor-flagged rows really look different from the rest, we split each (model, horizon) cell into anchor-identified and non-flagged subsets and report the mean entropy, mean peakedness, and mean Brier on each subset (Table 2). Gemini is the only model where anchor detection consistently lines up with a collapsed distribution and a much better Brier: on anchor-flagged rows, entropy drops from roughly 1.3–1.5 to $\sim$ 0.65, peakedness rises from $\sim$ 0.35–0.45 to $\sim$ 0.75, and mean Brier falls to $\sim$ 0.20 versus $\sim$ 0.80 on the unflagged rows. Opus and GPT-5.5 barely move on either subset; Kimi sometimes shows a higher peak on the flagged rows but is not reliably more correct.

## C. Entropy Diagnostics

Figure 4 reports mean predicted-distribution entropy by model and horizon. Gemini's mean entropy collapses monotonically with horizon ($1.40 \to 1.31 \to 1.12$), consistent with the anchor-identification and peakedness signals reported in Section 4; the other three models stay near 1.4–1.55 across horizons. The uniform reference is $\ln 5 \approx 1.609$.

## D. Alternative Elicitation Schemes

We considered two alternatives before settling on the discrete five-bucket output. A point estimate collapses the

model's belief into a single number, discarding uncertainty information and conflating forecasting skill with implicit risk attitude, since a conservative model that always predicts near zero scores well in low-volatility regimes regardless of whether it has extracted any signal. A parametric continuous distribution such as a Gaussian or Student-$t$ preserves uncertainty but imposes a shape assumption that is poorly matched to the target: yield changes around macroeconomic turning points are visibly asymmetric, with cutting cycles producing left-skewed distributions, hiking surprises producing right-skewed ones, and risk-off episodes generating fat left tails, all of which a Gaussian smooths away by construction. More flexible families (skew-normal, generalized normal, mixtures, chi, log-normal) restore shape flexibility at the cost of additional parameters that LLMs calibrate poorly and, in the non-negative cases, require an arbitrary location shift for signed changes. The discrete equal-mass scheme avoids both pathologies: it elicits a full belief, expresses any shape (including skew and bimodality) without a shape assumption, and pins the uninformative baseline at $(0.2, 0.2, 0.2, 0.2, 0.2)$, at the cost of within-bucket resolution that we view as appropriate to the noise level of monthly yield changes.

## References


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

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *International Conference on Learning Representations*, 2023.

Faust, J. and Wright, J. H. Forecasting inflation. In Elliott, G. and Timmermann, A. (eds.), *Handbook of Economic Forecasting*, volume 2A, pp. 2–56. Elsevier, 2013.

Federal Reserve Bank of St. Louis. FRED economic data. https://fred.stlouisfed.org/, 2026.

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

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. In *arXiv:2402.18563*, 2024.

Karger, E., Bastani, H., Yueh-Han, C., Jacobs, Z., Halawi, D., Zhang, F., and Tetlock, P. E. ForecastBench: A dynamic benchmark of AI forecasting capabilities. In *arXiv:2409.19839*, 2024.

Lopez-Lira, A., Tang, Y., and Zhu, M. The memorization problem: Can we trust LLMs' economic forecasts? *arXiv:2504.14765*, 2024.

Schölkopf, H. et al. Wisdom of the silicon crowd: LLM ensemble prediction capabilities rival human crowd accuracy. *arXiv:2402.19379*, 2024.

Stock, J. H. and Watson, M. W. Forecasting output and inflation: The role of asset prices. *Journal of Economic Literature*, 41(3):788–829, 2003.