

Betting on Reasoning: Predicting Forecast Reliability in Prediction Markets

Stanford CS224N Custom Project

Rahul Rejeev

Department of Computer Science
Stanford University
rejeev@stanford.edu

Abstract

Large language models can generate probability estimates for real-world events, but it remains unclear when these forecasts should be trusted. We investigate this question along two axes: (1) whether retrieval-augmented generation and structured reasoning improve LLM forecasting on prediction markets, and (2) whether the linguistic properties of reasoning traces predict forecast accuracy without access to ground truth. We build a forecasting pipeline over 100 resolved binary questions from Polymarket, which is an online prediction market where participants trade on the outcomes of events and then evaluate five prompting strategies in a 2×2 ablation of ($\pm$ RAG $\times$ $\pm$ CoT) plus self-consistency sampling. Retrieval was found to be the dominant factor in the accuracy of LLMs in predicting outcomes. RAG strategies achieve Brier scores of 0.17, a 32% improvement over the naive baseline and roughly 40% better than non-retrieval strategies. Chain-of-thought and self-consistency provide marginal additional benefit. For meta-prediction, a logistic regression classifier over 19 hand-crafted linguistic features achieves 66% accuracy and 0.60 AUC-ROC under leave-one-out cross-validation, outperforming a sentence-embedding baseline (54%, 0.48 AUC), suggesting that reasoning traces encode interpretable but only modest signals of epistemic reliability.

1 Key Information

- Mentor: Qinan Yu
- External Collaborators: N/A
- Sharing project: No
- Late Days (if you are using any):

Team member	Late days remaining
Rahul Rejeev	1
Total pooled	1
Team size	1
Late days applied for the report	1

2 Introduction

Prediction markets aggregate crowd beliefs into probability estimates for future events and have been shown to produce well-calibrated forecasts across politics, sports, finance, and science [1]. Recent work has explored whether LLMs can match this crowd accuracy. Schoenegger et al. [2] showed that an ensemble of 12 LLMs achieves aggregate Brier scores statistically indistinguishable from a 925-person human crowd, while Halawi et al. [3] demonstrated that retrieval-augmented LLM systems can approach human-level forecasting performance.

However, these studies focus on *whether* LLMs can forecast well, not on *when* their forecasts should be trusted. In practice, a forecasting system is only useful if users can distinguish reliable predictions from unreliable ones. This motivates our central research question: **do the linguistic properties of LLM reasoning traces contain predictive signals about forecast accuracy?**

We frame this as a two-level problem. At the first level, we build a retrieval-augmented forecasting pipeline that generates probability estimates and reasoning traces for binary prediction market questions from Polymarket. At the second level, we train classifiers that take a reasoning trace as input and predict whether the associated forecast will be accurate, without access to the ground truth outcome. This meta-prediction task treats reasoning traces as standalone textual artifacts and investigates whether their structure (uncertainty language, evidence usage, reasoning depth) correlates with reliability.

Our contributions are: (1) a systematic 2×2 ablation study isolating the effects of retrieval and chain-of-thought reasoning on forecasting quality across 100 balanced markets; (2) a feature-based reliability classifier demonstrating that reasoning traces encode above-chance signals of forecast accuracy; and (3) a comparison between hand-crafted linguistic features and sentence-embedding representations for reasoning-trace classification at small sample sizes.

3 Related Work

LLM forecasting. Schoenegger et al. [2] introduced the “wisdom of the silicon crowd” approach, aggregating predictions from 12 diverse LLMs to match 925 human forecasters on 31 Metaculus questions (Brier ≈ 0.20). They identified a systematic acquiescence bias where LLMs tend to predict probabilities above 0.5 regardless of base rates. Halawi et al. [3] showed that a retrieval-augmented system using GPT-4 with access to news articles can approach human-crowd accuracy, motivating the use of RAG for forecasting. Our work extends both by conducting controlled ablations to isolate the contributions of retrieval versus reasoning strategy.

Chain-of-thought reasoning. Wei et al. [4] demonstrated that prompting LLMs to produce intermediate reasoning steps improves performance on arithmetic, commonsense, and symbolic reasoning tasks. Wang et al. [5] extended this with self-consistency decoding, sampling multiple reasoning paths and aggregating answers via majority vote or averaging. We apply both techniques to probabilistic forecasting, where the output is a continuous probability rather than a discrete answer, and evaluate whether the diversity of reasoning paths improves calibration.

Retrieval-augmented generation. Lewis et al. [6] introduced RAG, combining parametric knowledge in pretrained models with non-parametric retrieval from external documents. For forecasting, retrieval is especially important because events may postdate the model’s training data. Our pipeline retrieves time-filtered news articles published before market resolution, providing the model with relevant context it could not have seen during training.

Calibration and uncertainty. Forecasting evaluation centers on the Brier score [7], which measures the mean squared error between predicted probabilities and binary outcomes. A model is considered calibrated if its predicted probabilities match actual outcomes: for example, among all events where the model predicts 70%, roughly 70% should actually occur. Prior work evaluates calibration by looking at the predicted probabilities themselves. Our reliability classifier takes a different approach: instead of examining the probability output, it analyzes the *reasoning text* that led to the prediction and asks whether linguistic patterns in that text (such as hedging, citing of evidence, or using confident language) can predict whether the forecast will turn out to be accurate.

4 Approach

Our system has two components: a forecasting pipeline that generates probability estimates and reasoning traces (Section 4.1), and a reliability classifier that predicts forecast accuracy from traces alone (Section 4.2).

4.1 Forecasting Pipeline

Data collection. We collected 100 resolved binary markets from Polymarket’s Gamma API, balanced across five categories (politics, crypto, sports, science/technology, economics) with exactly 10 YES and 10 NO resolutions per category. Markets were filtered to require binary outcomes, unambiguous resolution (final prices at exactly 0 or 1), at least \$10,000 trading volume, resolution dates between January 2025 and February 2026, and a minimum 14-day window between creation and closure to ensure sufficient news coverage.

Retrieval. We use a search-based retrieval approach: for each market, we query the Serper.dev Google News API using the market question as the search string and filter results to dates before market resolution. The top 5 articles (title, source, date, and snippet) are formatted as numbered sources and concatenated into a string which is then appended to the prompt. This is simpler than embedding-based RAG [6], as we do not have a vector store or a learned retriever, but it provides the model with relevant, time-filtered external information it would not have from training alone. All 100 markets have at least one retrieved article.

Prompting strategies. We implement five strategies in a 2×2 ablation grid plus self-consistency:

1. **Direct (no RAG):** The model receives only the question and outputs a single probability with no reasoning and no retrieved context.
2. **CoT (no RAG):** The model reasons step-by-step (analyzing YES factors, NO factors, evidence, and uncertainties) but receives no retrieved context, relying only on parametric knowledge.
3. **Direct + RAG:** The model receives retrieved news context and outputs a single probability with no intermediate reasoning.
4. **CoT + RAG:** The model receives retrieved context and reasons step-by-step before outputting a probability. The CoT prompt elicits structured analysis:
“Think through this step by step: 1. What are the key factors that would make this resolve YES? 2. What are the key factors that would make this resolve NO? 3. What does the available evidence suggest? 4. Are there any important uncertainties or conflicting signals?”
5. **Self-Consistency + RAG:** Following Wang et al. [5], we generate $N = 5$ independent CoT traces per market at temperature 1.0 (vs. 0.3 for other strategies) and average the extracted probabilities to produce the final estimate.

All strategies use the same calibration-focused system prompt: *“Be calibrated: a 70% prediction should resolve YES about 70% of the time. Do NOT default to extreme probabilities; use the full range from 0.05 to 0.95.”*

4.2 Reliability Classifier

The reliability classifier addresses a different question from the forecasting pipeline: given only the reasoning trace (the text the model wrote while thinking through a question), can we predict whether the resulting forecast will be accurate? The classifier receives no information about the actual outcome. We label each CoT+RAG forecast as *accurate* if its Brier score is below 0.25 (better than a coin flip) or *inaccurate* otherwise, then train two classifiers to predict this label.

Feature-based classifier. We extract 19 hand-crafted features from each reasoning trace, organized into five groups:

- **Uncertainty language** (6 features): counts and ratios of hedge words (e.g., *might*, *could*, *possibly*, *unlikely*, *uncertain*), certainty words (e.g., *definitely*, *clearly*, *guaranteed*, *confident*), the hedge-to-certainty ratio, and a negation count (*not*, *never*, *cannot*).
- **Evidence and reasoning markers** (3 features): counts of evidence phrases (e.g., *according to*, *data suggests*, *historically*), contrast markers (e.g., *however*, *although*, *on the other hand*), and causal markers (e.g., *because*, *therefore*, *as a result*).

- **Structural features** (4 features): number of question marks, numbered list items, bullet points, and total structured items.
- **Surface statistics** (4 features): word count, sentence count, average sentence length, and lexical diversity (unique words divided by total words).
- **Probability features** (2 features): count of numeric references in the text and probability extremity ($|p - 0.5|$), which captures how far the forecast is from maximum uncertainty. A prediction of 0.95 has extremity 0.45; a prediction of 0.50 has extremity 0.

All features are standardized to zero mean and unit variance. We use L_2 -regularized logistic regression ($C = 1.0$) with balanced class weights, which up-weights the minority class (inaccurate forecasts) so the model does not simply learn to predict “accurate” for everything.

Embedding-based classifier. As a neural baseline, we use the pretrained all-MiniLM-L6-v2 sentence-transformer [8] as a frozen feature extractor: each reasoning trace is encoded into a 384-dimensional dense vector, without any fine-tuning of the transformer weights. These embeddings are then standardized and classified with the same logistic regression setup as the feature-based approach. The key difference is the input representation: 19 interpretable features versus 384 learned embedding dimensions.

Evaluation. With only 100 labeled samples, a traditional train/test split would leave too little data for either training or evaluation. Instead, we use leave-one-out cross-validation (LOO-CV): for each of the 100 reasoning traces, we train the classifier on the other 99 and predict the held-out trace. This is repeated 100 times so every trace gets a prediction from a model that never saw it during training. We report accuracy (fraction of correct predictions), AUC-ROC (which measures how well the classifier ranks accurate forecasts above inaccurate ones, where 0.5 is random and 1.0 is perfect), and confusion matrices.

5 Experiments

5.1 Data

Our dataset consists of 100 resolved binary markets from Polymarket spanning five categories with balanced resolutions (50 YES, 50 NO). Table 1 summarizes the dataset.

Category	YES	NO	Total
Politics	10	10	20
Crypto	10	10	20
Sports	10	10	20
Science/Tech	10	10	20
Economics	10	10	20
Total	50	50	100

Table 1: Dataset summary. All markets resolved between January 2025 and February 2026 with $\geq \$10K$ trading volume. Each market has at least one pre-resolution news article.

5.2 Evaluation Method

Because this project has two distinct tasks, we use different metrics for each.

Forecasting is evaluated with the Brier score: $BS = \frac{1}{N} \sum_{i=1}^N (p_i - o_i)^2$, where p_i is the predicted probability and $o_i \in \{0, 1\}$ is the binary outcome. For example, predicting 0.8 on an event that resolves YES ($o = 1$) gives a Brier score of $(0.8 - 1.0)^2 = 0.04$, while predicting 0.5 gives $(0.5 - 1.0)^2 = 0.25$. A naive predictor that always outputs $p = 0.5$ achieves an overall Brier score of 0.25; perfect prediction achieves 0. Lower is better.

Reliability classification is evaluated with accuracy and AUC-ROC under LOO-CV. Accuracy measures the fraction of traces correctly classified as accurate or inaccurate. AUC-ROC measures

how well the classifier *ranks* forecasts: if we pick one truly accurate forecast and one truly inaccurate forecast at random, AUC-ROC is the probability the classifier assigns a higher confidence score to the accurate one. An AUC of 0.5 means the ranking is no better than random; 1.0 means perfect separation. AUC-ROC is useful here because our classes are imbalanced (74 accurate vs. 26 inaccurate), and unlike accuracy, it does not depend on the choice of classification threshold.

5.3 Experimental Details

All forecasts use gemini-2.0-flash-001 via Google Vertex AI with temperature 0.3 and max 2048 output tokens (except self-consistency, which uses temperature 1.0 for diverse reasoning paths). We used gemini-2.0-flash-001 as it has a knowledge cutoff of June 2024. News retrieval uses the Serper.dev Google News API with up to 5 articles per market. The classifier uses scikit-learn’s LogisticRegression with StandardScaler preprocessing.

5.4 Results

Strategy	RAG	CoT	Overall	vs. Naive	Crypto	Econ.	Politics	Sci/Tech
Naive ($p=0.5$)	—	—	0.250	—	0.250	0.250	0.250	0.250
Direct	No	No	0.297	+19%	0.307	0.231	0.341	0.268
CoT	No	Yes	0.287	+15%	0.353	0.193	0.302	0.319
Direct + RAG	Yes	No	0.171	−32%	0.276	0.112	0.063	0.241
CoT + RAG	Yes	Yes	0.176	−30%	0.194	0.159	0.116	0.249
SC + RAG	Yes	Yes	0.177	−29%	0.224	0.162	0.078	0.255

Table 2: Brier scores across forecasting strategies (lower is better). Bold indicates best per column. “vs. Naive” shows relative change from the 0.25 baseline (negative = improvement). Sports category omitted for space (see Appendix). All strategies evaluated on $n = 100$ markets.

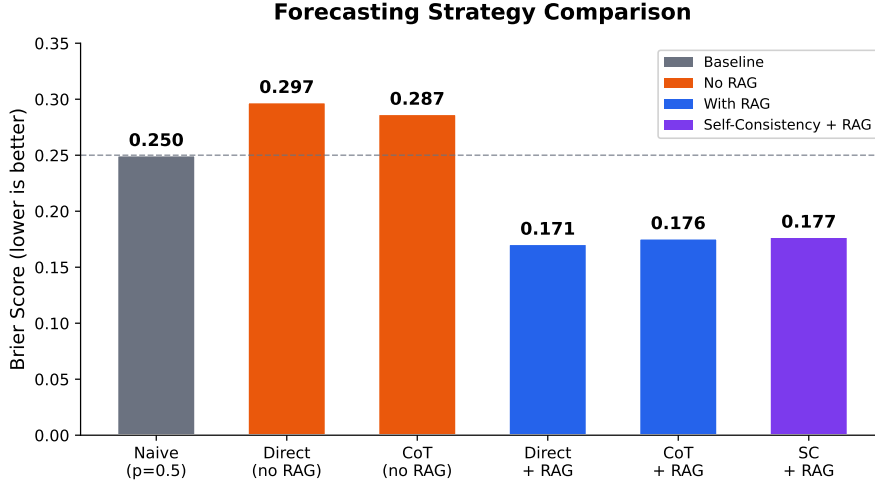


Figure 1: Brier scores across forecasting strategies. Orange bars (no RAG) perform worse than the naive baseline; blue/purple bars (with RAG) achieve ~ 0.17 . The gap between RAG and non-RAG dwarfs differences among RAG strategies.

Forecasting. The results in Table 2 and Figure 1 reveal three key patterns. First, **retrieval is the dominant factor**: all RAG strategies achieve Brier scores near 0.17, representing a 30–32% improvement over the naive baseline, while non-RAG strategies perform *worse* than random (Brier > 0.25). The gap between RAG and non-RAG strategies (~ 0.11 Brier points) dwarfs the gap between different RAG strategies (< 0.006 points).

Second, **chain-of-thought provides marginal benefit with RAG**. Without retrieval, CoT slightly improves over direct prompting (0.287 vs. 0.297), but with RAG, CoT+RAG (0.176) slightly *under-*

performs Direct+RAG (0.171). This suggests that when the model has access to relevant evidence, structured reasoning adds little, and may introduce errors through overthinking.

Third, **self-consistency does not work to improve over a single-pass CoT+RAG strategy** (0.177 vs. 0.176). Self-consistency works by generating multiple reasoning paths and aggregating their answers. In tasks like math, where there is one correct answer, the majority vote filters out wrong paths [5]. But in forecasting, each path produces a different probability, and averaging them can push the result toward the middle. If one path correctly predicts 0.9 and two others predict 0.5, the average (0.63) is worse than the single correct path. This may explain why sampling five paths at high temperature does not outperform a single pass at low temperature.

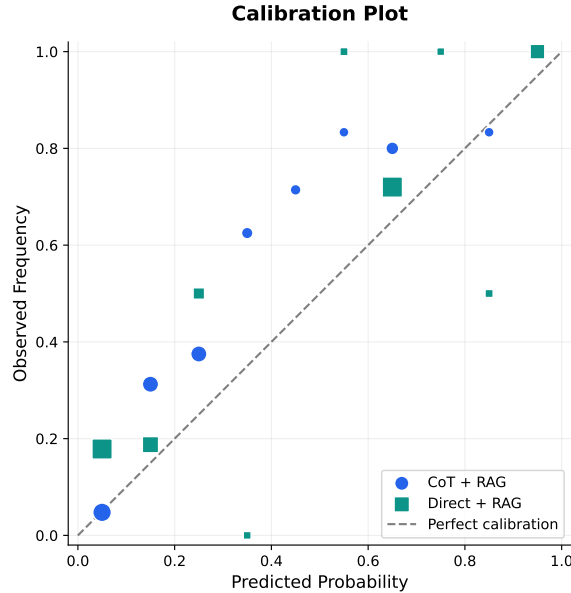


Figure 2: Calibration plot for CoT+RAG and Direct+RAG. Each point represents a bin of predictions: the x-axis is the average predicted probability in that bin, and the y-axis is the fraction of events in that bin that actually resolved YES. A perfectly calibrated model would place all points on the dashed diagonal. Points above the diagonal indicate underconfidence (events resolve YES more often than predicted); points below indicate overconfidence. Both strategies are generally underconfident in the 0.1–0.5 range, meaning the model hedges toward moderate probabilities when it should be more decisive. Point size reflects the number of predictions in each bin.

Category variation. Politics shows the largest RAG benefit ($0.25 \rightarrow 0.063$ for Direct+RAG), likely because political questions such as election outcomes and inaugurations generate clear, decisive news coverage that directly answers the question. Economics also benefits substantially ($0.25 \rightarrow 0.112$ for Direct+RAG), as most economics markets in our dataset concern Fed interest rate decisions, which are well-covered by financial reporting that includes explicit forward guidance. Science/technology is the hardest category across all strategies (Brier ≈ 0.25 even with RAG). Examining the worst predictions reveals the issue: questions like “Will Tesla launch a driverless Robotaxi service by October 31?” or “Will AI be TIME’s Person of the Year?” involve novel, speculative outcomes where even current news articles do not resolve the underlying uncertainty.

Reliability classification. The feature-based classifier achieves 66% accuracy and 0.60 AUC-ROC under LOO-CV (Table 3). At first glance, 66% accuracy appears worse than the majority-class baseline of 74%, which always predicts “accurate.” However, the majority baseline achieves its high accuracy by never identifying any inaccurate forecast (0% recall on the minority class). Our classifier, trained with balanced class weights, correctly identifies 15 out of 26 inaccurate forecasts (58% recall) while still correctly classifying 51 out of 74 accurate ones (69% recall). The AUC-ROC of 0.60, above the 0.50 random baseline, confirms that the classifier is learning a real signal: when it assigns a

Model	Accuracy	AUC-ROC	n	Accurate	Inaccurate
Feature-based LogReg	0.660	0.600	100	74	26
Embedding LogReg	0.540	0.476	100	74	26
Majority-class baseline	0.740	0.500	100	74	26

Table 3: Reliability classifier results (LOO-CV on CoT+RAG traces). The majority-class baseline always predicts “accurate.”

higher confidence score to one forecast over another, the higher-scored forecast is more likely to be genuinely accurate.

The embedding-based classifier performs near chance (54% accuracy, 0.48 AUC-ROC). This suggests that generic sentence embeddings, which are trained to capture semantic similarity between texts, do not pick up on the specific linguistic patterns (such as hedging or confidence language) that correlate with forecast reliability. Additionally, the logistic regression must learn 384 weights from only 100 samples, compared to just 19 weights for the feature-based classifier, making it much harder to fit a reliable decision boundary with so little data.

6 Analysis

Why does RAG dominate? Without retrieved context, the model can only draw on what it learned during training. But many of our markets concern events that either happened after the training cutoff or are too specific to appear in training data, such as whether a particular cryptocurrency token will launch or which candidate will win a local election. In these cases, the model essentially has to guess, which explains why the non-RAG strategies actually perform worse than the naive baseline of always predicting 0.5. When we provide even a few relevant news articles, the model can ground its predictions in actual reporting about the event, which is why all three RAG strategies perform similarly well regardless of the reasoning approach used.

Why doesn’t CoT help with RAG? When the model already has relevant news articles, it can often extract the answer directly from the context. Adding step-by-step reasoning on top of that may actually hurt, because the extra reasoning steps give the model opportunities to second-guess itself or over-weight minor details in the articles. This is consistent with findings that CoT can hurt performance on tasks where the answer is straightforward [4].

Feature importance. The top features in the logistic regression classifier reveal interpretable patterns (Figure 3, Table 4):

Feature	Coeff.	Direction
Question count	+1.14	→ accurate
Prob. extremity	+0.84	→ accurate
Numbered items	−0.61	→ inaccurate
Certainty count	+0.51	→ accurate
Numeric references	+0.50	→ accurate

Table 4: Top 5 features by absolute coefficient magnitude (logistic regression trained on all data). Positive coefficients indicate association with accurate forecasts.

The strongest predictor is question count (+1.14): traces that pose more rhetorical questions tend to accompany accurate forecasts, perhaps because the model is actively interrogating the evidence rather than passively summarizing it. Probability extremity (+0.84) is the second strongest feature, meaning that when Gemini assigns a confident probability (close to 0 or 1), that prediction tends to be correct. In other words, the model appears to “know what it knows.” The positive coefficient for certainty words (+0.51) reinforces this pattern: traces with more words like “definitely” or “clearly” are associated with accurate forecasts.

On the other side, more numbered items (−0.61) correlate with *inaccurate* forecasts. This may reflect the model producing more elaborate, structured enumeration when it is uncertain about a question,

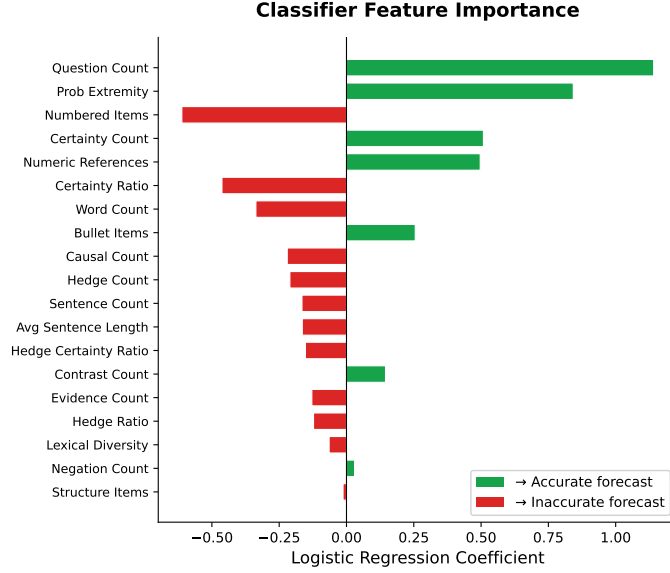


Figure 3: Logistic regression coefficients for the 19 reasoning-trace features. Green bars indicate features associated with accurate forecasts; red bars with inaccurate forecasts.

essentially compensating for lack of clarity with more organization. Word count (-0.34) shows a similar pattern: longer traces tend to be less accurate, consistent with the idea that the model writes more when it is less sure of the answer.

Qualitative error patterns. We examined the classifier’s mistakes to understand what it gets wrong. Among false positives (classifier predicted accurate, but the forecast was bad), a common pattern is the model assigning very low probability to events that actually happened. For example, it predicted only 0.10 for “Will Tesla launch a driverless Robotaxi service by October 31?” and 0.10 for “OpenAI browser by October 31?” (both resolved YES). The reasoning traces for these predictions read as confident and well-structured, which fooled the classifier, but the model was confidently wrong. These are inherently speculative markets where even well-informed humans would struggle to predict the outcome, so the model’s failure is not surprising. Several sports predictions follow the same pattern: the model assigned 0.20 to the Seattle Seahawks winning Super Bowl 2026 and 0.20 to Liverpool winning the Premier League (both resolved YES). In competitive sports, any individual team winning is unlikely by default, so the model’s low probabilities are reasonable priors that just happened to be wrong in these cases.

Among false negatives (classifier predicted inaccurate, but the forecast was actually good), many involve predictions that were nearly perfect but whose reasoning traces contained hedging language. For example, the model predicted 0.99 for “Will Nicușor Dan win the Romanian presidential election?” (correct), but the trace apparently contained enough uncertainty markers to trigger an inaccurate prediction from the classifier.

Limitations. There are several important limitations to acknowledge.

Dataset size. With only 100 markets, our results are based on a relatively small sample. The classifier in particular suffers from this: only 26 of the 100 traces are labeled as inaccurate, which makes it difficult to learn reliable patterns for the minority class. A larger dataset would allow for a proper train/test split instead of leave-one-out cross-validation, and would give more statistical confidence in the per-category comparisons (which currently have only 20 markets each).

No crowd-price baseline. We originally planned to compare our LLM forecasts against Polymarket’s own crowd prices, which would have been a strong external benchmark. However, the Polymarket API does not retain price history for markets that have already resolved, so we were unable to obtain these prices. This means we cannot say whether our system performs better or worse than the prediction market itself.

Single model. All forecasts use Gemini 2.0 Flash. Different LLMs may exhibit different strengths, biases, and reasoning styles, so our findings about CoT and RAG may not generalize to other models like GPT-4 or Claude.

Retrieval quality. Our retrieval approach is simple: we use the market question as a Google News search query and take the top 5 results. We do not verify whether retrieved articles are actually relevant or high-quality, and some articles may contain speculative commentary rather than factual reporting. A more sophisticated retrieval pipeline with relevance filtering could improve results.

Classifier scope. The reliability classifier only operates on CoT+RAG traces. We do not know whether the same linguistic features would predict accuracy for other strategies (e.g., Direct+RAG or self-consistency). Additionally, the Brier threshold of 0.25 for labeling forecasts as accurate or inaccurate is somewhat arbitrary. Choosing a stricter threshold (e.g., 0.15) would change the class balance and potentially the classifier’s behavior.

Category overlap in economics. Most of our economics markets concern Fed interest rate decisions, which are closely related questions with similar news coverage. This lack of diversity within the category may inflate the apparent performance of RAG on economics questions, since a single article about Fed policy could be relevant to multiple markets.

7 Conclusion

We investigated two linked questions about LLM forecasting on prediction markets: whether retrieval and reasoning strategies improve forecast quality, and whether reasoning traces can predict forecast reliability. Our 2×2 ablation across 100 balanced Polymarket questions shows that **retrieval is the critical ingredient**. RAG strategies achieve Brier scores around 0.17, a 30% improvement over the naive baseline, while chain-of-thought and self-consistency provide only marginal additional benefit. For meta-prediction, a feature-based logistic regression classifier over 19 hand-crafted linguistic features achieves 66% accuracy and 0.60 AUC-ROC, outperforming a sentence-embedding baseline. This tells us that reasoning traces do carry interpretable signals of reliability, even if those signals are modest in size. Probability extremity and certainty language turn out to be the strongest predictors, and the feature-based approach beats dense representations, suggesting that at small sample sizes, simple and interpretable features capture the relevant signal better than high-dimensional embeddings.

7.1 Practical Applications

These findings have direct implications for the growing prediction market ecosystem. As event contract exchanges like Polymarket and Kalshi scale (with over \$44 billion in trading volume in 2025 alone), there is increasing demand for tooling that helps traders and automated agents make better decisions. The infrastructure for prediction market trading is still in its early stages compared to equities or crypto, and the gap between exchange APIs and usable strategy development tools remains wide.

Our results suggest two concrete ways this research could inform such tooling. First, the dominance of retrieval in our ablation points to a clear design principle for forecasting agents: *context matters more than reasoning strategy*. Any system that generates LLM-based predictions on event contracts should prioritize feeding relevant, timely information into the prompt rather than investing in elaborate reasoning chains. For backtesting engines that replay historical markets against trading strategies, this means that simulating the information environment (what news and context was available at each point in time) is just as important as simulating order books and fills. A backtest is only as good as the context it provides, and our work shows that the gap between informed and uninformed LLM predictions is substantial (Brier scores of 0.17 vs. 0.29).

Second, the reliability classifier opens a path toward confidence-aware trading. Even at 66% accuracy, a meta-classifier that flags unreliable forecasts before they are acted on could help filter LLM-generated signals. In a trading pipeline, this means an agent could choose to abstain or reduce position size when its own reasoning trace exhibits linguistic markers associated with inaccuracy (high hedge word counts, moderate rather than extreme probabilities, lack of concrete evidence citations). As prediction market strategies move from hand-built bots toward LLM-powered agents, this kind of self-assessment becomes increasingly valuable.

7.2 Future Work

Several directions would strengthen these results. Scaling the dataset beyond 100 markets would enable proper train/test splits and support more expressive classifiers such as fine-tuned BERT models. Comparing LLM forecasts against Polymarket crowd prices would provide an important external benchmark, testing whether LLMs can match or complement human collective intelligence. Extending the analysis to multiple LLMs (GPT-4, Claude, Llama) would test whether our reliability signals generalize across model families or are specific to Gemini’s reasoning patterns. Finally, integrating point-in-time news retrieval into backtesting frameworks would allow researchers to evaluate forecasting strategies under realistic historical conditions, where the model only sees information that was actually available at prediction time.

References

- [1] Kenneth J. Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D. Nelson, George R. Neumann, Marco Ottaviani, Thomas C. Schelling, Robert J. Shiller, Vernon L. Smith, Erik Snowberg, Cass R. Sunstein, Paul C. Tetlock, Philip E. Tetlock, Hal R. Varian, Justin Wolfers, and Eric Zitzewitz. The promise of prediction markets. *Science*, 320(5878):877–878, 2008.
- [2] Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, and Philip Tetlock. Wisdom of the silicon crowd: LLM ensemble prediction capabilities rival human crowd accuracy. *Science Advances*, 2024.
- [3] Danny Halawi, Fred Zhang, Yueh-Han Chen, and Jacob Steinhardt. Approaching human-level forecasting with language models. *arXiv:2402.18563*, 2024.
- [4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.
- [5] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*, 2022.
- [6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.
- [7] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
- [8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*, 2019.
- [9] Google DeepMind. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2024.

A Prompt Templates

System prompt (all strategies). “You are a skilled forecaster. Given a question about a future event [and relevant news context], [reason carefully step by step before estimating / estimate] the probability that the event will resolve YES. Be calibrated: a 70% prediction should resolve YES about 70% of the time. Do NOT default to extreme probabilities; use the full range from 0.05 to 0.95.”

Direct prompt (no RAG). “Question: {question} \n\n Based on your knowledge, what is the probability this event resolves YES? \n Respond with ONLY a decimal number between 0.0 and 1.0.”

CoT prompt (with RAG). “*Question: {question} \n\n Relevant context: {context} \n\n Think through this step by step: \n 1. What are the key factors that would make this resolve YES? \n 2. What are the key factors that would make this resolve NO? \n 3. What does the available evidence suggest? \n 4. Are there any important uncertainties or conflicting signals? \n\n After your analysis, state your final probability on a new line in exactly this format: \n PROBABILITY: <number between 0.0 and 1.0>*”

B Full Per-Category Results

Strategy	Crypto	Econ.	Politics	Sci/Tech	Sports
Naive ($p=0.5$)	0.250	0.250	0.250	0.250	0.250
Direct (no RAG)	0.307	0.231	0.341	0.268	0.340
CoT (no RAG)	0.353	0.193	0.302	0.319	0.268
Direct + RAG	0.276	0.112	0.063	0.241	0.162
CoT + RAG	0.194	0.159	0.116	0.249	0.161
SC + RAG	0.224	0.162	0.078	0.255	0.167

Table 5: Full per-category Brier scores (20 markets per category, lower is better).