FUTURESIM: Replaying World Events to Evaluate Adaptive Agents

Shashwat Goel [*1,2] **Nikhil Chandak** [*2] **Arvindh Arun** [*3] **Ameya Prabhu** [4,5] **Steffen Staab** [3,6] **Moritz Hardt** [2,4] **Maksym Andriushchenko** [1,2] **Jonas Geiping** [1,2,4]

Abstract

AI agents are being increasingly deployed in dynamic, open-ended environments that require adapting to new information as it arrives. To efficiently measure this capability for realistic use-cases, we propose building grounded simulations that replay real-world events in the order they occurred. We build `FutureSim`, where agents forecast world events beyond their knowledge cutoff while interacting with a chronological replay of the world: real news articles arriving and questions resolving over the simulated period. We evaluate frontier agents in their native harness, testing their ability to predict world events over a three-month period from January to March 2026. `FutureSim` reveals a clear separation in their capabilities, with the best agent's accuracy being 25%, and many having worse calibration than a constant predictor. Through careful ablations, we show how `FutureSim` offers a realistic setting to study emerging research directions like long-horizon test-time adaptation, search, memory and reasoning about uncertainty. Overall, we hope our benchmark design paves the way to measure AI progress on long-horizon open-ended adaptation spanning multiple months in the real world.

1. Introduction

Language model agents have achieved resounding success across diverse benchmarks. Yet, they are held back by their inability to adapt over a long-horizon in dynamic environments (ARC Prize Foundation, 2026; Silver & Sutton, 2025). While recent game and simulation benchmarks like ARC-AGI 3 and VendingBench (Backlund et al., 2025) provide informative proxies, adaptation should be measured in environments that align with how our world evolves (Farquhar & Gal, 2019). To fill this gap, we ask:

> *Can we evaluate AI agent adaptation in settings that require general reasoning, grounded in how our world evolves?*

We propose building simulations that replay real-world events in the temporal order they occurred to evaluate adaptive agents. We introduce `FutureSim`, an interactive environment where agents have to predict world events beyond the underlying language model's knowledge cutoff. Each day, agents receive new information from news articles and ground-truth feedback on earlier predictions, based on which they can learn to update future predictions. This makes forecasting suitable for studying adaptation in two key ways. First, it assesses whether agents' priors align with how the world evolves, and whether they appropriately update them in light of new information. Indeed, a decade of research on human expert forecasters has found a strong relationship between a forecaster's adaptivity and accuracy (Mellers et al., 2015; Atanasov et al., 2020). Second, more practically, limiting to a predictive setting where agents' actions cannot change the underlying environment sidesteps the challenge of realistically simulating counterfactual worlds (Bruce et al., 2024).

For our experiments, we show such simulations can be bootstrapped from just timestamped source documents. To achieve this, we use the methodology proposed in (Chandak et al., 2026) to create prediction questions from news articles. In Figure 1, we benchmark agents in a 90-day period from January to March 2026 on 330 forecasts. We find that `FutureSim` cleanly discriminates the capabilities of both, closed and open-weight frontier models, in their recommended agent harnesses. All models utilize new information to improve their predictions over the course of the simulation. The best performing agent, GPT 5.5 in Codex, consumes 3700 turns and 12.4M tokens spanning multiple sequential context window compactions in a single

[*]Equal contribution [1]ELLIS Institute Tübingen [2]Max Planck Institute for Intelligent Systems [3]Institute for AI, University of Stuttgart [4]Tübingen AI Center [5]University of Tübingen [6]University of Southampton. Correspondence to: Shashwat Goel <shashwat-now@gmail.com>.

Proceedings of the ICML 2026 Workshop on RLxF: Reinforcement Learning from World Feedback, Seoul, South Korea. 2026.
Copyright 2026 by the author(s).

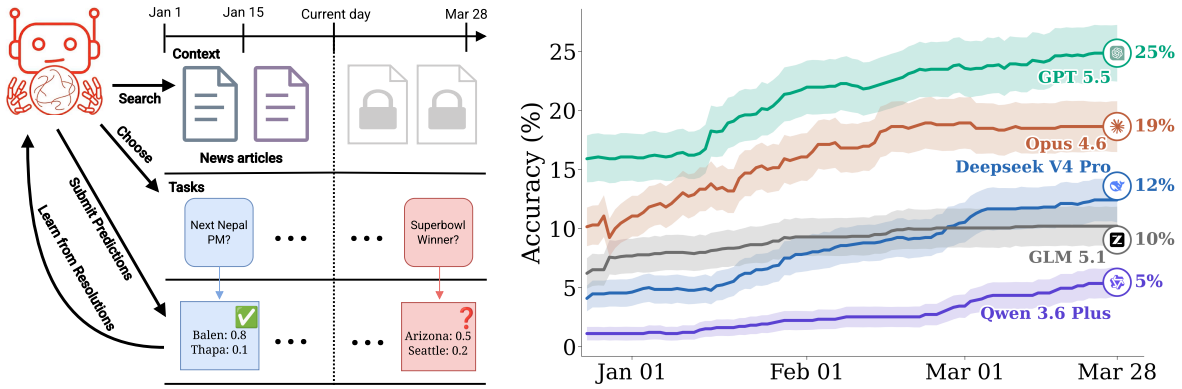

Figure 1. In FutureSim, agents are evaluated in an interactive forecasting environment. They can search news up to the current simulation date, and choose when to update their predictions. We evaluate all models in their recommended harness at maximum reasoning effort, finding GPT 5.5 performs the best, while DeepSeek V4 Pro overtakes GLM 5.1 through better test time adaptation.

run showing the long-horizon nature of FutureSim.

Overall, our main contributions are as follows:

- To realistically evaluate adaptation in AI agents on the economically valuable task of forecasting, we build FutureSim, which replays how the world evolved after the agents' knowledge cutoff. To prevent leakage of future information, we carefully sandbox agents, while providing them access to reliably dated offline snapshots of news articles.
- FutureSim is open-ended: agents choose which questions to make forecasts on, and when. The questions are free-form, and agents have to come up with multiple possible outcomes with an incentive to report calibrated probability distributions and make timely updates.
- The core implementation of FutureSim is flexible, allowing users to benchmark any combination of models, harnesses, and chronological event domains. We show significant room for improvement in test time adaptation for frontier models, even in their native agent harnesses.
- We show how FutureSim can also support research on emerging capabilities like reasoning to search under uncertainty, memory, harness design, and multi-agent dynamics by designing ablations that isolate the effect of each of these capabilities.

2. Related Work

Long-horizon environments. Recent benchmarks study long-horizon agent interaction in games and simulations. For example, BALROG (Paglieri et al., 2025) uses games to evaluate open-ended multimodal agents, while ARC-AGI 3 (ARC Prize Foundation, 2026) uses interactive games for testing learning on the fly. GAIA-2 (Froger et al., 2026) evaluates agents in human-authored mobile-use scenarios, while Vending-Bench (Backlund et al., 2025) and YC-Bench (He et al., 2026) study agents in economic simulations. The dynamics in these environments are either human-designed or simulated by a model, making them imperfect proxies for societal evolution (Seshadri et al., 2026). Live deployments such as Anthropic's Project Vend (Anthropic, 2025) provide the most realism, but are slow, expensive, can raise safety concerns, and are not exactly reproducible across models. To circumvent such issues, we propose drawing environment dynamics from real-world event timestamps. This is related to benchmarks that replay the evolution of real artifacts, such as SWE-Evo (Thai et al., 2026) for coding agents, but we show this can be generalized across domains by creating prediction tasks from timestamped source documents (Chandak et al., 2026).

Judgemental Forecasting benchmarks. Language model forecasting has recently received substantial attention in the domain of *judgemental forecasting*, i.e., predicting discrete real-world events, unlike *statistical forecasting* of time-series data (Webby & O'Connor, 1996). Early evaluations (Halawi et al., 2024; Karger et al., 2025) tested models on static forecasting questions with retrieved evidence. More recently, benchmarks such as ProphetArena (Yang et al., 2026) and PredictionArena (Zhang et al., 2026b) evaluate agents through live trading on prediction markets like Kalshi and Polymarket. FutureSim can subsume such evaluations by allowing benchmarking on any data, including theirs, while having a crucial advantage: live-market evaluations are difficult to reproduce and ablate because market conditions change over time, and reliable signals require long evaluation windows. FutureSim instead provides a replayable environment that allows controlled ablations of search, memory, harness design,

Table 1. **Comparison to related benchmarks.** `FutureSim` is the only benchmark that tests models on open-ended adaptive reasoning about general, real-world events over a long horizon while being efficiently reproducible. We mark benchmarks where the agent has to decide what to do, rather than being given a single, fully defined task, as open-ended. Benchmarks that can benefit from sequentially learning across tasks are marked as testing adaptation. *We estimate horizon lengths as the maximum actions used in a single model trajectory reported in the original papers, with N/R = not reported.

Benchmark	General world reasoning	Real event data	Reproducible	Tests adaptation	Open-ended	Horizon Length*
GAIA-2 (Froger et al., 2026)	✗	✗	✓	✗	✗	25
τ^3-Bench (Shi et al., 2026)	✗	✗	✓	✗	✗	33
ARC-AGI-3 (ARC Prize Foundation, 2026)	✗	✗	✓	✓	✓	7,800
BALROG (Paglieri et al., 2025)	✗	✗	✓	✓	✓	100,000
SWE-Evo (Thai et al., 2026)	✗	✓	✓	✓	✗	N/R
Vending-Bench-2 (Backlund et al., 2025)	✗	✗	✓	✓	✓	6,000
KellyBench (Grady et al., 2026)	✗	✓	✓	✓	✓	1,000
ForecastBench (Karger et al., 2025)	✓	✓	✓	✗	✗	1
ProphetArena (Yang et al., 2026)	✓	✓	✗	✗	✗	1
PredictionArena (Zhang et al., 2026b)	✓	✓	✗	✓	✓	N/R
`FutureSim` (Ours)	✓	✓	✓	✓	✓	4,000

and adaptation. Parallel work KellyBench (Grady et al., 2026) studies long-horizon betting on Premier League matches through statistical model building. In contrast, `FutureSim` evaluates agents' ability to forecast diverse, general world events from evolving evidence.

3. The `FutureSim` Environment

`FutureSim` is a chronological environment, and in our work, we set a daily cadence, with time-steps corresponding to real-world dates. We now describe the mechanics of `FutureSim`.

3.1. Environment Design

The task state and context state of the environment are updated at each time step as follows.

Tasks.

Sample Forecasting Question

Question. Who will be sworn in as Nepal's new prime minister?

Resolution Date: March 6, 2026
Answer Type. String (Name)
Ground-Truth Answer. Balendra Shah
Source. Al Jazeera

The current state of tasks is maintained as a CSV file, with each row containing data about one forecasting question. This includes the question's background information, resolution criteria, resolution date, and the agent's most recent forecast for the question. Unlike prior forecasting evaluations that expect a single predicted outcome per question, we allow agents to submit a probability distribution over multiple possible outcomes that it has to come up with itself. Given the inherent uncertainty in forecasting, this allows a more complete measurement of the agent's beliefs. Once the simulation date passes a question's resolution date, the ground-truth outcome is added to the state file. We provide an example question above.

Context. At each time-step, the context consists of documents that became available by then. We use Common Crawl News (Nagel, 2016) as it provides reliably dated snapshots of news articles which are not affected by future updates (Chandak et al., 2026).

Agent Interaction. The environment itself is minimal, allowing the definition of arbitrary *agents*. By an agent, we mean the tools, prompts, and orchestration that models can use to interact with the environment, which we describe later. The environment itself provides only two actions:

1. `submit_forecast(question_id, outcomes)`: register or update the probability distribution over predicted outcomes for one active question.

2. `next_day()`: ends the current time-step and advances the simulation, running any necessary evaluations. The task and context state are updated for the next time-step.

Evaluation. Let $\mathcal{Q}$ denote the set of forecasting questions, with each $q \in \mathcal{Q}$ having a ground-truth answer y_q. Formally, an agent's forecast is a set of outcomes $\Omega_q = \{o_1, \ldots, o_k\}$ with probabilities $p_q(o_i) \geq 0$ satisfying $\sum_{i=1}^{k} p_q(o_i) \leq 1$. We use language model-based answer matching (Chandak et al., 2025) to check if an outcome o matches the ground-truth y_q. We report the following metrics:

Brier Skill Score (BSS). For our free-form questions with no predefined set of choices, we need to adapt the multi-

category Brier score (Mucsányi et al., 2023) as follows,

$$\mathrm{BSS}(q) = 1 - \sum_{o \in \Omega_q \cup \{y_q\}} \left(p_q(o) - \mathbf{1}[o = y_q]\right)^2,$$

where $p_q(o) = 0$ for $o \notin \Omega_q$. In Appendix C.1, we prove that this is a proper scoring rule. Higher score is better: 1 is for a fully confident correct answer, 0 for abstaining by reporting zero probability, and -1 is for assigning all probability to wrong guesses.

Accuracy. We measure the accuracy of the outcome assigned the most probability (top-1 accuracy)

$$\mathrm{Acc} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}\left[\arg\max_o p_q(o) = y_q\right]$$

While the Brier skill score incorporates both the correctness of predicted outcomes and the calibration of the probabilities distributed over them, the accuracy is agnostic of the probability distribution and only depends on the correctness of the top outcome predicted for each question. In our results, we report the mean across all questions $\mathcal{Q}$, reporting the projected score based on the current prediction for unresolved questions, and scoring 0 for questions where an agent has registered no prediction.

4. Benchmarking Frontier Agents in `FutureSim`

We now describe the current `FutureSim` benchmark setup and results on frontier agents.

4.1. Experimental setup

Forecasting Questions. Forecasting evaluations require fresh questions about events that occur after the knowledge cutoffs of models being tested. We follow the scalable automated methodology proposed by Chandak et al. (2026) to curate short-answer forecasting questions without predefined choices, from news articles. We make multiple refinements to this pipeline as described in Appendix A, such as using stronger models, filtering out easy questions, and improving the reliability of resolution dates. We use Al Jazeera articles as source documents, as we found this to be the highest quality source with a large number of articles freely accessible via CCNews. Finally, we obtain 330 forecasting questions resolving between January 1st and March 28th 2026, which is after the knowledge cutoffs of the models we evaluate. Each question is active from the start of the simulation (2025-12-24) until its resolution date. Agent predictions to each question are evaluated by DeepSeek v3.2 as the answer matcher, with prompts in Appendix F.5. We limit the outcome set size predicted by agents to ≤ 5 per question for answer matching efficiency.

Search Corpus. The search corpus is a deduplicated snapshot of CCNews containing 7.36M articles from 141 distinct news sources between January 2023 and March 2026. Agents have access to articles published on or before the current simulation date. About 7.12M articles are available on day 0, and 244K new articles enter the corpus over the 88-day simulation window.

We provide the topic distribution of questions in Table 2, noting it is more representative of important events than prediction markets. That said, the prediction questions and search corpus can be changed easily in `FutureSim`, and we will keep updating it over time to evaluate newer models.

Tools available to agents. All agents are run with full access to the harness's built-in shell and file tools (eg: `Bash`, `Read`, `Write`, `Grep`). These commands can be used to access the raw articles, which are organized into folders by date for ease of browsing, and create, edit or execute files in the agents own workspace. To prevent leakage of future information, we do not allow even read access beyond the agent's workspace and accessible article folders up to the current simulation date and disable web search tools and commands like `curl`, with sandbox details in Appendix B.1.2. Finally, we provide access to a hybrid semantic + keyword search tool over the news corpus, implemented using LanceDB, that returns 5 chunks of 512 tokens. We use Qwen3 8B embeddings for the semantic search. The search tool allows defining the period between which to fetch chunks, in the format `search_news(query, from_date, to_date)`.

Agents tested. Unless specified, we evaluate at maximum reasoning effort, in the recommended harness for open-weight models, and native code harness for closed ones: We use Codex for GPT 5.5; OpenCode for Qwen3.6 Plus; and Claude Code for Opus 4.6, DeepSeek V4 Pro, and GLM 5.1.

Our Baseline Harness. While it is interesting to study agent performance without any task-specific hand-holding (ARC Prize Foundation, 2026), better orchestration of tools and prompts can help elicit agent potential on any given task (Wang et al., 2025). Thus, we also build a custom harness as an improved baseline for future work on our benchmark. For developing our harness, we iterated on DeepSeek V3.2 performance on OpenForesight (May to August 2025) (Chandak et al., 2026) as a validation setting. Our final baseline harness incorporates features like context consumption feedback, a forced memory update phase with structured memory tools and per-question memory, and procedural forecasting guidelines. We provide details in Appendix B.1. We also load the task state as a dataframe, allowing agents to manipulate and process it with restricted python query execution.

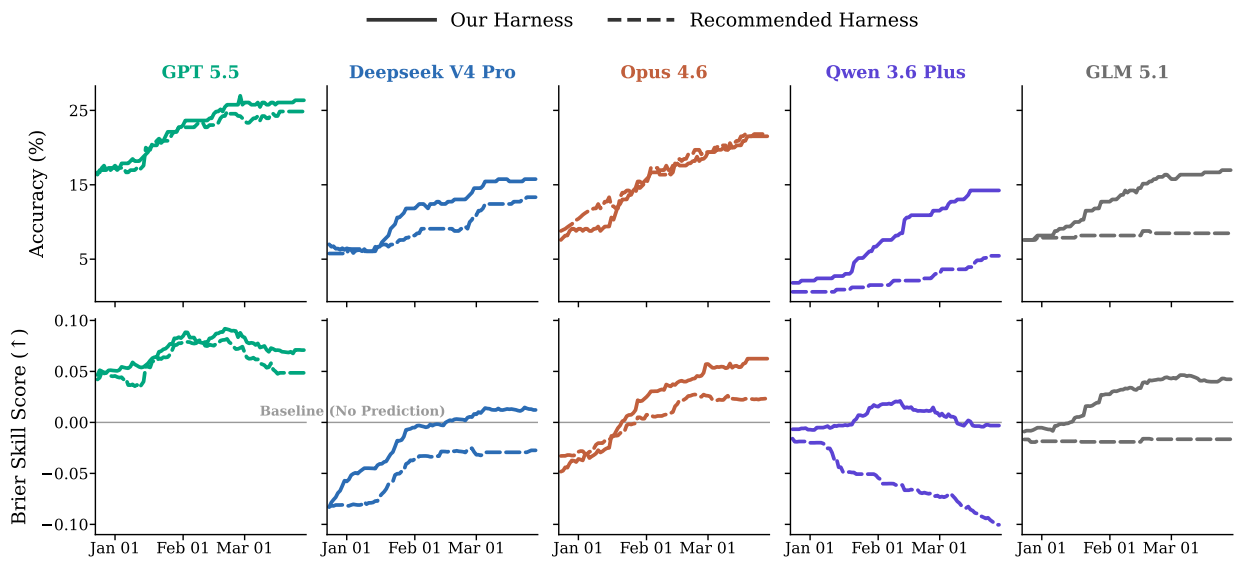

Figure 2. **Agent performance on `FutureSim`.** GPT 5.5 by far performs the best in both accuracy and Brier skill score. (Top) Except for GPT 5.5, all models start at a negative Brier skill score, and while they fail to improve significantly in their recommended harness, they cross over to a positive Brier skill score in our harness. (Bottom) Even for accuracy, models perform significantly better in our harness than their recommended harness, improving consistently over the course of the simulation.

4.2. Results

Figure 2 shows the Brier skill score and accuracy over the course of the simulation for different models, both in their default-recommended harness and in ours. We find that GPT 5.5 outperforms all other models by a large margin, in both accuracy and Brier skill score. It starts with the best initial predictions on day 0, and continues improving its predictions throughout. DeepSeek V4 Pro also shows significant improvements over the course of the simulation in both harnesses, whereas Qwen3.6 Plus and GLM 5.1 improve only in our harness. In contrast, Qwen3.6 Plus actually deteriorates in calibration in its recommended OpenCode harness. This shows the importance of good harnesses in eliciting test-time adaptation capabilities from some models. Finally, even Claude Opus 4.6 performs significantly better on our harness than its native and widely used Claude Code harness, showing that coding harnesses may not be as generally optimal as one might think.

These results demonstrate the importance of harness engineering, for which `FutureSim` provides a realistic long-horizon test-bed. This includes emerging context management methodologies like recursive language models (Zhang et al., 2026a), and also autoresearch approaches that enable agents to self-improve their own harness (Zhang et al., 2026c; Lou et al., 2026).

4.3. Analyzing Agent Behaviours

We now analyze the trajectories and reasoning traces of agents and report some interesting strategies and failure modes we observe. GPT 5.5 has the best Top-1 accuracy, but we find among its incorrect final predictions, 27.4% assign at least 0.5 probability to the wrong top answer, and 9.1% assign at least 0.75, indicating significant overconfidence. DeepSeek V4 pro updates its prediction on almost every question during the simulation, which helps it improve substantially from its initial weak predictions. However, it often places placeholder predictions like *no new appointment*. GLM 5.1 is relatively conservative and has the smallest overconfidence rate among wrong answers (only 3.7% above 0.5), but performs poorly as it makes the lowest number of updates among the models tested. Qwen3.6 Plus abstains from more questions than any other model, only registering predictions on 36.7% questions, while all other models submit prediction on all the questions. Note that even when we subset on the questions on which all agents update their predictions at least once (Appendix D.5), the agent rankings are still consistent with what we show in Figure 1. Across models, we observe instances of "self-conditioning" (Sinha et al., 2026), where models start treating prior rationales and lessons they wrote in memory as hard-truths, leading to subsequent overconfident mistakes.

Comparing Agents to Human Aggregate Forecasts. While we use a generalizable recipe to create forecasting questions from news reporting, some questions also have corresponding prediction markets on Polymarket. In Figure 3 (left), we compare model predictions to the human aggregate on these questions at corresponding times. We find that GPT 5.5's prediction updates are not only aligned with movements in the human aggregate for multiple mar-

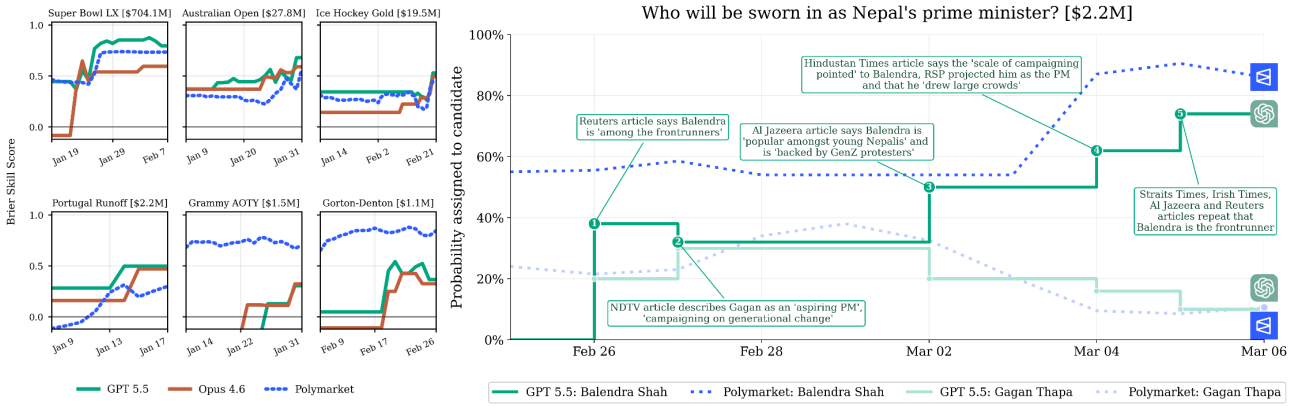

Figure 3. **Comparisons of frontier agent prediction to human crowd aggregates on Polymarket.** (Left) We find that GPT 5.5 leads the real market aggregate for some questions, including the Super Bowl market, which traded 700M in total volume. That said, it performs relatively poorly on some other markets, with Claude Opus 4.6 closely tracking GPT 5.5 predictions but usually slightly worse. (Right) We zoom in on the market about the Nepal PM election, annotating rationales from GPT 5.5's prediction update trajectory. Notably, GPT 5.5 cites reasonable evidence, and its updates on candidate probabilities are aligned with, albeit lagging the human aggregate.

kets, sometimes they even lead the market (e.g. predicting the Super Bowl and Portugal Runoff winner), which could be potentially used to make large profits given the millions in volume traded on these markets. At the same time, the model's predictions are also significantly worse on the Grammy and UK constituency election (Gorton and Denton) markets. Indeed, it is perhaps natural to expect prediction market aggregates to be harder to beat for events that aggregate human opinion.

In Figure 3 (right), we visualize GPT 5.5's update trajectory on the Nepal Prime Minister market, showing its cited evidence for each update is appropriate. Of course, given the millions in volume traded for these questions on Polymarket, the human aggregate updates are smoother than a single agent run, but GPT 5.5's updates are closely aligned with their movements, and only a bit lagging. This could be enhanced by indexing the search corpus across fresher sources, such as social media. Overall, this analysis shows that our simulation realistically tracks economically valuable capabilities.

5. Capabilities Tested

Performing well on `FutureSim` requires agents to be capable on various fronts: agents should filter the right questions to focus on at each time-step based on the new context available, creatively search for relevant evidence, remember useful information over the course of the simulation, and learn from environment feedback as past predictions resolve. In this section, we show how `FutureSim` ablations can isolate the effect of these capabilities to enable research on these emerging directions.

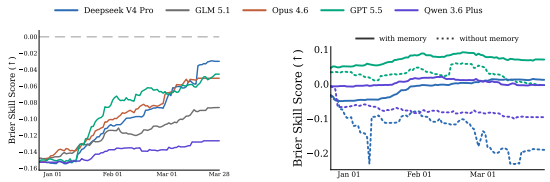

Figure 4. **(Left) Comparing test-time adaptation across agents.** We start different models in our improved harness at the lowest performing agent's (Qwen 3.6 Plus) initial prediction set, to maximize scope for improvement over time. We find agents fail to update enough to even reach the constant predictor performance of 0 brier skill score despite their own capabilities being stronger for some of them. This shows agents get anchored to the initial prediction set and fail to adapt sufficiently. **(Right) Benefits from memory.** By ablating the ability to write and fetch memories at test time, we find models clearly benefit from inference-time memory.

5.1. Test-time adaptation

Each day, as the news corpus grows in the simulation and the available context repository expands, agents should update their forecasts based on the new information. Moreover, as some forecasts are resolved on the date the ground-truth becomes known, agents can reflect on their predictions to learn forecasting meta-skills, akin to on-the-job learning. This leads to two natural research questions: 1) How do different agents compare in their ability to perform test-time adaptation? and, 2) How can we know for a single agent how much room there is to improve at test time adaptation?

Comparing test-time adaptation across agents. In the standard setup of `FutureSim` presented until now, each agent has a different starting performance, making it hard to compare test-time adaptation ability across agents. To isolate and compare adaptation, in the left panel of Figure 4, we fix the initial forecasts to the worst agent's (Qwen3.6 Plus)

and evaluate all agents in our harness, which maximizes the scope for improvement at test-time. We observe that GPT 5.5, Claude Opus 4.6, and DeepSeek V4 Pro all show similar levels of improvement, while GLM 5.1 does worse relative to them, and finally Qwen3.6 Plus barely improves its Brier skill score over time. At the end of the simulation, all agents fail to reach even the baseline Brier skill score (0) of not predicting at all, let alone their original positive Brier skill score performance. This is despite being informed that their predictions are obtaining a negative Brier skill score as they resolve, showing how frontier agents fail to adapt away from bad initial anchors.

Can agents match full knowledge performance with sequential arrival of information? During our simulation, each day new news arrives sequentially, and agents have to incorporate it by updating forecasts up to one day before the resolution of each question. One way to test how efficiently agents update their predictions is by checking whether they can match their own performance when directly asked to predict each question one day before its resolution date, when they have maximal information for a forecast and no prior prediction to anchor to. We perform this experiment for GPT 5.5 xhigh in Codex, where comparing the left bars of both subplots in the left panel of Figure 5, we find that directly searching with full information available (green) leads to much higher accuracy (31.2% vs 24.8%) than sequentially updating predictions as information becomes available in the simulation (blue). This showcases a clear inefficiency in the agent's ability to adapt at test time and also shows a floor on the maximum accuracy achievable.

We hope these results spark work on test-time adaptation in models using FutureSim, including continued finetuning (Yang et al., 2025) to internalize the large amount of new news arriving in the simulation, and test-time training (Hardt & Sun, 2024) on forecast resolution signals.

5.2. Memory

Given the limits of the context window size in transformer-based language models, agents need alternate ways to implement long-term memory. The current approach to frontier agent deployments uses a file-based memory with structured access, which we implement in our harness with carefully designed, minimal yet informative guidelines. We ablate agents having write access to any memories in the right panel of Figure 4 and find that all three models tested perform worse without memory. This shows the importance of memory for our task, as observed from qualitative traces: agents store and use post-resolution feedback, information found via search over the context, and summaries of past rationale. Memory also protects against drift: when searches returned weak or stale evidence, agents with memory often retained a calibrated prior rather than re-reasoning from scratch and over-updating toward a plausible but unsupported alternative.

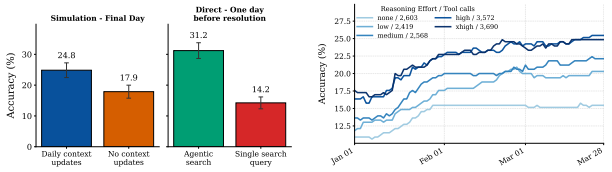

Figure 5. **Capability ablations for search and inference scaling.** (Left) Benefits from search. We evaluate GPT 5.5 xhigh reasoning effort in four different settings to isolate the benefits of agentic search over updating context in FutureSim. Measuring accuracy at the last day of the simulation, we find large improvements from daily context updates (blue) compared to when no articles beyond the first date are added during the simulation (orange). That said, directly evaluating each question one day before its resolution (green) leads to the best performance of 31.2%, which is a floor on the accuracy attainable in our environment that no simulation run achieves. However, when agentic search is replaced by a single retrieval per question (red), accuracy halves, highlighting the importance of reasoning sequentially about what information to seek in forecasting. (Right) Benefits from scaling test-time compute. We run GPT 5.5 in all five available reasoning efforts, finding more inference compute leads to better accuracy on FutureSim.

5.3. Search

Unlike many existing search benchmarks (Yang et al., 2018; Wei et al., 2025), in FutureSim, the search is not for past facts knowable perfectly from the accessible documents. Rather, (i) the document corpus evolves, adding more context each day, and (ii) agents have to creatively reason to come up with *what to search for*, based on the scattered evidence across documents seen until then. To measure the effect of these properties on FutureSim performance, we ablate full agentic search over the evolving corpus in the simulation in two key ways, with results in the left panel of Figure 5. First, we ablate the daily arrival of news articles during the simulation, finding that this leads to significantly worse accuracy on the last day (24.8% in blue with daily updates vs 17.9% in orange without context updates). This shows the importance of continued search for fresh evidence as the corpus evolves in the environment. Second, we compare full agentic search (green) to retrieving only articles with a single semantic search query using the question title (red), where for each question, we perform the search 1 day before it resolves. We find full agentic search leads to double the accuracy of a single query, demonstrating the importance of sequential information-seeking for forecasting (Brier skill score in Figure 9).

These results show FutureSim can support research on reasoning-intensive sequential search agents (Jin et al., 2025), as well as better underlying search tools (Khattab & Zaharia, 2020; Shao et al., 2025) for the dynamic, and uniquely bayesian search setting of forecasting (Murphy, 2026).

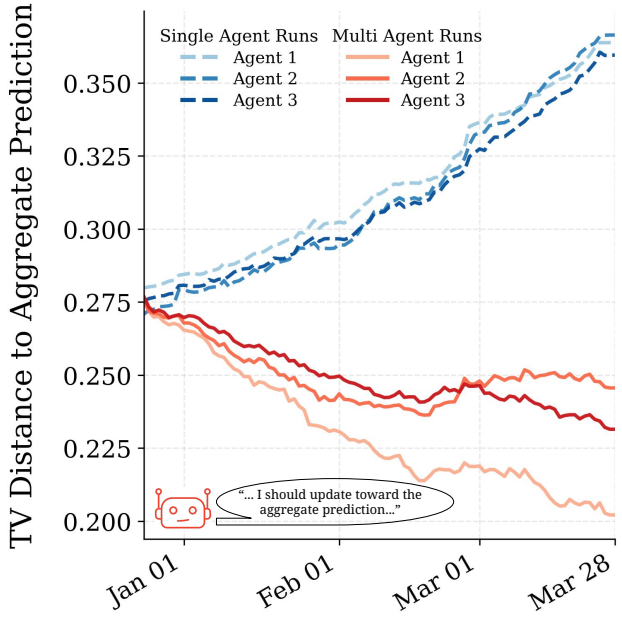

Figure 6. **Multi-agent dynamics.** When we run multiple copies of DeepSeek V3.2 agents simultaneously, we see agent predictions start moving toward the aggregate, unlike independent single agent runs where predictions diverge over time.

5.4. Inference Scaling

For economically valuable tasks like forecasting world events, scaling test-time compute can provide beneficial gains in performance worth the extra cost (Snell et al., 2025). To demonstrate this, we compare GPT 5.5 at five different reasoning efforts in the right panel of Figure 5, observing consistent improvements in accuracy at higher reasoning budgets (similar trends hold for brier score in Figure 10). We note that GPT 5.5 at high effort uses much more tool calls and obtains significantly higher accuracy than lower efforts, though xhigh does not lead to further improvements over high.

We are excited to support research on the performance vs cost scaling behaviour of various test-time compute paradigms like parallel aggregation (Venkatraman et al., 2026), multi-agent systems (Tran & Kiela, 2026), and scaling environment interactions (Shen et al., 2025) on `FutureSim`.

5.5. Multi agent dynamics

As we move towards a world where agents owned by different users interact and compete (Bansal et al., 2018), it is interesting to study how agents adapt in the presence of other agents. To demonstrate how `FutureSim` can support multi-agent experiments, we demonstrate a simple example, where three identical DeepSeek v3.2 agents compete simultaneously. The agents only depend on each other through an information bottleneck, which is the current aggregate prediction for each question, similar to crowd aggregates on prediction markets. In Figure 6 we observe that over the course of multi-agent simulations, agents converge towards similar predictions, while in independent single agent runs their predictions diverge over time. This is despite prompting agents that they will be graded on the *peer score*, which incentivizes informative predictions beyond the crowd aggregate. We provide details on aggregate prediction, peer score and TV distance computation in Appendix C.3. The absolute agent performance itself is similar in single-agent and multi-agent runs as shown in Figure 11.

By providing a simulation grounded in real-world dynamics, we hope `FutureSim` supports multi-agent research, such as improvements from self-play (Silver et al., 2018), and studies on how diverse agents evolve (Park et al., 2023) when equipped with richer communication channels (Brown et al., 2022).

6. Conclusion

In this work, we propose evaluating the ability of frontier agentic systems to adapt in dynamic environments by building simulations that replay real-world events. We build one such simulation, `FutureSim` which uses forecasting questions from news documents, and updates the context available to agents at each time-step, tasking them to maintain a distribution of possible outcomes and their probabilities for each question. Current frontier agents are far from saturating the benchmark, and demonstrate multiple interesting behaviors in our testing. We also show how `FutureSim` can be efficiently ablated and reproduced to study a range of emerging capabilities. A key advantage of our benchmark is flexibility: it exposes a minimal set of actions that allows benchmarking of any model and harness, and the data can easily be updated, or extended to other domains where chronological events are available. The only constraint is that we focus on purely predictive settings, where agent actions cannot significantly change environment dynamics, as otherwise counterfactual worlds would have to be simulated accurately. While this limits applicability to certain decision making domains like public policy, we show that predictive tasks like ours still offer valuable insights about the world models of agents, and how they adapt to new information (Ha & Schmidhuber, 2018). We are thus hopeful that our work can guide AI progress towards the next frontier, of building agents that continually adapt and learn in the real-world.

References

Anthropic. Project Vend: Can Claude Run a Small Shop? (And Why Does That Matter?). https://www.anthropic.com/research/project-vend-1, June 2025. Accessed: 2026-05-03.

ARC Prize Foundation. Arc-agi-3: A new challenge for frontier agentic intelligence, 2026. URL https://arxiv.org/abs/2603.24621.

Atanasov, P., Witkowski, J., Ungar, L., Mellers, B., and Tetlock, P. Small steps to accuracy: Incremental belief updaters are better forecasters. *Organizational Behavior and Human Decision Processes*, 160:19–35, 2020. doi: 10.1016/j.obhdp.2020.02.001.

Backlund, A. et al. Vending-bench: A benchmark for long-term coherence of autonomous agents, 2025. URL https://arxiv.org/abs/2502.15840.

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mordatch, I. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sy0GnUxCb.

Brown, N., Bakhtin, A., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022. Meta Fundamental AI Research Diplomacy Team (FAIR).

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Chandak, N., Goel, S., Prabhu, A., Hardt, M., and Geiping, J. Answer matching outperforms multiple choice for language model evaluation, 2025. URL https://arxiv.org/abs/2507.02856.

Chandak, N., Goel, S., Prabhu, A., Hardt, M., and Geiping, J. Curating the future: A scalable recipe for training open-ended forecasters. In *ICML*, 2026. URL https://openreview.net/forum?id=SiMYGtHfxT.

Damani, M., Puri, I., Slocum, S., Shenfeld, I., Choshen, L., Kim, Y., and Andreas, J. Beyond binary rewards: Training LMs to reason about their uncertainty. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

Farquhar, S. and Gal, Y. Towards robust evaluations of continual learning, 2019. URL https://arxiv.org/abs/1805.09733.

Froger, R., Andrews, P., Bettini, M., Budhiraja, A., Cabral, R. S., Do, V., Garreau, E., Gaya, J.-B., Laurençon, H., Lecanu, M., Malkan, K., Mekala, D., Menard, P., Bertran, G. M.-T., Piterbarg, U., Plekhanov, M., Rita, M., Rusakov, A., Vorotilov, V., Wang, M., Yu, I., Benhalloum, A., Mialon, G., and Scialom, T. Gaia2: Benchmarking LLM agents on dynamic and asynchronous environments. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9gw03JpKK4.

Grady, T., Parker, K., Zarov, I., Course, H., Taylor, C., and Taylor, R. Kellybench: A benchmark for long-horizon sequential decision making, 2026. URL https://arxiv.org/abs/2604.27865.

Ha, D. and Schmidhuber, J. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models, 2024. URL https://arxiv.org/abs/2402.18563.

Hardt, M. and Sun, Y. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CNL2bku4ra.

He, M., Jain, A., Kumar, A., Tu, V., Bakshi, S., Patro, S., and Rajani, N. YC-Bench: Benchmarking ai agents for long-term planning and consistent execution, 2026. URL https://arxiv.org/abs/2604.01212.

Jin, B., Zeng, H., Yue, Z., Yoon, J., Arık, S. O., Wang, D., Zamani, H., and Han, J. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Proceedings of the 2nd Conference on Language Modeling*, Montreal, Canada, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Karger, E., Bastani, H., Yueh-Han, C., Jacobs, Z., Halawi, D., Zhang, F., and Tetlock, P. Forecastbench: A dynamic benchmark of AI forecasting capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lfPkGWXLLf.

Khattab, O. and Zaharia, M. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020. URL https://arxiv.org/abs/2004.12832.

Lou, X., Lázaro-Gredilla, M., Dedieu, A., Wendelken, C., Lehrach, W., and Murphy, K. P. Autoharness: improving llm agents by automatically synthesizing a code harness, 2026. URL https://arxiv.org/abs/2603.03329.

Mellers, B., Stone, E., Murray, T., Minster, A., Rohrbaugh, N., Bishop, M., Chen, E., Baker, J., Hou, Y., Horowitz, M., Ungar, L., and Tetlock, P. Identifying and cultivating superforecasters as a method of improving probabilistic predictions. *Perspectives on Psychological Science*, 10 (3):267–281, 2015. doi: 10.1177/1745691615577794.

Mucsányi, B., Kirchhof, M., Nguyen, E., Rubinstein, A., and Oh, S. J. Proper/strictly proper scoring rule, 2023. URL https://trustworthyml.io/.

Murphy, K. Agentic forecasting using sequential bayesian updating of linguistic beliefs, 2026. URL https://arxiv.org/abs/2604.18576.

Nagel, S. Common crawl news dataset, 2016. URL https://data.commoncrawl.org/crawl-data/CC-NEWS/index.html.

Paglieri, D., Cupiał, B., Coward, S., Piterbarg, U., Wolczyk, M., Khan, A., Pignatelli, E., Kuciński, Ł., Pinto, L., Fergus, R., Foerster, J. N., Parker-Holder, J., and Rocktäschel, T. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

Paleka, D., Goel, S., Geiping, J., and Tramèr, F. Pitfalls in evaluating language model forecasters. In *ICLR*, 2026. URL https://openreview.net/forum?id=z85kARAoyD.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

Seshadri, P., Cahyawijaya, S., Odumakinde, A., Singh, S., and Goldfarb-Tarrant, S. Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations, 2026. URL https://arxiv.org/abs/2601.17087.

Shao, R., Qiao, R., Kishore, V., Muennighoff, N., Lin, X. V., Rus, D., Low, B. K. H., Min, S., tau Yih, W., Koh, P. W., and Zettlemoyer, L. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*, 2025. URL https://arxiv.org/abs/2504.20595.

Shen, J., Bai, H., Zhang, L., Zhou, Y., Setlur, A., Tong, S., Caples, D., Jiang, N., Zhang, T., Talwalkar, A., and Kumar, A. Thinking vs. doing: Agents that reason by scaling test-time interaction, 2025. URL https://arxiv.org/abs/2506.07976.

Shi, Q., Zytek, A., Razavi, P., Narasimhan, K., and Barres, V. τ-knowledge: Evaluating conversational agents over unstructured knowledge, 2026. URL https://arxiv.org/abs/2603.04370.

Silver, D. and Sutton, R. S. Welcome to the era of experience, 2025. URL https://storage.googleapis.com/deepmind-media/Era-of-Experience%20/The%20Era%20of%20Experience%20Paper.pdf. Preprint of a chapter to appear in *Designing an Intelligence*, MIT Press.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Sinha, A., Arun, A., Goel, S., Staab, S., and Geiping, J. The illusion of diminishing returns: Measuring long horizon execution in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3lm8lWYxiq.

Snell, C. V., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Thai, M. V. T., Le, T., Manh, D. N., Nhat, H. P., and Bui, N. D. Q. Swe-evo: Benchmarking coding agents in long-horizon software evolution scenarios, 2026. URL https://arxiv.org/abs/2512.18470.

Tran, D. and Kiela, D. Single-agent llms outperform multi-agent systems on multi-hop reasoning under equal thinking token budgets, 2026. URL https://arxiv.org/abs/2604.02460.

Venkatraman, S., Jain, V., Mittal, S., Shah, V., Obando-Ceron, J., Bengio, Y., Bartoldson, B. R., Kailkhura, B., Lajoie, G., Berseth, G., Malkin, N., and Jain, M. Recursive self-aggregation unlocks deep thinking in large language models, 2026. URL https://arxiv.org/abs/2509.26626.

Wang, X., Li, B., Song, Y., Xu, F. F., Tang, X., Zhuge, M., Pan, J., Song, Y., Li, B., Singh, J., Tran, H. H., Li, F., Ma, R., Zheng, M., Qian, B., Shao, Y., Muennighoff, N., Zhang, Y., Hui, B., Lin, J., Brennan, R., Peng, H., Ji, H., and Neubig, G. Openhands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OJd3ayDDoF.

Webby, R. and O'Connor, M. Judgemental and statistical time series forecasting: a review of the literature. *International Journal of Forecasting*, 12(1):91–118, 1996. URL https://www.sciencedirect.com/science/article/pii/0169207095006443.

Wei, J., Sun, Z., Papay, S., McKinney, S., Han, J., Fulford, I., Chung, H. W., Passos, A. T., Fedus, W., and Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025. URL https://arxiv.org/abs/2504.12516.

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

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018. URL https://aclanthology.org/D18-1259/.

Yang, Z., Band, N., Li, S., Candes, E., and Hashimoto, T. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=07yvxWDSla.

Zhang, A. L., Kraska, T., and Khattab, O. Recursive language models, 2026a. URL https://arxiv.org/abs/2512.24601.

Zhang, J., Liu, G., Johansson, O., Yitayew, H., Ohly, K., and Li, G. Prediction arena: Benchmarking ai models on real-world prediction markets, 2026b. URL https://arxiv.org/abs/2604.07355.

Zhang, J., Zhao, B., Yang, W., Foerster, J., Clune, J., Jiang, M., Devlin, S., and Shavrina, T. Hyperagents, 2026c. URL https://arxiv.org/abs/2603.19461.

Appendix

Contents

A Data **13**

A.1 Question creation methodology 13

A.2 Example of a Discarded Question 13

A.3 Question Distribution 14

A.4 Additional Details 14

A.5 Mitigating Leakage of Future Information . 14

B Environment Details **14**

B.1 Harness details 14

B.2 Cost 15

B.3 Context Details: CCNews 16

C Metrics **16**

C.1 Properness of Brier Skill Score 16

C.2 Time-Weighted and Peer Scores 16

C.3 Aggregate Predictions and Update Size . . . 17

C.4 Variance Computation 17

D Additional experiments **17**

D.1 Main Results 17

D.2 Search Ablations Brier skill score 17

D.3 Inference Scaling Brier skill score 18

D.4 Multi-agent performance 18

D.5 Performance on Questions where all models update predictions 18

E Licenses and Social Impacts **19**

E.1 Licenses and terms of use 19

E.2 Broader impacts 19

F Prompts **19**

F.1 Native Harness 19

F.2 Our Harness 20

F.3 Multi-Agent 22

F.4 Earliest-Date Repair Prompt 22

F.5 Answer Matching Evaluation Prompts . . . 22

A. Data

A.1. Question creation methodology

A natural way to obtain resolved forecasting questions is to reuse questions from prediction markets (Yang et al., 2026). While being grounded in the real world, prediction markets also have their limitations, as highlighted by recent work (Chandak et al., 2026; Paleka et al., 2026): they cover a skewed subset of real-world events, often concentrated in sports, cryptocurrency, politics, and entertainment (Paleka et al., 2026), and they mostly use binary yes/no or multiple-choice formats. In `FutureSim` , we instead take a more scalable approach of synthesizing short-answer forecasting questions from any (news) source document introduced in Chandak et al. (2026).

The question generation pipeline takes timestamped articles as source documents and prompts an LLM to generate free-form short-answer questions whose answer is contained in the source article, while the question title, background, and resolution criteria are written as if the question were being asked before the answer is known. The pipeline then applies leakage checks, validates the question format, extracts and revises dates, filters invalid answer types, and removes questions that fail quality checks.

Our Data. The current question set is generated from Al Jazeera articles in the first quarter of 2026. We use this source because it produced a high-quality yield of globally relevant, freely accessible articles in CCNews, while still covering a broader event mix than prediction markets. We initiated our question generation from 10,000+ source articles eventually narrowing down to just 330 high quality questions (3%).

Additional Refinement. We curated the question set through multiple rounds of filtering and date repair once the initial set was obtained. Across these iterations, we removed duplicates, tautological or leaky questions, questions that resolved too early, questions that were already answerable by models too far in advance, and questions that appeared to be noisy after post hoc checks. We also repeated the resolution-date update loop multiple times, re-running the earliest-date inference procedure after each pass to further tighten dates or eliminate remaining early-resolving items.

We make several refinements for this benchmark:

- We create a test set resolving from January 1 to March 28, 2026, so the benchmark targets events after the knowledge cutoffs of the frontier models we evaluate.
- We revise each resolution date to the earliest date on which the answer could be inferred with certainty from public evidence. Our pipeline prompts models like GPT 5.5 with web-search to answer the question and identify the earliest inferable date; only judged-correct answers are used for date repair, and questions whose repaired date falls before January 1, 2026 are removed.
- We add answerability filters at both ends of the difficulty spectrum. Questions that models can answer confidently using search capped around June 2025 are removed as stale or too easy, while questions that models still fail to answer even with full web search as of April 2026 are removed as likely label noise.
- We keep the recipe source-agnostic: the same pipeline can be run on any timestamped document corpus, although this benchmark uses news articles as the source domain.

A.2. Example of a Discarded Question

The following example illustrates why the earliest-resolution pass is stricter than checking the final source article alone. At first sight, the question looks like a valid April 2026 forecasting target: the official PAHO meeting was announced for April 13, 2026, and the answer appeared explicitly in a January 2026 PAHO notice. However, a model with web search judged that the answer was already inferable before our evaluation window, so we discarded it.

Discarded Forecasting Question

Question. Which country will PAHO review alongside the United States at its virtual meeting on measles elimination status on April 13, 2026?

Background. In the Americas, countries can lose measles elimination status if local transmission continues for too long. A regional health body is monitoring countries whose status may be at risk.

Response of GPT-5.5 with web search

Answer. Mexico.

Earliest inferable date. November 11, 2025.

The official PAHO announcement on January 16, 2026 explicitly named the United States and Mexico, but the answer was already strongly determined by public evidence on November 11, 2025. By then, Canada had just lost its measles-free status, PAHO's director had publicly discussed which countries were next, and the United States and Mexico were the two remaining countries approaching the 12-month local-transmission threshold. The model concluded that the January announcement was confirmation rather than the first date on which the answer could be inferred.

Topic	# Questions	Fraction (%)
International Politics & Diplomacy	91	27.6
Conflict & War	78	23.6
Sports	75	22.7
Crime & Justice	26	7.9
US Politics	20	6.1
Disasters & Accidents	14	4.2
Other	26	7.9
Total	330	100.0

Table 2. Topic distribution of the 330 forecasting questions in `FutureSim`. "Other" bundles the long tail of low-frequency topics (Technology & AI, Entertainment & Culture, Business & Economy, Health & Medicine, Climate & Environment, Religion, Science & Space).

Each retained question becomes one forecasting task. In the main experiments, all tasks are visible from the start of the simulation on December 24, 2025 and remain active until their resolution date. Agents may submit and revise forecasts while a question is active. Each submission is a probability distribution over up to five free-form outcomes.

A.3. Question Distribution

Topic Distribution Table 2 summarizes the topic distribution of the 330 forecasting questions in `FutureSim`.

Resolution distribution. The questions resolve over 84 distinct calendar dates between January 1 and March 28, 2026. Of the 330 questions, 122 resolve in January, 92 in February, and 116 in March. This spread is intentionally denser than a small hand-written benchmark, giving agents many overlapping active questions while still allowing every question to resolve within a compact three-month simulation window.

A.4. Additional Details

Empirical difficulty. We also summarize difficulty of our question set by how often the final top prediction matches the resolved answer across the frontier models in our harness. By this measure, 229 questions are answered correctly by no run, 35 by exactly one run, 17 by exactly two runs, 16 by exactly three runs, and 33 by all four runs. Equivalently, 101 questions are solved by at least one run.

A.5. Mitigating Leakage of Future Information

`FutureSim` is designed so that agents can use large-scale retrieval without directly seeing future information. We mitigate leakage at three stages.

Question-text leakage. The question generation pipeline from Chandak et al. (2026) explicitly checks whether the title, background, or resolution criteria reveal the answer. Questions that leak the answer through wording, source-of-truth descriptions, answer format, or date choices are revised or removed before inclusion.

Resolution-date leakage. We revise resolution dates to the earliest public date on which the answer could be inferred with certainty. This avoids giving agents extra post-resolution time in which the answer may already be easy to recover from contemporaneous reporting. In practice, we ask online models to find both the answer and the earliest date on which the answer was publicly inferable, judge whether the answer matches the ground truth, and update dates using only judged-correct responses. We provide the prompt in Appendix F.4.

Environment leakage. During simulation, articles and search results are filtered by the current simulation date. Agents cannot query the live web, and sandboxing prevents direct reads from the hidden dataset, future article folders, search database, and sibling run directories. The implementation details are described in Appendix B.1.2.

B. Environment Details

B.1. Harness details

Our baseline harness. The motivation for our custom harness is that long-horizon forecasting tests agents on several frontiers at once: they must repeatedly revisit a large state, search a growing corpus, remember partially useful evidence over many days, and still leave enough context budget for reasoning and forecast submission. A minimal "just give the model shell and tool access" setup leaves ample performance on the table for open-weights models, especially, so our baseline harness adds lightweight structure intended to improve reliability without solving the task for the agent. In particular, it incorporates the following features:

1. **Context consumption feedback:** After each tool call, the agent receives feedback about remaining context budget and approximate context occupancy. This is useful because the task spans thousands of turns, and without explicit budget awareness, agents often spend too much context browsing or performing repeated file reads, leaving too little room for final reasoning and forecast submission. We also clear the live interaction context each day and explicitly tell the agent this, encouraging it to externalize useful information rather than assuming that important evidence will remain in-window indefinitely.
2. **Structured memory tools:** Instead of asking the agent to maintain free-form notes arbitrarily in its workspace, we expose external memory through explicit tool calls

with named entries and bounded fields. The goal is to make memory writing and retrieval a deliberate action rather than an accidental byproduct of shell usage. This structure also makes it easier for the agent to store compact, queryable summaries of evidence, lessons from resolved questions, and reusable forecasting heuristics.

3. **Per-question memory:** In addition to global notes, the harness maintains memory entries attached to individual questions. This is motivated by the fact that forecasting requires a mix of cross-question lessons and question-specific evidence: a general lesson about overconfidence should be stored differently from a candidate list or event-specific rationale for one market. Per-question memory helps agents revisit an active question after many simulated days without having to reconstruct all prior reasoning from scratch.

4. **Forced memory phase:** When the agent ends a day, or when the context budget becomes too tight, the harness enters an explicit memory-update phase before actually advancing. During this phase the agent is encouraged to compress what it learned into persistent notes and leave a cleaner state for the next day. The motivation is to prevent a common failure mode in long runs where agents defer summarization until too late, lose transient evidence to compaction, and then repeatedly rediscover the same information.

5. **Procedural forecasting scaffolding:** The prompt encourages a concrete workflow: inspect the active questions, prioritize the ones most worth updating, search for relevant evidence, submit forecasts, update memory, and only then proceed to the next day. This scaffolding is intentionally lightweight: it does not tell the model what the answer is, but it does reduce dithering and helps it allocate attention to imminent resolutions, stale forecasts, and questions where new evidence is most likely to matter. In our experience this kind of process guidance is especially important for getting models to repeatedly revise forecasts rather than treating an early answer as final.

B.1.1. SIMULATION LOGIC.

The simulation progresses in discrete time steps. At the beginning of the simulation, the agents are initialized with a prompt specifying the task, the scoring rules, the available tools and context, and the active questions; the native harness prompt is provided in Appendix F.1, with additional prompt variants in Appendix F. At each timestep, the agent can observe and interact with the environment via MCP tool calls, following the OpenReward Standard: `https://openrewardstandard.io/`. The agents can observe the currently active questions, search over the latest news corpus, and make predictions. When the agent wants to proceed to the next timestep, it can call the `next_day` action, which advances the environment by one timestep and updates the state accordingly. The state update reflects the new simulation date, the number of newly available articles, the number of active questions, and any questions that resolved on that date. For each resolved prediction, the environment exposes the ground-truth outcome and the score the agent accrued, in chronological order, which the agents can use to calibrate later predictions and learn new strategies. The simulation terminates after the last question resolves, at which point the final metrics are computed.

B.1.2. SANDBOXING AGENTS TO AVOID CONTAMINATION.

To make sure agents cannot access future information, we sandbox them carefully, using `bwrap` on a Linux server. Each harness runs inside a sandbox, an isolated process with its own private filesystem view and with controlled network access, to prevent leakage of unintended post-cutoff information. We ensure (i) *No live web search*: the harness has no direct network access, and the only access to external links is LLM provider endpoints used to run the model. `WebSearch`, `WebFetch`, and any commands like `curl` to any other endpoints are blocked. (ii) *Date-gated article corpus*: only the `articles/YYYY/MM/DD/` directories up to the current simulation date are exposed inside the sandbox, and the `search_news` MCP tool automatically caps its `to_date` so the underlying index cannot return future-dated articles. (iii) *Read-only environment state*: `state.csv` containing the questions, their metadata, and the agent's forecasts is read-only, and the environment codebase, dataset, and other run directories are not visible from inside the sandbox at all. The agent has read-write access only to its own sandboxed workspace.

B.2. Cost

Our experiments mix direct API usage, answer matching, and local retrieval infrastructure. Frontier models like GPT 5.5 and Opus 4.6 are very expensive to run for long evaluations. For GPT 5.5, Opus 4.6, and GLM 5.1, we therefore evaluated the agents through their providers' coding plans, each costing roughly $220 per month. These prices are not directly comparable to token-metered API execution, but we believe the equivalent cost of running the same long-horizon experiments through the parallel API would be substantially higher, potentially on the order of $10\times$ more for GPT 5.5 and Opus 4.6. For models run through metered inference, DeepSeek-v4-pro costs about $50 per simulation run, while Qwen through OpenRouter costs about $150 per run. In addition, each run requires answer matching over many candidate outcomes; we use DeepSeek V3.2 for this step because it is cheap and reliable for answer matching (Chandak

et al., 2025), costing less than \$50 for over 10,000 queries in a full run. Finally, our retrieval setup requires hosting the Qwen3 8B embedding model used by the LanceDB search pipeline on a single A100 or H100 GPU. These infrastructure costs are modest relative to the main agent runs for closed models, but they are part of the full end-to-end cost of reproducing our experiments.

B.3. Context Details: CCNews

The external context in `FutureSim` is a dated snapshot of Common Crawl News (CCNews), which lets us replay what evidence was available on each simulated day without relying on live web access. We first convert the raw crawl into a deduplicated article corpus, preserving article text, source, URL, and publication date. At runtime, agents only interact with the portion of this corpus whose dates are at or before the current simulation day.

In our experiments, agents access the corpus through a local LanceDB-backed retrieval tool rather than unrestricted browsing. Articles are split into moderate-sized text chunks, embedded with Qwen3 Embedding 8B, and indexed for hybrid retrieval combining semantic search and keyword matching. The search interface also supports explicit date bounds, so both the filesystem view and the retrieval layer enforce the same temporal cutoff. We treat this context store as part of the environment rather than part of any particular agent design: the benchmark is compatible with different retrieval, context-management, and memory strategies as long as they respect the date restrictions.

C. Metrics

We follow the notation from Section 3. For a question $q \in \mathcal{Q}$, the ground-truth answer is y_q, and an agent forecast is a set of outcomes $\Omega_q = \{o_1, \dots, o_k\}$ with probabilities $p_q(o_i)$ satisfying $p_q(o_i) \geq 0$ and $\sum_i p_q(o_i) \leq 1$. We assume all the outcomes in Ω_q are (semantically) distinct from each other. We write $p_q(o) = 0$ for outcomes not named in Ω_q. In implementation, equality to y_q is checked by the answer matcher, so the formulas below should be read up to semantic equivalence of answer strings.

As in the main text, the score for a resolved question is

$$\mathrm{BSS}(q) = 1 - \sum_{o \in \Omega_q \cup \{y_q\}} \left(p_q(o) - \mathbf{1}[o = y_q]\right)^2.$$

Higher is better: a fully confident correct forecast receives 1, an abstention with no probability mass receives 0, and assigning all probability to a wrong outcome receives -1. There is a daily column `avg_brier` in the tasks CSV which shows the mean of $\mathrm{BSS}(q)$ over all questions, with questions that have no forecast contributing 0. Once a question resolves, its final score is held fixed in later daily aggregates.

C.1. Properness of Brier Skill Score

Theorem C.1 (Strict properness for subprobability forecasts)**.** *Fix a question q. Suppose the forecaster's belief over the named outcomes Ω_q is given by probabilities $\pi_q(o)$ satisfying $\sum_{o \in \Omega_q} \pi_q(o) \leq 1$, with the remaining mass corresponding to the event that the true answer matches no outcome in Ω_q. Then among all reports p_q with $p_q(o) \geq 0$ and $\sum_{o \in \Omega_q} p_q(o) \leq 1$, the expected Brier skill score is uniquely maximized by reporting $p_q(o) = \pi_q(o)$ for every $o \in \Omega_q$.*

Proof. If the realized answer matches $o \in \Omega_q$, then the Brier skill score is

$$\mathrm{BSS}(p_q, o) = 1 - \sum_{u \in \Omega_q} \left(p_q(u) - \mathbf{1}[u = o]\right)^2 = 2p_q(o) - \sum_{u \in \Omega_q} p_q(u)^2.$$

If the realized answer matches no named outcome, then the scoring rule includes the true answer with forecast probability 0, so the score is

$$\mathrm{BSS}(p_q, Y) = 1 - \left(1 + \sum_{u \in \Omega_q} p_q(u)^2\right) = -\sum_{u \in \Omega_q} p_q(u)^2.$$

Taking expectation under the forecaster's belief gives

$$\begin{aligned}
\mathbb{E}\,\mathrm{BSS}(p_q, Y) &= \sum_{o \in \Omega_q} \pi_q(o) \left(2p_q(o) - \sum_{u \in \Omega_q} p_q(u)^2\right) - \left(1 - \sum_{o \in \Omega_q} \pi_q(o\right. \\
&= 2 \sum_{o \in \Omega_q} \pi_q(o) p_q(o) - \sum_{u \in \Omega_q} p_q(u)^2 \\
&= \|\pi_q\|_2^2 - \|p_q - \pi_q\|_2^2.
\end{aligned}$$

The first term in the final line is independent of the output, while the second is minimized uniquely at $p_q = \pi_q$. Since $\sum_{o \in \Omega_q} \pi_q(o) \leq 1$, this honest report is feasible. Therefore the Brier skill score is strictly proper even when both beliefs and reports are subprobabilities over the named outcomes. □

C.2. Time-Weighted and Peer Scores

For this section, let $p_{q,t}$ denote the forecast for question q that an agent is holding on day t, using the same notation as above at each snapshot. Let $T_q = \{o_q, \dots, r_q - 1\}$ be the days on which question q is open, and let $\mathrm{BSS}_t(q)$ be the Brier skill score of $p_{q,t}$, or 0 if no forecast has yet been submitted. The single-agent time-weighted score is

$$\mathrm{TW} = 100 \sum_{q \in \mathcal{Q}} \frac{1}{|T_q|} \sum_{t \in T_q} \mathrm{BSS}_t(q).$$

This rewards positive brier forecasts that are made earlier (and also penalizes negative forecasts made earlier), because a forecast earns its score on every day it is held until it is updated or the question resolves.

In multi-agent runs, each daily Brier skill score is first made peer-relative. For agent a, define

$$\mathrm{Peer}_{a,t}(q) = \mathrm{BSS}_{a,t}(q) - \overline{\mathrm{BSS}}_{-a,t}(q),$$

where $\overline{\mathrm{BSS}}_{-a,t}(q)$ is the average score of the other agents with active forecasts on q at day t; if there is no other active forecast, the baseline is 0. The time-weighted peer score applies the same single 100 multiplier as the single-agent time-weighted score:

$$\mathrm{TWPeer}_a = 100 \sum_{q \in \mathcal{Q}} \frac{1}{|T_q|} \sum_{t \in T_q} \mathrm{Peer}_{a,t}(q).$$

C.3. Aggregate Predictions and Update Size

When multiple agents forecast the same question, the market aggregate shown in the environment is the coordinate-wise mean of their current probabilities:

$$\bar{p}_q(o) = \frac{1}{n_q} \sum_{a=1}^{n_q} p_q^{(a)}(o),$$

again treating missing outcomes as probability 0.

For two forecasts p_q and p'_q with outcome sets Ω_q and Ω'_q, the total variation distance is

$$d_{\mathrm{TV}}(p_q, p'_q) = \frac{1}{2} \sum_{o \in \Omega_q \cup \Omega'_q} \left| p_q(o) - p'_q(o) \right|.$$

We report this metric when studying multi-agent dynamics in Figure 11.

C.4. Variance Computation

Most metrics and error bands in time plots like Figure 1 are computed from question-level scores, not from the already-averaged daily curves. For each model group, we first restrict attention to the questions that are present in all selected runs. On each date, the central curve is the average score on this common question set, averaging over runs within each question and then over questions.

The uncertainty band uses a two-level bootstrap. Each bootstrap replicate resamples questions with replacement from the common question set. For every sampled question, we also sample one of the selected runs, so the replicate captures both finite-question variation and run-to-run stochasticity. We then average the sampled question scores to obtain one bootstrap value for that date. For example, the plotted band in Figure 1 is 1 standard deviation of these bootstrap values around the central curve (mean).

D. Additional experiments

D.1. Main Results

Figure 7 shows the Brier-skill-score analogue of the main benchmark results in Figure 2. The trend is consistent with the main paper: GPT 5.5 remains clearly ahead, Opus 4.6 is the only other model ending with a positive average Brier skill score, GLM 5.1 stays close to the abstention baseline, and DeepSeek V4 Pro and Qwen3.6 Plus remain negative despite some improvement over time. Figure 8 shows that these differences are not explained simply by more environment interaction: GPT 5.5 makes the most tool calls, but Opus 4.6 and DeepSeek V4 Pro use similar order-of-magnitude tool budgets while ending with substantially different Brier scores.

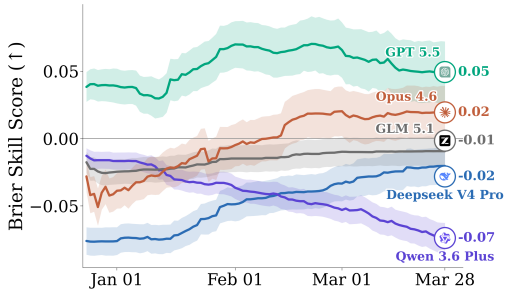

Figure 7. **Main-results brier skill score.** We plot the Brier skill score trajectories for the main frontier-agent benchmark in `FutureSim`. Consistent with the main accuracy and harness results, GPT 5.5 remains clearly ahead throughout the simulation, Opus 4.6 ends with a smaller positive score, and the remaining models stay near or below the abstention baseline.

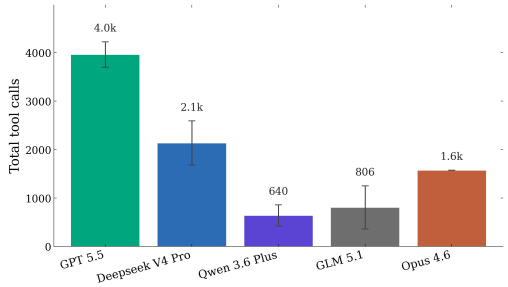

Figure 8. **Tool calls in the main benchmark.** We report the total number of tool calls made by each model during the simulation. GPT 5.5 uses the most tools, but tool-use volume alone does not explain performance: Opus 4.6 and DeepSeek V4 Pro use similar order-of-magnitude tool budgets while ending with substantially different Brier skill scores.

D.2. Search Ablations Brier skill score

Figure 9 gives the Brier-skill-score version of the search ablation in the left panel of Figure 5. It preserves the main qualitative conclusion that fresh daily context and agentic search matter: the full simulation with daily updates is much

better than freezing the corpus at day 0, and the one-day-before-resolution agentic search setting outperforms a single retrieval query. Unlike the accuracy plot, however, the one-day-before setting does not translate its accuracy gain into a strong Brier score, showing that late access to evidence helps top-1 prediction more than calibration.

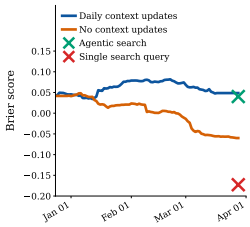

Figure 9. **Benefits from search.** We evaluate GPT 5.5 xhigh reasoning effort in four different settings to isolate the benefits of agentic search over updating context in `FutureSim`, this time measuring brier skill score. Consistent with the accuracy trend, we find large improvements from daily context updates (blue line) compared to when no articles beyond the first date are added during the simulation (orange line), and agentic search (green cross) outperforms just performing a single search query (red cross) one day before each question resolves. However, evaluating each question one day before its resolution does not improve brier skill score despite the large gain in accuracy, which can be attributed to poor distribution of probabilities over possible outcomes.

D.3. Inference Scaling Brier skill score

Figure 10 mirrors the inference-scaling accuracy result in the right panel of Figure 5. Increasing GPT 5.5's reasoning effort improves Brier skill score as well as accuracy, with the largest gains coming from moving away from no reasoning and diminishing returns at the highest efforts. This supports the same interpretation as the main figure: extra test-time compute helps the agent search and reason, but the benefit plateaus for this model in `FutureSim`.

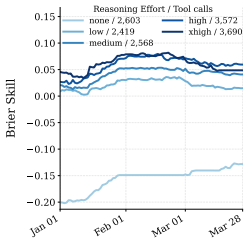

Figure 10. **Effect of scaling test-time compute on brier skill score.** We run GPT 5.5 in all five available reasoning efforts to see how additional inference compute changes brier skill score on `FutureSim`. We find higher reasoning effort consistently leads to better brier skill score, although the effect plateaus for this model after reasoning effort high. Notably, reasoning effort "none" has extremely poor brier skill score while not being as bad in accuracy, indicating the importance of reasoning for calibration (Damani et al., 2026).

D.4. Multi-agent performance

Figure 11 reports the accuracy and Brier-skill-score trajectories for the multi-agent experiment described in Section 5.5. The multi-agent runs reach higher accuracy than independent single-agent runs for all three DeepSeek v3.2 agents, matching the main-text observation that interaction through the aggregate changes agent behavior. The Brier curves are more mixed, but they remain in the same overall range as the single-agent runs, so the multi-agent setup changes prediction dynamics without producing a large absolute performance shift in this initial experiment.

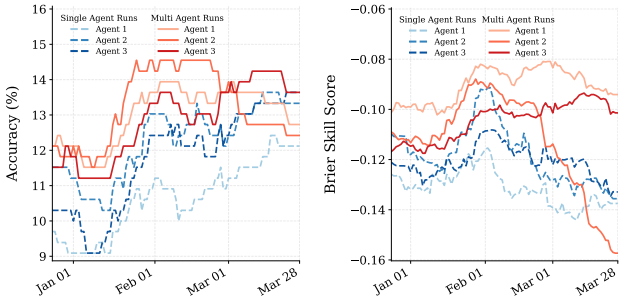

Figure 11. **Multi-agent performance.** When we run multiple copies of DeepSeek v3.2 agents simultaneously, we see individual agent predictions obtain slightly higher accuracy than the independent agent runs.

D.5. Performance on Questions where all models update predictions

One concern with our full benchmark is that some models may abstain on many more questions than others, so part of the overall comparison could in principle be skewed and driven by coverage differences rather than by forecast quality on shared questions. To control for this, we measure performance on the subset of questions on which every model makes at least one prediction during the simulation. This leaves only 46 questions and we recompute the trajectories on this common set and report performance in Figure 12.

The key result is that the accuracy rankings remain the same even after this restriction. In particular, the relative ordering from the main benchmark does not disappear when all models are evaluated on only the questions they all chose to engage with. This suggests that the main comparison is not merely a coverage effect from models such as Qwen3.6 Plus abstaining more often, but reflects real differences in forecast quality on shared questions. The brier score curves tell a similar story: absolute values enlarge on the easier shared subset, but the overall separation between stronger and weaker models remains.

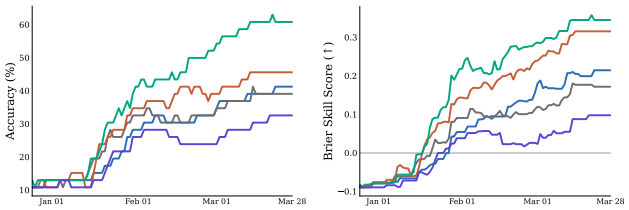

Figure 12. **Performance on the shared-prediction subset.** We restrict to the questions on which all models submit at least one forecast. (Left) Accuracy over time on this subset. The ranking remains consistent with the main results, showing that the overall accuracy ordering is not an artifact of differential abstention. (Right) Brier skill score on the same subset, which shows a broadly similar calibration story, with GPT 5.5 still clearly ahead of the others.

E. Licenses and Social Impacts

E.1. Licenses and terms of use

FutureSim uses several existing third-party assets and services. The news corpus is based on Common Crawl News (Nagel, 2016), which is made available under the Common Crawl Terms of Use (https://commoncrawl.org/terms-of-use). Common Crawl provides crawl data, but the underlying crawled page contents may remain subject to the rights and terms of the original content owners. The Al Jazeera articles used as source documents for question generation are treated as copyrighted source material under Al Jazeera's terms/legal notices (https://www.aljazeera.com/news/2010/4/28/legal-notices). We use these articles for question generation and date-gated retrieval/evaluation, and we do not redistribute the full raw article text.

For retrieval and evaluation infrastructure, we use LanceDB under the Apache License 2.0, Qwen3-Embedding-8B under the Apache License 2.0, bubblewrap/bwrap under LGPL-2.0-or-later, and OpenCode under the MIT License. Closed or hosted model services, including GPT 5.5/Codex, Claude Opus 4.6/Claude Code, DeepSeek V4 Pro, DeepSeek v3.2 as the answer matcher, GLM 5.1, and Qwen3.6 Plus, were accessed through the applicable provider subscription/API terms; we do not redistribute their model weights. Third-party prediction-market aggregate data is used only for the comparison analysis in Figure 3, is not redistributed, and remains subject to the platform's terms of use.

E.2. Broader impacts

FutureSim is intended to support reproducible research on adaptive forecasting agents by evaluating how they update beliefs under uncertainty in an offline, date-gated replay of real-world events. This could improve understanding of calibration, search, memory, and test-time adaptation, but stronger forecasting agents may also be misused for information advantage in competitive settings, persuasive manipulation around unfolding events, or overconfident decision support in high-stakes domains. Even when used as intended, results may reflect biases in the underlying news corpus and should not be taken as evidence that an agent is reliable for consequential policy, financial, legal, medical, or security decisions. We mitigate these risks by using an offline benchmark, which requires us to take special measures to prevent future-information leakage, such as sandboxing and date-gating access to context.

F. Prompts

F.1. Native Harness

The following template is rendered for the main forecasting runs. Runtime fields such as dates and question counts are filled in by the simulation harness.

```
You are a forecasting agent. Today is
↪  <current_date>. Your goal is to make accurate
↪  and calibrated predictions.

## UPDATE CADENCE
You have the chance to update your predictions
↪  every <timegap_days> day(s). Your workspace
↪  files (memory/, scripts, notes) persist across
↪  days -- use them to track reasoning and lessons
↪  learned. Articles are available via the search
↪  tool and in the articles/ directory. Current
↪  date: <current_date>. Next scheduled update:
↪  <next_date>.
<optional_resolution_cadence_note>

## SCORING (Brier Skill Score)
You have to output a distribution of (outcome,
↪  probability) pairs for each question you make a
↪  forecast on.
You are evaluated on the Brier Skill Score = 1 -
↪  sum_i (p_i - y_i)^2 summed over all outcomes,
↪  where:
- p_i = your probability for outcome i
- y_i = 1 if your outcome i is TRUE, 0 otherwise
- Higher is better: 1.0 = perfect, 0.0 = abstaining
↪  from guessing, negative = worse than
↪  abstaining.

Key Mechanics:
1. Accuracy + Calibration: assign calibrated
↪  probabilities that reflect true likelihoods.
2. Time-Weighted Score: forecasts made earlier
↪  matter, but updating is rewarded when new
↪  evidence arrives.
3. Prediction-Count Incentive: unanswered active
↪  questions receive zero contribution.
4. End-of-Session Metrics are shown after each
↪  session.
5. Max Outcomes: submit at most
↪  <max_outcomes_per_question> outcomes per
↪  question.
6. No Placeholders: "Unknown", "TBD", and "Other"
↪  hurt your score.

## AVAILABLE DATA
You have access to a news article database, which
↪  is updated daily through a search tool, that
↪  you can use to find evidence for your
↪  forecasts.
```

```
You can access the market.csv file (READ-ONLY) in
↪  your workspace containing <num_questions>
↪  questions (<num_active> active/unresolved,
↪  <num_resolved> resolved).

Column descriptions of the DataFrame (market.csv):
- qid, title, background, resolution_criteria,
↪  answer_type
- resolution_date, is_resolved, ground_truth
- num_predictions, options, my_prediction,
↪  my_prediction_date

Note: my_prediction contains your current forecast
↪  as a dict, or None if not yet predicted.
↪  ground_truth contains the resolved answer, or
↪  None if not yet resolved.

## TOOLS AVAILABLE FOR YOUR USE
- mcp__forecast__search_news(query, from_date?,
↪  to_date?): search the news corpus for evidence.
↪  to_date is capped at today's date.
- mcp__forecast__submit_forecasts(question_id,
↪  outcomes): submit exactly one forecast for
↪  exactly one question ID.
- mcp__forecast__next_day(): end the current
↪  session and proceed to the next one.

## Workspace:
- market.csv -- Read-only snapshot of all
↪  questions, refreshed each day.
- articles/ -- Browsable news articles organized by
↪  date as articles/YYYY/MM/DD/articles.jsonl.
- memory/ -- Persistent notes directory. Read and
↪  write freely.

You have full control over your workspace. You may
↪  create files or directories that help you
↪  perform better, such as forecasting strategies,
↪  calibration notes, scripts, or per-question
↪  research notes.

## SUBMISSION RULES
- qid must be from an active unresolved question
↪  identified from market.csv.
- Each mcp__forecast__submit_forecasts call must
↪  contain exactly one forecast for one question
↪  ID.
- You may submit again later in the same session to
↪  update that qid.
- Maximum of <max_outcomes_per_question> outcomes
↪  allowed per question.
- Outcome names must be real predicted answers.
- Never use placeholders like "Unknown", "TBD",
↪  "Other", or "N/A".
- Probabilities must sum to <= 1.0.

## Rules
- No web access is available. Use
↪  mcp__forecast__search_news and articles/ for
↪  information.
- market.csv is read-only. Do not modify it.
- You can use Bash, Read, Write, Grep, Glob, and
↪  other tools freely in your workspace.
- Your job is to maximize your time-weighted score.

Begin.
```

F.2. Our Harness

This prompt is used for our custom forecasting harness. It adds daily feedback, memory tools, update priorities, and action/context-budget guidance around the same forecasting task.

```
You are a forecasting agent. Today is
↪  <current_date>. Your goal is to make accurate
↪  and calibrated predictions.

<results_since_last_session>
## RESULTS SINCE YOUR LAST SESSION (<last_date> ->
↪  <current_date>)
- "<resolved_question_title>"
  Your prediction distribution:
  ↪  <prediction_distribution> | Truth:
  ↪  <ground_truth>
  Brier: <brier_skill_score> | TW-Score:
  ↪  <time_weighted_score>
...

## YOUR CUMULATIVE PERFORMANCE TILL TODAY
- Total Predictions: <num_predictions>
↪  (<num_resolved> resolved)
- accuracy: <accuracy>% | brier skill score:
↪  <mean_brier_skill_score>
 accuracy = fraction of resolved questions where
 ↪  your top outcome matched the truth; brier
 ↪  skill score = mean brier skill score across
 ↪  resolved questions

<source_context>
<source_specific_rules>

## UPDATE CADENCE
You can make updates every <timegap_days> day(s).
↪  Your context is cleared after every session and
↪  your memory (along with past predictions) is
↪  the only information retained between sessions.
↪  <new_articles_text><last_update_text>Current
↪  date: <current_date>. Next scheduled update:
↪  <next_date>.
<tomorrow_resolution_reminder>

IMPORTANT: You have predictions on
↪  <predicted_count> out of <active_count> active
↪  questions.
Tip: You can check your existing predictions by
↪  reading market.csv and filtering rows where
↪  `my_prediction` is not null.

UPDATE RULES:
- Do NOT re-predict questions from scratch unless
↪  you find specific new evidence.
- Only update a prediction if you find SPECIFIC NEW
↪  evidence (news, data) that updates your view.

PRIORITIES FOR UPDATES:
1. Questions resolving the next day (filter
↪  `market.csv` by `resolution_date` == tomorrow)
↪  -- make sure your prediction is up-to-date
↪  before calling next_day.
2. Questions without predictions (if any)
3. Questions where today's news search reveals new
↪  information
4. Questions approaching resolution date that you
↪  haven't checked recently
5. Skip questions where there is no new evidence

## YOUR MEMORY
Current meta-insights with their indices:
<meta_insight_index>

`mem_df` holds your per-question notes (reasoning,
↪  evidence, calibration) -- 1 row per question.
Columns: qid (str), question (str), last_updated
↪  (str), memory (str), category (str)
<prior_memory_location_or_empty_memory_note>

Inspect `mem_df` by reading <prior_mem_csv>. Edit
↪  per-question notes with
↪  `mcp__forecast__mem_add`,
↪  `mcp__forecast__mem_update`,
↪  `mcp__forecast__mem_delete`.
```

```
Manage meta-insights with
↪  `mcp__forecast__memory_retrieve` (using the
↪  indices), `mcp__forecast__memory_new`,
↪  `mcp__forecast__memory_update`,
↪  `mcp__forecast__memory_delete`.
Caps: meta-insights <= 500 entries; per-question
↪  `mem_df` memory <= 1000 chars per row.

## SCORING (Brier Skill Score)
You have to output a distribution of (outcome,
↪  probability) pairs for each question you make a
↪  forecast on.
You are evaluated on the Brier Skill Score = 1 -
↪  sum_i (p_i - y_i)^2 summed over all outcomes,
↪  where:
- p_i = your probability for outcome i
- y_i = 1 if your outcome i is TRUE, 0 otherwise
- Higher is better: 1.0 = perfect, 0.0 = abstaining
↪  from guessing, negative = worse than
↪  abstaining.

Key Mechanics:
1. Accuracy + Calibration: assign calibrated
↪  probabilities that reflect true likelihoods.
2. Time-Weighted Score: forecasts made earlier
↪  matter, but updating is rewarded when new
↪  evidence arrives.
3. Prediction-Count Incentive: unanswered active
↪  questions receive zero contribution.
4. End-of-Session Metrics are shown after each
↪  session.
5. Max Outcomes: submit at most
↪  <max_outcomes_per_question> outcomes per
↪  question.
6. No Placeholders: "Unknown", "TBD", and "Other"
↪  hurt your score.

## AVAILABLE DATA
You have access to a news article database which is
↪  updated daily through a search tool, that you
↪  can use to find evidence for your forecasts.
You also have access to a read-only `market.csv`
↪  file in your workspace with <num_questions>
↪  questions (<num_active> active/unresolved,
↪  <num_resolved> resolved).

Column descriptions of market.csv:
- qid, title, background, resolution_criteria,
↪  answer_type
- resolution_date, is_resolved, ground_truth
- num_predictions, options, my_prediction,
↪  my_prediction_date

Note: `my_prediction` contains your current
↪  forecast as a dict, or None if not yet
↪  predicted. `ground_truth` contains the resolved
↪  answer, or None if not yet resolved.

## TOOLS AVAILABLE FOR YOUR USE
- `mcp__forecast__search_news(query, from_date?,
↪  to_date?)`: search the news corpus for
↪  evidence. `to_date` is capped at today's date.
- `mcp__forecast__memory_retrieve` /
↪  `mcp__forecast__memory_new` /
↪  `mcp__forecast__memory_update` /
↪  `mcp__forecast__memory_delete`: manage
↪  meta-insight entries.
- `mcp__forecast__mem_add` /
↪  `mcp__forecast__mem_update` /
↪  `mcp__forecast__mem_delete`: manage
↪  question-specific notes in `mem_df`.
- `mcp__forecast__submit_forecasts(question_id,
↪  outcomes)`: submit exactly one forecast for
↪  exactly one question ID (`qid`).
- `mcp__forecast__next_day()`: first call enters
↪  memory-update mode; call it a second time after
↪  your memory updates to actually proceed to the
↪  next day.
```

```
You also have access to native tools Bash/Read/Grep
↪  etc. -- use them to read market.csv and browse
↪  articles/. The MCP server persists today's
↪  `mem.csv` + `meta.yaml` automatically on
↪  `mcp__forecast__next_day`; do not write under
↪  `memory/` yourself.

## Workspace:
- market.csv -- Read-only snapshot of all
↪  questions, refreshed each day.
- articles/ -- Browsable news articles organized by
↪  date as articles/YYYY/MM/DD/articles.jsonl.
- predictions/ -- Read-only record of your past
↪  submissions, one file per day as
↪  `predictions/YYYY-MM-DD.json`.
- memory/ -- Read-only persisted notes
↪  (`memory/YYYY-MM-DD/{mem.csv, meta.yaml}`).
↪  Read prior days' files for context; edit memory
↪  only through the MCP memory tools.

## INTERACTION FLOW
You have <max_actions> actions per day. Each query,
↪  search, memory operation, or submission uses 1
↪  action.
You have a context budget of <max_total_tokens>
↪  tokens for this session. This tracks both the
↪  input prompt and cumulative output tokens spent
↪  so far.
Keep at least <submit_reserve_tokens> tokens free
↪  for a final submit. Force-submit once the
↪  remaining context budget is at or below
↪  <force_submit_threshold_tokens>.
If both budgets are configured, both are enforced
↪  and the session ends when either one is
↪  exhausted.

You can interleave reads, searches, memory
↪  operations, and submissions as needed. Consider
↪  reading `memory/<last_date>/mem.csv` early to
↪  recall prior reasoning and identify which
↪  questions need attention.

## MEMORY WORKFLOW
Treat memory as two layers:
- `mem_df`: question-specific reasoning, evidence,
↪  and calibration notes for a single QID.
- meta-insights: reusable cross-question patterns,
↪  lessons, and calibration rules that should help
↪  on future days.

Before calling `mcp__forecast__next_day()`:
1. Update `mem_df` for questions you researched or
↪  forecasted today using `mcp__forecast__mem_add`
↪  / `mcp__forecast__mem_update`.
2. If today's work revealed a reusable pattern,
↪  lesson, or calibration rule, promote it into a
↪  meta-insight.
3. If a prior meta-insight is stale or
↪  contradicted, revise or delete it.

Do not use meta-insights as a daily activity log.
↪  If you learned nothing reusable today, it is
↪  fine to skip meta-insight writes.

## SUBMISSION RULES
- qid must be from an active (`is_resolved=False`)
↪  question you identified from market.csv.
- Each `mcp__forecast__submit_forecasts` call must
↪  contain exactly one forecast for one question
↪  ID.
- You may submit again later in the same session to
↪  update that qid.
- Maximum of <max_outcomes_per_question> outcomes
↪  allowed per question.
- Outcome names must be real predicted answers.
- Never use placeholders like "Unknown", "TBD",
↪  "Other", or "N/A".
- Probabilities must sum to <= 1.0.
```

```
Tip: After submitting a forecast, consider saving
↪  your reasoning and key evidence for that QID
↪  using `mcp__forecast__mem_add`/`mcp__forecast_⌋
↪  _mem_update`.

---
Budget at start:
Actions remaining: <max_actions>
Context tokens remaining: <max_total_tokens>
↪  (estimated current context 0 /
↪  <max_total_tokens>)
Note: current message tokens are not accounted for
↪  yet.
Begin.
```

F.3. Multi-Agent

When the simulation is run with multiple agents, the daily prompt is modified as follows. These changes are inserted in addition to the base forecasting prompt.

```
## MULTI-AGENT SETTING
You are competing against <N-1> other forecasting
↪  agents on the same set of questions.
You each predict independently on every wakeup day.
↪  After each day, your predictions
are averaged with the others' into a market
↪  aggregate (the `market_aggregate`
column), which you can see starting the following
↪  day.
You are scored relative to your competitors: to
↪  earn a positive time-weighted peer
score, your predictions need to be more accurate
↪  than the group average.

...

## SCORING (Time-Weighted Peer Score (Brier-Skill
↪  Based))

...

- **Time-Weighted Peer Score (TW-Peer)**: On each
↪  day a prediction is held, your
Brier Skill Score is compared to the mean of all
↪  other agents' scores for the same
question. These daily differences are summed over
↪  the lifetime of the prediction.
A positive TW-Peer indicates predictions that were
↪  consistently more accurate than
the group average.

...

**Relative Performance (multi-agent)**: Final
↪  scoring is relative, so you have to
outperform the market aggregate to gain positive
↪  peer score.

...

Note: `market_aggregate` and `my_prediction`
↪  columns contain Python dicts
(or None). You can access them directly, e.g.
`row['market_aggregate']['outcome_name']`.

- `market_aggregate`: the mean probability
↪  distribution across all agents' latest
predictions from the previous day. `None` on the
↪  first day.
- `my_prediction`: your own latest forecast, or
↪  None if you have not predicted this
question yet.
- `num_predictions`: total number of prediction
↪  submissions made on this question
across all agents and all days.
```

F.4. Earliest-Date Repair Prompt

The following prompt is used to figure out potentially already-resolved questions and infer the earliest date on which the answer could have been determined from public information.

```
You are provided with a forecasting question (which
↪  might be from the past). You have to find not
↪  only the answer to the question, but also the
↪  earliest date on which the answer to the
↪  question could be inferred. Be smart in your
↪  inference. The question might contain extra
↪  details about the situation/event being asked
↪  but I want you to find out the earliest date by
↪  which the answer could have been figured out
↪  (even without extra details). For example, if
↪  you had seen the question 6 months back, could
↪  you have figured out the answer confidently.

Question Title: <question_title>
Question Background: <background>
Expected Answer Type: <answer_type>

Think step by step about the information provided
↪  and put the answer to the question in <answer>
↪  </answer> tags and the earliest date on which
↪  the answer to the question could be inferred
↪  with certainty in <date> </date> tags. The date
↪  should be in the format YYYY-MM-DD.
Once you find the answer, please make sure to find
↪  THE EARLIEST DATE the answer could have been
↪  guessed. Try to search as much as possible
↪  across sites/pages to find out when was the
↪  earliest time the answer to the question was
↪  basically known/determined (or could have been
↪  inferred confidently from public knowledge).
```

F.5. Answer Matching Evaluation Prompts

This prompt is used at resolution time to decide whether a submitted free-form outcome matches the ground truth answer for scoring.

Resolved-answer equivalence prompt

```
You are an objective judge of forecasting
↪  predictions.

Question: "<question_title>"
Predicted outcome: "<predicted_outcome>"
Ground truth (actual answer): "<ground_truth>"

Does the predicted outcome match the ground truth?
↪  Rules:
- YES if predictions are semantically equivalent
↪  (same meaning, different wording)
- YES if predicted outcome is MORE SPECIFIC than
↪  ground truth (e.g. "David Raya" matches "Raya")
- NO if predicted outcome contains generic text
↪  like "Unknown" or "Answer 1" or "Option 1"
- NO if predicted outcome is VAGUER/MORE GENERAL
↪  than ground truth (e.g., "a goalkeeper" does
↪  NOT match "David Raya")
- NO if they refer to different things
```

```
Essentially, you have to grade whether the
↪  forecaster correctly predicted the ground truth
↪  answer for the question.
Answer strictly "Yes" or "No".
```

This prompt is used before scoring to cluster semantically equivalent free-form predictions, so differently worded outcomes are treated as the same candidate answer.

Prediction-clustering match prompt

```
You are an objective judge of forecasting
↪  predictions.

Question: "<question_title>"

New prediction: "<candidate_prediction>"

Existing predictions:
1. <existing_prediction_1>
2. <existing_prediction_2>
...
N. <existing_prediction_N>

Does the new prediction match any of the existing
↪  predictions semantically?
- Match if they mean the same thing or if new
↪  prediction is more specific
- Do NOT match if new prediction is vaguer/more
↪  general

If yes, respond with ONLY the number (e.g., "1" or
↪  "3").
If no match exists, respond with "None".

Answer:
```