FutureWorld: A Live Reinforcement Learning Environment for Predictive Agents with Real-World Outcome Rewards

Zhixin Han[1,6*], **Yanzhi Zhang**[2,6*], **Chuyang Wei**[3,6], **Maohang Gao**[3,6], **Xiawei Yue**[1,6], **Kefei Chen**[5,6], **Yu Zhuang**[4,6], **Haoxiang Guan**[3,6], **Jiyan He**[6], **Jian Li**[5], **Yitong Duan**[6†], **Yu Shi**[6†], **Mengting Hu**[1†], **Shuxin Zheng**[6†]

[1]**College of Software, Nankai University**
[2]**Academy of Mathematics and Systems Science, Chinese Academy of Sciences**
[3]**School of Computer Science and Technology, University of Science and Technology of China**
[4]**Institute of Automation, Chinese Academy of Sciences**
[5]**IIIS, Tsinghua University** [6]**Zhongguancun Academy, Beijing, China**

zhixinhan@mail.nankai.edu.cn, zhangyanzhi20@mails.ucas.ac.cn,
duanyitong@zgci.ac.cn, shiyu@bza.edu.cn, mthu@nankai.edu.cn, sz@zgci.ai

Abstract

Live future prediction refers to the task of making predictions about real-world events before they unfold. This task is increasingly studied using large language model-based agent systems, and it is important for building agents that can continually learn from the real world. It can provide a large number of prediction questions grounded in diverse real-world events, while preventing answer leakage. To leverage the advantages of future prediction, we present FutureWorld, a live agentic reinforcement learning environment that closes the training loop between prediction, outcome realization, and parameter updates. Specifically, we modify and extend `verl-tool`, resulting in a new framework that we call `verl-tool-future`. Unlike standard reinforcement learning training frameworks that rely on immediate rewards, `verl-tool-future` stores prediction-time rollouts, backfills rewards after real-world outcomes become available, and then replays the completed trajectories for policy update. Across three open-source agents, successive FutureWorld training rounds lead to consistent improvements in prediction accuracy, probabilistic scoring, and calibration, demonstrating that delayed real-world outcome feedback can serve as an effective reinforcement learning signal.

1 Introduction

Live future prediction refers to the task of making predictions about real-world events before they unfold, where the results are unknown at prediction time but can be verified based on subsequent real-world outcomes Zeng et al. [2025], Liu et al. [2026], Karger et al. [2025]. Recent works Zeng et al. [2025], Liu et al. [2026], Karger et al. [2025], Turtel et al. [2025], Metaculus [2025], UniPat AI [2026], Jeen et al. [2026], Su et al. [2026] increasingly study this task using large language model (LLM)-based agent systems, often in the form of agentic prediction pipelines that retrieve information, reason, and generate predictions. This task naturally supports the development of agent systems that make predictions, observe realized outcomes, and update their policy accordingly. In this way, they can remain aligned with evolving real-world dynamics instead of relying solely on static knowledge

*These authors contributed equally to this work.
†Corresponding authors.

Preprint.

acquired during training. It is important for building agents that can continually learn from real-world feedback, adapt to changing conditions, and remain reliable in the dynamic real world.

Recent works on LLM agents have explored interactive environments for agentic capabilities, including web navigation, application use, and knowledge-work automation Zhou et al. [2024], Trivedi et al. [2024], Drouin et al. [2024], Majumder et al. [2023], Chen et al. [2025]. Similarly, live future prediction involves a temporal interaction loop in which agents make predictions about unresolved real-world events, receive outcome-grounded feedback after those events are realized, and improve their policies over time. This naturally motivates treating it as a learning environment. Such an environment should satisfy three requirements. **First**, the stream of questions should remain live, because new prediction targets continually emerge from ongoing real-world developments, and a static collection of historical questions cannot meet the need for live continual learning. **Second**, the learning signal should be grounded in realized outcomes, rather than relying solely on proxy signals such as process rewards, so that optimization remains aligned with the true objective of prediction. **Third**, training should include the agent's own information retrieval and analysis, because in realistic future prediction the agent must learn not only how to make predictions from given information, but also what information to seek and how to interpret it. In this way, live future prediction defines a learning environment in which agents can continually adapt their policies through real-world feedback and continually update their understanding of an evolving world.

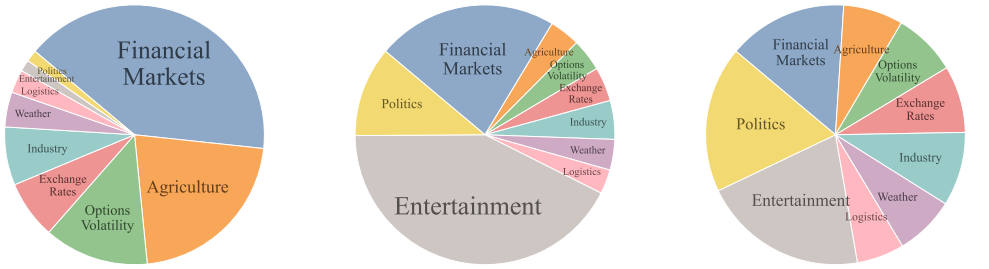
Prior works have explored future prediction from different aspects. Existing works Zeng et al. [2025], Karger et al. [2025], Liu et al. [2026] take **an important step** by providing live evaluation infrastructures that continually introduce new prediction questions over time. **Another step** is to place learning and agentic rollouts inside a live setting. UniPat AI [2026] moves in this direction by combining reinforcement learning (RL) with agentic rollouts in a live prediction setup. However, its learning signal is based on rubric-based process rewards rather than realized outcomes, leaving a gap between the training signal and the true objective of prediction. **A further step** is to optimize learning directly against realized outcomes. Recent works Turtel et al. [2025], Jeen et al. [2026] move toward this objective by exploring outcome-based RL. However, they do so only on static datasets of previously resolved questions, where the evidence available for prediction is fixed in advance. As a result, the agent is not truly trained to search for, select, and interpret evidence in a live environment. Taken together, prior works have explored different important aspects of live future prediction, but no existing work has yet established it as a unified learning environment.

A closer look at live future prediction reveals several properties that make it particularly well suited as a learning environment. Feedback can be derived directly from real-world outcomes, without requiring human annotation. This greatly reduces cost and makes it possible to scale training over a large number of prediction questions. In addition, the diversity of the real world allows prediction questions to span many domains. Notably, data leakage is prevented by construction.

To leverage the advantages of live future prediction, we present FutureWorld, a live agentic RL environment that closes the training loop between prediction, outcome realization, and parameter updates. Each day, the system automatically generates a large number (2,047.29 on average, see Section 3.6) of prediction questions from a broad set of carefully selected, high-value event sources spanning many domains, as illustrated in Figure 1 (a). The generated questions are first filtered to remove low-quality items, after which the remaining questions are resampled to balance domain proportions while reducing the number of similar questions within each domain. For each prediction question, the agent executes a rollout that may involve issuing search queries, reading retrieved information, reasoning, and producing a prediction, while the environment records the trajectory. The ground-truth outcome is provided later, when the event resolves, at which point it is matched to the corresponding stored trajectory and used to compute a reward. To support this delayed-feedback loop, we modify and extend `verl-tool` Jiang et al. [2025], and introduce the resulting framework as `verl-tool-future`. This proposed framework decouples prediction-time rollout collection from later outcome retrieval, reward backfilling, and policy update. The accumulated rewards are then used to update the model's parameters. The entire pipeline runs autonomously on a daily cycle.

In our environment, we take three open-source base language models and train them using outcome-based RL. The results show that training is effective. In summary, our contributions are as follows:

- We introduce FutureWorld, an agentic RL learning environment for live future prediction. To the best of our knowledge, it is the first open learning environment in which agents learn directly from the outcomes of their own predictions about the future.



(a) Domain distribution of website sources. (b) Domain distribution of questions before resampling. (c) Domain distribution of questions after resampling.

Figure 1: Domain distributions of website sources (a), questions before resampling (b), and questions after resampling (c). After resampling, questions are more evenly distributed across domains.

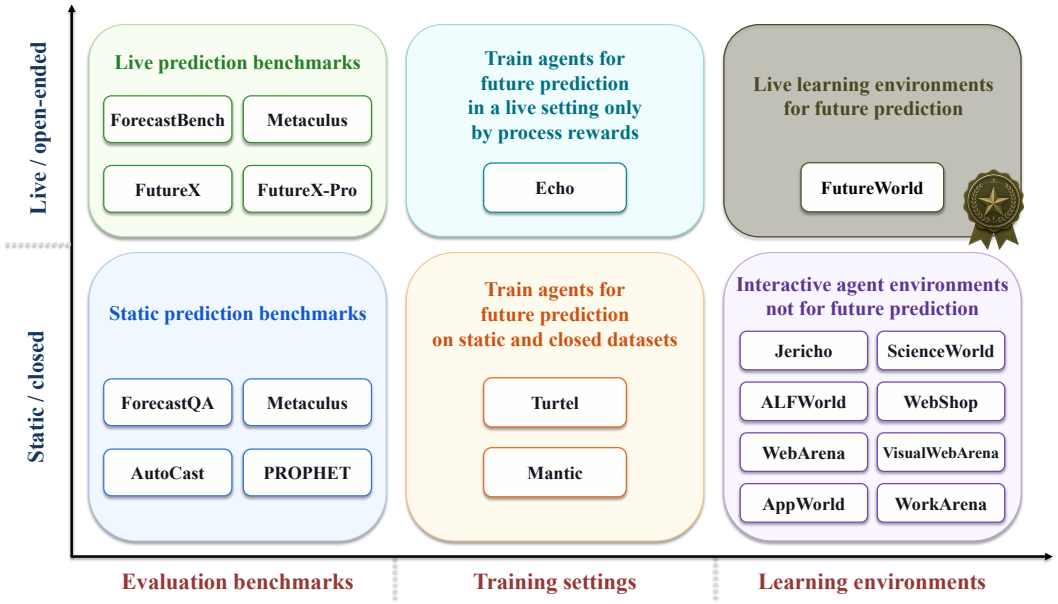


Figure 2: FutureWorld fills the gap for predictive agents.

- We propose `verl-tool-future`, our modified and extended version of `verl-tool` for delayed-feedback RL in live future prediction.
- We provide empirical evidence that outcome-based RL in a live environment can improve prediction ability over time. Open-source models trained in FutureWorld achieve progressively better performance across successive training days.

2 Related work

Prior works fall into three lines. Future prediction benchmarks evaluate models on unresolved real-world events. Recent methods for training predictive agents have improved performance, but often rely on static non-live datasets, pre-retrieved information, or indirect process-based reward signals. Agent environments, meanwhile, highlight the importance of interaction with timely and reproducible feedback, but are usually not built for live future prediction with direct outcome-grounded learning signals. Figure 2 highlights the remaining gap.

2.1 Benchmarks for future prediction

Future prediction benchmarks evaluate models on real-world events. Earlier benchmarks such as ForecastQA Jin et al. [2021] and AutoCast Zou et al. [2022] use historical prediction questions, while

PROPHET Tao et al. [2025] further emphasizes inferability by constructing prediction questions paired with supporting news evidence. More recent efforts have moved prediction evaluation closer to live settings. ForecastBench Karger et al. [2025] and FutureX Zeng et al. [2025] introduce automatically updated benchmarks of unresolved prediction questions, while FutureX-Pro Liu et al. [2026] extends this live paradigm to higher-value vertical domains.

2.2 Training agents for future prediction

Existing methods have already explored the problem of improving future prediction ability in LLM-based agent systems. **Some works** apply outcome-based RL to static historical datasets of already resolved prediction questions. Turtel et al. [2025] demonstrate that fine tuning with negative Brier score as a reward signal yields substantial accuracy gains on a 14B parameter model trained over 110,000 resolved Polymarket events,[1] but the agent operates only on a fixed prompt that contains pre-collected information, rather than acquiring information through its own web search process, so the information gathering policy remains entirely untrained. Jeen et al. [2026] apply Group Relative Policy Optimization (GRPO) Shao et al. [2024] to a 120B-parameter model using the Brier score reward on approximately 10,000 historical binary prediction questions with resolved outcomes, achieving notable improvements on the Metaculus benchmark Metaculus [2025]. However, its research phase is also performed before training. **Other work**, exemplified by UniPat AI [2026], operates in a live setting with agentic rollouts and a daily rolling cycle, but uses rubric-based process rewards rather than direct outcome-based rewards, introducing an indirection between the training signal and the target objective. Across these efforts, no existing system simultaneously trains on outcome-derived rewards, performs agentic information retrieval inside the training loop, and operates over a live, rolling stream of questions.

2.3 Agent environments

Agent environments have established the value of interaction and feedback for training and evaluating large language agents. Prior works have studied a range of environments for interactive agent tasks with timely feedback and reproducible evaluation. Early text-based settings such as Jericho Hausknecht et al. [2020] provide interactive fiction games for language-conditioned action. ScienceWorld Wang et al. [2022] focuses on grounded scientific reasoning in an interactive text environment, and ALFWorld Shridhar et al. [2021] connects text-based interaction with simulated embodied household tasks. Other works study web and software interaction in more realistic yet still static and closed settings. WebShop Yao et al. [2023] formulates grounded web interaction as an online shopping task, WebArena Zhou et al. [2024] recreates functional websites in a reproducible sandbox, VisualWebArena Koh et al. [2024] extends this paradigm to visually grounded web tasks, AppWorld Trivedi et al. [2024] builds a controllable ecosystem of apps populated with fictitious users, and WorkArena Drouin et al. [2024] targets browser-based knowledge-work tasks in enterprise software. However, all of these environments remain static (non-live) and closed, which limits their ability to expose agents to open-ended real-world dynamics.

3 FutureWorld environment

3.1 Problem definition

We formulate live future prediction as a delayed-feedback interaction problem. Each instance is a prediction prompt q about a future event whose outcome is unknown at prediction time. At prediction time t_q^{pred}, an agent interacts with external information sources through a sequence of search actions and observations, and then produces a final probability estimate for whether the target event will occur. In this formulation, each live future prediction instance is converted into a binary classification problem. For the k-th stochastic rollout of question q, FutureWorld represents the completed trajectory as

$$\tau_{q,k} = \big(q,\, t_q^{\text{pred}},\, a_{q,k,1}, o_{q,k,1}, \ldots, a_{q,k,m}, o_{q,k,m}, \hat{\pi}_{q,k},\, t_q^{\text{resolve}},\, z_q,\, r_{q,k}\big), \tag{1}$$

where $a_{q,k,i}$ is the i-th search action, $o_{q,k,i}$ is the corresponding search observation, m is the number of search steps in the rollout, $\hat{\pi}_{q,k} \in [0,1]$ is the final probability estimate, t_q^{resolve} denotes the

[1] `https://polymarket.com`

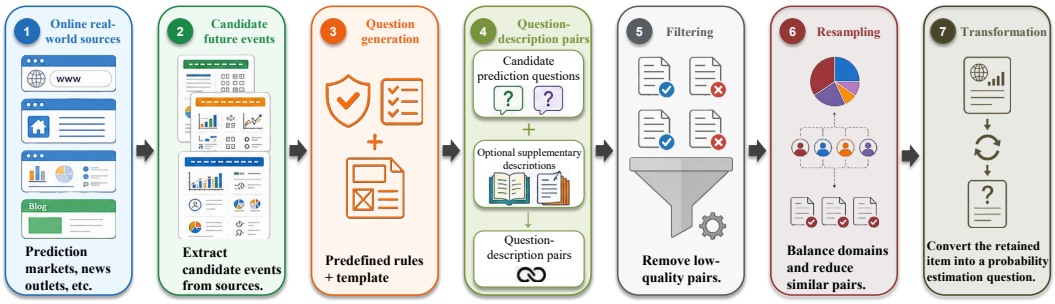


Figure 3: Overview of the FutureWorld pipeline for constructing prediction prompts.

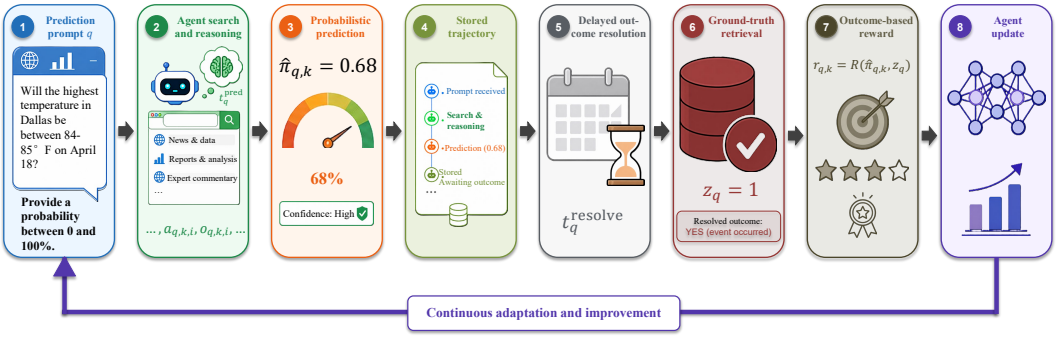


Figure 4: Overview of the FutureWorld training loop.

scheduled time at which FutureWorld attempts to retrieve the outcome of q, $z_q \in \{0, 1\}$ is an indicator of whether the target event actually occurs, and $r_{q,k}$ is the delayed trajectory-level reward.

The trajectory is completed in two distinct stages. Before the event resolves, FutureWorld stores the prediction-time prefix containing q, t_q^{pred}, the action-observation sequence, and $\hat{\pi}_{q,k}$. At the scheduled resolution time t_q^{resolve}, FutureWorld attempts to retrieve the event outcome. If the outcome is available, the environment backfills the realized label

$$z_q = \begin{cases} 1, & \text{if the target event occurs,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

It then computes the delayed reward $r_{q,k} = R(\hat{\pi}_{q,k}, z_q)$. This schema makes FutureWorld a live environment with timestamps, tool interaction, delayed outcome labels, and reward backfilling.

3.2 Environment overview

Building on the definition above, FutureWorld implements a live environment that supports the complete pipeline from question generation to delayed reward assignment. As illustrated in Figure 3, the process begins by collecting data from a broad set of public online sources about candidate future events, and then constructs questions using predefined rules or templates and, when applicable, associates them with supplementary descriptions that provide additional background information. These pairs are further processed through filtering and resampling so that the final retained set is higher quality, more balanced across domains, and has lower within-domain question similarity. The retained questions are then converted into final prediction prompts, where the agent is asked to estimate the probability that the specified future event will actually occur.

Once deployed in the environment, as shown in Figure 4, an agent receives a prediction prompt q at prediction time t_q^{pred}, interacts with external information sources through a sequence of search actions and observations $\{(a_{q,k,i}, o_{q,k,i})\}_{i=1}^{m}$, and then produces a final probability estimate $\hat{\pi}_{q,k}$. FutureWorld records this prediction-time process as the prefix of a rollout trajectory $\tau_{q,k}$. At the scheduled resolution time t_q^{resolve}, when the corresponding real-world outcome is expected to be available, the environment attempts to retrieve the ground truth, backfills the resolved label z_q, and

computes the outcome-based reward $r_{q,k}$. In this way, FutureWorld serves as a continuously refreshed delayed-feedback environment that maintains a persistent stream of future prediction questions and closes the loop among agent interaction, realized outcomes, and policy improvement.

3.3 Data sources

FutureWorld begins by collecting candidate future events from a broad set of online sources through automated network requests.[2] We maintain a pool of 72 websites. These source websites cover a wide range of domains. Figure 1 (a) shows the domain distribution of the source websites. Most of these websites cover consequential real-world developments.

3.4 Construction of question-description pairs

After collecting data from source websites, FutureWorld instantiates candidate prediction questions using predefined rules or templates. These questions are formulated as binary prediction questions that ask whether a specific event will occur. For example, a question may take the form *Will the highest temperature in Dallas be between 84-85°F on April 18?* Each question may also be paired with an optional supplementary description to provide additional background information. For some websites, the description can be directly extracted from the source page, whereas for websites that periodically publish a fixed set of data types, we pre-generate type-level descriptions with the assistance of an LLM, GPT-5.4.[3]

3.5 Question-description pair filtering

Based on our exploration, we identify three criteria that high-quality prediction questions should satisfy. **First**, it should be objectively resolvable. The outcome should be verifiable from publicly accessible evidence. **Second**, it should concern a meaningful real-world outcome. **Third**, it should be safe for public release. The question should exclude sensitive, harmful, or otherwise inappropriate content. To enforce all these quality requirements, FutureWorld applies a filtering stage.

To operationalize the three criteria above, we design three filters for quality control. Each filter asks the LLM to judge whether a question-description pair violates the corresponding eligibility criteria. The optional description included in the input helps the LLM assess eligibility more reliably even without web search, thereby reducing the cost. A pair is removed if any filter flags it as ineligible. We use `bytedance-seed/seed-1.6` as the filtering model.[4]

3.6 Question-description pair resampling

After filtering, FutureWorld applies resampling to the remaining question-description pairs. It is designed to balance the proportions across different domains while reducing question similarity within each domain. The target number of questions retained after resampling can be manually specified. The detailed resampling procedure is provided in Appendix A.

After performing the resampling procedure, we obtain the final set of questions. In practice, before resampling, we typically obtain around 2,047.29 questions on average across seven observations. We set the target number of questions retained to 500. Figure 1 (b) and (c) show the domain distributions before and after resampling, respectively. As shown in the figure, the distribution of questions across domains becomes more balanced after resampling.

3.7 Prompt construction for probabilistic prediction

The resampled set is then passed to the prompt construction stage. FutureWorld applies a predefined prompt template to convert each retained binary question into a probability estimation question. This probabilistic formulation allows the agent to express uncertainty explicitly and provides a richer, more fine-grained supervision signal for learning. Although some questions are paired with supplementary

[2] FutureWorld only uses publicly accessible sources. The collected data are used only for academic research, and not for any illegal or commercial purpose.

[3] `https://openai.com/index/introducing-gpt-5-4`

[4] `https://seed.bytedance.com/en/seed1_6`

descriptions in earlier stages to support filtering and resampling, these descriptions are omitted from the final prompts given to the agent, so that the agent is not given additional hints.

3.8 Ground-truth retrieval

Outcome retrieval is implemented as a set of source-specific resolvers rather than a single universal rule, since different sources expose outcomes through different interfaces and encode resolution states in different formats. For sources supported by `AkShare` King [2022], the corresponding resolver queries `AkShare` to obtain the structured data used for outcome resolution. Each question record stores the information needed to route the question to the appropriate resolver and verify the returned outcome, including the source website, source URL, expected resolution time, source-specific metadata when available, and other fields. If the source has not yet published a valid outcome, or if the returned information cannot be reliably matched to the original question, the question is marked as unresolved and excluded from scoring.

3.9 Training loop through verl-tool-future

To decouple data collection, prediction-time rollout generation, outcome acquisition, and model updating, we modify `verl-tool` Jiang et al. [2025] to support storing LLM rollouts. We refer to our modified framework as `verl-tool-future`.

At 20:00 on day t, we use the FutureWorld environment to obtain a batch of prediction questions whose outcomes are expected to be resolved on the following day, i.e., day $t+1$. The LLM agent is required to perform at least one web-search action before giving its final prediction, and the prediction-time trajectory prefix defined in Section 3.1 is saved. At 20:30 on day $t+1$, FutureWorld attempts to retrieve answers for the batch of prediction questions issued on day t. For questions whose outcomes are successfully obtained, it backfills the resolved label z_q and the corresponding reward $r_{q,k}$ into the stored trajectories. The completed trajectories are then used to update the model parameters through reinforcement learning.

FutureWorld runs every day to collect new prediction questions and attempt to retrieve outcomes for the questions issued yesterday. However, real-world events and public reporting are not perfectly synchronized. Even when an event has already resolved, its outcome may not yet be available on the corresponding source website because of delays. Furthermore, some prediction targets may be postponed or canceled. Overall, when ground-truth retrieval is performed in the evening, some previous prediction questions may still remain unresolved. Across five consecutive days of observation, an average of 35.65% of questions do not have retrievable ground truth at the scheduled retrieval time. When ground truth cannot be successfully retrieved, our strategy is to discard the corresponding questions and exclude them from training.

4 Training agents in FutureWorld

4.1 Reward and replay

We define trajectory-level reward as the negative Brier loss Jeen et al. [2026], Turtel et al. [2025]:

$$r_{q,k} = -\left(\hat{\pi}_{q,k} - z_q\right)^2. \tag{3}$$

If the agent fails to output a valid probability in $[0,1]$, we set $r_{q,k} = -1$, the lowest possible reward.

We optimize the policy with GRPO Shao et al. [2024]. During rollout generation, the agent's search queries are treated as policy actions, while the information returned by the search tool is treated as external observations. These tool observations are masked in the policy loss because they are produced by the external environment rather than by the policy itself.

4.2 Experiments

4.2.1 Performance gains across consecutive training days

We save the resulting model checkpoint after each day of training. After training for 8 consecutive days, we evaluate all daily checkpoints on the same set of 500 prediction questions to ensure a

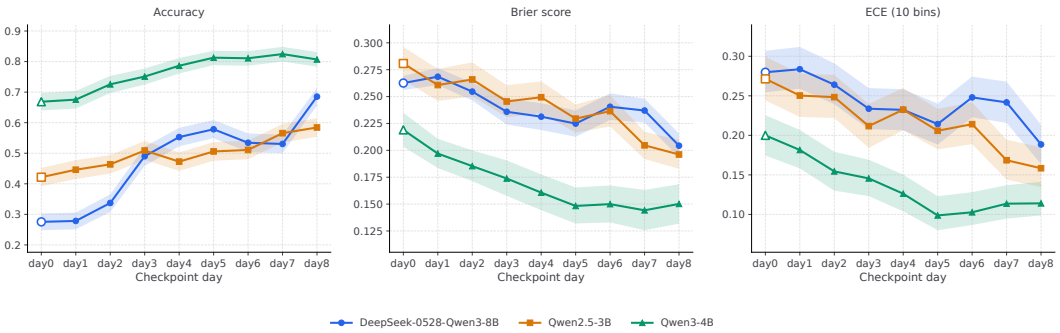


Figure 5: Prediction performance across model checkpoints saved on different days. Shaded regions indicate 95% bootstrap confidence intervals.

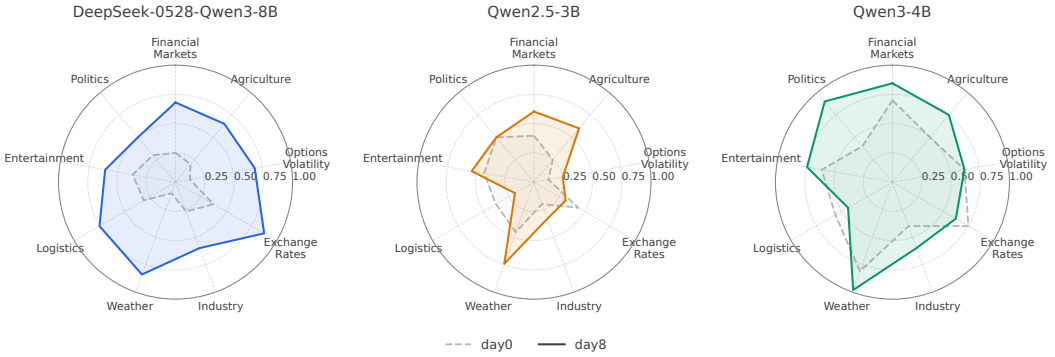


Figure 6: Domain-wise prediction performance before and after FutureWorld training.

consistent comparison. All nine checkpoints (day0-day8) are evaluated on the same day. Figure 5 summarizes the evaluation results. It reports accuracy after converting each probabilistic estimation into a binary prediction using a 0.5 threshold. When the predicted probability is 0.5, we treat it as a positive prediction, i.e., the model is considered to predict that the specified event will occur in the future. Accuracy generally improves over the consecutive training days. Figure 5 also reports the Brier score, which directly evaluates the quality of the probabilistic predictions. For the Brier score, lower values indicate better predictions. Furthermore, we use 10 equal-width probability bins to compute expected calibration error (ECE) Guo et al. [2017]. ECE decreases gradually over time. All results indicate that RL training in FutureWorld can improve agents' ability to predict future events.

4.2.2 Domain-wise performance gains after FutureWorld training

Figure 6 provides a domain-wise comparison between the initial models and the checkpoints after 8 days of FutureWorld training. The figure shows that the day-8 checkpoints generally outperform the initial models across most domains, indicating that the gains are not driven by a single domain of questions. This suggests that FutureWorld training improves the agents' general ability to gather information and reason, rather than merely adapting to a narrow domain-specific pattern.

4.2.3 Generalization beyond binary questions

Our RL training is conducted on binary questions. We therefore further examine whether the acquired predictive ability transfers beyond this binary format. To this end, we design the FutureWorld daily benchmark as a more general-purpose benchmark for evaluating predictive agents, as described in detail in Appendix C. As summarized in Table 1, the benchmark covers four question types: binary choice, simple multiple choice, difficult multiple choice, and numeric prediction. Representative examples of each type are shown in Table 2. Unlike the binary probabilistic prompts used during training, the daily benchmark requires agents to answer in heterogeneous formats. The prompt templates used for these four benchmark formats are provided in Appendix D. This allows us to test

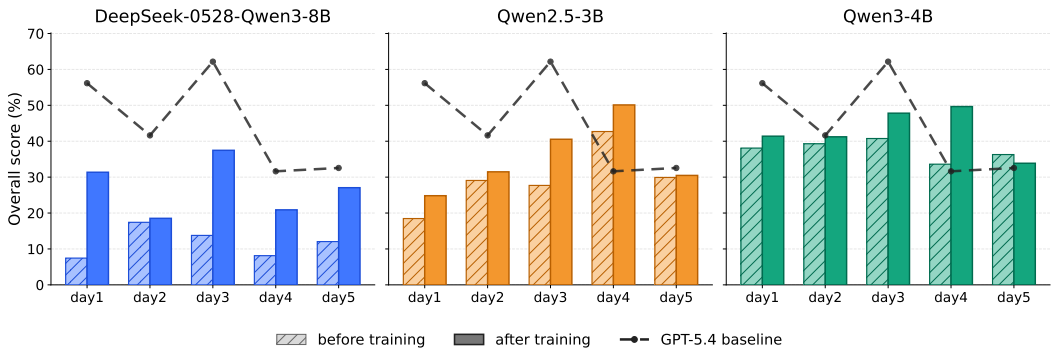


Figure 7: Day-level overall scores on the FutureWorld daily benchmark over five consecutive days. Bars compare each agent before and after FutureWorld RL training, and the dashed line shows the GPT-5.4 web-search baseline.

whether RL improves the ability to gather information and reason about uncertain future events, rather than simply overfitting to the binary prediction questions. We evaluate our trained agents on five consecutive days, and compare their performance with that of the corresponding untrained models. The results are reported in Figure 7. As shown in the figure, all agents achieve performance gains after RL training, and in some cases the trained agents even outperform a strong GPT-5.4 baseline equipped with a web-search plugin. This suggests that RL training in FutureWorld indeed enhances agents' general predictive ability.

5 Conclusion

We introduce FutureWorld, a live environment for training predictive agents with real-world outcome rewards. Unlike prior works that rely on static collections of resolved questions or proxy process-based rewards, FutureWorld keeps the prediction stream live, places agentic search and reasoning inside the training loop, and learns directly from realized future outcomes. We develop `verl-tool-future`, our modified and extended version of `verl-tool`, which decouples prediction-time rollout from later outcome retrieval, reward backfilling, and policy update. Our experiments show that delayed real-world feedback can serve as an effective learning signal, leading to improved prediction performance over successive days. We hope FutureWorld and `verl-tool-future` can serve as useful steps toward agent systems that continually improve by making predictions, observing outcomes, and adapting to an evolving real world.

6 Limitations & future work

The availability of prediction markets and odds aggregation websites such as Polymarket, Kalshi,[5] Metaculus,[6] and Manifold[7] can provide agents with a substantial advantage, as these sites often expose crowd-aggregated beliefs about some future events. This potential source of advantage is not specific to FutureWorld. Rather, it reflects a broader design choice shared by existing agentic prediction systems Su et al. [2026], UniPat AI [2026], which typically do not explicitly exclude prediction markets or odds aggregation websites during information retrieval. To remain consistent with this established practice, our implementation adopts the same setting. A systematic analysis of how publicly available information from prediction markets and odds aggregators affects agents' prediction policy remains an important open problem, and we view it as a direction for future work.

Another limitation concerns longer delayed feedback. In our implementation, when ground truth cannot be successfully retrieved at the scheduled retrieval time, we discard the corresponding questions and exclude them from reward computation. This strategy is simple and practical, but it inevitably wastes computation and potentially valuable supervision signals, because some questions

[5] `https://kalshi.com`
[6] `https://www.metaculus.com/`
[7] `https://manifold.markets/`

may be discarded after one failed retrieval attempt even though their outcomes could become available on later days. Developing more effective mechanisms for incorporating such longer delayed feedback remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This work is supported by the Zhongguancun Academy, (Grant No.s C20250210).

References

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents, 2025. URL https://arxiv.org/abs/2502.01600.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks?, 2024. URL https://arxiv.org/abs/2403.07718.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7903–7910, Apr. 2020. doi: 10.1609/aaai.v34i05.6297. URL https://ojs.aaai.org/index.php/AAAI/article/view/6297.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Scott Jeen, Matthew Aitchison, and Mantic. Training llms to predict world events. *Thinking Machines Lab: News*, 2026. https://thinkingmachines.ai/news/training-llms-to-predict-world-events/.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. Forecastqa: A question answering challenge for event forecasting with temporal text data, 2021. URL https://arxiv.org/abs/2005.00792.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip E. Tetlock. Forecastbench: A dynamic benchmark of ai forecasting capabilities, 2025. URL https://arxiv.org/abs/2409.19839.

Albert King. Akshare. https://github.com/akfamily/akshare, 2022.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jiashuo Liu, Siyuan Chen, Zaiyuan Wang, Zhiyuan Zeng, Jiacheng Guo, Liang Hu, Lingyue Yin, Suozhi Huang, Wenxin Hao, Yang Yang, Zerui Cheng, Zixin Yao, Lingyue Yin, Haoxin Liu, Jiayi Cheng, Yuzhen Li, Zezhong Ma, Bingjie Wang, Bingsen Qiu, Xiao Liu, Zeyang Zhang, Zijian Liu, Jinpeng Wang, Mingren Yin, Tianci He, Yali Liao, Yixiao Tian, Zhenwei Zhu, Anqi Dai, Ge Zhang, Jingkai Liu, Kaiyuan Zhang, Wenlong Wu, Xiang Gao, Xinjie Chen, Zhixin Yao, Zhoufutu Wen, B. Aditya Prakash, Jose Blanchet, Mengdi Wang, Nian Si, and Wenhao Huang. Futurex-pro: Extending future prediction to high-value vertical domains, 2026. URL https://arxiv.org/abs/2601.12259.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. Clin: A continually learning language agent for rapid task adaptation and generalization, 2023. URL https://arxiv.org/abs/2310.10134.

Metaculus. Ai forecasting benchmark series q2 (2025). Metaculus, 2025. URL https://www.metaculus.com/aib/2025/q2/.

Nous Research. Hermes agent. https://github.com/nousresearch/hermes-agent, 2026.

OpenClaw Contributors. Openclaw. https://github.com/openclaw/openclaw, 2026.

Tianrui Qin, Qianben Chen, Sinuo Wang, He Xing, King Zhu, He Zhu, Dingfeng Shi, Xinxin Liu, Ge Zhang, Jiaheng Liu, et al. Flash-searcher: Fast and effective web agents via dag-based parallel execution. *arXiv preprint arXiv:2509.25301*, 2025. URL https://arxiv.org/abs/2509.25301.

Qwen Team. Qwen3-max: Just scale it, September 2025.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Shiqian Su, Sen Xing, Xuan Dong, Muyan Zhong, Bin Wang, Xizhou Zhu, Yuntao Chen, Wenhai Wang, Yue Deng, Pengxiang Zhu, Ziyuan Liu, Tiantong Li, Jiaheng Yu, Zhe Chen, Lidong Bing, and Jifeng Dai. Miroflow: Towards high-performance and robust open-source agent framework for general deep research tasks, 2026. URL https://arxiv.org/abs/2602.22808.

Zhengwei Tao, Zhi Jin, Bincheng Li, Xiaoying Bai, Haiyan Zhao, Chengfeng Dou, Xiancai Chen, Jia Li, Linyu Li, and Chongyang Tao. Prophet: An inferable future forecasting benchmark with causal intervened likelihood estimation, 2025. URL https://arxiv.org/abs/2504.01509.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents, 2024. URL https://arxiv.org/abs/2407.18901.

Benjamin Turtel, Danny Franklin, Kris Skotheim, Luke Hewitt, and Philipp Schoenegger. Outcome-based reinforcement learning to predict the future, 2025. URL https://arxiv.org/abs/2505.17989.

UniPat AI. Echo: Towards general ai prediction, 2026. URL https://unipat.ai/blog/Echo.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader?, 2022. URL https://arxiv.org/abs/2203.07540.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657878. URL https://doi.org/10.1145/3626772.3657878.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, Yixiao Tian, Jinpeng Wang, Zaiyuan Wang, Yang Yang, Lingyue Yin, Mingren Yin, Zhenwei Zhu, Tianle Cai, Zehui Chen, Jiecao Chen, Yantao Du, Xiang Gao, Jiacheng Guo, Liang Hu, Jianpeng Jiao, Xiangsheng Li, Jingkai Liu, Shuang Ni, Zhoufutu Wen, Ge Zhang, Kaiyuan Zhang, Xin Zhou, Jose Blanchet, Xipeng Qiu, Mengdi Wang, and Wenhao Huang. Futurex: An advanced live benchmark for llm agents in future prediction, 2025. URL https://arxiv.org/abs/2508.11987.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023. URL https://arxiv.org/abs/2304.11277.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. Forecasting future world events with neural networks, 2022. URL https://arxiv.org/abs/2206.15474.

A Resampling procedure

We categorize the filtered question-description pairs into domains. Specifically, we use keyword-matching rules to determine which domain each pair belongs to. For each domain, we then select a representative subset while avoiding pairs that are semantically too similar to one another. The selected pairs are retained as the final set.

Let M denote the manually specified target number of pairs to retain after the resampling stage. Given the domain assignment, we first allocate this total budget across non-empty domains. For each domain d, let $\mathcal{P}_d = \{p_1, \ldots, p_{N_d}\}$ denote the set of filtered pairs assigned to this domain, and let M_d denote the target number of pairs to retain from this domain. The values of M_d are chosen to

Table 1: Question types in the FutureWorld daily benchmark.

Question type	Prediction format	Correct answer structure	Daily cap
Binary choice	Select from 2 options	Exactly one option	$\leqslant 5$
Simple multiple choice	Select from 3–4 options	One or multiple options	$\leqslant 10$
Difficult multiple choice	Select from 5–26 options	One or multiple options	$\leqslant 15$
Numeric prediction	Predict a specific value	A numeric value	$\leqslant 20$

make the domain distribution as balanced as possible, subject to the capacity constraint $M_d \leqslant N_d$, and satisfy

$$\sum_d M_d = M,$$

where the summation is taken over all non-empty domains.

We then perform the selection procedure separately for each domain. For each pair $p_i \in \mathcal{P}_d$, we construct a joint textual representation s_i by concatenating the question with its associated description, and embed s_i into a dense vector.[8] Including the description allows the resulting embedding to capture richer semantic information about the target event. We then run K-means clustering over these embeddings with $K = M_d$, and sample one representative pair from each cluster. This yields M_d retained pairs for domain d.

B Implementation details

All experiments are conducted on two NVIDIA A100 GPUs, each with 80 GiB of memory. We implement the delayed-feedback training loop with `verl-tool-future`, our modified `Verl-Tool`-based framework, using FSDP Zhao et al. [2023] as the distributed training backend and vLLM Kwon et al. [2023] for rollout generation. All computation is performed in BF16 precision with FlashAttention-2 Dao [2023] enabled. We use the AdamW optimizer Loshchilov and Hutter [2019] with a learning rate of 1×10^{-6} and weight decay of 0.01. The GRPO mini-batch size is set to 32 and the per-GPU micro-batch size is set to 2. For rollout generation, we sample four stochastic trajectories for each question with temperature 1.0 and top-p $= 1.0$, so nucleus truncation is not applied Holtzman et al. [2020]. The agent is equipped with a lightweight Google-search interface backed by the Serper API.[9] We train three agents initialized from `Qwen3-4B-Instruct-2507` Team [2025], `Qwen2.5-3B-Instruct` Team [2024], Yang et al. [2024], and `DeepSeek-R1-0528-Qwen3-8B` DeepSeek-AI [2025]. We set the number of prediction questions to 500 per day.

C FutureWorld daily benchmark

C.1 Benchmark design

Beyond its role as a learning environment, FutureWorld also supports a live benchmark. Compared with the questions used for RL training, it features both greater diversity in question formats and higher overall difficulty. We limit the benchmark to at most 50 questions per day. The FutureWorld daily benchmark contains four question types, summarized in Table 1. Table 2 provides representative examples. The prompt templates used for these four question types are provided in Appendix D. Each day, the benchmark is refreshed with a new batch of prediction questions whose answers are expected to be revealed on the following day. We retrieve the ground-truth outcomes for the batch of questions released two days earlier. This design increases the fraction of questions that can be resolved.

[8] We encode each s_i into a dense semantic representation using a pretrained embedding model, `BAAI/bge-small-en-v1.5` Xiao et al. [2024].

[9] `https://serper.dev`

Table 2: Examples of the four question types in the FutureWorld daily benchmark.

Question type	Example
Binary choice	Will Trump visit North Korea by April 30? A. Yes B. No
Simple multiple choice	Shimizu S-Pulse vs. V-Varen Nagasaki A. Shimizu S-Pulse B. Draw (Shimizu S-Pulse vs. V-Varen Nagasaki) C. V-Varen Nagasaki
Difficult multiple choice	What will Google say during their next earnings call? A. Banana B. Translate / Translation C. Autonomous / Autonomously D. Dividend E. CAPTCHA / reCAPTCHA F. Flash / Flash-Lite G. Lens H. Google Maps I. NVIDIA J. Maps K. Circle to Search L. Gemini Live M. Token N. Anthropic O. Pixel 10 / Pixel 10a P. Alphabet Q. YouTube R. TikTok S. Cutting-edge T. ChatGPT / OpenAI U. Advertising / Advertisement V. Find the Look / Virtual Try-On
Numeric prediction	On 2026-04-29 (UTC+8), what will Internal Three-Way Crossbred Hog hog price be, in CNY per kilogram?

C.2 Scoring rules

We use type-specific scoring rules for different question formats. Only resolved questions with valid ground-truth answers are scored. If a question remains unresolved, it is excluded from scoring. As a result, the reported metrics reflect performance only on questions with confirmed outcomes.

C.2.1 Choice-question scoring

Suppose a choice-question has m options. We denote the gold vector by $\mathbf{y} \in \{0,1\}^m$ and the prediction vector by $\hat{\mathbf{y}} \in \{0,1\}^m$, where

$$y_j = \begin{cases} 1, & \text{if option } j \text{ is correct,} \\ 0, & \text{if option } j \text{ is wrong,} \end{cases}$$
$$\hat{y}_j = \begin{cases} 1, & \text{if option } j \text{ is selected by the agent,} \\ 0, & \text{if option } j \text{ is not selected by the agent.} \end{cases} \tag{4}$$

The question-level score is then computed as the option-level F1 score

$$S_{\mathrm{F1}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2\,\mathbf{y}^\top \hat{\mathbf{y}}}{\|\mathbf{y}\|_1 + \|\hat{\mathbf{y}}\|_1}. \tag{5}$$

Table 3: Average performance of several frontier agents on the FutureWorld daily benchmark over four consecutive days. The best results are **bold-typed** and the second best ones are underlined.

Agents	S_{bin}	S_{smc}	S_{dmc}	S_{num}	S_{overall}
`x-ai/grok-4.20`	70.00	36.70	6.33	15.24	32.07
`z-ai/glm-5.1`	71.25	45.71	2.79	15.24	33.75
`openai/gpt-5.4`	**81.25**	40.31	**20.78**	5.17	36.88
`anthropic/claude-opus-4.6`	63.75	47.14	20.28	16.84	37.00
`google/gemini-3.1-pro-preview`	**81.25**	42.77	7.95	**18.84**	37.70
`qwen/qwen3-max-thinking`	**81.25**	**47.38**	15.73	11.70	**39.01**

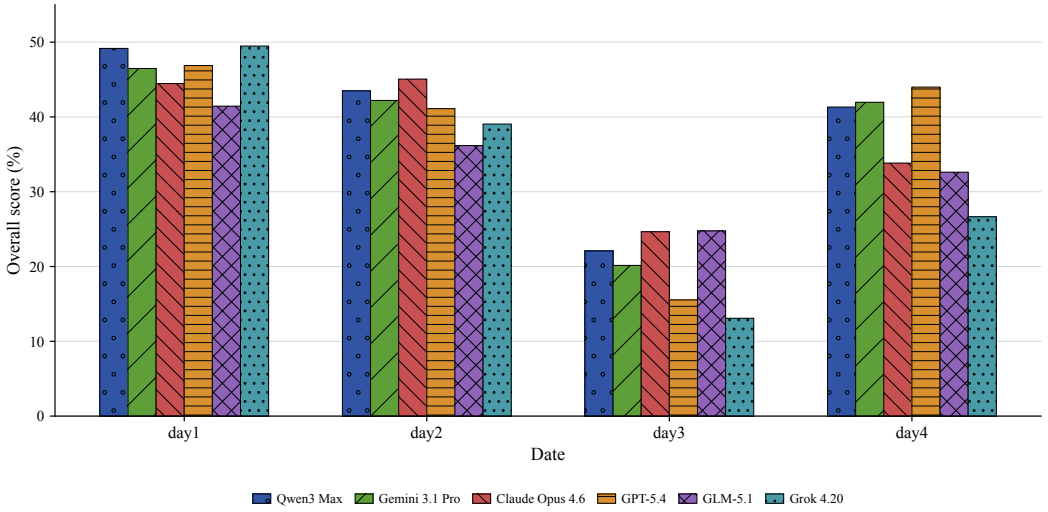


Figure 8: Overall scores of agents on the FutureWorld daily benchmark over 4 consecutive days.

For binary choice questions, the prediction must contain exactly one selected option. Any prediction that selects more than one option is treated as invalid and assigned an F1 score of 0.

C.2.2 Numeric-question scoring

For numeric prediction questions, we evaluate the agent relative to the recent variability of the prediction target. Let $\hat{v}$ denote the value predicted by the agent. Let

$$\mathcal{V} = \{v_1, v_2, \ldots, v_8\} \tag{6}$$

denote eight consecutive historical values associated with the prediction target, where $v_8 = v$ is the resolved true value on the target date. We define the score as

$$S_{\text{num}} = \max\left(0,\ 1 - \left(\frac{\hat{v} - v}{3\sigma(\mathcal{V}) + \varepsilon}\right)^2\right), \tag{7}$$

where $\sigma(\mathcal{V})$ denotes the sample standard deviation of $\mathcal{V}$, and ε is a small number introduced for numerical stability. This score is bounded in $[0, 1]$ and drops as the prediction value moves farther from the true value, but it drops more slowly for targets that fluctuate more.

C.2.3 Overall scoring

After computing question-level scores, we average them separately within each question type. Let S_{bin}, S_{smc}, S_{dmc}, and S_{num} denote the mean scores for binary choice, simple multiple choice, difficult multiple choice, and numeric prediction questions, respectively. The final overall score is computed by assigning equal weight to the four question types:

$$S_{\text{overall}} = \frac{1}{4}\left(S_{\text{bin}} + S_{\text{smc}} + S_{\text{dmc}} + S_{\text{num}}\right). \tag{8}$$

Table 4: Daily scores of agent frameworks on the FutureWorld daily benchmark, where all frameworks use GPT-5.4 as the base model. The best score is **bold-typed**.

Date	Method	S_{bin}	S_{smc}	S_{dmc}	S_{num}	S_{overall}
2026-04-25	smolagents	**75.00**	**71.43**	39.17	62.18	61.95
	Flash-Searcher	**75.00**	57.14	**60.61**	**62.35**	**63.78**
2026-04-26	smolagents	**100.00**	**44.44**	29.80	**54.17**	57.10
	Flash-Searcher	**100.00**	**44.44**	**53.94**	47.52	**61.48**
	OpenClaw	50.00	**44.44**	49.35	12.79	39.15
2026-04-27	smolagents	**100.00**	57.14	49.26	43.30	62.43
	Flash-Searcher	80.00	85.71	35.86	**79.59**	70.29
	OpenClaw	**100.00**	**100.00**	**71.11**	52.68	**80.95**
	Hermes	**100.00**	71.43	38.03	45.63	63.77
2026-04-28	smolagents	80.00	**36.67**	41.33	39.83	49.46
	Flash-Searcher	60.00	**36.67**	44.80	66.33	51.95
	OpenClaw	**100.00**	25.00	28.71	19.78	43.37
	Hermes	**100.00**	**36.67**	**58.93**	**67.02**	**65.65**
2026-04-30	smolagents	**100.00**	**60.00**	48.21	53.22	65.36
	Flash-Searcher	**100.00**	40.00	45.22	–	61.74
	OpenClaw	**100.00**	0.00	41.05	17.74	39.70
	Hermes	**100.00**	40.00	**50.36**	**77.45**	**66.95**
2026-05-02	Flash-Searcher	**50.00**	**55.56**	63.90	52.34	**55.45**
	OpenClaw	**50.00**	44.44	43.25	27.23	41.23
	Hermes	25.00	44.44	**67.55**	**65.15**	50.54
	GPT-5.4 (web search)	**50.00**	44.44	18.51	41.99	38.74
2026-05-03	smolagents	**100.00**	**44.44**	**52.80**	48.62	61.47
	Flash-Searcher	**100.00**	**44.44**	43.08	54.00	60.38
	OpenClaw	**100.00**	**44.44**	36.92	24.67	51.51
	Hermes	**100.00**	**44.44**	51.83	**72.17**	**67.11**
	GPT-5.4 (web search)	75.00	**44.44**	19.64	50.64	47.43
2026-05-04	smolagents	**100.00**	**62.50**	50.95	43.16	64.15
	Flash-Searcher	**100.00**	**62.50**	**69.05**	**59.00**	**72.64**
	Hermes	60.00	50.00	21.94	23.30	38.81
	GPT-5.4 (web search)	40.00	50.00	4.76	19.82	28.64
2026-05-05	smolagents	20.00	14.29	36.11	45.31	28.93
	Flash-Searcher	**60.00**	7.14	**48.61**	**60.60**	**44.09**
	OpenClaw	**60.00**	**21.43**	43.19	19.36	36.00
	Hermes	**60.00**	**21.43**	42.22	31.97	38.90
	GPT-5.4 (web search)	40.00	0.00	12.50	34.37	21.72

C.3 Evaluation of frontier agents

We evaluate several frontier agents[10] Qwen Team [2025] with live web search capabilities on four consecutive days, and report the four-day average results in Table 3. On average, `qwen/qwen3-max-thinking` achieves the best overall performance. Figure 8 further shows the daily overall score achieved by each agent over the four consecutive days.

C.4 Evaluation of open-source agent frameworks

We additionally evaluate agents built with several open-source agent frameworks on the FutureWorld daily benchmark. The evaluated frameworks cover different agent-system design choices.

[10] `https://docs.z.ai/guides/llm/glm-5.1`, `https://grok.com/`
`https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`
`https://www.anthropic.com/news/claude-opus-4-6`

smolagents is a lightweight Hugging Face library[11] that exposes a compact interface for building tool-using agents and connecting language models with external tools and code execution Roucher et al. [2025]. Flash-Searcher organizes web search and evidence reading as a DAG-based parallel execution process, aiming to reduce serial search latency and improve evidence coverage Qin et al. [2025]. OpenClaw is a local personal-assistant framework with a unified execution gateway, multi-agent routing, and skill-extension mechanisms OpenClaw Contributors [2026]. Hermes emphasizes long-term memory and self-improvement, including cross-session retrieval, summary, user modeling, skill generation, and parallel sub-agent execution Nous Research [2026]. Table 4 summarizes the performance of different agent frameworks, where all frameworks use GPT-5.4 as the base model.

D Prompt templates for FutureWorld daily benchmark

The following templates are used to construct prompts for the four question formats in the FutureWorld daily benchmark. In each template, <QUESTION> is replaced by the concrete prediction question.

D.1 Binary choice

```
You are an agent that can predict future events. The event to be predicted:
"""
<QUESTION>
"""

Your goal is to identify the single correct option based on your analysis.

NOTE: If you notice a potential time conflict in the question, do not worry-this may be
due to different time zones, and it can still refer to the same moment in time.

IMPORTANT: Your final answer MUST end with this exact format:
\boxed{A} or \boxed{B}

Do not use any other format. Do not refuse to make a prediction. Do not say "I cannot
predict the future." You must make a clear prediction based on the best data currently
available, using the box format specified above.
```

D.2 Simple multiple choice

```
You are an agent that can predict future events. The event to be predicted:
"""
<QUESTION>
"""

Your goal is to identify all the correct option(s) based on your analysis. There may be
exactly one correct option or multiple correct options; you must determine the correct
set.

Please list all correct option(s) you have identified, separated by commas. If you
believe there is only one correct option, output that option alone (without any commas).

Output only the option label(s). Do not output the full option text. If you output the
option text in addition to the option label(s), your answer will be marked incorrect even
 if you chose the correct option(s).

NOTE: If you notice a potential time conflict in the question, do not worry-this may be
due to different time zones, and it can still refer to the same moment in time.

IMPORTANT: Your final answer MUST end with this exact format:
\boxed{label 1} or \boxed{label 2, label 3, ...}

For example: \boxed{A} for a single correct option, or \boxed{B, C} for multiple correct
options.

Do not use any other format. Do not refuse to make a prediction. Do not say "I cannot
predict the future." You must make a clear prediction based on the best data currently
available, using the box format specified above.
```

[11] https://huggingface.co

D.3 Difficult multiple choice

```
You are an agent that can predict future events. The event to be predicted:
"""
<QUESTION>
"""

Your goal is to identify all the correct option(s) based on your analysis. There may be
exactly one correct option or multiple correct options; you must determine the correct
set.

Please list all correct option(s) you have identified, separated by commas. If you
believe there is only one correct option, output that option alone (without any commas).

Output only the option label(s). Do not output the full option text. If you output the
option text in addition to the option label(s), your answer will be marked incorrect even
 if you chose the correct option(s).

NOTE: If you notice a potential time conflict in the question, do not worry-this may be
due to different time zones, and it can still refer to the same moment in time.

IMPORTANT: Your final answer MUST end with this exact format:
\boxed{label 1} or \boxed{label 2, label 3, ...}

For example: \boxed{A} for a single correct option, or \boxed{B, C} for multiple correct
options.

Do not use any other format. Do not refuse to make a prediction. Do not say "I cannot
predict the future." You must make a clear prediction based on the best data currently
available, using the box format specified above.
```

D.4 Numeric prediction

```
You are an agent that can predict future events. The event to be predicted:
"""
<QUESTION>
"""

Your goal is to make a numeric prediction.

NOTE: If you notice a potential time conflict in the question, do not worry-this may be
due to different time zones, and it can still refer to the same moment in time.

IMPORTANT: Your final answer MUST end with this exact format:
\boxed{number}

Do not use any other format. Do not refuse to make a prediction. Do not say "I cannot
predict the future." You must make a clear prediction based on the best data currently
available, using the box format specified above.

Your output must be a plain numeric value only. Do NOT include any units, currency
symbols, percent signs, commas, spaces, scientific notation (e.g., 1e6), or any other
extra characters. Only digits are allowed, with an optional leading minus sign and an
optional decimal point (e.g., 123, -45, 67.89).
```

E Broader impacts

FutureWorld may have positive societal impacts by supporting the development of predictive agents that learn from real-world feedback and adapt to changing conditions. Such agents may help improve decision support in domains where anticipating future events is useful, such as public-interest monitoring, scientific analysis, operations planning, and risk assessment.

At the same time, stronger predictive agents may also create risks if used inappropriately. For example, they could be used to support speculative trading, political targeting, or other decisions that over-rely on uncertain predictions. Incorrect predictions may also mislead users if their uncertainty is not properly communicated. To mitigate these risks, FutureWorld focuses on probabilistic prediction rather than deterministic claims, filters out sensitive or otherwise inappropriate questions, and discusses possible advantages from prediction-market and odds-aggregation websites. We also apply safeguards for responsible release. FutureWorld only uses publicly accessible sources and filters out sensitive, harmful, or otherwise inappropriate questions.

Table 5: Assets used in this work.

Asset	License / terms	Usage
`verl-tool`	MIT License	Used as the base tool-agent RL framework and modified into `verl-tool-future`.
PyTorch FSDP	BSD-style / BSD-3-Clause	Used as the distributed training backend.
vLLM	Apache-2.0 License	Used for efficient rollout generation.
FlashAttention-2	BSD-3-Clause License	Used to accelerate attention computation during training and rollout.
`Qwen3-4B-Instruct-2507`	Apache-2.0 License	Used as one of the open-source base models for training predictive agents.
`Qwen2.5-3B-Instruct`	Qwen Research License	Used as one of the open-source base models for training predictive agents.
`DeepSeek-R1-0528-Qwen3-8B`	MIT License	Used as one of the open-source base models for training predictive agents.
`BAAI/bge-small-en-v1.5`	MIT License	Used to encode question-description pairs for semantic resampling.
Serper API	Serper API terms of service	Used to provide the Google-search interface during agent rollouts.
`AkShare`	MIT License	Used to retrieve part of the outcome data.
Public source websites	Source-specific access conditions	Used to collect candidate future events from publicly accessible sources.

F Existing assets, licenses, and terms of use

We use several existing assets, including open-source code packages, pretrained models, APIs, and public online data sources. We cite the original creators where applicable and follow the corresponding licenses or terms of use. Table 5 summarizes the main existing assets used in this work.