Future-as-Label: Scalable Supervision from Real-World Outcomes

Anonymous Authors[1]

Abstract

Time creates free supervision: forecasts about real-world events eventually resolve to verifiable outcomes. We use this temporal structure to train language models for probabilistic forecasting from causally masked information, assigning rewards only after events resolve. Our approach applies reinforcement learning with proper scoring rules to outcome-based supervision, using rank-32 LoRA fine-tuning of gpt-oss-120b. On a held-out synthetic future-events benchmark, the trained model improves over strong frontier and open-weight baselines, achieving the best log score, Brier score, and calibration error. These results suggest that real-world outcome resolution can provide scalable supervision for improving forecasting accuracy and calibration without human annotation.

1. Introduction

Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective approach for improving language models in domains such as mathematics, code generation, and formal reasoning, where correctness can be checked automatically. By replacing human annotation with deterministic reward functions, these methods scale efficiently and yield strong empirical gains. However, existing RLVR settings typically depend on immediate, closed-form verification, limiting them to tasks where correctness can be resolved at training time.

Many real-world forecasting problems have a different but equally useful source of verification: time. Forecasts about elections, court cases, markets, scientific milestones, or corporate events eventually resolve to objective outcomes. This temporal structure creates a natural information asymmetry: at prediction time, the model observes only information available up to a cutoff time t; after resolution, the outcome can be evaluated using information revealed by a later time $s > t$. Thus, the passage of time provides labels without requiring human annotation.

We use this structure to train language models from real-world outcome resolution. The model makes probabilistic forecasts from causally masked inputs, and once events resolve, receives rewards from a proper scoring rule. Unlike supervised fine-tuning, which fits fixed target outputs, our approach optimizes sampled reasoning trajectories using only retrospective outcome-based rewards. This emphasizes calibrated decision-making under uncertainty rather than imitation of labeled answers.

This setting connects RLVR to LLM-based forecasting. Prior work has used prompting, retrieval, ensembling, and supervised fine-tuning over historical forecasting questions (Halawi et al., 2024). Other work uses language models as judges or evaluators when objective rewards are unavailable (Liu et al., 2025b). In contrast, our supervision comes from externally resolved outcomes: the resolver may use post-cutoff evidence unavailable to the predictor, but it does not score model reasoning or provide endogenous preference judgments. The reward is grounded in new evidence and outcomes revealed over time.

We make three contributions. First, we formulate temporally resolved real-world events as a source of verifiable and scalable reward for probabilistic forecasting. Second, we introduce an annotation-free training procedure that uses outcome-based rewards under a strict causal information constraint. Third, we show that this approach improves forecasting accuracy and calibration over strong pretrained baselines, including frontier and open-weight models, on held-out future-event prediction benchmarks.

2. Method

We study future-event prediction under a causal information constraint. At prediction time t, the model observes only information available up to that cutoff and predicts whether an event will occur by a later resolution time $s > t$. Although training is performed offline on events whose outcomes are already known, all inputs are temporally masked to exclude post-t information. Rewards are computed only from the realized outcome at resolution time.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Each episode consists of a single binary forecasting question. The **predictor** is the language model being trained. Given a temporally masked information state, it samples a reasoning trajectory that terminates in a probabilistic forecast $p \in (0, 1)$. The **resolver** is a fixed external process that determines the realized outcome $y \in \{0, 1\}$ using post-t evidence unavailable to the predictor. The resolver is used only to determine whether the event occurred; it does not observe model outputs, score reasoning quality, or provide preference judgments. Thus, supervision comes from the information gap between the predictor's pre-t view and the resolver's post-t view.

Once the event resolves, we assign a terminal reward using the log score:

$$R(p, y) = y \log(p) + (1 - y) \log(1 - p).$$

This strictly proper scoring rule incentivizes calibrated probabilistic predictions and provides a continuous learning signal under uncertainty. There are no intermediate rewards.

We optimize the policy using Group Relative Policy Optimization (GRPO) (Liu et al., 2025a). For each state, the model samples a group of K reasoning trajectories, each producing a forecast. After resolution, we compute a reward for each trajectory and subtract the group mean reward to obtain a relative advantage:

$$A(\tau_i) = R(\tau_i) - \frac{1}{K} \sum_j R(\tau_j).$$

Policy updates increase the likelihood of trajectories with positive relative advantage. This compares alternative forecasts generated under the same pre-t information, reducing variance from outcome noise while assigning credit across sampled reasoning trajectories.

We enforce causal masking throughout training. The predictor receives only timestamped information available at or before t, while outcome verification uses independent post-t sources. All training and evaluation events resolve after the base model's reported knowledge cutoff, reducing the risk that outcomes are encoded in parametric memory. In our experiments, we fine-tune gpt-oss-120b using rank-32 LoRA adapters, making the outcome-based training procedure parameter-efficient.

3. Experimental Setup

All experiments are conducted offline using historical data. For each example, predictions are backdated to a cutoff time t, while the realized outcome is withheld from the model and used only to compute reward after resolution.

Dataset. We construct a binary future-event prediction dataset designed to preserve strict temporal separation between prediction and verification. The pre-cutoff information state consists of an English-language news corpus aggregated from publicly accessible sources, including international newspapers, wire services, and financial news outlets. Articles are timestamped using publisher-provided publication times and normalized to UTC.

For each example, we freeze the corpus at cutoff time t and generate a natural-language question about an event expected to resolve strictly after that cutoff. Generated events span multiple domains, including politics, economics, technology, and corporate actions. Model inputs are constructed exclusively from sources published at or before t; articles with missing or ambiguous timestamps are excluded to reduce temporal leakage.

Outcome verification uses independent post-t sources that are not included in the model's input context. A separate frozen resolver has access to post-cutoff news and archival sources and is used solely to determine whether the event occurred. The resolver does not observe model outputs or training dynamics, so resolution errors may introduce noise but do not create endogenous reward signals. Examples that cannot be resolved with high confidence are discarded, and each retained example is assigned a resolution time s based on the earliest dated source supporting the resolved outcome.

The dataset contains 8,049 binary prediction examples, with 7,147 used for training and 903 held out for evaluation under a temporally disjoint split. Training examples cover cutoffs from July 2024 through November 2025, while test examples use cutoffs on or after November 2025. This ensures that evaluation occurs strictly after the training period. Prediction horizons range from one day to six months, covering events that resolve over days to weeks.

Training. We fine-tune **gpt-oss-120b** using rank-32 LoRA adapters to produce probabilistic forecasts from causally masked inputs. Each generation terminates in a scalar probability $p \in (0, 1)$, which is scored against the realized outcome using the log score. Training uses GRPO with groups of four sampled trajectories per event. Additional optimization details are provided in Appendix A.

Baselines. We compare against strong prompted forecasting baselines evaluated under the same temporal information constraint. These include **gpt-oss-120b** and **GPT-5**, which are prompted to produce probabilistic predictions without task-specific fine-tuning. The trained model is evaluated at **step 150**. All models receive the same pre-t information and produce forecasts in the same output format.

Metrics. We evaluate probabilistic forecasts using four metrics: log score, Brier score, Brier skill score, and ex-

Table 1. Forecasting performance on synthetic benchmark

Model	Log ↑	Brier ↓	ECE ↓
Synthetic future-events			
gpt-oss-120b	-0.5740	0.1924	0.0843
gpt5	-0.5455	0.1733	0.0787
gpt-oss-120b-RL (150)	**-0.5266**	**0.1700**	**0.0321**

pected calibration error (ECE). Log score is the training reward; Brier score measures squared error between predicted probabilities and binary outcomes; Brier skill score measures relative improvement in Brier score over a naive baseline that always predicts the mean outcome rate in the training sample; and ECE measures calibration by comparing predicted confidence to empirical outcome frequencies across 10 probability bins.

4. Results

We evaluate on a synthetic future-event benchmark constructed under strict temporal controls. We report log score, Brier score, and expected calibration error (ECE).

Table 1 compares three inference regimes: prompted **gpt-oss-120b**, prompted **GPT-5**, and **gpt-oss-120b** fine-tuned with rank-32 LoRA adapters using resolved-outcome rewards. On the synthetic future-events benchmark, the trained model achieves the best score on all three metrics. Relative to prompted gpt-oss-120b, it improves log score from -0.5740 to -0.5266, Brier score from 0.1924 to 0.1700, and ECE from 0.0843 to 0.0321. It also outperforms prompted GPT-5 on log score, Brier score, and ECE, despite using the same gpt-oss-120b base model as the weaker prompted baseline.

The largest relative gain is in calibration: ECE falls by more than half compared with both prompted baselines. This indicates that training on resolved outcomes improves not only aggregate forecast accuracy, but also the reliability of the model's probability estimates.

Overall, the results support the central premise of our approach: real-world outcome resolution can serve as scalable supervision for probabilistic forecasting. Rank-32 LoRA fine-tuning on temporally resolved events yields improvements over both the underlying prompted model and a stronger frontier baseline.

5. Discussion and Conclusion

This work studies a scalable supervision regime in which feedback is provided by the eventual resolution of real-world events. By training probabilistic forecasts against realized outcomes using proper scoring rules, our approach aligns optimization with the temporal structure of forecasting: mod-

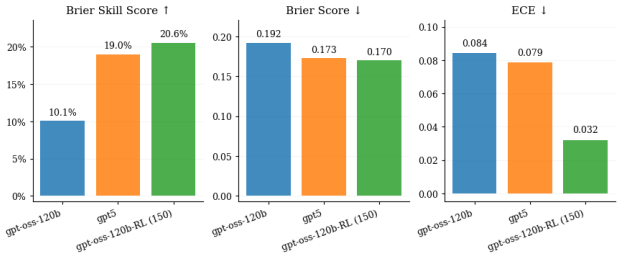

Figure 1. Model calibration and accuracy metrics

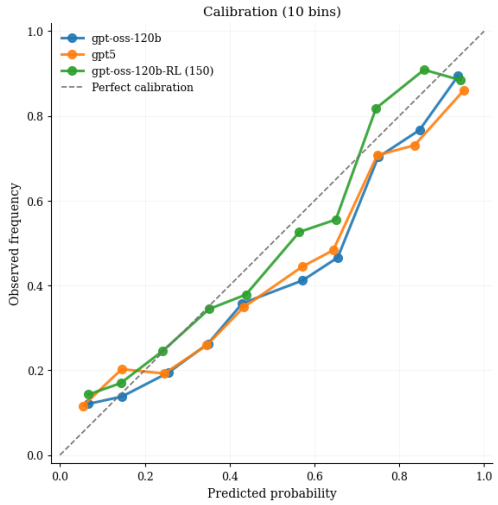

Figure 2. Calibration curves over 10 confidence bins.

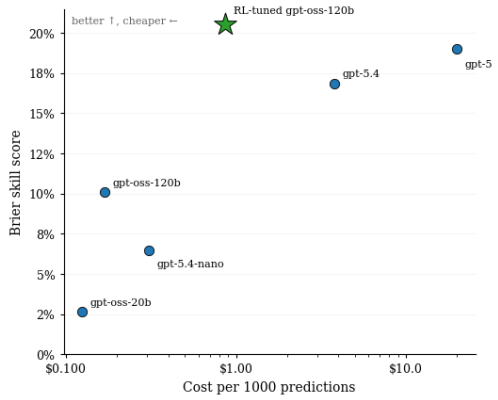

Figure 3. Cost–performance tradeoff across models on our test set. Each point shows a model's cost per 1000 predictions versus its Brier skill score. Our model achieves the best Brier skill score while remaining substantially cheaper than GPT-5 and GPT-5.4.

els act under a causal information constraint, and feedback arrives only after the world reveals the answer.

Empirically, rank-32 LoRA fine-tuning of gpt-oss-120b improves forecasting performance over strong prompted baselines. On the synthetic future-events benchmark, the trained model achieves better log score, Brier score, and expected calibration error than both prompted gpt-oss-120b and prompted GPT-5.

A central benefit is improved calibration. Because rewards are assigned only after outcomes resolve, overconfident incorrect predictions incur large penalties, while appropriately uncertain predictions are penalized less severely. This encourages probability estimates that better track empirical event frequencies, rather than merely producing plausible forecast rationales.

Relative to conventional RLVR settings, this approach operates in open-world domains with sparse and delayed feedback. Group-relative optimization helps stabilize learning by comparing alternative trajectories generated under identical pre-resolution information and rewarding those that assign better probabilities to realized outcomes.

This study has several limitations. Training is performed offline on resolved events rather than in a fully online deployment loop. Event generation and outcome resolution rely on automated pipelines that may introduce biases, coverage gaps, or resolution errors. The current experiments also focus on binary outcomes; extending the framework to continuous, multi-class, and richer structured outcomes remains an important direction for future work.

Overall, the results suggest that chronologically evolving real-world data can provide scalable supervision for forecasting models. More broadly, outcome resolution offers a path for extending verifiable reward-driven learning beyond closed-world tasks and toward open-ended real-world prediction under uncertainty.

References

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.

Liu, S., Liu, H., Liu, J., Xiao, L., Gao, S., Lyu, C., Gu, Y., Zhang, W., Wong, D. F., Zhang, S., and Chen, K. Compassverifier: A unified and robust verifier for llms evaluation and outcome reward, 2025a. URL https://arxiv.org/abs/2508.03686.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective, 2025b. URL https://arxiv.org/abs/2503.20783.

A. Hyperparameters

Hyperparameter	Value
Optimization algorithm	Dr GRPO (Liu et al., 2025a)
Policy optimizer	AdamW
Learning rate	4e-5
Batch size (events)	32
Trajectories per event	4
Reward	Log score
Reward timing	Terminal only
Gradient application	All tokens in sampled trajectories
Probability range	$[0.001, 0.999]$
Training regime	RL with temporally masked inputs

Table 2. Training and optimization hyperparameters