For example, prediction markets are dominated by binary (yes/no) or multiple choice questions. While this design is easy to score, the most foresight often lies in predicting the unexpected, or when a large number of possibilities could occur. The most important questions to forecast—such as scientific breakthroughs, geopolitical shocks, or technological disruptions—often emerge as *unknown unknowns*: possibilities not anticipated, and hard to enumerate. Thus, in this work, we focus on training models to make open-ended predictions like "Which company will the US Government buy a >5% stake in by September 2025?". Such questions require exploration and imagination, rewarding novel hypotheses that turn out to be correct, rather than just distributing probabilities over a known set of outcomes.

Background. LLM weights are frozen after training, especially for open-weight models. Any event that happens after the last date in the training corpus is in the future for the LLM. This provides a time window to collect questions for training models to reason about future events. Similarly, their evaluation involves testing on questions resolving after the cutoff date of the training data, called *backtesting* (Tashman, 2000). While prior work has relied on prediction market questions as training data, this has three key problems:

1. The questions are created by humans, which makes them low in number (Paleka et al., 2025a). This becomes a bottleneck for scaling training data, which has been an essential component in the success of LLMs (Kaplan et al., 2020; Lu, 2025).
2. Most questions have binary outcomes, which creates a 50% baseline success rate. This leads to noisy rewards in outcome-based RL, which means even incorrect reasoning has a high chance of being reinforced.
3. Each platform has a skewed distribution of events. All overrepresent US political news, along with their specific focus such as crypto-currency price movements in Polymarket, technology in Metaculus, personal life of users in Manifold, and sports events on Kalshi (Paleka et al., 2025a).

These limitations motivate us to explore alternate ways to create forecasting questions.

Setup. Let $\mathcal{X}$ be the set of open-ended forecasting questions; and $\mathcal{Y}$ the set of short textual answers. We provide a language model π_θ a question $x \in \mathcal{X}$, for which we already know the ground-truth outcome $y^\star$ as it has resolved in the real-world. We ask the model to report its best guess prediction y, and the probability q of it being the true outcome.

Measuring Accuracy. We measure accuracy by checking if the model's attempted answer y matches with the ground truth outcome $y^\star$, using another language model to test for semantic equivalence (for example "Geoffrey Hinton" = "Geoffrey Everest Hinton") consistent with recent frontier benchmarks (Wei et al., 2024; Phan et al., 2025). For evaluations, we use Llama-4-Scout (Meta AI, 2025), as in a recent study (Chandak et al., 2025), it at matching answers, it has inter-human levels of alignment with human judgments. During training, we use Qwen3-4B in non-thinking mode, as it achieves high alignment levels for its size in the same study. We find the two models agree on $\sim 97\%$ grading responses, and manual validation ensures they are accurate in $\geq 95\%$ cases.

Measuring Calibration. We adapt the multi-class Brier scoring rule (Mucsányi et al., 2023) for free-form responses as follows (details in Section A):

$$S'(q, y, y^*) = \begin{cases} 1 - (q-1)^2, & \text{if } y \equiv y^* \\ -q^2, & \text{if } y \neq y^* \end{cases}$$

Interpretation. Predicting an event with a probability $q = 0$ returns a baseline score of 0 regardless of the prediction y. Correct predictions receive positive scores while incorrect predictions negative. For brevity, we call $S'(q, y, y^*)$ *Brier score* throughout this paper. Our Brier score is equivalent to the reward metric used by Damani et al. (2025). They show this is a proper scoring rule, incentivizing both high accuracy and truthful reporting of probability on the answer that seems most likely. For completeness, we discuss this further in Section A.

Training Algorithm: GRPO (Shao et al., 2024). We train LLMs using outcome-based reinforcement learning on our dataset. For each prompt x, we draw K completions

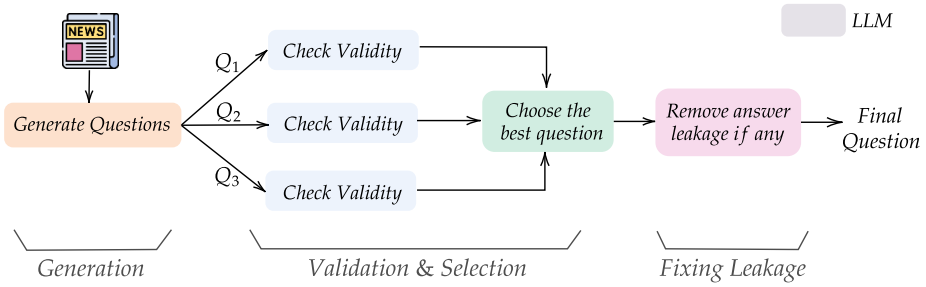

Figure 2: **Our question generation methodology.** We use DeepSeek-v3 to generate multiple forecasting questions per news article. Then, we use Llama-4-Maverick to check if questions follow all guidelines, choose the best question, and remove any hints revealing the answer.

$\{(y_i, p_i)\}_{i=1}^{K} \sim \pi_\theta(\cdot \mid x)$ and compute rewards $r_i = R(y_i, p_i; y^\star)$. However, following prior work (Damani et al., 2025; Turtel et al., 2025b), we do not divide by *remove* the group standard deviation when computing advantages, as this stabilizes updates in settings like ours where reward variance can be small.

Initial Policy: Qwen3 Thinking (Yang et al., 2025). We start with the 8B thinking model. For Qwen3 models, no official knowledge–cutoff date is reported. When queried directly, the models return inconsistent cutoff dates (most often *October 2023* or *June 2024*). Usually, they treat questions about 2024 as being in the future. Since the model weights were released and frozen in April 2025, we train up to this date, and use the period between May to August 2025 for testing. In the Appendix, we show large improvements from our training on even the Llama and Gemma models in Section B.2.

4 Generating Open-Ended Forecasting Questions from News

We now discuss our methodology to convert daily news articles into forecasting questions using language models. Any fixed forecasting dataset loses value as newer base models with training cutoffs after the dataset was created are adopted. Thus, we first describe the general methodology which can be used in the future, and then describe the specific instantiations we used to create our training data `OpenForesight`which has questions until April 2025. We conclude by demonstrating forecasting improvements due to our data filtering steps.

4.1 Methodology for Generating Forecasting Questions

We generate short-answer, open-ended forecasting questions from individual news articles as illustrated in Figure 2. We describe each step in detail below:

Sourcing Event Information. News outlets establish global infrastructure for reporting salient events as they occur. Unfortunately, Paleka et al. (2025a) show that sourcing news via online search engines is unreliable. While search engines provide date cutoffs, future information can leak through updates to articles after the publish date, and even search engine ranking. This compromises the reliablity of backtests, and leaks future information in training, which can hurt Deep Learning models that easily overfit to spurious correlations. Fortunately, the CommonCrawl News (CCNews) Corpus (Nagel, 2016) provides static monthly snapshots of global news with accurate dates. This makes it free and easy to obtain news articles for creating forecasting questions.

Generating samples from documents. Based on each news article, we ask a *sample creator* model to generate up to three diverse forecasting samples. Each sample consists of:

1. **Question:** Asks about the prediction of an event.
2. **Background:** Provides brief context, and defines uncommon terms.
3. **Resolution criteria:** Fixes a source of truth, proposes a resolution date for the question, and the expected answer format.

4. **Answer:** Drawn verbatim from the article, unique, short (usually 1–3 words), and non-numeric (usually a name or location).
5. **Source article link:** Obtained from article metadata for future reference.

Sample Generated Forecasting Question

Question. Who will be confirmed as the new prime minister of Ukraine by 17 July 2025?

Background. Ukraine's parliament is scheduled to vote to appoint a new prime minister.

Resolution Criteria.

- **Source of Truth:** Official announcement from the Verkhovna Rada (Ukraine's parliament) confirming the appointment, via parliamentary records or government press release.
- **Resolution Date:** 17 July 2025, the date on which the parliamentary vote occurs and results are published.
- **Accepted Answer Format:** Full name of the individual exactly as given in the parliamentary announcement.

Answer Type. String (Name)

Ground-Truth Answer. Yulia Svyrydenko

Source. The Guardian (live blog): Ukraine live updates — 17 July 2025

A challenging issue we face is that sometimes news articles talk about past events, or report an event late. This is why we ask the sample creator to propose a resolution date, and set the final resolution date as `min(model_generated_date, article_publish_date)`. We perform additional steps, including manual review, to address this issue for evaluation questions, as described later in Section 6. For training data, we do not add more complex steps to fix resolution dates due to cost constraints.

Filtering samples. For each question, we use another LLM, *the sample selector*, to verify:

1. The question-answer pair is fully based on information in the source article.
2. The question is forward-looking, for e.g. it is in future tense
3. The answer is definite, unambiguous, and resolvable by the publication date.

We mark a question as valid only if it passes these checks. If multiple questions from a single article remain, we ask the sample selector to pick the best one, favoring questions with clear, unique answers and high relevance. This is to ensure data diversity, and enhance quality.

Editing to fix leakage. At this stage, we find that even the filtered samples sometimes leak information about the answer. This can create shortcuts during training. To fix this, we do a final editing stage where we ask the sample selector to scan the title, background, and resolution criteria to check if they reveal the answer. When it finds leakage, we ask it to rewrite only the offending spans, replacing specifics with generic placeholders. Finally, we re-scan using exact string matching any remaining mentions of the answer, and discard those samples.

Overall, this pipeline can continually ingest news articles and generate open-ended forecasting questions. We use the same methodology to create train, validation and test splits, but use *different news sources* to check if our model learns generalizable forecasting skills.

4.2 `OpenForesight`: An Open, Large-Scale Forecasting Training Dataset

We now describe the specific composition of our training dataset. We use DeepSeek v3 as the sample creator and Llama-4-Maverick as the sample selector, with prompts in Section F.1.