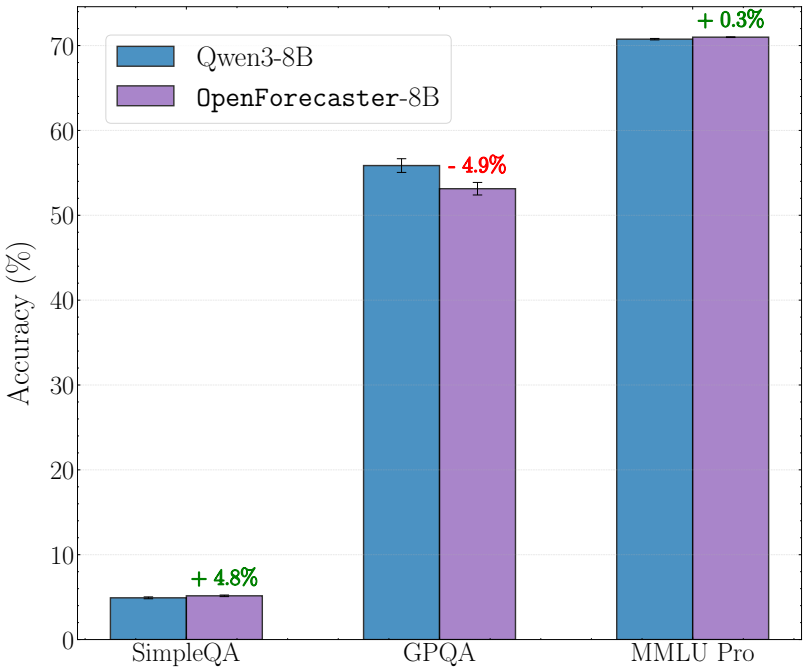

Figure 15: Accuracy comparison on general benchmarks. The number highlighted in green (red) shows *relative* improvement (degradation) over the pre-RL model.

B.5 Evaluation on Metaculus Questions

Prediction Market Dataset. We also source questions from the Metaculus prediction market from May - Nov. 2025. We filter questions which are either meta-prediction questions (prediction about some other prediction question on Metaculus), stock price prediction or below a certain trading volume leaving 449 high-interest questions. Each question is binary so a naive baseline gets 50% accuracy. We evaluate models on this benchmark by computing both accuracy and brier score.

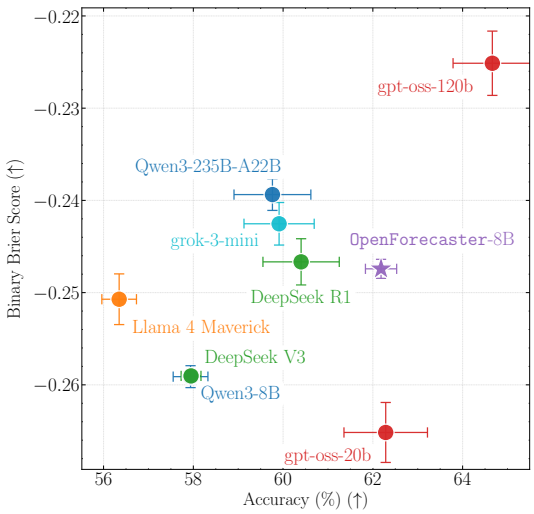

Figure 16: Performance of models on Metaculus questions from May 01 to Nov. 18 2025.

Results on Real Market Questions. As shown in Figure 16, `OpenForecaster` 8B performs better than many larger models like DeepSeek-R1 and Llama-4-Maverick while GPT-OSS-120B is significantly better than rest of the models. **Note:** The brier score reported here is the *standard brier score* used in prediction markets ranging from -1 to 0 (with -0.25 being the baseline for a constant prediction of 50%) and this is different from the brier score we have reported earlier for freeform questions.

C Experimental Setup

C.1 Training Details

Models. We primarily train and perform all our ablations on Qwen3-8B (Yang et al., 2025) thinking model. No official knowledge–cutoff date for Qwen-3 is reported. When queried directly, the models returns inconsistent cutoff dates (most often *October 2023* or *June 2024*). Although the family was released in April 2025, the models frequently treat events from 2024 as future, suggesting a practical cutoff date. This behavior is acceptable for training—even when some prompts refer to events that lie in the model's past. We also train models for Llama and Gemma family to understand how much post-training on `OpenForesight` helps models which haven't undergone RL training already.

Framework. We perform RL training using the `VeRL` package with GRPO algorithm (Shao et al., 2024) for optimization.

Policy/backbone. Unless noted, the trainable policy is `Qwen3-8B`. Prompts (with retrieval of up to 5 chunks) are truncated to 4,096 tokens and responses are capped at 8,192 tokens.

Sampling. We generate with a vLLM-based sampler (chunked prefill enabled). Training uses temperature 1.0 with K=8 samples per prompt.

Optimization. We use AdamW (Loshchilov et al., 2017) with learning rate 5×10^{-6}, cosine decay, 1% warmup, and a minimum LR ratio of 0.1. FSDP parameter *and* optimizer offloading are enabled; gradient checkpointing, padding removal, and dynamic batch sizing are used. Global train batch size is 256 (PPO mini-batch 64). Training runs are performed a node of 8 H100 GPUs for 5 epochs.

Data and Training Order. We train on all 52K samples from `OpenForesight` and also include additional 2K resolved binary questions sourced from Metaculus (essential for performing well on binary questions as we show in Section B.1). Importantly, we do not shuffle all the 54K samples randomly as that would lead to less than 10 binary samples in each batch (of size 256) on average. We instead create two separate groups for freeform and binary questions (randomly shuffling within each group). We order our training set such that first 52K questions are free-form and last 2K being binary.

Note: We found having separate batches of binary and freeform questions is crucial to get good performance on binary questions like those found in prediction markets.

Advantages and losses. GRPO with group-centered advantages (no standard-deviation normalization). PPO clipping uses ϵ_{low}=0.20, ϵ_{high}=0.28, and clip-c=10.0. We apply a low-variance KL penalty with coefficient β=0.005.

Rewards. We use the `Qwen3-4B` with greedy decoding (temperature=0) as the judge for assessing answer correctness. We instruct it to enforce strict, reference-guided matching with tolerance for case and common aliases, and prompt it in `non-thinking` mode.

For freeform questions, the reward function is the sum of accuracy (0 to 1) and brier score (-1 to 1). For binary questions, the reward is just the (negated) binary brier score (-1 to 0 with -0.25 being the baseline for a prediction of 50%). We also include a format reward penalty of -1 if the answer and/or probability values cannot be extracted or parsed succesfully and 0 otherwise. Thus, the final (total) reward range on freeform questions is $[-2, 2]$ and on binary questions is $[-2, 0]$.

C.2 Evaluation Details

For all our main results, we sample $n = 3$ response per prompt for each model with temperature 0.6 and top-$p = 0.95$.

Freeform Evaluation. To ensure that our model is not exploiting any LLM-as-a-judge biases in training, we employ `Llama-4-Scout` (instead of `Qwen3-4B` judge used in training) with greedy decoding (to minimize variance) for checking model's responses against the ground truth using answer matching.

Binary Evaluation. For binary questions, we ask the model to report its best probability estimate p for the event to resolve `YES`. Let o be the actual outcome of the event (1 if `YES` and 0 otherwise). The brier score is computed as $-(p-o)^2$ with 0 being the best and -1 being the worst.

C.3 Details on Compute

To improve transparency around data and compute, we report approximate token counts, training steps, and GPU-hours for both SFT and RL. Our curated `OpenForesight`training set contains 52,183 samples. The average sample (just the question without any retrieval) has about 1000 characters and corresponds to roughly 400 tokens under the Qwen3 tokenizer for the question text, yielding approximately 2×10^7 prompt tokens in total.

For SFT, fine-tuning `Qwen3-8B` for 3 epochs took 5 hours on 8 H100 GPUs, corresponding to roughly 40 H100 GPU-hours. RL training is substantially more expensive: Our final run lasted for 5 epochs over the training set, resulting in about 1,300 optimization steps, for an estimated total of $\sim$ 1,000 H100 GPU-hours. Including all ablations, we estimate we used $\sim$ 20,000 H100 GPU-hours.