REWARD HACKING REVEALS ITSELF IN JUDGE CONFIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards has driven major advances in reasoning. Yet reliable rule-based verification remains challenging for many tasks, motivating the use of LLM judges as scalable proxy reward providers. However, optimizing against an LLM judge can induce reward hacking, where policies exploit its evaluation errors to increase proxy reward while task performance stagnates or deteriorates. Existing approaches to monitoring reward hacking often require external evaluation, additional inference, or access to model internals. In contrast, we uncover a monitoring signal already present in the LLM judge's outputs: verdict confidence systematically declines as reward hacking develops. Building on this finding, we propose **balanced verdict margin (BVM)**, a lightweight and task-agnostic metric that tracks reward hacking using the judge's verdict-token log-probabilities. We further develop a **margin-guided reward intervention (MRI)** that mitigates reward hacking without requiring gold or reference feedback during training. Extensive experiments show that BVM consistently tracks reward hacking across tasks, while MRI substantially improves task performance over unmodified LLM-judge training, even matching the rule-based control in mathematical reasoning.

1 INTRODUCTION

Effective reinforcement learning for language-model reasoning depends on rewards that reliably reflect task performance. Automatically verifiable answers enable such feedback in mathematics, coding, and scientific question answering (Guo et al., 2025; Yu et al., 2025; Liu et al., 2026). For many open-ended tasks, however, response quality depends on multiple criteria that are difficult to capture through programmatic checks. LLM judges offer a flexible alternative for evaluating responses (Zheng et al., 2023) and providing training rewards based on task-specific criteria (Gunjal et al., 2026). Optimizing against a judge, however, introduces a challenge beyond evaluation: training may favor responses that exploit the judge's errors. This can lead to *reward hacking*, in which proxy rewards increase without corresponding improvements in task performance (Amodei et al., 2016; Gao et al., 2023). This risk reflects a mismatch between an evolving policy and a fixed LLM judge (Kenton et al., 2026).

Existing approaches to monitoring reward hacking draw on external reference feedback, policy-output diagnostics, or model-internal signals. External evaluation depends on reliable reference feedback (Gao et al., 2023). Output-based methods may miss exploitation that is not expressed in observable behavior or require additional inference (Baker et al., 2025; Wang et al., 2026b; Mahmoud et al., 2026). Methods based on model internals require access to internal representations or gradients (Miao et al., 2025; Wang et al., 2026a). Taken together, these approaches provide valuable evidence but impose additional requirements. This highlights the need for a general, easily obtained and training-free metric for monitoring reward hacking.

In this paper, we investigate whether the reward judge's own outputs can reveal reward hacking. Our analysis shows that reward hacking is accompanied by declining response likelihood under the judge, suggesting that policy outputs are moving away from regions familiar to the judge (Section 3.1). This observation leads us from response predictability to the judge's confidence in its verdicts: as responses become less familiar, does the judge also become less certain about accepting or rejecting them? To quantify this confidence, we define the **verdict margin** as the absolute

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

gap between the log-probabilities of positive and negative verdict tokens. A larger margin indicates a more decisive judgment, while a smaller margin indicates greater uncertainty. To track judgment confidence across responses, we introduce **balanced verdict margin (BVM)**, which averages the mean verdict margins of accepted and rejected judgments. Across our experiments, BVM declines as proxy rewards diverge from gold or reference rewards, providing a signal of developing reward hacking. Because it is computed directly from the reward judge's existing verdict-token log-probabilities, BVM requires no auxiliary training or stronger external evaluator, making it a simple and broadly applicable signal.

The confidence captured by verdict margin also supports intervention at the individual-judgment level. Our analysis suggests that reward hacking is largely driven by false positives, which tend to have lower verdict margins than true positives. We therefore introduce margin-guided reward intervention (MRI), which attenuates low-confidence positive rewards. In a controlled mathematical reasoning setting with verifiable answers, LLM-judge training with MRI achieves performance close to rule-based training across Qwen and GLM models. To assess its broader applicability, we extend MRI to open-ended medical and scientific question answering, where training uses rubric-based rewards (Gunjal et al., 2026). Training with MRI improves reference scores by 6.67 points on HealthBench and 5.97 points on ResearchQA. In summary, our main contributions are as follows:

- We propose balanced verdict margin (BVM), a lightweight, task-agnostic monitoring metric derived from the reward judge's verdict-token log-probabilities.
- We show that declining BVM tracks reward hacking and develop a margin-guided reward intervention (MRI) to suppress unreliable positive feedback.
- Experiments across controlled mathematical reasoning and open-ended question answering demonstrate BVM's effectiveness in detecting reward hacking. Across these settings, MRI mitigates reward hacking while improving task performance.

2 Related Work

Reward Hacking in RL. Reward hacking occurs when an agent exploits a proxy reward, increasing measured return without improving the intended objective (Amodei et al., 2016; Hadfield-Menell et al., 2017; Skalse et al., 2022). This problem is particularly relevant to preference-based and language-model reinforcement learning, where learned reward models provide the feedback used for policy optimization (Christiano et al., 2017; Ziegler et al., 2019; Bai et al., 2022). Gao et al. (2023) characterize how reward-model overoptimization develops as optimization pressure increases. Rafailov et al. (2024) show that similar overoptimization also occurs in direct alignment algorithms, even without an explicit reward model. These findings highlight the need to identify when reward signals become unreliable as optimization proceeds. Focusing on LLM judges, our work investigates whether their verdict confidence can reveal reward hacking and guide reward intervention.

Mitigating Reward Hacking. Existing mitigation methods primarily improve reward estimation or constrain policy optimization. Reward-model ensembles, uncertainty penalties, and pessimistic reward estimates reduce the influence of unreliable reward predictions (Coste et al., 2024; Eisenstein et al., 2024; Houliston et al., 2024; Zhang et al., 2024). Constrained RLHF and KL regularization limit optimization or policy drift to reduce exploitation of imperfect reward models (Moskovitz et al., 2024; Stiennon et al., 2020; Gao et al., 2023). Unlike approaches that aggregate multiple reward models or constrain policy updates, MRI selectively adjusts individual rewards using the existing judge's confidence, requiring neither additional reward models nor changes to the policy optimization algorithm.

3 Method

Our framework uses the judge's verdict margins for both monitoring and intervention (Figure 1). BVM aggregates these margins across responses to monitor reward hacking, while MRI uses individual margins to adjust training rewards.

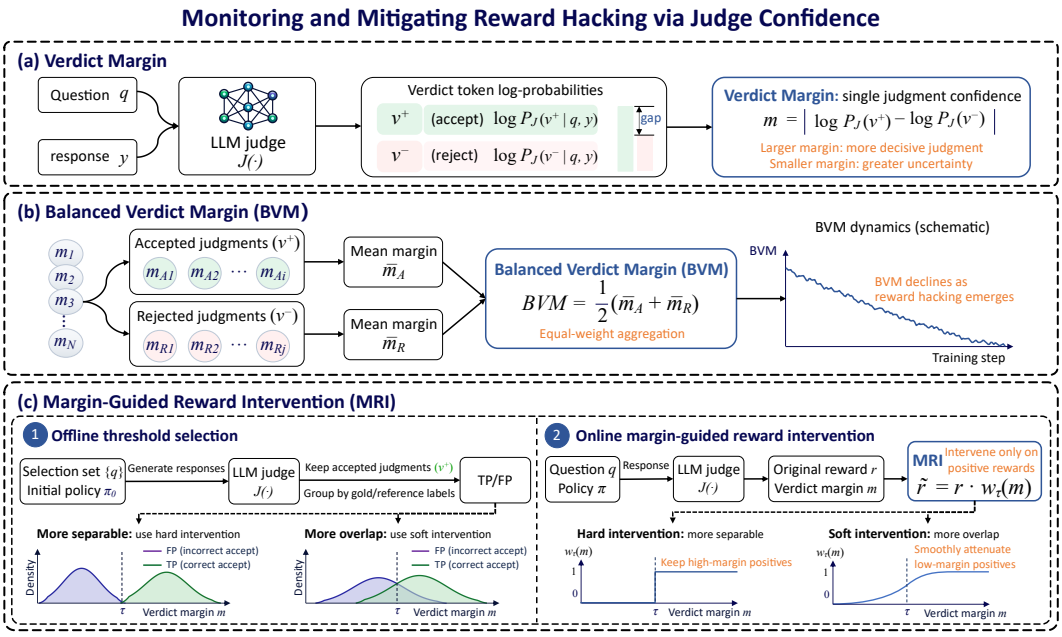


Figure 1: **Overview of our framework.** (a) The verdict margin is the absolute log-probability gap between accept and reject verdict tokens. A larger margin indicates a more confident judgment. This signal supports both monitoring and intervention. (b) For monitoring, BVM equally weights the mean margins of accepted and rejected judgments. (c) For intervention, gold or reference labels distinguish true positives from false positives among accepted judgments before training. The margin distributions of true positives and false positives guide threshold selection. Their degree of overlap determines the choice of hard or soft intervention.

3.1 From Response Likelihood to Verdict Confidence

During reinforcement learning, the policy's response distribution evolves while the reward judge remains fixed. We therefore track the likelihood of policy-generated responses under the judge throughout training. We feed each policy-generated response to the judge together with its scoring context. Let $y = (y_1, \ldots, y_n)$ denote the response tokens and c the context preceding them. We then compute the average log-probability that the judge assigns to these response tokens:

$$\ell(y \mid c) = \frac{1}{n} \sum_{k=1}^{n} \log P_J(y_k \mid c, y_{<k}). \tag{1}$$

Here, P_J denotes the judge's token probabilities, $y_{<k}$ denotes the response tokens before position k, and n is the number of response tokens under the judge's tokenizer. A lower $\ell(y \mid c)$ indicates that the response is less predictable under the judge.

At training step t, we generate N_t responses from policy π_t to a fixed set of monitoring questions and compute their mean response log-likelihood, $L_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_{t,i} \mid c_{t,i})$. This summarizes how predictable the current policy's responses are under the judge.

Figure 2 shows how response log-likelihood changes across response positions during training. We compare LLM-judge training with a rule-based control, using the same LLM judge to compute response log-likelihoods for both. In the LLM-judge run, reward hacking (Figure 3) is accompanied by declining log-likelihood across much of the response. Despite receiving higher proxy rewards, the generated responses become less predictable under the judge. In contrast, the rule-based run exhibits no reward hacking and mostly stable or increased response log-likelihood. This contrast suggests that reward hacking is associated with responses becoming less familiar to the judge. We next examine whether this decline in response predictability is accompanied by changes in the judge's verdict probabilities.

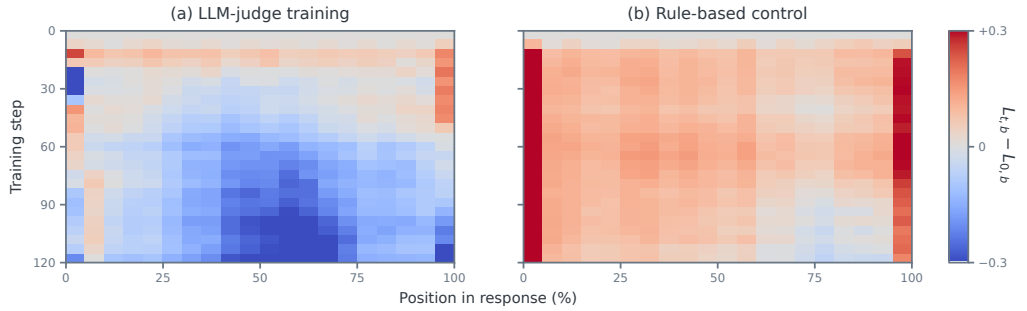


Figure 2: **Response-likelihood change under a fixed judge.** Responses from both Qwen2.5-32B training runs on AIME 2024 are evaluated by the same Qwen2.5-32B-Instruct judge. Rows show training steps, and columns show 20 equal bins from the start to the end of each response. $L_{t,b}$ denotes the mean token log-probability in position bin b at training step t. Colors show $L_{t,b} - L_{0,b}$. Blue indicates a decrease and red indicates an increase in response likelihood from step 0.

3.2 Balanced Verdict Margin

Inspired by prior work on prediction confidence (Guo et al., 2017; Kadavath et al., 2022), we quantify judgment confidence using the absolute gap between the log-probabilities of two candidate verdict tokens (Figure 1(a)). Let v^+ and v^- denote the positive and negative candidates, respectively. We define the **verdict margin** as

$$m = \left|\log P_J(v^+) - \log P_J(v^-)\right|. \tag{2}$$

Both probabilities are computed at the same verdict position given the same context. A larger margin indicates a stronger preference for either acceptance or rejection, while a margin near zero indicates uncertainty between the two (Zhu et al., 2026).

Although a large margin does not guarantee that a judgment is correct, higher-margin judgments generally agree more often with independent gold or reference labels (Figure 6), which supports using verdict margin as a reliability signal. However, monitoring reward hacking requires tracking confidence across the policy's responses, not just individual judgments. We therefore aggregate verdict margins to summarize how the judge's confidence changes during training.

The most direct way to aggregate verdict margins is to average over all judgments. At training step t, this pooled mean is $M_t^{\text{pool}} = \frac{1}{K_t}\sum_{i=1}^{K_t} m_{t,i}$, where K_t is the number of judgments and $m_{t,i}$ is the margin of judgment i. However, as the judge's acceptance rate rises during policy optimization, the pooled mean becomes increasingly dominated by accepted judgments. This can obscure changes in rejected judgments' margins, which are important for monitoring reward hacking (Section 5). We address this by averaging the two groups' mean verdict margins with equal weight (Figure 1(b)). Let $\bar{m}_{t,A}$ and $\bar{m}_{t,R}$ denote the mean margins of judgments accepted and rejected by the reward judge, respectively. We define balanced verdict margin (BVM) as

$$\text{BVM}_t = \frac{1}{2}\left(\bar{m}_{t,A} + \bar{m}_{t,R}\right). \tag{3}$$

By assigning equal weight to the mean margins of accepted and rejected judgments, BVM prevents the more frequent verdict from dominating the summary. A decline in BVM indicates that the judge is becoming less confident when evaluating policy-generated responses. BVM uses only the judge's verdicts and verdict-token log-probabilities, requiring no reference labels for monitoring.

Section 4.2 evaluates BVM's effectiveness in monitoring reward hacking and compares it with accepted-only, rejected-only, and pooled margin aggregation methods. We report each metric's percentage change from its value at step 0. For BVM, this gives $C_t = (\text{BVM}_t - \text{BVM}_0)/\,\text{BVM}_0 \times 100\%$, where $\text{BVM}_0 > 0$.

3.3 Margin-Guided Reward Intervention

BVM provides a distribution-level monitoring signal, while individual verdict margins enable instance-level reward adjustment. We focus on positive rewards because the growth in judge errors

is dominated by increasing false acceptance, while false negatives remain substantially less frequent (Appendix C.1). Since false positives generally have lower margins than true positives (Figure 7), we conservatively attenuate low-margin positive rewards while largely preserving high-margin ones.

Figure 1(c) illustrates the two stages of MRI: offline threshold selection and online reward intervention. Before training, the initial policy generates responses on a threshold-selection set. The reward judge evaluates these responses, and we collect the verdict margins of accepted judgments to form the distribution $\mathcal{M}_0^+$. Using gold or reference labels, we distinguish false positives from true positives within this group. We then choose the quantile level α so that the low-margin intervention region covers many false positives while including as few true positives as possible. The threshold is set to $\tau = Q_\alpha(\mathcal{M}_0^+)$, where Q_α denotes the α-quantile. This threshold guides the attenuation of low-margin positive rewards. We keep τ unchanged throughout training and use only verdict margins to intervene on positive rewards.

For a positive reward r with margin m, we modify its contribution as

$$\tilde{r} = r\, w_\tau(m). \tag{4}$$

We consider two forms of intervention. **Hard intervention** uses a binary cutoff, while **soft intervention** applies gradual attenuation:

$$\begin{cases} \text{Hard:} & w_\tau(m) = \mathbf{1}[m > \tau], \quad (5) \\ \text{Soft:} & w_\tau(m) = 1 - \lambda\, \sigma\left(\dfrac{\tau - m}{\beta}\right), \quad (6) \end{cases}$$

where $\sigma(z) = (1 + e^{-z})^{-1}$; $\lambda \in [0, 1]$ controls the attenuation strength, and $\beta > 0$ controls the smoothness. Offline analysis shows varying degrees of separation between the verdict-margin distributions of true positives and false positives. Hard intervention is suitable when the two distributions are relatively separable, whereas soft intervention provides a smoother adjustment when substantial overlap remains.

4 Experiments

Experiments address two questions: whether BVM tracks emerging reward hacking (Section 4.2), and whether margin-guided reward intervention improves independently evaluated task performance (Section 4.3).

4.1 Experimental Setup

Settings. We consider two complementary reinforcement learning settings: controlled mathematical reasoning, which provides verifiable answers for precise evaluation, and open-ended question answering, which uses rubric-based rewards to test generalization beyond verifiable tasks.

Training environments. For mathematical reasoning, monitoring experiments use Qwen2.5-7B-Base and Qwen2.5-32B-Base (Yang et al., 2025b), trained on DAPO-Math using DAPO (Yu et al., 2025). We compare a rule-based control with LLM-judge training using corresponding-scale Qwen2.5-Instruct judges. Intervention experiments include these two models and additionally cover Qwen3-4B and Qwen3-8B (Yang et al., 2025a), as well as GLM-4-9B (Team GLM et al., 2024). For open-ended QA, we train Qwen2.5-7B-Base on the Medicine and Science subsets of RubricHub (Li et al., 2026) using GRPO (Shao et al., 2024). Monitoring experiments compare LLM-judge training using Qwen2.5-7B-Instruct with a GPT-OSS-120B control (OpenAI, 2025). Intervention experiments apply MRI to the Qwen2.5-7B-Instruct rewards. Following prior work (Mahmoud et al., 2026), we treat GPT-OSS-120B as a stronger reference evaluator, not a ground-truth oracle. Training parameters are provided in Appendix A (Table 4).

Evaluation protocol. We score the same generated responses using the training-time proxy reward and an independent gold or reference evaluation. For mathematical reasoning, we monitor reward hacking on AIME 2024. For open-ended QA, we monitor it on 300-example subsets of HealthBench (Arora et al., 2025) and ResearchQA (Yifei et al., 2026). In mathematics, a rule-based verifier assesses correctness. For open-ended QA runs using a 7B judge, GPT-OSS-120B provides

offline reference judgments at the criterion level. BVM is computed using the reward judge for each LLM-judge run. For the rule-based controls, which use no LLM judge for training rewards, we compute BVM offline with corresponding-scale Qwen2.5-Instruct judges. Appendix A details model and evaluator settings (Table 5).

Intervention protocol. We select thresholds on AIME 2024 for Math and the corresponding RubricHub subsets for Medicine and Science. The choice of hard or soft intervention depends on the separation between true-positive and false-positive margin distributions. We use hard intervention for mathematical reasoning and evaluate its effect on AIME 2024, AIME 2025, AIME 2026, AMC 2023, MATH-500 (Lightman et al., 2024), Omni-MATH (Gao et al., 2025), OlympiadBench (He et al., 2024), and MinervaMath (Lewkowycz et al., 2022). For open-ended QA, we use soft intervention and evaluate it on RubricHub-Medicine-300, HealthBench-300, RubricHub-Science-300, and ResearchQA-300. Appendix C reports the selected quantiles, threshold selectivity, and intervention results under different threshold settings.

4.2 Monitoring Reward Hacking

We first evaluate whether BVM tracks reward hacking as it emerges during policy optimization.

In mathematical reasoning, proxy rewards rise while gold accuracy collapses under LLM-judge training, revealing reward hacking (Figure 3). Figure 4(a) shows the corresponding contrast in BVM: it contracts by roughly 50–60% in these runs but rises by roughly 20–50% in the rule-based controls. For each model size, BVM for the control run and the corresponding LLM-judge run is computed using the same fixed judge, so the opposing trends cannot be attributed to differences in the monitoring judge. Taken together, the proxy–gold divergence and the controlled BVM comparison show that BVM helps distinguish genuine task improvement from optimization that exploits an imperfect reward judge.

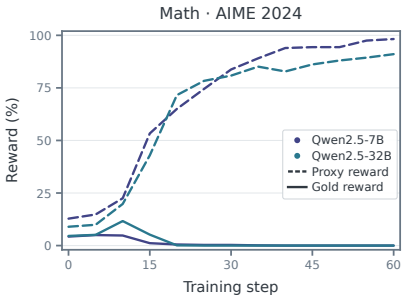


Figure 3: **Proxy and gold rewards during LLM-judge training.** Dashed curves show proxy rewards from the LLM judges; solid curves show offline rule-verified reward on the same responses. Their divergence illustrates reward hacking.

Open-ended QA provides a complementary test under rubric-based feedback. In Medicine and Science, rewards from the 7B judge increase while diverging from GPT-OSS-120B reference evaluations, revealing reward hacking (Appendix Figure 9(b,c)). BVM decreases by about 5% on HealthBench and 26% on ResearchQA, while remaining close to its initial value in the GPT-OSS-120B controls (Figure 4(b,c)). Across settings, BVM declines in reward-hacking runs but remains stable or increases in the controls, supporting its use as a monitoring signal.

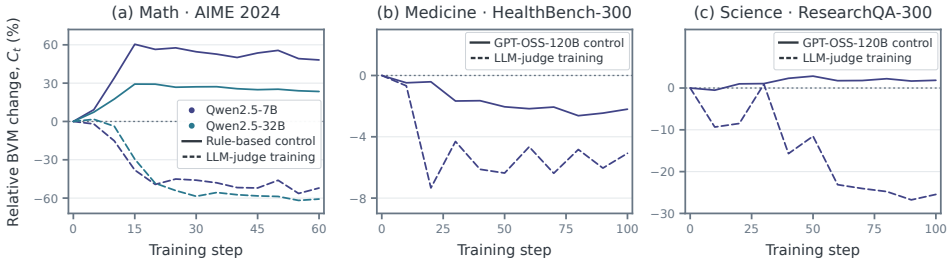


Figure 4: **Relative BVM change during training.** C_t is the percentage change in BVM relative to its value at step 0. Dashed curves represent LLM-judge training. Solid curves represent rule-based controls in Math and GPT-OSS-120B controls in Medicine and Science.

Why balance accepted and rejected margins? We compare BVM with accepted-only, rejected-only, and pooled margin aggregation in the Qwen2.5-32B mathematical reasoning setting (Figure 5).

Accepted-only aggregation fails to capture reward hacking, whereas rejected-only and pooled aggregation also decline in the improving rule-based control. BVM instead increases in the control and decreases under LLM-judge training ($+20.3\%$ versus -42.8%). This comparison shows why tracking both verdict classes with equal weight matters for distinguishing reward hacking from task improvement. Appendix B extends the comparison to additional settings.

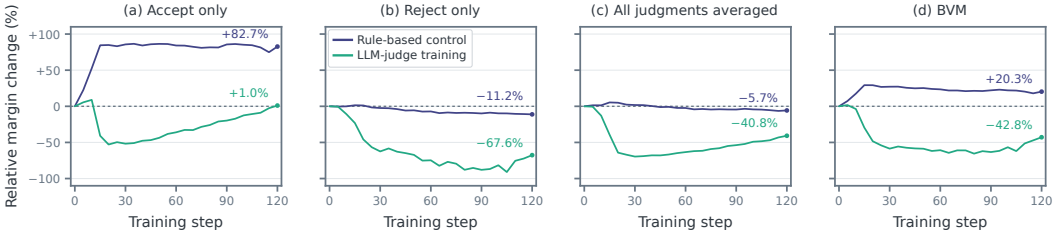


Figure 5: **Comparison of verdict-margin aggregation methods.** Panels compare accepted-only, rejected-only, pooled, and BVM aggregation for Qwen2.5-32B on AIME 2024, using Qwen2.5-32B-Instruct as the judge. Purple and green denote the rule-based control and LLM-judge training, respectively. Curves show percentage changes relative to their values at step 0. Endpoint annotations show the final percentage changes.

4.3 Mitigating Reward Hacking with Margin-Guided Intervention

We next evaluate whether verdict margins can be used not only to monitor reward hacking but also to improve training feedback. The key is to improve actual task performance, rather than merely lowering proxy rewards.

Improving training feedback through margin-guided intervention relies on a relationship between verdict margin and judgment reliability. Figure 6 shows that higher-margin judgments are generally more accurate across all four experiments, with a sharper increase in Math and a more gradual trend in Medicine and Science. This supports margin as a reliability signal, although a large margin does not guarantee a correct judgment.

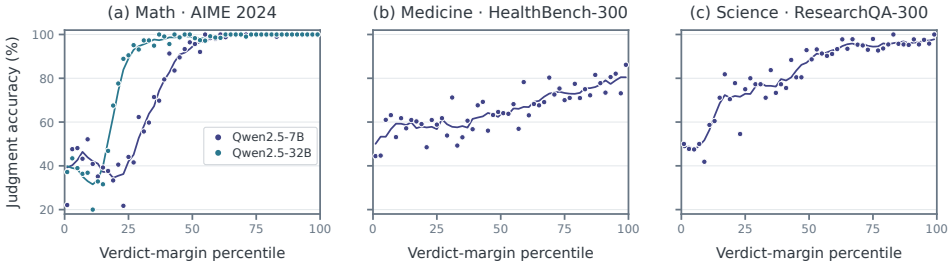


Figure 6: **Judgment accuracy by verdict-margin percentile.** We determine judgment correctness using rule-based verification in Math and GPT-OSS-120B reference judgments in Medicine and Science. Both correct acceptances and correct rejections count as correct judgments. Each point shows accuracy within a 2-percentile margin bin, and curves show centered five-bin rolling averages.

In mathematical reasoning, Hard MRI improves accuracy over unmodified LLM-judge training in **37/40** model–benchmark combinations and achieves the best result among the three reward schemes in **19/40** comparisons (Table 1). These results cover five Qwen and GLM models across eight benchmarks. Performance generally approaches the rule-based control and exceeds it on some benchmarks. In the Qwen2.5 runs, BVM increases under Hard MRI, following a trend similar to the rule-based control rather than the decline under unmodified LLM-judge training (Appendix Figure 11). Together with the performance gains, this pattern supports MRI's effectiveness in mitigating reward hacking.

For open-ended QA, Soft MRI improves reference scores on **all four datasets** while retaining the same reward judge during training (Table 2). For Medicine, it yields relative gains of **24.5%** on

RubricHub-Medicine and **23.4%** on HealthBench. For Science, it improves performance by **28.1%** on RubricHub-Science and **17.3%** on ResearchQA. On HealthBench, Soft MRI reaches 35.21, exceeding the GPT-OSS-120B control by 0.86 points. These results show that margin-guided intervention can improve training with a weaker judge without replacing it with a stronger model. Together, these results support MRI's effectiveness across different task settings.

Table 1: **Mathematical reasoning performance on eight benchmarks.** We compare rule-based training, unmodified LLM-judge training, and LLM-judge training with Hard MRI across five policy models. Values are accuracy (%) evaluated using rule-based verifiers. Best results are in bold and second-best are underlined.

Policy model	Reward scheme	AIME 24/25/26	AMC23	MATH-500	Omni-MATH	OlyBench	MinervaMath
Qwen2.5-7B	Rule-based	**15.52** / **9.79** / 8.33	**61.80**	55.20	16.02	28.93	19.12
	LLM judge	4.06 / 1.98 / 2.50	20.31	34.60	10.70	14.54	9.56
	Hard MRI	13.33 / 8.65 / **8.85**	56.48	**68.00**	**20.39**	**35.31**	**21.69**
Qwen2.5-32B	Rule-based	**25.52** / **21.88** / **18.12**	73.36	58.40	**20.93**	**35.01**	15.07
	LLM judge	11.15 / 5.63 / 6.88	47.03	57.40	18.76	28.04	**22.43**
	Hard MRI	22.81 / 20.73 / 16.98	**74.22**	**60.60**	19.53	32.94	15.44
Qwen3-4B	Rule-based	**24.58** / **24.17** / **19.90**	**69.77**	60.20	20.03	**35.61**	20.22
	LLM judge	13.85 / 13.54 / 13.33	43.36	58.20	15.70	27.74	21.69
	Hard MRI	21.88 / 19.79 / 15.00	66.25	**60.60**	**21.66**	35.31	**24.63**
Qwen3-8B	Rule-based	**26.35** / 23.33 / 21.46	**74.53**	**76.80**	**27.40**	**44.36**	**30.51**
	LLM judge	22.60 / 17.29 / 19.17	60.39	72.40	21.27	39.91	25.00
	Hard MRI	24.69 / **23.85** / **22.50**	71.48	70.20	25.19	40.36	23.16
GLM-9B	Rule-based	12.29 / **10.00** / 10.00	50.78	55.60	15.66	29.23	24.26
	LLM judge	8.65 / 5.31 / 6.77	30.47	45.60	11.18	20.92	9.19
	Hard MRI	**12.81** / 8.23 / **10.94**	**52.50**	**67.20**	**20.07**	**36.94**	**24.63**

Table 2: **Open-ended QA performance on four datasets.** We compare training with a GPT-OSS-120B judge, unmodified LLM-judge training, and LLM-judge training with Soft MRI. Values are GPT-OSS-120B reference scores (%). RubricHub-Medicine and RubricHub-Science are in-domain evaluations, while HealthBench and ResearchQA are out-of-domain evaluations. Best results are in bold and second-best are underlined.

	Medicine		**Science**	
Reward scheme	RubricHub-Medicine	HealthBench	RubricHub-Science	ResearchQA
GPT-OSS-120B	**44.01**	34.35	**41.18**	**47.66**
LLM judge	31.05	28.54	28.89	34.50
Soft MRI	38.66	**35.21**	37.01	40.47

5 Analysis of BVM Dynamics

Margin dynamics by judgment outcome. To understand which judgments contribute to BVM contraction, we analyze margin dynamics by judgment outcome. Using gold or reference labels, we group judgments into true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs), and examine how each group's mean verdict margin changes during training. Figure 7 shows that FPs generally have lower mean verdict margins than TPs, while FNs have lower margins than TNs in open-ended QA. These differences suggest that the judge tends to be less confident when its judgments are incorrect. During training, TN margins decline substantially, whereas the accepted categories show no consistent downward trend across experiments. The decline in TN margins weakens rejection-side confidence and contributes to BVM contraction.

A shift toward false acceptance. To examine what the declining TN margins reflect, we focus on responses identified as incorrect by the independent evaluator. We use the signed verdict margin, $s = \log P_J(v^+) - \log P_J(v^-)$. The sign of s indicates acceptance or rejection preference, while $|s|$ equals the verdict margin m, measuring judgment confidence.

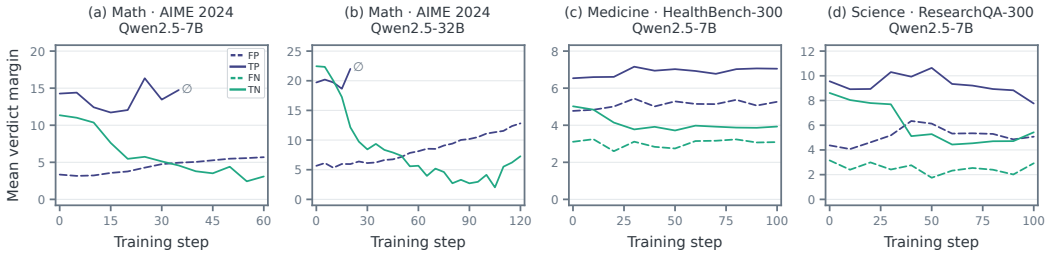


Figure 7: **Mean verdict margins by confusion-matrix category.** Curves show mean verdict margins for TPs, FPs, TNs, and FNs during training. Purple and green respectively denote judge acceptance and rejection. Solid curves represent correct judgments (TPs/TNs), and dashed curves represent incorrect judgments (FPs/FNs). Judgment correctness is assessed using rule-based verification in Math and GPT-OSS-120B reference judgments in Medicine and Science. FNs are virtually absent in Math. The symbol $\emptyset$ indicates that the category has no samples, so its mean margin is undefined.

Figure 8 shows that, during LLM-judge training of Qwen2.5-32B, the margin distribution of incorrect responses shifts from the rejection side toward and across zero, accompanied by increasing false acceptance. Because these responses are incorrect according to the rule-based verifier, this shift cannot be explained by improved answer correctness. It instead indicates that the judge increasingly assigns higher acceptance probabilities to incorrect responses.

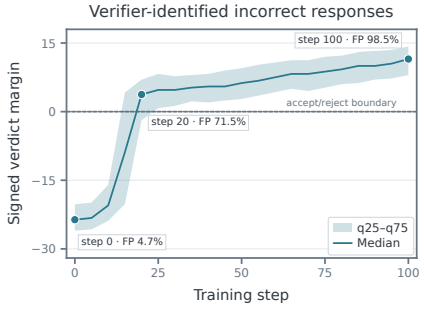


Figure 8: **Signed verdict margins for incorrect math responses from Qwen2.5-32B.** We consider responses identified as incorrect by the rule-based verifier. The line and shaded band show the median and 25th–75th percentiles of signed verdict margins. Positive and negative margins indicate higher probabilities for acceptance and rejection.

Why both verdict classes matter. These observations help explain BVM contraction without implying that rejection confidence alone is sufficient for monitoring. In the reward-hacking runs, rejection-side mean margins decline, and changes on the acceptance side do not offset this decline. BVM therefore decreases. In the rule-based control shown in Figure 5, rejection-side margins also decline, but acceptance-side margins increase enough for BVM to rise. Thus, the acceptance side provides essential context for interpreting weakened rejection confidence. By weighting both sides equally, BVM captures their combined changes without allowing the increasingly frequent accepted judgments to dominate.

6 Conclusion

Reward scores indicate what the policy is learning to satisfy, but do not establish whether the underlying judgments remain trustworthy. This work shows that verdict confidence provides a complementary signal for monitoring reward hacking. We capture this signal with balanced verdict margin (BVM), which declines alongside divergence between proxy and gold or reference rewards across controlled mathematical reasoning and open-ended question answering. Margin-guided reward intervention (MRI) uses individual verdict margins to suppress unreliable positive feedback, bringing mathematical performance close to rule-based training across Qwen and GLM model families and improving task performance on HealthBench and ResearchQA.

A remaining limitation is that MRI uses gold or reference labels to select its intervention threshold before training. Future work could explore whether verdict-confidence dynamics alone can guide threshold selection and adapt intervention strength as policies evolve.

AI USE STATEMENT

In preparing this submission, we used generative AI tools only to aid and polish the writing and for retrieval and discovery, including finding related work. The authors reviewed and verified all AI-assisted text and citations and take responsibility for the final content.

ETHICS STATEMENT

This work studies reward hacking to improve the monitoring and control of optimization against imperfect LLM-based feedback. We use existing models and benchmark datasets and conduct no new human-subject data collection. The Medicine experiments are research evaluations and are not intended for clinical decision-making. Our analysis is intended to support the detection and mitigation of reward hacking, and we recommend independent evaluation before deploying LLM-judge-based training systems.

REPRODUCIBILITY STATEMENT

Section 3 defines verdict margin, BVM, and margin-guided reward intervention, and Section 4.1 specifies the models, datasets, and evaluation protocol. Appendix A provides the optimization, sampling, and judgment details; Appendices B and C report additional monitoring evidence and intervention analyses. The anonymous supplementary code provides the verdict-margin extraction, BVM, threshold-selection, and hard and soft intervention implementations, together with configuration files, integration interfaces, and validation tests. Distributed rollout, policy optimization, and checkpointing rely on VERL (Sheng et al., 2025), while judge inference uses an OpenAI-compatible serving backend, as documented in the supplement.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. doi: 10.48550/arXiv.1606.06565. URL https://arxiv.org/abs/1606.06565.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. doi: 10.48550/arXiv.2505.08775. URL https://arxiv.org/abs/2505.08775.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. doi: 10.48550/arXiv.2204.05862. URL https://arxiv.org/abs/2204.05862.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025. doi: 10.48550/arXiv.2503.11926. URL https://arxiv.org/abs/2503.11926.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03741.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dcjtMYkpXx.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5u1GpUkKtG.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07985.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.17746.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1711.02827.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 3828–3850, 2024. doi: 10.18653/v1/2024.acl-long.211.

Sam Houliston, Alizée Pace, Alexander Immer, and Gunnar Rätsch. Uncertainty-penalized direct preference optimization. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024. doi: 10.3929/ethz-b-000719933.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. doi: 10.48550/arXiv.2207.05221. URL https://arxiv.org/abs/2207.05221.

Zachary Kenton, Lili Janzer, Rory Greig, Tian Huey Teh, Kirill Tyshchuk, Jonah Brown-Cohen, Harri Edwards, Senthooran Rajamanoharan, Noah Y Siegel, Natasha Jaques, et al. Debate training reduces reward hacking in RLAIF. *arXiv preprint arXiv:2608.17776*, 2026. doi: 10.48550/arXiv.2608.17776. URL https://arxiv.org/abs/2608.17776.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022. doi: 10.52202/068431-0278.

Sunzhu Li, Jiale Zhao, Huimin Ren, Zhenlin Wei, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Wei Chen. RubricHub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 31320–31344, 2026. doi: 10.18653/v1/2026.acl-long.1445.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Tengxiao Liu, Deepak Nathani, Zekun Li, Kevin Yang, and William Yang Wang. WildSci: Advancing scientific reasoning from in-the-wild literature. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11677–11695, 2026. doi: 10.18653/v1/2026.findings-acl.567.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2605.12474*, 2026. doi: 10.48550/arXiv.2605.12474. URL https://arxiv.org/abs/2605.12474.

Yuchun Miao, Liang Ding, Sen Zhang, Rong Bao, Lefei Zhang, and Dacheng Tao. Information-theoretic reward modeling for stable RLHF: Detecting and mitigating reward hacking. *arXiv preprint arXiv:2510.13694*, 2025. doi: 10.48550/arXiv.2510.13694. URL https://arxiv.org/abs/2510.13694.

Ted Moskovitz, Aaditya Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D. Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained RLHF. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04373.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. doi: 10.48550/arXiv.2508.10925. URL https://arxiv.org/abs/2508.10925.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems*, volume 37, pp. 126207–126242, 2024. doi: 10.52202/079017-4009.

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. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9460–9471, 2022. doi: 10.52202/068431-0687.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020. URL https://arxiv.org/abs/2009.01325.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024. doi: 10.48550/arXiv.2406.12793. URL https://arxiv.org/abs/2406.12793.

Songtao Wang, Quang Hieu Pham, Fangcong Yin, Xinpeng Wang, Jocelyn Qiaochu Chen, Greg Durrett, and Xi Ye. Detecting and suppressing reward hacking with gradient fingerprints. In *Conference on Language Modeling*, 2026a. URL https://arxiv.org/abs/2604.16242v2.

Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=Gk7gLAtVDO.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Li S. Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. ResearchQA: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Transactions of the Association for Computational Linguistics*, 14:1365–1389, 2026. doi: 10.1162/tacl.a.732.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775.

Xiaoying Zhang, Jean-François Ton, Wei Shen, Hongning Wang, and Yang Liu. Mitigating reward overoptimization via lightweight uncertainty estimation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 81717–81747, 2024. doi: 10.52202/079017-2596.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Kun Xu, and Yang You. CAMEL: Confidence-gated reflection for reward modeling. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306. PMLR, 2026. URL https://arxiv.org/abs/2602.20670v2.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. doi: 10.48550/arXiv.1909.08593. URL https://arxiv.org/abs/1909.08593.

A Experimental Details

This section provides the experimental details needed to interpret and reproduce the main results. In MRI runs, gold or reference labels are used only for offline evaluation and threshold selection. After a threshold is fixed, MRI uses only the reward judge's verdict and verdict-token log-probabilities. The rule-based and stronger-judge controls instead use their specified supervision directly during training.

A.1 Data and Evaluation

Controlled mathematical reasoning. Policies are trained on the processed DAPO-Math corpus of 17,398 prompts. AIME 2024 is used for training-time monitoring and for selecting the intervention threshold from the initial policy's margin distribution. The broader intervention evaluation uses AIME 2024, AIME 2025, AIME 2026, AMC 2023, MATH-500, Omni-MATH, OlympiadBench, and MinervaMath. A rule-based verifier evaluates mathematical correctness. AIME contains 30 problems and AMC 2023 contains 40 problems, with 32 sampled responses per problem; the reported metric is average single-response accuracy, not pass@32. The other four benchmarks use one response per problem.

Open-ended question answering. Policies are trained separately on the Medicine and Science subsets of RubricHub. We hold out 300 RubricHub examples in each domain for in-domain evaluation and threshold selection. HealthBench-300 and ResearchQA-300 are held-out evaluation sets for Medicine and Science, respectively. The 7B reward judge supplies online training feedback. GPT-OSS-120B has two separate roles: it supplies offline criterion-level reference judgments for the 7B-judge and intervention runs, and it supplies online rewards in the stronger-judge control. We treat it as a stronger evaluator rather than an error-free oracle.

Table 3: Dataset roles. "MRI eval." denotes intervention evaluation; held-out evaluation uses an independent reference channel rather than absolute ground truth.

Setting	Dataset	Role
Math	DAPO-Math	Training
Math	AIME 2024	Monitoring / threshold selection / MRI eval.
Math	AIME 2025	MRI eval.
Math	AIME 2026	MRI eval.
Math	AMC 2023	MRI eval.
Math	MATH-500	MRI eval.
Math	Omni-MATH	MRI eval.
Math	OlympiadBench	MRI eval.
Math	MinervaMath	MRI eval.
Medicine	RubricHub-Medicine	Training
Medicine	RubricHub-Medicine-300	Monitoring / threshold selection / MRI eval.
Medicine	HealthBench-300	Held-out monitoring / MRI eval.
Science	RubricHub-Science	Training
Science	RubricHub-Science-300	Monitoring / threshold selection / MRI eval.
Science	ResearchQA-300	Held-out monitoring / MRI eval.

A.2 Optimization and Sampling

Table 4 summarizes the principal Qwen2.5 configurations. Within each setting, the rule-based control, LLM-judge training, and LLM-judge training with intervention use the same policy-optimization configuration and differ only in how rewards are produced or modified. The Qwen3 and GLM experiments follow the same mathematical reasoning protocol with model-specific judge templates.

Table 4: Principal policy-training configurations. A dash indicates that the corresponding regularizer is disabled.

Configuration	Math 7B	Math 32B	Open-ended QA
Prompt / response tokens	2,048 / 8,192	2,048 / 8,192	4,096 / 4,096
Prompts / batch	64	64	128
Rollouts / prompt	8	16	16
PPO mini-batch	16	16	128
PPO epochs	1	1	1
Actor LR	1.0×10^{-6}	1.0×10^{-6}	4.2×10^{-6}
Warmup	First 10 steps	First 10 steps	First 10%
Loss aggregation	token mean	token mean	seq.-mean–token-mean
KL coefficient	–	–	0.01
Gradient clip	1.0	1.0	0.1
Train (T, p, k)	$(1.0, 1.0, -1)$	$(1.0, 1.0, -1)$	$(1.0, 1.0, -1)$
Eval. decoding	$(T, p, k) = (1.0, 0.7, -1)$	$(T, p, k) = (1.0, 0.7, -1)$	Greedy ($T = 0$)
Group filter	accuracy	accuracy	disabled
Eval. interval	5 steps	5 steps	10 steps

Here T, p, and k denote temperature, top-p, and top-k, respectively; $k = -1$ disables top-k filtering. Mathematical-reasoning groups are resampled until they contain reward variation, with at most ten generation batches. Open-ended QA evaluation uses deterministic greedy decoding with one response per prompt. No group filtering is used for open-ended QA. Warmup spans the first 10 steps for mathematical reasoning and the first 10% of training steps for open-ended QA.

A.3 Models, Evaluators, and Reported Steps

In mathematical reasoning, severe reward hacking eventually causes unmodified LLM-judge training to collapse, so we report its peak-performing checkpoint. For the rule-based controls and Hard MRI, we instead report checkpoints after training has stabilized. In open-ended QA, we compare all three conditions at step 50. We continue these runs to step 100 to examine subsequent reward-hacking dynamics and BVM evolution. Table 5 lists the policy–judge pairings and training steps used in the main result tables.

Table 5: Policy models, reward judges, and training steps for Tables 1 and 2. The reward-judge column applies to unmodified LLM-judge training and MRI. Controls use rule-based rewards in Math and GPT-OSS-120B rewards in Medicine and Science.

		Reported training step		
Policy model	Reward judge	Control	LLM judge	MRI
Mathematical reasoning				
Qwen2.5-7B-Base	Qwen2.5-7B-Instruct	150	10	150
Qwen2.5-32B-Base	Qwen2.5-32B-Instruct	200	10	200
Qwen3-4B-Base	Qwen3-4B-Instruct-2507	270	60	270
Qwen3-8B-Base	Qwen3-8B[a]	180	90	180
GLM-4-9B-0414	GLM-Z1-9B-0414[b]	150	30	150
Medicine and Science (trained separately)				
Qwen2.5-7B-Base	Qwen2.5-7B-Instruct	50	50	50

[a] Thinking disabled.

[b] 128-token reasoning budget.

A.4 POLICY AND JUDGE PROMPTS

The mathematical policy receives the following continuation prompt, where braced text denotes a substituted field:

Mathematical reasoning: policy prompt

```
Solve the following math problem step by step.  The last line of your
response should be of the form Answer:  $Answer (without quotes)
where $Answer is the answer to the problem.
{problem}
Remember to put your answer on its own line after ``Answer:''.
```

Mathematical reasoning: judge prompt

System message.

```
You are a strict math answer judge.  Compare the candidate response
with the reference answer and decide whether the candidate response
is correct.
```

User message.

```
Problem:
{problem}

Reference answer:
{reference_answer}

Candidate response:
{response}

Output format:
Return exactly one line.
The line must be exactly one of these two strings:
\boxed{1}
\boxed{-1}

Use \boxed{1} if the candidate final answer is mathematically
correct.
Use \boxed{-1} if it is wrong.
Do not output explanations, JSON, Markdown, or any other text.
```

The `Problem` field contains the complete policy input, including its answer-format instructions. Qwen judges use this template directly. The full GLM-Z1 template is shown below.

Mathematical reasoning: GLM-Z1 judge prompt

System message.

```
You are a strict math answer judge.  Compare the candidate response
with the reference answer and decide whether the candidate response
is correct.

Use only a very brief internal verification.  The reasoning inside
<think> must not exceed 128 tokens (at most 2 short sentences).  Do
not restate the problem, the reference answer, or the candidate
response.  Stop reasoning as soon as you can determine the verdict
and then emit the required boxed verdict.
```

User message.

```
Problem:
{problem}

Reference answer:
{reference_answer}

Candidate response:
{response}

Output format:
Return exactly one line.
The line must be exactly one of these two strings:
\boxed{1}
\boxed{-1}
```

```
Use \boxed{1} if the candidate final answer is mathematically
correct.
Use \boxed{-1} if it is wrong.
Do not output explanations, JSON, Markdown, or any other text.
```

Open-ended QA: policy prompt

System message.
`You are a helpful assistant.`
User message.
The original question.

RubricHub training and ResearchQA evaluation use the following user message. The input lists numbered evaluation criteria, not their truth values. The judge returns a JSON array of `id`/`satisfied` decisions, for example `[{"id": 1, "satisfied": true}, {"id": 2, "satisfied": false}]`.

RubricHub and ResearchQA: judge prompt

User message.

```
You are an expert evaluator. Given a user question, a candidate
response, and a list of evaluation criteria, decide for each
criterion whether the response satisfies it.
=== Question ===
{question}
=== Response ===
{response}
=== Criteria ===
1. {criterion_1}
2. {criterion_2}
...
For each criterion, output a JSON object {"id": <int>, "satisfied":
<true|false>}. Output nothing other than a JSON array of these
objects.
```

HealthBench evaluation uses a separate user message. The conversation contains the question context followed by the candidate assistant response. Each input criterion is preceded by its ID and score weight in square brackets. The output maps each ID to a `criteria_met` Boolean, as illustrated in the required format below.

HealthBench: judge prompt

User message.

```
You are an expert evaluator. Judge the last assistant response
against every numbered HealthBench rubric below.
For each rubric, criteria_met is true exactly when the assistant
response exhibits or satisfies the stated criterion. This rule also
applies to negative-point rubrics: criteria_met=true means the
undesirable behavior is present and its negative points will be
applied.
Return only one JSON object. Every supplied rubric ID must appear
exactly once. Each value must be an object containing exactly one
boolean field named criteria_met. Do not add fields or text outside
the JSON object.
Required format:
{
  "1": {"criteria_met": true},
  "2": {"criteria_met": false}
}
<conversation>
{conversation_with_candidate_response}
</conversation>
```

```
<rubrics>
1.  [{weight_1}] {criterion_1}
2.  [{weight_2}] {criterion_2}
...
</rubrics>
```

A.5 Judgments and Score Construction

The mathematical judge receives the problem, reference answer, and candidate response and returns one binary verdict. The open-ended QA judge evaluates all criteria for a response and returns one Boolean decision per criterion. Malformed or missing decisions receive no positive credit and do not enter margin statistics. For criteria with negative weights, verdict polarity is reversed before forming accepted and rejected groups so that "accept" always denotes the reward-favorable decision.

RubricHub and ResearchQA rewards are computed per response by taking a normalized weighted average over its criteria; response rewards are then averaged across the evaluation set. HealthBench likewise yields a response-level score but retains signed criterion weights, so satisfying a negative-weight criterion lowers the score. BVM is instead computed at the criterion level: each criterion produces a separate accept/reject verdict and margin, and all valid criterion decisions receive equal weight without reapplying criterion weights.

BVM requires both accepted and rejected judgments with valid margins. If either group is empty, we leave BVM undefined rather than substituting zero for the missing group mean.

For the response-likelihood analysis in Figure 2, tokens in each response are first assigned to 20 equal relative-position bins. Within each response, token log-probabilities are averaged separately in each bin. The resulting per-response bin means are then averaged equally across responses to obtain an aggregate mean for each relative-position bin. Each cell reports this mean minus its step-0 counterpart.

B Additional Monitoring Evidence

B.1 Reward Divergence across Settings

Figure 9 brings together additional examples of reward hacking under LLM-judge training. In mathematical reasoning with GLM-4-9B, proxy reward rises while independently verified accuracy collapses. In Medicine and Science, rewards from the Qwen2.5-7B judge increasingly diverge from GPT-OSS-120B reference scores on the same responses.

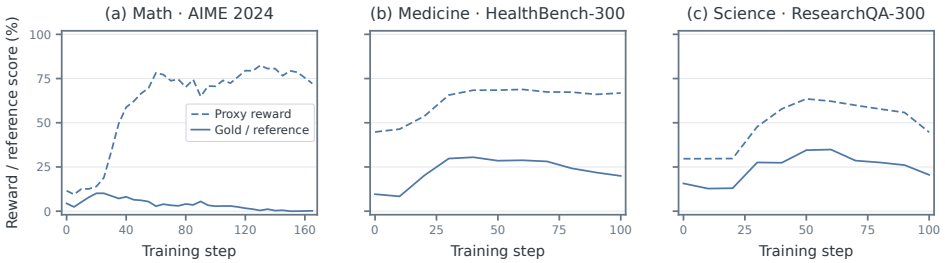


Figure 9: **Proxy–gold/reference divergence during LLM-judge training.** Dashed curves show proxy rewards; solid curves show independent evaluations of the same responses. Policies are GLM-4-9B in Math and Qwen2.5-7B in Medicine and Science. Math uses rule-based verification, while Medicine and Science use GPT-OSS-120B reference judgments.

B.2 Why Balanced Aggregation?

The pooled mean combines changes in conditional confidence with changes in the judge's acceptance rate. At training step t, let ρ_t be the accepted fraction and let $\bar{m}_{t,A}$ and $\bar{m}_{t,R}$ be the accepted

and rejected mean margins. Then

$$M_t^{\text{pool}} - \text{BVM}_t = (\rho_t - \tfrac{1}{2})(\bar{m}_{t,A} - \bar{m}_{t,R}). \tag{7}$$

Thus, pooled and balanced aggregation can differ solely because one verdict class becomes more frequent, even when the two class-conditional means are unchanged.

Table 6 presents selected cases in which BVM decreases under LLM-judge training and increases in the control. In Qwen2.5-32B, accepted-only margins increase in both conditions, while rejected-only and pooled margins decrease in both; BVM is the only one of these four aggregations that moves in opposite directions across the two conditions. In Science, pooled margins also decrease slightly in the stronger-judge control, whereas BVM increases. Qwen2.5-7B provides a further example of BVM separating the conditions, although all four aggregations do so in that setting.

Table 6: Selected comparisons with opposite BVM trends in LLM-judge training and controls. Values are percentage changes from each metric's initial value. Math uses AIME 2024; Science uses ResearchQA. Each pair uses the same evaluation window: steps 0–60 for Qwen2.5-7B, 0–120 for Qwen2.5-32B, and 0–100 for Science.

Setting / policy	Training condition	Accepted-only	Rejected-only	Pooled	BVM
Math / Qwen2.5-7B	LLM judge	−18.6	−72.8	−47.5	−52.1
	Rule-based	+118.2	+5.6	+14.4	+48.2
Math / Qwen2.5-32B	LLM judge	+1.0	−67.6	−40.8	−42.8
	Rule-based	+82.7	−11.2	−5.7	+20.3
Science / Qwen2.5-7B	LLM judge	−10.1	−37.8	−29.1	−25.5
	GPT-OSS-120B	+3.7	+0.1	−0.5	+1.8

C Intervention and Threshold Analysis

C.1 Why Intervene on Positive Rewards?

MRI targets reward-favorable judgments because the growth in judge errors is dominated by false acceptance. Figure 10 traces all four confusion-matrix categories throughout the unmodified LLM-judge runs, rather than comparing endpoints alone. False positives increase sharply in every setting, whereas false negatives remain much smaller and either decline or grow only modestly. This pattern supports attenuating unreliable positive rewards while leaving negative feedback unchanged. At each evaluated training step t, we generate new responses and count judgments in each category. The curves therefore summarize different response batches rather than tracking individual samples changing class.

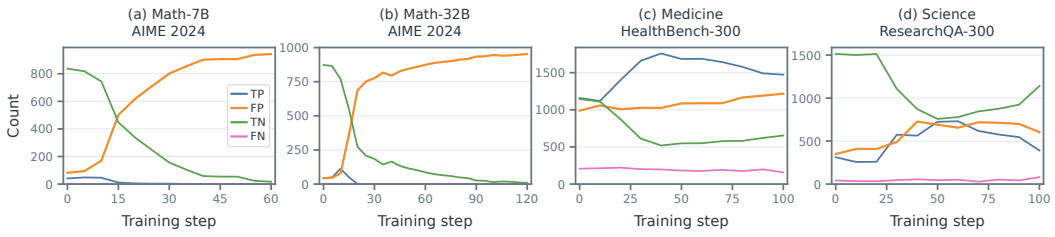


Figure 10: **Confusion-matrix composition during LLM-judge training.** Curves show true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) on AIME 2024, HealthBench, and ResearchQA. Math panels count responses; open-ended QA panels count criterion decisions, so vertical magnitudes should be compared only within a panel.

C.2 THRESHOLD SELECTION

For each intervention, the initial policy is evaluated once and the accepted verdict margins form $\mathcal{M}_0^+$. Offline gold or reference labels are used to select a quantile that covers many false positives (FPs) while exposing few true positives (TPs). The corresponding numerical threshold is then fixed for the entire training run. Table 7 lists the quantiles used in the reported experiments.

Table 7: MRI configurations used in the main experiments. Thresholds are computed on the initial accepted-margin distribution of the listed selection set.

Setting	Policy	Selection set	Form	Quantile
Math	Qwen2.5-7B	AIME 2024	Hard	Q70
Math	Qwen2.5-32B	AIME 2024	Hard	Q50
Math	Qwen3-4B	AIME 2024	Hard	Q50
Math	Qwen3-8B	AIME 2024	Hard	Q30
Math	GLM-4-9B	AIME 2024	Hard	Q50
Medicine	Qwen2.5-7B	RubricHub-Med.-300	Soft	Q35
Science	Qwen2.5-7B	RubricHub-Sci.-300	Soft	Q35

For the soft intervention, both domains use $\lambda = 1$ and $\beta = 0.5$. For every setting, the numerical threshold is determined from its own initial accepted-margin distribution $\mathcal{M}_0^+$ at the selected quantile in Table 7. Invalid margin values retain their original reward rather than being suppressed.

C.3 SELECTIVITY OF THE INTERVENTION REGION

Table 8 evaluates the fixed thresholds after selection. We call the dataset used to select the threshold the *threshold-selection set*: AIME 2024 for Math and the corresponding RubricHub subset for Medicine or Science. HealthBench and ResearchQA provide held-out evaluations of the selected thresholds. FP coverage is the fraction of all accepted FPs below the threshold, while TP retention is the fraction of accepted TPs above it. The low-margin region covers substantially more FPs than TPs in both settings, and the same enrichment remains visible on HealthBench and ResearchQA.

Table 8: Selectivity of the low-margin intervention region. Open-ended QA statistics are computed at the criterion-decision level.

Configuration	Threshold-selection set	FP coverage (%)	TP retention (%)
Qwen2.5-7B Hard Q70	AIME 2024	98.80	86.05
Qwen2.5-32B Hard Q50	AIME 2024	83.33	92.11
Medicine Soft Q35	RubricHub-Medicine-300	41.56	82.25
Science Soft Q35	RubricHub-Science-300	43.10	84.13
Configuration	**Held-out evaluation set**	**FP coverage (%)**	**TP retention (%)**
Medicine Soft Q35	HealthBench-300	26.60	86.79
Science Soft Q35	ResearchQA-300	39.84	88.54

C.4 INTERVENTION RESULTS UNDER DIFFERENT THRESHOLD SETTINGS

In controlled mathematical reasoning, the true-positive and false-positive margin distributions exhibit a comparatively clear separation (Figure 6 and Table 8). We therefore use hard cutoffs and do not conduct an extensive threshold-sensitivity analysis for Math. We focus this analysis on open-ended QA, where the greater overlap makes the choice of intervention form and threshold more consequential.

For open-ended QA, the same Qwen2.5-7B base policy and training protocol were used with Hard Q40/Q50 and Soft Q35/Q40/Q50. In Table 9, step 30 is the latest evaluation shared by every condition. We additionally compare the three soft thresholds at their later common step 50. The No-MRI and Soft-Q35 results use the same runs for each dataset as Table 2. For each condition, both datasets are evaluated using the same training run.

Table 9: Intervention results under different threshold settings for open-ended QA. Medicine is shown on the left and Science on the right. Values are GPT-OSS-120B reference scores (%). All runs use Qwen2.5-7B-Base; best intervention results at each step are in bold.

		Medicine		Science	
Step	Condition	RubricHub	HealthBench	RubricHub	ResearchQA
30	No MRI	31.29	29.76	24.18	27.55
30	Hard Q40	30.27	26.17	27.07	29.54
30	Hard Q50	0.00	−16.06	25.61	28.33
30	Soft Q35	35.55	29.86	26.11	28.24
30	Soft Q40	**35.67**	**30.27**	**27.65**	29.48
30	Soft Q50	32.85	28.42	26.25	**29.98**
50	No MRI	31.05	28.54	28.89	34.50
50	Soft Q35	**38.66**	**35.21**	**37.01**	**40.47**
50	Soft Q40	34.08	32.00	27.74	36.11
50	Soft Q50	37.67	32.84	24.81	29.72

The common-horizon comparison shows substantial sensitivity to intervention strength and form. In particular, Medicine Hard Q50 collapses by step 30; its offline parse-error rate is zero, so the negative HealthBench score is a valid signed score rather than missing output. At step 50, Soft Q35 gives the best intervention score on all four datasets, whereas Q50 lowers both Science scores below the unmodified LLM-judge condition. These single-run diagnostics motivate setting-specific threshold selection rather than a universal quantile.

C.5 Intervention Outcomes

Figure 11 presents two comparisons in which Hard MRI produces BVM trajectories similar to the rule-based control. In the Qwen2.5 runs, both conditions increase BVM, unlike unmodified LLM-judge training. These selected comparisons show that MRI can preserve control-like confidence dynamics while avoiding the contraction observed in the reward-hacking runs.



Figure 11: **BVM under three reward schemes.** Selected comparisons on AIME 2024 show the rule-based control, unmodified LLM-judge training, and Hard MRI. Each curve reports percentage change from its own initial BVM, using the same judge within each panel.