# UNDERSTANDING REWARD OVER-OPTIMIZATION IN RUBRIC-BASED RL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reinforcement Learning (RL) with rubric-based rewards has become a canonical framework for post-training large language models (LLMs) in open-ended domains. However, recent studies have shown that rubric-based RL remains susceptible to reward over-optimization, where the RL-trained policy exploits spurious patterns that satisfy the rubric without necessarily improving the underlying quality of its responses. We first demonstrate the prevalence of this phenomenon in medical and scientific tasks, observing substantial reward over-optimization under both rubric-based and rubric-free evaluation. Motivated by this observation, we systematically investigate the underlying causes through a series of controlled experiments. Our results rule out several potential explanations, showing that neither suboptimal reference guidance or limited coverage of diverse reference responses, nor the use of stronger judges for rubric construction, can fully account for the observed over-optimization. In contrast, we find that factors such as the number of rubrics associated with each prompt, or using preference-based reward with rubric guidance reward, can substantially mitigate reward over-optimization. Taken together, our findings highlight the importance of rubric quantity and reward design in understanding and mitigating reward over-optimization in rubric-based RL.

## 1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has been the canonical framework for Large Language Models (LLMs) in verifiable domains such as mathematics and coding (Lambert et al., 2025; Guo et al., 2025; Zeng et al., 2025). While effective in these domains, it remains challenging in open-ended domains such as medicine (Arora et al., 2025; Li et al., 2025) and instruction following (Zhou et al., 2023; Wen et al., 2024), where straightforward ground-truth labels are typically unavailable.

Early solutions employ preference-based reward models (Ouyang et al., 2022; Zhong et al., 2025). However, they tend to overfit to spurious patterns (e.g., positional bias, verbosity bias) and suffer from reward hacking issues (Yu et al., 2025; Huang et al., 2025; Gao et al., 2023). Recent work instead utilizes prompt-specific rubrics to assess multi-dimensional aspects of the response quality (Gunjal et al., 2026; Viswanathan et al., 2025), providing a more scalable and interpretable signal that extends RLVR to non-verifiable domains. Despite these encouraging results, recent works show that rubric-based RL remains susceptible to reward over-optimization (Mahmoud et al., 2026; Wang et al., 2026), where the policy exploits spurious features to satisfy the rubric and could deviate from the intended goal. While these studies highlight and diagnose this failure mode, they primarily characterize *how* reward hacking manifests during RL post-training, leaving the underlying cause of reward over-optimization poorly understood. This motivates a fundamental question:

*What factors during rubric-based RL training are responsible for reward over-optimization?*

Understanding these factors is crucial for designing more reliable rubric generation and training strategies that can mitigate reward over-optimization while preserving the benefits of rubric-based RL. To answer this question, we begin with a controlled rubric-based RL setting (Sec. 3.2), following Mahmoud et al. (2026), where a *weak verifier* provides the training reward, while a stronger *golden verifier* is used only for evaluation. We observe consistent reward over-optimization across different tasks under both rubric-based (with length-control) and rubric-free evaluation. At the same time,



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.

training exhibits an increasing verbosity bias: response lengths continuously grow as optimization progresses, which sets the stage for subsequent study.

To better understand the sources of reward over-optimization in rubric-based RL, we structure our investigation around a set of *hypotheses*. We first identify several plausible factors that may contribute to reward over-optimization (Sec. 4), and then carefully design controlled experiments to verify whether the hypotheses are valid (Sec. 5). Our investigation focuses on three main dimensions of rubric-based reward design. Firstly, we examine whether the quality and quantity of generated rubrics affect over-optimization. Specifically, we investigate three hypotheses:

- Can constructing rubrics using high-quality reference guidance mitigate reward over-optimization?
- Can generating more rubrics per prompt increase the likelihood of capturing spurious patterns?
- Can using a stronger judge produce higher-quality rubrics and consequently reduce over-optimization?

Our experiments show that neither incorporating high-quality reference guidance (Fig. 3) nor using a more capable judge (Fig. 6) during rubric generation substantially mitigates reward over-optimization. In contrast, reducing the number of rubrics generated per prompt consistently alleviates the problem (Fig. 8), suggesting that rubric quantity plays a more important role than rubric quality in this setting. Secondly, we investigate how the choice of reward formulation influences over-optimization, particularly when rewards are constructed from preference-based rubrics. We find that simply changing the way rubric-level rewards are aggregated can substantially mitigate reward hacking. Taken together, these findings provide a systematic view of the factors underlying reward over-optimization in rubric-based RL and offer practical insights for designing more effective RL fine-tuning methods in open-ended domains.

**We make the following contributions:** First, we systematically examine factors in rubric construction that incentivize reward over-optimization. We demonstrate empirically that neither factors such as stronger reference guidance nor a more capable judge substantially alleviates the problem. In contrast, reducing the number of rubrics generated per prompt consistently reduces over-optimization. Next, by investigating the impacts of reward aggregation strategies on over-optimization, we reveal that the reward formulation itself can substantially shape the severity of reward hacking: even when the underlying rubric-level judgments remain identical, different aggregation schemes can induce distinct optimization pressures and consequently lead to different degrees of reward hacking.

## 2 Related Work

**Rubric Construction for RL Training.** Constructing criteria for evaluating open-ended tasks was initially developed in the context of evaluation (Arora et al., 2025; Yifei et al., 2025). Gunjal et al. (2026) later extended this rubric-based paradigm to RL post-training, using rubrics as reward signals for open-ended tasks. Subsequent works have focused on scaling rubric construction (Liu et al., 2026), or leveraging rubrics to scaffold exploration during RL (Zhou et al., 2026). More recently, He et al. (2026) identified reward hacking during rubric-based RL and manually designed rubrics to mitigate the observed failure modes, where the policies exploit spurious patterns to obtain high rewards. Despite these advances, prior work has primarily focused on improving rubric quality without considering its effect on reward hacking or observing reward hacking issues, leaving how *rubric generation itself* shapes reward over-optimization largely unexplored. We instead systematically study how different rubric generation strategies influence reward over-optimization during RL.

**Reward Hacking and Over-optimization.** Reward hacking, or reward over-optimization in RL (Skalse et al., 2022), occurs when a trained policy maximizes the specified reward through behaviors that deviate from the intended objective, and has been extensively studied in various settings (Taylor, 2016; Everitt et al., 2017), particularly in the context of LLM alignment (Gao et al., 2023; Rafailov et al., 2024; Tang et al., 2024). Early work establishes scaling laws for reward-model over-optimization in RLHF, characterizing how reward-model and preference-data scale affect the severity of reward hacking (Rafailov et al., 2024). Park et al. (2024) show that offline alignment algorithms can overfit to response length at the expense of performance, which is linked to verbosity bias in training data. Subsequent work further examines how preference-data quality, preference loss functions, and the choice of offline versus online training algorithms influence robustness to reward

over-optimization (Tang et al., 2024). Building on this line of work, we investigate reward hacking in rubric-based RL, where rewards are formed by aggregating multiple rubric-level judgments. In particular, we study how the choices in rubric generation and reward formulation shape the policy's optimization and, consequently, the severity and nature of reward hacking.

**Reward Over-optimization in Rubric-based RL.** Early works provide initial evidence that reward hacking emerges in rubric-based RL. He et al. (2026) show that the trained policy can exploit rubric-based rewards by producing artifacts and verbose self-evaluations that inflate their judged scores. Wang et al. (2026) further reproduce such failure modes in controlled settings by injecting systematic biases into the judge and detecting when the policy learns to exploit them. More recently, Mahmoud et al. (2026) attribute rubric reward hacking to a mismatch between the training verifier and a stronger judge; they then develop a framework for diagnosing reward over-optimization in rubric-based domains, showing that improvements under rubric-based evaluation do not necessarily transfer to preference-based judgments. They attribute this gap in part to incomplete rubric coverage, which leaves important failure modes unspecified, and to gains concentrating on presence-based criteria such as completeness, thereby incentivizing verbosity. Another work (Goel et al., 2026) notes related concerns where self-graded rubric gains may not transfer to stronger evaluators. While these works primarily observe and diagnose the prevalence of reward over-optimization, we go a step further by investigating its underlying causes within the rubric-based RL pipeline.

# 3 Experimental settings

## 3.1 Rubric-based RL background

**Rubric Generation.** Rubric-based reinforcement learning replaces exact-matching answers with a scale judge score conditioned on prompt-specific criteria (Gunjal et al., 2026; Arora et al., 2025). Specifically, for each prompt $x$ sampled from a dataset $\mathcal{D}$, a language model (LM) $\mu$ is first prompted to generate a reference guidance $y_{\text{ref}} \sim \mu(\cdot|x)$, representing an ideal response to the prompt. Conditioned on the prompt-reference pair $(x, y_{\text{ref}})$, a rubric generator $\mathcal{R}$ then constructs a set of rubric items $\mathcal{C}$:

$$\mathcal{C} \sim \mathcal{R}(\cdot|x, y_{\text{ref}}) \tag{1}$$

where $\mathcal{C} = \{c_i, w_i\}_{i=1}^{d}$, with $c_i$ as the $i$-th rubric criteria, $w_i$ as its associated weight, and $d = |\mathcal{C}|$ as the number of criteria for prompt $x$.

**RL Fine-tuning.** During RL post-training, the policy model $\pi_\theta$ generates a response $y \sim \pi_\theta(\cdot \mid x)$ for a given prompt $x$. A verifier $v$ then evaluates the response against each criterion in the rubric $\mathcal{C}$, producing a binary judgment vector $g(x, y, \mathcal{C}) = (g_1, \ldots, g_d) \in \{0, 1\}^d$, where $g_i = 1$ indicates that the response satisfies the criterion $c_i$ and 0 otherwise. The overall scalar rubric reward is calculated as a normalized weighted sum of the judgment vector and its corresponding weights (Gunjal et al., 2026; Mahmoud et al., 2026):

$$r(x, y, \mathcal{C}) = \frac{w^\top g(x, y, \mathcal{C})}{||w||_1} \tag{2}$$

where $w = (w_1, \ldots, w_n)$ is a vector of rubric weights. The resulting scalar reward is then used as the training signal for RL fine-tuning.

## 3.2 Controlled setting to study Reward Hacking

**Setup.** We follow the experimental setup from Mahmoud et al. (2026) to study reward hacking in rubric-based RL (see Figure 1 for a visual depiction) with fully controlled settings from the rubric generation stage to the fine-tuning stage. Specifically, during training, the LMs are optimized against a *weak verifier* model $v^{\text{weak}}$, which produces a binary judgment vector $g^{\text{weak}}(x, y, \mathcal{C})$ to aggregate the weak reward $r^{\text{weak}}(x, y)$ using Eq. 2. During evaluation, a better *gold verifier* $v^{\text{gold}}$ will be used to evaluate the underlying improvement of the fine-tuned policy.

**Datasets and Evaluation Metrics.** We study reward hacking in two domains, medicine and science. For training, we use RaR-Medicine and RaR-Science (Gunjal et al., 2026). For evaluation, we use HealthBench-Hard (Arora et al., 2025) and ResearchQA (Yifei et al., 2025). We adopt a length-controlled, rubric-based evaluation protocol to mitigate verbosity bias, following prior work (Chiu

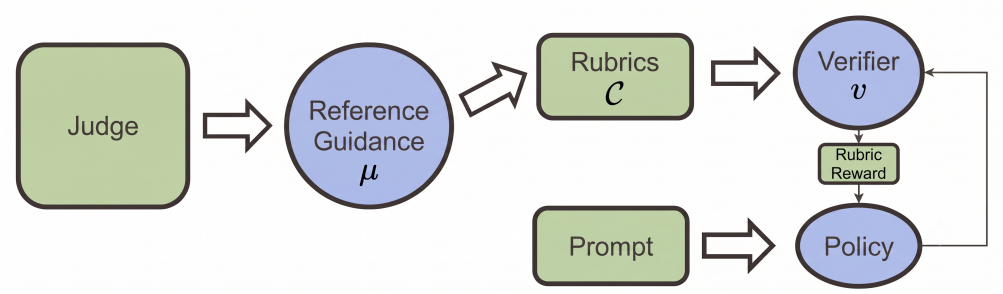



Figure 1: Controlled Experimental Setting in this work, following Mahmoud et al. (2026), we consider a controlled experimental setting in which a given judge uses reference guidance generated by $\mu$ to generate a set of rubrics. These rubrics are then used by a verifier $v$ to provide reward signals for training the policy.

et al., 2026; Dubois et al., 2024). Additionally, we report preference-based scoring against the base model. This evaluation is motivated by previous findings that even rubric-based scoring with a strong verifier can induce reward hacking under *rubric-free* evaluation when rubrics leave important failure modes unspecified (Mahmoud et al., 2026).

**Model Specification.** We use Qwen2.5-3B-Instruct Qwen et al. (2025) as the training policy. During training, Qwen2.5-14B-Instruct serves as the *weak judge*, and GPT-OSS-120B OpenAI et al. (2025) serves as the *gold verifier* for evaluation. Further details of hyperparameters and evaluation settings are provided in Appendix A.

## 4 Hypotheses for Reward Over-optimization in Rubric-based RL.

We first examine the over-optimization behavior that emerges in rubric-based RL. As shown in Fig. 2, across two evaluation types and two domains, we observe a substantial degradation in performance as measured by the golden verifier during RL training, consistent with prior observations in Mahmoud et al. (2026).

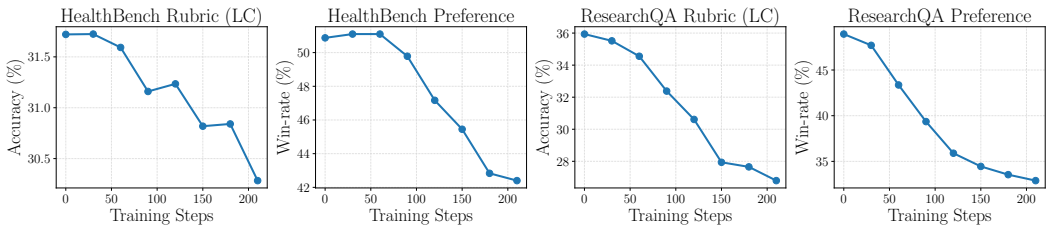



Figure 2: **Reward Over-optimization in rubric-based RL.** The performance of the trained policy significantly degrades over the course of RL training, as measured by the golden verifier $v^{\text{gold}}$ using preference-based and length-controlled reward scores.

The degradation in Fig. 2 raises a central question: *Why does reward hacking occur in rubric-based RL?* This question is particularly important because rubric-based evaluation has been shown to provide more informative guidance than preference-based methods, while offering interpretable evaluation criteria and scalability (Gunjal et al., 2026; Liu et al., 2026). In this section, we propose a set of hypotheses that characterize potential sources of reward hacking in rubric-based RL. These hypotheses are not intended to provide mathematically precise explanations, but rather to offer intuitive perspectives on how the design and optimization of rubric-based rewards may give rise to reward-hacking behavior.

**Hypothesis 1.** ***Sub-optimal reference guidance.*** *Reward over-optimization occurs when the reference model $\mu$ used for rubric generation is suboptimal or ambiguous, leading to misaligned rubrics.*

Since rubric generators typically rely on reference guidance to identify desirable properties and construct evaluation criteria, deficiencies or ambiguities in the reference can propagate directly into the generated rubrics. In particular, even a factually correct reference may be ambiguous or underspecified, causing the rubric generator to focus on superficial or unintended characteristics, increasing the risk of reward over-optimization.

**Hypothesis 2.** ***Better judge, better rubrics.*** *Using a stronger and more capable rubric generator $\mathcal{R}$ can mitigate reward over-optimization.*

The second hypothesis is based on the intuition that a stronger rubric generator $\mathcal{R}$ can better capture the underlying task objective or be better at identifying errors in the reference guidance, leading to more accurate rubrics. In contrast, a weaker rubric generator may generate overly simplistic, noisy, or incomplete criteria, creating opportunities for the policy to exploit gaps between the rubric reward and the true objective.

**Hypothesis 3.** ***The more rubrics there are, the noisier the evaluation becomes.*** *Increasing the number of rubric generator criteria can increase evaluation noise, thereby increasing the risk of reward over-optimization.*

While adding more rubric criteria may provide broader coverage of the task objective, it can also introduce redundancy or low-quality criteria into the evaluation. We therefore hypothesize that there exists a trade-off between rubric coverage and evaluation reliability: beyond a certain number of criteria, additional rubrics may introduce more noise than useful information, increasing the gap between the rubric reward and the true objective.

**Hypothesis 4.** ***The reward function is not the right choice to utilize rubrics.*** *How much of the observed over-optimization can be attributed to the reward aggregation function?*

The current dominant aggregation function computes the reward as a weighted sum of a binary judgment vector and its corresponding weight vector. We therefore question to what extent reward hacking is driven by this particular aggregation function, rather than by the rubric-based evaluation framework itself. Notably, prior work on preference-based RLHF has similarly observed that alternative reward aggregation functions can mitigate or even prevent reward hacking (Munos et al., 2024; Rosset et al., 2024).

# 5 Investigating the hypotheses

## 5.1 Evidence Against Hypothesis 1 (Sub-optimal reference guidance)

In practice, we have little control over prompts and reference guidance used for rubric generation; it might be just that the reference responses were sub-optimal or ambiguous to provide detailed rubrics. If this hypothesis were true, reward hacking can be mitigated by carefully designing reference responses using a more capable language model.

**Task Setting.** To investigate the effect of reference guidance, we construct three types of reference responses for each prompt in $\mathcal{D}$. First, we use Gemma-4-31B-IT Team et al. (2026) as a more capable reference model $\mu^{\text{strong}}$ with strong knowledge and performance on STEM benchmarks to generate *strong reference guidance*. Second, we use Llama-3.2-3B-Instruct Grattafiori et al. (2024) as a weak reference model $\mu^{\text{weak}}$ to generate *weak reference guidance*, with suboptimal performance. We also include reference guidance from the original dataset $\mu_{\text{orig}}$ to serve as an additional baseline.

**High-quality guidance does not mitigate reward over-optimization.** As shown in Fig 3, interestingly, the results show that across various tasks and evaluation metrics, strong reference responses yield the worst performance at the end of training. This implies that high-quality reference guidance is not the right solution to mitigate reward over-optimization.

**What types of reference responses can truly mitigate reward over-optimization?** If the quality of the reference guidance itself is not the primary factor, what properties of the reference responses instead determine their ability to mitigate over-optimization? To answer this question, we investigate two key aspects: *Coverage* measures the diversity of reference guidance available to the rubric generator: rather than conditioning on a single response, we provide a set of reference guidance $\{y^i_{\text{ref}}\}_{i=1}^N$ sampled from different capable models. *Proximity*, in contrast, concerns how closely the

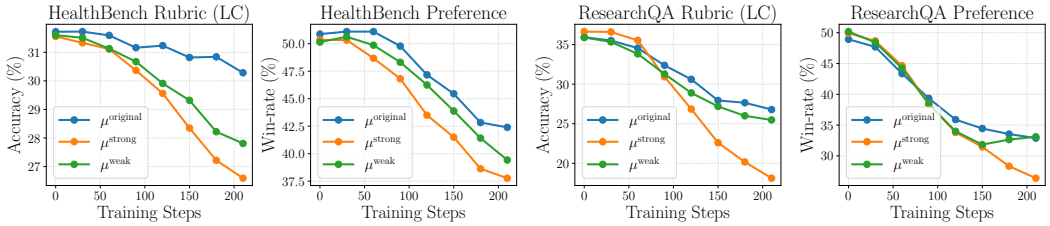



Figure 3: **Evaluation-set golden reward dynamics throughout RL training.** Regardless of the response type used during training, reward over-optimization consistently emerges. Notably, training with strong reference-guidance responses leads to the most pronounced performance degradation compared with other response types.

reference guidance matches the policy being optimized. Specifically, we sample reference guidance directly from the base policy itself, motivated by prior work showing that closer reference policies can mitigate over-optimization in preference-based learning (Tang et al., 2024). Additionally, in the context of rubric-based learning, proximity may be particularly important as off-policy reference responses can produce informative but difficult-to-satisfy rubric criteria, leaving the learned policy with little or no gradient signal for improving on those criteria during training (Tyagi et al., 2026).

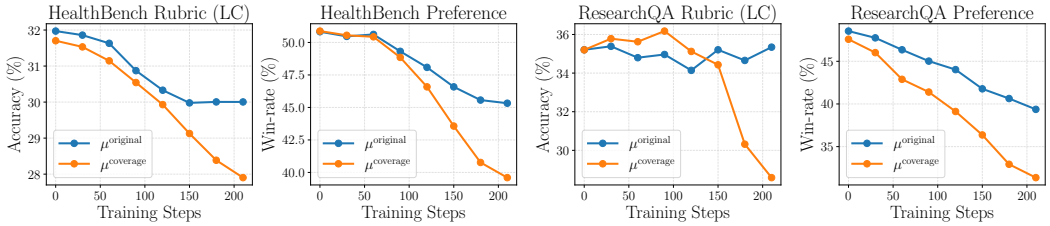



Figure 4: Increasing the number of reference guidance examples (coverage) during rubric generation can actually exacerbate reward over-optimization, resulting in worse final performance than using a single reference.

**Enhancing Coverage could even incentivize reward over-optimization.** We provide our coverage results in Fig. 4, where we find that increasing the coverage of reference guidance consistently worsens performance. In particular, providing more diverse reference responses leads to substantially faster reward over-optimization compared with using a single reference. This result suggests that, rather than mitigating reward over-optimization, additional coverage in reference guidance can provide the rubric generator with more opportunities to identify and exploit spurious features, ultimately accelerating the degradation of the policy.

**Does making the reference guidance proximal to the learning policy help?** We next investigate whether the proximity of reference guidance to the learning policy can mitigate reward over-optimization. As shown in Fig. 5, incorporating proximity information likewise fails to prevent over-optimization and, in fact, performs worse than the original reference-guidance setting.

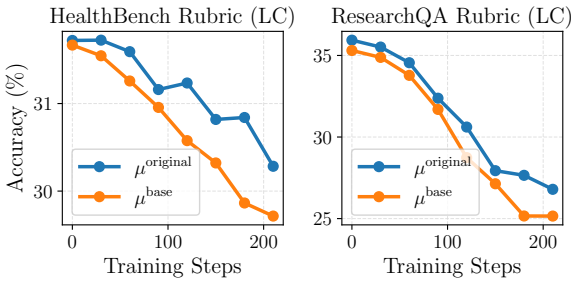



Figure 5: **Performance gap between base model and original reference guidance.** Using the base model as the reference also degrades performance compared to the original reference-guidance setting.

Together, these findings suggest that neither broader coverage nor greater proximity is sufficient to mitigate reward over-optimization. More broadly, they indicate that reference guidance itself may play a limited role in preventing reward over-optimization in rubric-based learning.

### 5.2 Evidence Against Hypothesis 2 (Better rubric generator, better rubrics)

Prior works have investigated the role of verifiers in mitigating reward over-optimization during RL training (Mahmoud et al., 2026). However, the role of the rubric generator $\mathcal{R}$, in particular, whether stronger rubric generation capabilities can mitigate reward hacking, remains under-explored. This perspective may also help explain Hypothesis 1: if a stronger rubric generator might effectively identify issues in its reference guidance, it may produce more reliable rubrics, thereby reducing reward hacking during RL training.

**Task Setting.** To investigate this hypothesis, we consider two rubric generators with different capabilities: a strong rubric generator $\mathcal{R}^{\text{strong}}$ (GPT-OSS-120B) and a weak rubric generator $\mathcal{R}^{\text{weak}}$ (Qwen2.5-14B-Instruct). We use each generator to construct rubrics and examine how rubric-generator capability affects reward over-optimization during RL training.

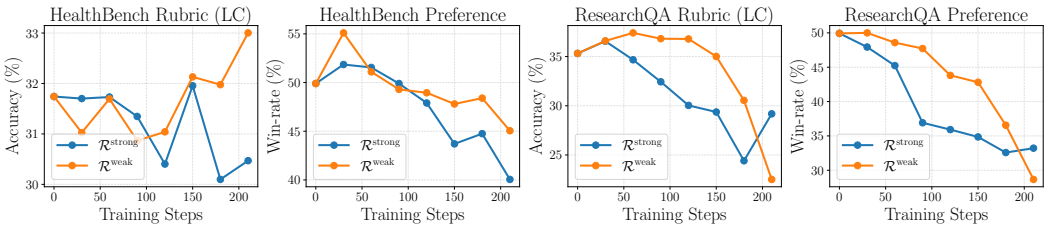



Figure 6: **Policy performance under weak rubric generator $\mathcal{R}^{\text{weak}}$, and strong rubric generator $\mathcal{R}^{\text{strong}}$.** We observe that using the strong rubric generator does not provide clear improvement in both performance and preventing reward over-optimization.

**Stronger judge capabilities in rubric generation do not necessarily translate into greater robustness to reward over-optimization.** The results are shown in Fig. 6. On HealthBench-hard, the strong verifier achieves the best performance and substantially mitigates reward over-optimization compared to the weak verifier. However, on ResearchQA, the fine-tuned models perform similarly, despite the strong rubric generator being significantly more capable than the weak rubric generator in this domain (Fig. 7). This suggests that stronger LLM judge capabilities alone do not necessarily translate into better mitigation of reward over-optimization during RL training.

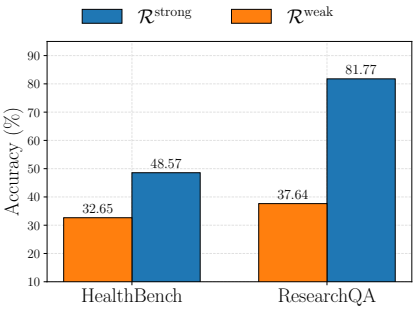



Figure 7: Performance gap between strong and weak rubric generators, using GPT-5.6-Terra as a judge.

### 5.3 Evidence for hypothesis 3 (Reducing the number of rubrics can reduce over-optimization)

To understand the motivation behind this hypothesis, consider each criterion $c_i$ and suppose that the weak and gold verifiers disagree on this criterion with probability $p_i = P(g_i^{\text{weak}} \neq g_i^{\text{gold}})$. Assuming independence across criteria, the probability that *at least one exploitable disagreement exists* is

$$P(\exists i : g_i^{\text{weak}} \neq g_i^{\text{gold}}) = 1 - \prod_{i=1}^{d}(1 - p_i). \tag{3}$$

This probability increases monotonically with the number of criteria $d$ whenever $p_i > 0$. Thus, as the number of rubrics per prompt increases, the policy has more opportunities to increase its reward by exploiting spurious rubric criteria on which the weak and gold verifiers strongly disagree.

**Task Setting.** To test whether the number of rubric criteria per prompt is a key factor driving reward over-optimization, we construct two datasets with different numbers of rubric criteria while keeping the underlying prompts and all other training settings fixed. Specifically, $\mathcal{D}_{1\text{-}5}$ contains examples with between 1 and 5 rubric criteria, while $\mathcal{D}_{7\text{-}20}$ contains examples with between 7 and 20 criteria. This setup allows us to isolate the effect of rubric dimensionality and examine whether increasing the number of criteria systematically increases the severity of reward over-optimization.

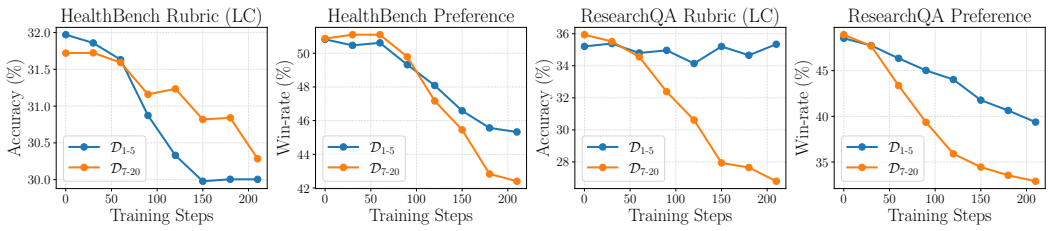



Figure 8: **Reward over-optimization as a function of the number of rubric criteria.** Beyond the rubric-score evaluation on HealthBench, we observe that using fewer rubric criteria can mitigate reward over-optimization under preference-based evaluation, particularly in the Science domain.

**Fewer rubrics can mitigate reward over-optimization.** As shown in Fig. 8, the reward over-optimization effect is substantially reduced compared with a larger number of rubrics. In particular, $\mathcal{D}_{1\text{-}5}$ exhibits a slower degradation in gold-verifier performance throughout RL training, whereas $\mathcal{D}_{7\text{-}20}$ shows considerably stronger over-optimization behavior, consistent with the hypothesis. Importantly, the result also suggests that rubric dimensionality alone is not sufficient to eliminate reward over-optimization, as the weak verifier can still contain exploitable discrepancies. Rather, the number of criteria appears to modulate the extent to which these discrepancies can be exploited.

**Fewer rubrics improve learning efficiency beyond weaker training signals.** One argument is that fewer rubrics simply provide a weaker training signal, causing the policy to remain closer to the base model $\pi_0$ and therefore limiting both reward over-optimization and genuine improvement. To distinguish between these explanations, we examine the relationship between win rate and KL divergence from the base policy (Rafailov et al., 2024; Gao et al., 2023). As shown in Fig. 9. We find that, at the same KL budget, training with fewer rubrics consistently achieves larger improvements in win rate than training with more rubrics. Thus, the improved behavior of $\mathcal{D}_{1\text{-}5}$ cannot be attributed solely to making smaller updates from $\pi_0$.

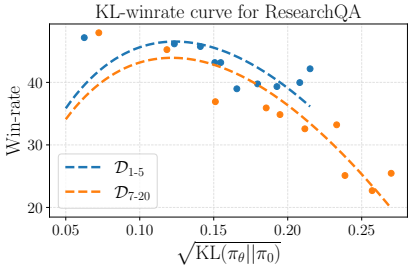



Figure 9: **KL-winrate scaling law curve fit on ResearchQA.** Using fewer rubrics enables greater improvement while reducing the policy's ability to exploit discrepancies between the weak and gold verifiers.

### 5.4 Evidence for Hypothesis 4

As discussed above, one might question whether reward over-optimization is primarily driven by the design of the reward function rather than by the rubric information itself. This raises a natural question: *if we replace the weighted-sum reward with an alternative learning signal that still incorporates rubric feedback, can we mitigate reward over-optimization?*

This question is motivated by the preference learning literature, which has highlighted several limitations of reward modeling, including vulnerability to distribution shift and limited expressivity in capturing complex human preferences (Munos et al., 2024; Rosset et al., 2024). To isolate the role of reward design, we propose an alternative RL objective based on a *preference model with rubric guidance*. Specifically, we optimize the policy according to following objective:

$$\max_{\pi_\theta} \mathbb{E}_{(x,\mathcal{C})\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x),\, y'\sim\pi'(\cdot|x)} \left[P(y \succ y' \mid x, \mathcal{C}) - \beta \mathrm{KL}(\pi_\theta \| \pi_0)\right], \tag{4}$$

where $P(y \succ y' \mid x, \mathcal{C})$ denotes the preference model, which estimate the preference that response $y$ is preferred over a reference response $y'$ based on the set of rubrics $\mathcal{C}$ and the prompt $x$.

**Task setting.** We compare the original rubric-based weighted-sum reward with this preference-based objective. This comparison allows us to examine whether changing the reward formulation, while keeping the underlying rubric information fixed, is sufficient to mitigate reward over-optimization. We summarize our observations below:

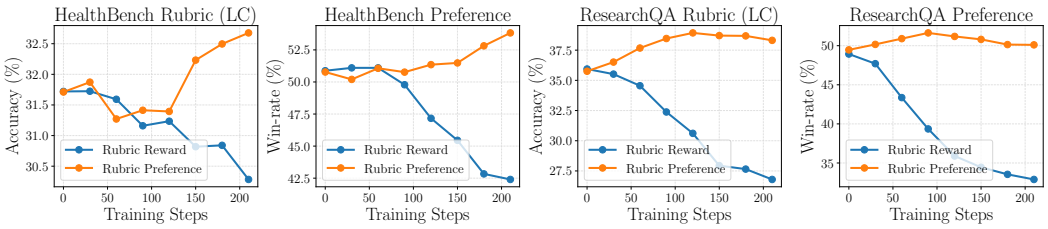



Figure 10: **Performance gap between Rubric reward aggregation and preference-based learning with rubric guidance.** Utilize preference effectively to mitigate reward hacking issues and exhibit little to no performance degradation throughout the RL training process.

**Preference-based rubric aggregation effectively addresses reward over-optimization.** As shown in Fig. 10, *preference-with-guidance-rubric* effectively mitigates reward over-optimization across all domains and evaluation strategies, while exhibiting consistent performance improvements throughout the RL training process. These results suggest that reward over-optimization is not inherent to rubric-based RL itself, but can be substantially mitigated through an appropriate reward design.

**Preference-based with rubric guidance effectively mitigates verbosity bias.** Verbosity has been shown to be strongly associated with presence-based rubric satisfaction during training, but can also increase the risk of hallucinations Gunjal et al. (2026), particularly in medical domains where factual accuracy is critical. To further examine this verbosity bias, we compare rubric-based reward optimization with preference-based learning by tracking both response length and factual correctness throughout RL training, as shown in Fig. 11. We find that optimizing against the rubric reward doubles response length while factual correctness consistently decreases, providing a clear indication of verbosity bias. In contrast, optimizing against the preference model substantially mitigates this effect. However, both approaches do not increase model factuality, highlighting the limitation of rubric-based learning in improving factuality.

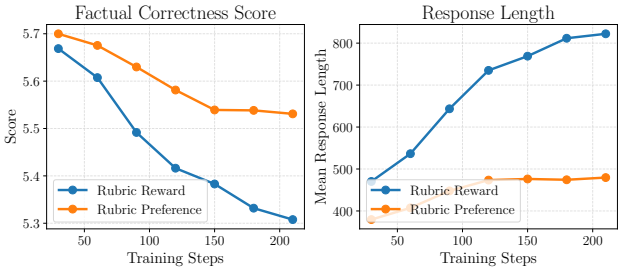



Figure 11: **Factual correctness and response length under rubric-reward aggregation and preference-based learning on HealthBench.** Preference-based learning substantially reduces verbosity bias, but neither approach yields improvements in factual correctness.

## 6 Conclusion

We investigate the sources of reward over-optimization in rubric-based reinforcement learning through *hypothesis testing*. We evaluate several plausible explanations for why policies exploit imperfections in rubric-based rewards using controlled empirical experiments. Our results show that stronger rubric generators or reference guidance provide limited mitigation, whereas the *number of rubrics per prompt* and the *aggregation of rubric-level rewards* have substantially greater impact on preventing reward over-optimization.

These findings offer practical guidance for designing more robust rubric-based rewards. More broadly, they suggest that improving rubric generation alone is insufficient; greater attention should be paid to how rubric-level signals are constructed and aggregated to develop more reliable reward mechanisms.

**Limitations.** While our analysis provides significant empirical insights – across different models, datasets, and objectives – into reward hacking mechanisms in rubric-based RL, a complete theoretical characterization of these findings remains beyond the scope of this work. Developing a formal theoretical framework that explains these observations and establishes the conditions under which they arise remains an important direction for future work. We therefore view our empirical findings as a foundation for such theoretical investigations.

### AI USE STATEMENT

In this work, we used AI models solely for revising the writing of the paper. Note that revision here strictly means enhancing the clarity and readability of the text (e.g., fixing typos or constructing LaTeX tables), and not for any other purposes.

### REPRODUCIBILITY STATEMENT

We provide detailed experimental settings, training procedures, models, datasets, evaluation protocols, and hyperparameters in Appendix A and B to facilitate reproducibility.

### ETHICS STATEMENT

This work investigates reward over-optimization in rubric-based RL using existing benchmark datasets and pretrained LLMs, without involving human subjects or collecting new personal data. We conduct controlled experiments in medical and scientific question-answering domains to study how rubric construction and reward design affect reward-hacking behavior. Our findings may contribute to more reliable and robust RL post-training methods, but insights into weaknesses of automated reward signals could also facilitate attempts to exploit such evaluators. We therefore encourage careful consideration of the reliability and safety of reward mechanisms when applying these methods in practice, particularly in high-stakes domains such as medical.

## REFERENCES

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, 2025. URL https://arxiv.org/abs/2505.08775.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024. URL https://arxiv.org/abs/2412.18925.

Yu Ying Chiu, Michael S. Lee, Rachel Calcott, Brandon Handoko, Paul de Font-Reaulx, Paula Rodriguez, Chen Bo Calvin Zhang, Ziwen Han, Udari Madhushani Sehwag, Yash Maurya, Christina Q Knight, Harry R. Lloyd, Florence Bacus, Mantas Mazeika, Bing Liu, Yejin Choi, Mitchell L Gordon, and Sydney Levine. Morebench: Evaluating procedural and pluralistic moral reasoning in language models, more than outcomes. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RMwJXp5Kb1.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel, 2017. URL https://arxiv.org/abs/1705.08417.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Shashwat Goel, RISHI HAZRA, Dulhan Jayalath, Timon Willi, Parag Jain, William F. Shen, Ilias Leontiadis, Francesco Barbieri, Yoram Bachrach, Jonas Geiping, and Chenxi Whitehouse. Training AI co-scientists using rubric rewards. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=bTySLCPXHO.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,

Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-

nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c1bTcrDmt4.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan

Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, Shengjie Bi, Shishir G Patil, Qi Qi, Shengyu Feng, Julian Katz-Samuels, Richard Yuanzhe Pang, Sujan Kumar Gonugondla, Hunter Lang, Yue Yu, Yundi Qian, Maryam Fazel-Zarandi, Licheng Yu, Amine Benhalloum, Hany Hassan Awadalla, and Manaal Faruqui. AdvancedIF: Rubric-based benchmarking and reinforcement learning for advancing LLM instruction following. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18003–18022, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.820. URL https://aclanthology.org/2026.acl-long.820/.

Rebecca Soskin Hicks, Mikhail Trofimov, Dominick Lim, Rahul K. Arora, Foivos Tsimpourlas, Preston Bowman, Michael Sharman, Chi Tong, Kavin Karthik, Arnav Dugar, Akshay Jagadeesh, Khaled Saab, Johannes Heidecke, Ashley Alexander, Nate Gross, and Karan Singhal. Healthbench professional: Evaluating large language models on real clinician chats, 2026. URL https://arxiv.org/abs/2604.27470.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo Ponti, and Ivan Titov. Post-hoc reward calibration: A case study on length bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Iu8RytBaji.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, and Yang Liu. Agent hospital: A simulacrum of hospital with evolvable medical agents, 2025. URL https://arxiv.org/abs/2405.02957.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.791. URL https://aclanthology.org/2026.acl-long.791/.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *CoRR*, abs/2501.15587, January 2025. URL https://doi.org/10.48550/arXiv.2501.15587.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning. In *Second Workshop on Agents in the Wild: Safety, Security, and Beyond*, 2026. URL https://openreview.net/forum?id=X5sPkeE60f.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea

Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Y5AmNYiyCQ.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4998–5017, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.297. URL https://aclanthology.org/2024.findings-acl.297/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pf4OuJyn4Q.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences, 2024. URL https://arxiv.org/abs/2404.03715.

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

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=yb3HOXO3lX2.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, and Will Dabney. Understanding the performance gap between online and offline alignment algorithms, 2024. URL https://arxiv.org/abs/2405.08448.

Jessica Taylor. Quantilizers: A safer alternative to maximizers for limited optimization. In *AAAI Workshop: AI, Ethics, and Society*, 2016. URL https://api.semanticscholar.org/CorpusID:10865865.

5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin, Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang, Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier

Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Utkarsh Tyagi, Xingang Guo, MohammadHossein Rezaei, Daniel George, Anas Mahmoud, Jackson Lee, Bing Liu, and Yunzhong He. Not every rubric teaches equally: Policy-aware rubric rewards for rlvr, 2026. URL https://arxiv.org/abs/2605.20164.

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RPRqKhjrr6.

Xuekang Wang, Zhuoyuan Hao, Shuo Hou, Hao Peng, Juanzi Li, and Xiaozhi Wang. Reproducing, analyzing, and detecting reward hacking in rubric-based reinforcement learning, 2026. URL https://arxiv.org/abs/2606.04923.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxing Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=U2aVNDrZGx.

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, 2025. URL https://arxiv.org/abs/2509.00496.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. Self-generated critiques boost reward modeling for language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11499–11514, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.573. URL https://aclanthology.org/2025.naacl-long.573/.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. Naturalreasoning: Reasoning in the wild with 2.8m challenging questions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=CF2OfwkYdM.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. SimpleRL-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vSMCBUgrQj.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future, 2025. URL https://arxiv.org/abs/2504.12328.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Chen Wei, Xie Yan, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for open-ended LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IM8ZHjhD0v.

# A Implementation Details

**Rubric generation.** For rubric generation, we use a temperature of 1.0 and a maximum generation length of 8192 tokens. To study the effect of reference-model coverage, we provide the judge model with five responses sampled from different capable models, spanning a range of model families and scales: Gemma-4-31B-IT Team et al. (2026), GLM-4.7-Flash Team et al. (2025), Mistral-Small-4-119B-2603, and Qwen3.6-35B-A3B. The judge model then generates rubrics conditioned on these diverse reference responses, allowing us to examine how broader coverage of capable models affects rubric quality and subsequent evaluation.

**RL training configurations.** All experiments are implemented using the `verl` framework Sheng et al. (2024) with a learning rate of $\eta = 10^{-6}$ and a KL coefficient of $\beta = 5 \times 10^{-3}$. We use the GRPO algorithm Shao et al. (2024) with a group size of $G = 8$ for 210 training steps. For rollout sampling, we use a temperature of 1.0 and a maximum response length of 2048 tokens for the medical domain and 3072 tokens for the science domains, following prior work Mahmoud et al. (2026). We set the training batch size of 64 prompts and a mini-batch size of 32. For reward calculation, the verifiers first generate a binary judgment vector $g$ greedily, which is then used to compute the reward using Eq. 2.

**Evaluation Details.** To quantify the variance of the evaluation set, we evaluate each model using multiple random seeds over the evaluation prompts for both preference-based and rubric-based evaluation. Specifically, we perform self-consistency evaluation with a judge temperature of 1.0, aggregating the resulting scores across multiple runs to obtain a more robust estimate of model performance. For preference-based evaluation, we use random flipping to mitigate positional bias.

**Length Control.** Similar criterion-fulfillment benchmarks, such as HealthBench Arora et al. (2025), may favor more verbose responses because longer responses provide more opportunities to satisfy individual evaluation criteria. To disentangle genuine improvements in criterion fulfillment from gains attributable to increased response length, we follow prior work (Chiu et al., 2026; Dubois et al., 2024; Hicks et al., 2026) and apply length control to the reward. Specifically, given a reward $r(x, y)$ and response length $l_i$, we define the length-controlled reward as

$$r_{\text{LC}}(x, y) = r(x, y) - \alpha(l_i - l_0), \tag{5}$$

where $l_0$ denotes the median response length of the base model. The coefficient $\alpha$ is estimated by fitting a linear regression of the reward against response length, thereby capturing the empirical relationship between verbosity and reward. This adjustment removes the component of the reward attributable to deviations from the base model's typical response length, allowing us to evaluate reward changes independently of response-length effects.

# B Dataset Details

## B.1 Training Dataset

We select two rubric datasets for training and to investigate the effect of reward over-optimization across 2 domains: RaR-Medicine and RaR-Science. Both datasets provide fine-grained, verifiable evaluation criteria.

**RaR-Medicine** Gunjal et al. (2026) is a medical reasoning dataset comprising 20K prompts curated from diverse medical reasoning sources, including medical-o1-reasoning-SFT Chen et al. (2024), Natural Reasoning Yuan et al. (2026), SCP-116K Lu et al. (2025), and GeneralThought-430K. To enable reliable scoring, RAR-Medicine provides instance-specific rubrics, which are automatically generated using GPT-4o, allowing each response to be evaluated against prompt-specific criteria.

**RaR-Science** Gunjal et al. (2026) Similar to RaR-Medicine, this dataset consists of 20K prompts curated to align with GPQA-Diamond categories. Prompts are sourced from natural_reasoning Yuan et al. (2026), SCP-116K Lu et al. (2025), and GeneralThought-430K, covering a broad range of scientific reasoning tasks.

## B.2 Evaluation Dataset

**HealthBench-Hard** Arora et al. (2025) is a medical benchmark designed to evaluate LLMs across several core medical competencies, including knowledge, understanding, reasoning, safety, and

medical text generation. Its hard subset consists of 1,000 questions selected for their difficulty on current frontier models. The benchmark also provides an automated evaluation pipeline that uses expert-developed checklists within an LLM-as-a-Judge framework, enabling more consistent evaluation by grounding judgments in explicit criteria.

**ResearchQA** Yifei et al. (2025) is a scientific question-answering benchmark derived from survey articles spanning 75 domains and containing approximately 21K questions. We use its validation subset, which consists of 703 examples, for evaluation. To avoid potential data contamination, we ensure that none of the validation examples overlap with the training examples used in the RaR-Science dataset.

## C Scaling-law curve fit details

Previous work on classical Reinforcement Learning from Humaan Feedback (RLHF) has established scaling laws relating reward-model scores to the KL divergence between the initial and optimized policies (Gao et al., 2023; Rafailov et al., 2024). Specifically, the reward follows the functional form $R(d)$:

$$R(d) = d(\alpha - \beta \log d) \tag{6}$$

where $\alpha, \beta$ are the fitted parameters, and $d = \sqrt{\mathrm{KL}(\pi_\theta || \pi_0)}$,

## D Prompt for Synthetic Rubric Generation

**Prompt for Synthetic Rubrics Generation**

You are an expert rubric writer. Your job is to generate a self-contained set of evaluation criteria ("rubrics") for judging how good a response is to a given question. Rubrics can cover aspects of a response such as, but not limited to, factual correctness, ideal-response characteristics, style, completeness, helpfulness, harmlessness, patient-centeredness, depth of reasoning, contextual relevance, and empathy. Each item must be self-contained; non-expert readers should not need to infer anything or consult external information. Begin each description with its category: "Essential Criteria: ...", "Important Criteria: ...", "Optional Criteria: ...", or "Pitfall Criteria: Does not mention ...".
Inputs:

- `question`: The full question text.
- `reference_answer`: The ideal answer, including any specific facts, explanations, or advice.

Total items:

- Choose `min_rubrics`–`max_rubrics` rubric items based on the complexity of the question.

Each rubric item:

- **title** (2–4 words).
- **description**: One sentence starting with its category prefix that explicitly states exactly what to look for. For example:
  - Essential Criteria: Identifies non-contrast helical CT scan as the most sensitive modality for ureteric stones.
  - Pitfall Criteria: Does not mention identifying (B) as the correct answer.
  - Important Criteria: Explains that non-contrast helical CT detects stones of varying sizes and compositions.
  - Optional Criteria: States "The final answer is (B)" or similar answer choice formatting.
- **weight**: For Essential/Important/Optional, use 1–5 (5 = most important); for Pitfall, use –1 or –2.

Category guidance:

- **Essential**: Critical facts or safety checks; if missing, the response is invalid (weight 5).
- **Important**: Key reasoning, completeness, or clarity; strongly affects quality (weight 3–4).
- **Optional**: Helpful style or extra depth; nice to have but not deal-breaking (weight 1–2).

- **Pitfall**: Common mistakes or omissions specific to this prompt—identify things a respondent often forgets or misstates. Each Pitfall description must begin with "Pitfall Criteria: Does not mention ..." or "Pitfall Criteria: Recommends ..." and use weight –1 or –2.

To ensure self-contained guidance:

- When referring to answer choices, explicitly say "Identifies (A)", "Identifies (B)", etc., rather than vague phrasing.
- If the format requires a conclusion like "The final answer is (B)", include a rubric item such as:
  - Essential Criteria: Includes a clear statement "The final answer is (B)".
- If reasoning should precede the answer, include a rubric like:
  - Important Criteria: Presents the explanation before stating the final answer.
- If brevity is valued, include a rubric like:
  - Optional Criteria: Remains concise and avoids unnecessary detail.
- If the question context demands mention of specific findings, include that explicitly (e.g., "Essential Criteria: Mentions that CT does not require contrast").

Output: Provide a JSON array of rubric objects. Each object must contain exactly three keys—title, description, and weight. Do not copy large blocks of the question or reference_answer into the text. Each description must begin with its category prefix, and no extra keys are allowed.
Now, given the question and reference_answer, generate the rubric as described. The reference answer is an ideal response but not necessarily exhaustive; use it only as guidance.

# E Prompt for Rubric Generation with Multiple Reference Guidance

**Prompt for Synthetic Rubrics Generation**

You are an expert rubric writer. Your job is to generate a self-contained set of evaluation criteria ("rubrics") for judging how good a response is to a given question. Rubrics can cover aspects of a response such as, but not limited to, factual correctness, ideal-response characteristics, style, completeness, helpfulness, harmlessness, patient-centeredness, depth of reasoning, contextual relevance, and empathy. Each item must be self-contained; non-expert readers should not need to infer anything or consult external information. Begin each description with its category: "Essential Criteria: ...", "Important Criteria: ...", "Optional Criteria: ...", or "Pitfall Criteria: Does not mention ...".
Inputs:

- `question`: The full question text.
- `reference_answers`: A list of ideal answers separated by newlines, including any specific facts, explanations, or advice.

Total items:

- Choose `min_rubrics`–`max_rubrics` rubric items based on the complexity of the question.

Each rubric item:

- **title** (2–4 words).
- **description**: One sentence starting with its category prefix that explicitly states exactly what to look for. For example:
  - Essential Criteria: Identifies non-contrast helical CT scan as the most sensitive modality for ureteric stones.
  - Pitfall Criteria: Does not mention identifying (B) as the correct answer.
  - Important Criteria: Explains that non-contrast helical CT detects stones of varying sizes and compositions.
  - Optional Criteria: States "The final answer is (B)" or similar answer choice formatting.
- **weight**: For Essential/Important/Optional, use 1–5 (5 = most important); for Pitfall, use –1 or –2.

Category guidance:

- **Essential**: Critical facts or safety checks; if missing, the response is invalid (weight 5).
- **Important**: Key reasoning, completeness, or clarity; strongly affects quality (weight 3–4).
- **Optional**: Helpful style or extra depth; nice to have but not deal-breaking (weight 1–2).
- **Pitfall**: Common mistakes or omissions specific to this prompt—identify things a respondent often forgets or misstates. Each Pitfall description must begin with "Pitfall Criteria: Does not mention ..." or "Pitfall Criteria: Recommends ..." and use weight –1 or –2.

To ensure self-contained guidance:

- When referring to answer choices, explicitly say "Identifies (A)", "Identifies (B)", etc., rather than vague phrasing.
- If the format requires a conclusion like "The final answer is (B)", include a rubric item such as:
  - Essential Criteria: Includes a clear statement "The final answer is (B)".
- If reasoning should precede the answer, include a rubric like:
  - Important Criteria: Presents the explanation before stating the final answer.
- If brevity is valued, include a rubric like:
  - Optional Criteria: Remains concise and avoids unnecessary detail.
- If the question context demands mention of specific findings, include that explicitly (e.g., "Essential Criteria: Mentions that CT does not require contrast").

Output: Provide a JSON array of rubric objects. Each object must contain exactly three keys—title, description, and weight. Do not copy large blocks of the question or reference_answer into the text. Each description must begin with its category prefix, and no extra keys are allowed.
Now, given the question and reference_answers, generate the rubric as described. The reference answers are a list of ideal responses but not necessarily exhaustive; use them only as guidance.

## F Prompt for Preference-based Evaluation and Training

**Evaluation Prompt Template**

**[SYSTEM]**
You are an expert evaluator. Given a user prompt, two candidate responses (A and B), and a list of quality rubrics, choose which response better satisfies the rubrics overall. Consider all rubrics holistically. Prefer the response that satisfies more of the important criteria and avoids more of the pitfalls. You must choose exactly one of A or B. Do not declare a tie. Start your response with a valid JSON object that contain a single key "preferred" and the value should be either "A" or "B". Example response:
{"preferred": "A"}"

**[USER]**
Given the following prompt, two responses, and rubrics, choose which response better satisfies the rubrics.

**[prompt]**
{prompt}

**[response_A]**
{response_A}

**[response_B]**
{response_B}

**[rubrics]**
{rubrics}

Your JSON Evaluation: