RubricHub: A Comprehensive and Highly Discriminative Rubric Dataset via Automated Coarse-to-Fine Generation

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has driven substantial progress in reasoning-intensive domains like mathematics. However, optimizing open-ended generation remains challenging due to the lack of ground truth. While rubric-based evaluation offers a structured proxy for verification, existing methods suffer from scalability bottlenecks and coarse criteria, resulting in a supervision ceiling effect. To address this, we propose an automated Coarse-to-Fine Rubric Generation framework. By synergizing principle-guided synthesis, multi-model aggregation, and difficulty evolution, our approach produces comprehensive and highly discriminative criteria capable of capturing the subtle nuances. Based on this framework, we introduce RubricHub, a large-scale (∼110k) and multi-domain dataset. We validate its utility through a two-stage post-training pipeline comprising Rubric-based Rejection Sampling Fine-Tuning (RuFT) and Reinforcement Learning (RuRL). Experimental results demonstrate that RubricHub unlocks significant performance gains: our post-trained Qwen3-14B achieves state-of-the-art (SOTA) results on HealthBench (69.3), surpassing proprietary frontier models such as GPT-5.

1 Introduction

Large Language Models (LLMs) are now widely deployed in real-world applications, making reliable evaluation of response quality increasingly important (Zheng et al., 2023b; Chang et al., 2024; Liang et al., 2022). In verifiable domains like mathematics and coding, Reinforcement Learning with Verifiable Rewards (RLVR) has driven substantial progress in complex reasoning, as seen in DeepSeek R1 (Guo et al., 2025; Lambert et al., 2024). In contrast, most real-world queries are open-ended and lack ground-truth answers, leading to subjective and unstable quality judgments. Recent studies (Arora et al., 2025; Team et al., 2025; Liu et al., 2025a) show that rubric-based evaluation mitigates this issue by decomposing quality into explicit, checkable criteria. By serving as a structured proxy for verification, rubrics yield interpretable assessments and more stable training signals, narrowing the gap between verifiable reasoning and open-ended generation (Gunjal et al., 2025; Huang et al., 2025; Zhou et al., 2025).

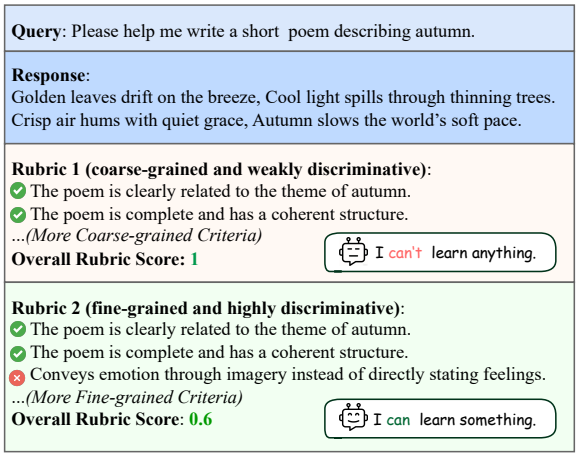

Figure 1: Motivating Example. Comparison between coarse-grained and fine-grained evaluation. Coarse rubrics (Rubric 1) result in indistinguishable high scores, whereas RubricHub (Rubric 2) utilizes highly discriminative criteria to reveal specific weaknesses, providing richer signals for alignment.

Despite their promise, existing rubrics face critical bottlenecks that hinder scalability. (i) *Reliance on Manual Expertise:* High-quality rubric creation demands expensive human effort, hindering its scalability. (Starace et al., 2025; Arora et al., 2025). (ii) *Narrow Domain Breadth:* Current datasets (Gunjal et al., 2025) are confined to specialized domains, restricting their utility for general-purpose LLMs. (iii) *Low Discriminability:* As illustrated in Figure 1, existing rubrics often rely on coarse, generic criteria that fail to capture subtle nuances. Consequently, they struggle to

distinguish superficially plausible responses from truly high-quality ones (Zhang et al., 2025a), creating *ceiling effects* in supervision signals.

To overcome these bottlenecks, we propose a fully automated **Coarse-to-Fine Rubric Generation** framework. First, we synthesize candidate criteria using a response-grounded and principle-guided strategy to maintain alignment with query intent. Second, we aggregate diverse perspectives from heterogeneous models to ensure comprehensiveness, mitigating single-source biases. Crucially, to increase discriminability, we employ a difficulty evolution mechanism. Instead of stopping at generic criteria, this mechanism evolves criteria to capture the discriminative nuances of exceptional responses, ensuring the rubric remains challenging enough to guide the alignment of top-tier models. Based on this framework, we construct **RubricHub**, a large-scale ($\sim$110k), and multi-domain rubric dataset characterized by fine-grained supervision and high discriminative power.

To validate the practical utility of RubricHub, we implement a two-stage post-training pipeline: (i) Rubric-based Rejection Sampling Fine-Tuning (RuFT), where rubrics act as robust filters to curate high-quality data; and (ii) Rubric-based Reinforcement Learning (RuRL), where rubric scores serve as reward signals for policy optimization. Experimental results demonstrate that RubricHub unlocks substantial gains. By post-training Qwen3-14B-Base, we achieve a 22.6-point lead over its official Instruct counterpart (Non-thinking) on HealthBench. Remarkably, our model even surpasses the frontier GPT-5 (69.3 vs. 67.2), despite being significantly smaller.

Our main contributions are as follows:

- We propose an automated Coarse-to-Fine Rubric Generation framework. It synergizes principle-guided and response-grounded synthesis, multi-model aggregation, and difficulty evolution to construct fine-grained criteria, thereby ensuring comprehensive evaluation coverage, capturing subtle quality nuances, and mitigating the supervision ceiling effect.
- We introduce RubricHub, a large-scale ($\sim$110k) and multi-domain rubric dataset, providing fine-grained and highly discriminative supervision for general-purpose LLMs.
- We validate RubricHub via a rubric-driven post-training pipeline (RuFT and RuRL), enabling Qwen3-14B to achieve SOTA performance on HealthBench, notably outperforming proprietary models (e.g., GPT-5).

2 Preliminaries

2.1 Rubric

Rubrics are structured scoring guides that define evaluation criteria and performance levels, widely used to assess output quality in education and model evaluation. For each query q, we define a fine-grained evaluation rubric $\mathcal{R}_q$ as a set of N_q weighted criteria:

$$\mathcal{R}_q = \{(c_i, w_i)\}_{i=1}^{N_q}, \tag{1}$$

where each criterion c_i encompasses semantic requirements and grader parameters. Criteria are categorized into two types: (1) *Verifiable Criteria*, representing objective constraints (e.g., format or word count) assessed via rule-based systems $\mathcal{G}_{\text{rule}}$; and (2) *Semantic Criteria*, capturing qualitative attributes (e.g., reasoning depth or tone) that require LLM-based evaluators $\mathcal{G}_{\text{LLM}}$. The weight w_i determines each criterion's importance, providing the basis for structured reward signals $r(q, o)$.

2.2 Task Formulation

We formulate rubric generation as a conditional task where an LLM $\mathcal{M}$ synthesizes a rubric $\mathcal{R}$ given input context I. By defining a prompt function $P(\cdot)$ that formats I into instructions, the process is:

$$\mathcal{R} = \mathcal{M}\big(P(I)\big). \tag{2}$$

In Section 3, we instantiate specific templates (e.g., $P_{\text{gen}}, P_{\text{agg}}$) to generate and refine rubrics through multiple stages.

3 Method

In this section, we introduce our automated **Coarse-to-Fine Rubric Generation** framework. As illustrated in Figure 2, we detail the core rubric generation pipeline, which operates in three phases: (1) *Principle-Guided & Response-Grounded Generation*, (2) *Multi-Model Aggregation*, and (3) *Difficulty Evolution*. Finally, we analyze the resulting dataset characteristics and detail how RubricHub is utilized for post-training.

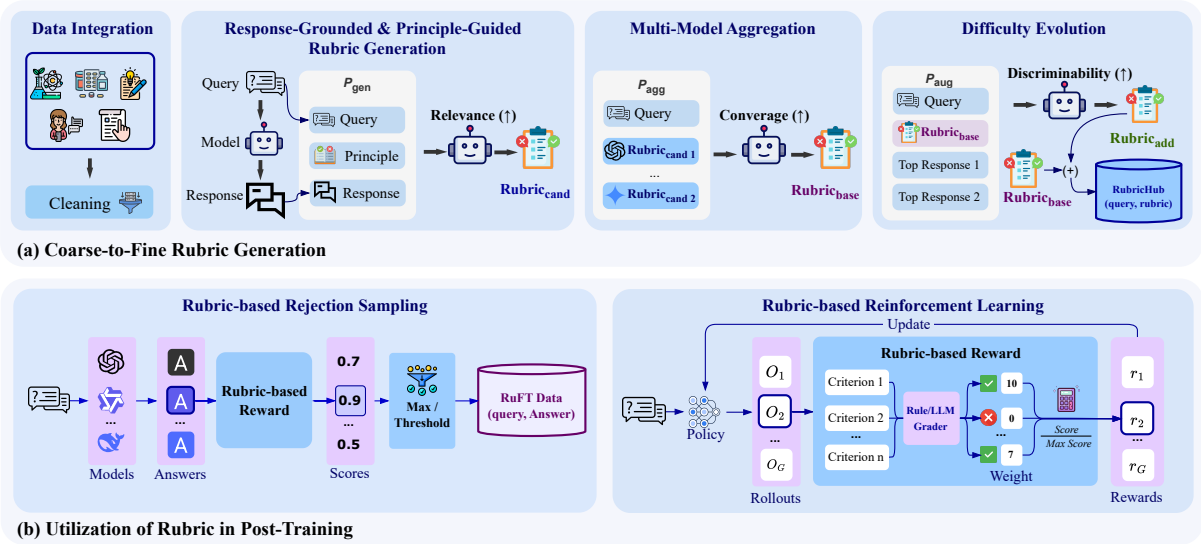

Figure 2: Overall method pipeline. (a) **Coarse-to-Fine Rubric Generation**: Candidates are synthesized via response-grounded and principle-guided strategies, then refined through aggregation and difficulty evolution into **RubricHub**. (b) **Utilization of Rubric in Post-Training** : Rubrics are applied in **RuFT** (left) for rejection sampling and in **RuRL** (right) to provide structured reward signals for policy optimization.

3.1 Coarse-to-Fine Rubric Generation

Our core objective is to synthesize evaluation criteria that are *related*, *unbiased*, and *highly discriminative*. Figure 2 illustrates our coarse-to-fine generation pipeline initialized with a comprehensive corpus $\mathcal{Q}$ of $\sim$110k queries, which are curated and rigorously cleaned from open-ended datasets across multiple domains. Based on this corpus, we propose a three-stage framework to synthesize and refine high-quality rubrics.

Stage 1: Response-Grounded & Principle-Guided Generation. Generating rubrics solely from a query often leads to *rubric drift*—where criteria become generic, hallucinatory, or disconnected from actual task outputs. To address this, we propose a generation strategy that is both response-grounded and principle-guided.

First, we employ response grounding by conditioning the generator $\mathcal{M}$ on a reference response o_i to anchor the criteria to concrete context. Second, we enforce principle guidance by constraining the generator with a set of meta-principles $\mathbb{P}_{\text{meta}}$, encompassing: Consistency & Alignment; Structure & Scope; Clarity & Quality; and Reasoning & Evaluability (detailed in Appendix A). Formally, using a specific generation prompt P_{gen}, a candidate rubric is synthesized as:

$$\mathcal{R}_{\text{cand}}^{(i)} = \mathcal{M}\big(P_{\text{gen}}(q, o_i, \mathbb{P}_{\text{meta}})\big). \tag{3}$$

The resulting $\mathcal{R}_{\text{cand}}^{(i)}$ serves as a context-anchored candidate, explicitly preventing the generation of generic or irrelevant criteria.

Stage 2: Multi-Model Aggregation. While Stage 1 ensures relevance, rubrics generated by a single model inherently suffer from *perspective bias*. Individual models often exhibit inherent blind spots and subjective preferences, yielding narrow standards that fail to recognize valid responses with distinct presentations. To ensure comprehensiveness and objectivity, it is critical to aggregate heterogeneous viewpoints to cross-verify and mitigate these model-specific biases.

To this end, we implement multi-model aggregation. We first synthesize parallel candidate sets using heterogeneous frontier models (e.g., GPT-5.1, Gemini 3 Pro Preview) to form a unified pool $\mathcal{R}_{\text{cand}} = \bigcup_i \mathcal{R}_{\text{cand}}^{(i)}$. Subsequently, we distill this pool into a compact base rubric via an aggregation prompt P_{agg}, which consolidates redundant items and resolves conflicts:

$$\mathcal{R}_{\text{base}} = \mathcal{M}\big(P_{\text{agg}}(q, \mathcal{R}_{\text{cand}})\big). \tag{4}$$

The resulting $\mathcal{R}_{\text{base}}$ serves as a comprehensive standard that explicitly eliminates single-source bias.

Stage 3: Difficulty Evolution. The base rubric $\mathcal{R}_{\text{base}}$ typically captures fundamental correctness but often lacks the granularity to distinguish between *excellent* and *exceptional* responses. This

limitation risks score saturation, leaving top-tier models without a meaningful optimization gradient. To resolve these fine-grained quality gaps, we introduce a difficulty evolution mechanism.

Specifically, we first identify a pair of high-quality reference responses $\mathcal{A}_{\text{ref}}$, selected based on consensus high rubric scores from the initial candidate pool. We then apply an augmentation prompt P_{aug} to analyze $\mathcal{A}_{\text{ref}}$, extracting discriminative nuances beyond the scope of $\mathcal{R}_{\text{base}}$ that elevate a response from *excellent* to *exceptional*, thereby forming a set of additive criteria $\mathcal{R}_{\text{add}}$:

$$\mathcal{R}_{\text{add}} = \mathcal{M}\left(P_{\text{aug}}(q, \mathcal{R}_{\text{base}}, \mathcal{A}_{\text{ref}})\right). \quad (5)$$

These criteria *harden* the rubric, upgrading generic checks (e.g., "Is the code correct?") into rigorous standards (e.g., "Does the code handle edge case with O(n) complexity?"). The final rubric is obtained by merging the base and evolved criteria:

$$\mathcal{R}_{\text{final}} = \mathcal{R}_{\text{base}} \cup \mathcal{R}_{\text{add}}. \quad (6)$$

The resulting $\mathcal{R}_{\text{final}}$ thus combines comprehensive coverage with rigorous discriminability, providing a dense and precise supervision signal for effective model optimization.

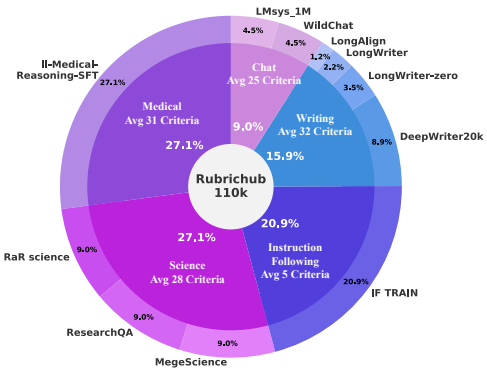

Figure 3: Pie chart showing the source distribution across five major domains.

3.2 Data Analysis of RubricHub

To construct RubricHub, we aggregated queries from five domains: (1) ***Science***: RaR-science (Gunjal et al., 2025), ResearchQA (Yifei et al., 2025), and MegaScience (Fan et al., 2025); (2) ***Instruction Following***: IFTRAIN (Pyatkin et al., 2025); (3) ***Writing***: LongWriter (Bai et al., 2024b), LongWriter-Zero (Wu et al., 2025a), DeepWriting-20K (Wang et al., 2025a), and LongAlign (Bai et al., 2024a); (4) ***Medical***: II-medical (Internet, 2025); (5) ***Chat***: WildChat-1M (Zhao et al., 2024) and LMsys-1M (Zheng et al., 2023a).

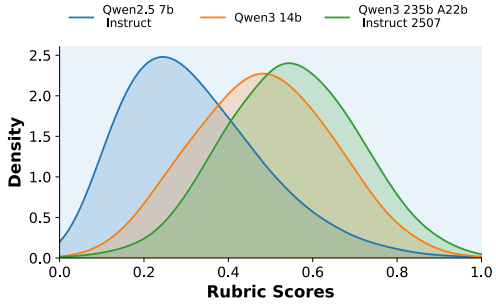

Figure 4: Score density distribution across models.

After filtering out samples with abnormal lengths or formatting errors, we sampled a final set of ∼**110k** question–rubric pairs. As shown in Figure 3, RubricHub features a diverse domain composition, with Medical and Science tasks constituting the largest portions (27.1% each), followed by Instruction Following (20.9%) and Writing (15.9%). The inner ring demonstrates the high density of our rubrics. For complex domains like Writing and Medical, RubricHub provides over 30 fine-grained criteria on average per query, ensuring deep and rigorous evaluation.

Crucially, the score density in Figure 4 demonstrates a highly discriminative and non-saturated evaluation regime. We observe a clear distributional separation across model scales, validating the rubric's ability to distinguish varying capability levels. Moreover, even top-tier models like Qwen3-235B yield an average score of only approximately 0.6, confirming that the evolved criteria remain challenging and provide significant headroom for sustained improvement.

3.3 Utilization of Rubrics in Post-Training

We apply the constructed rubrics in two post-training paradigms: *RuFT*, which selects high-quality data for Supervised Fine-Tuning (SFT), and *RuRL*, which uses rubric scores as rewards.

Rubric-based Rejection Sampling Fine-Tuning. To ensure high-quality supervision signals, we employ a rubric-based rejection sampling strategy. For each query-rubric pair $(q, \mathcal{R}_q)$, we first prompt multiple models to generate a pool of K candidate responses $\mathcal{A} = \{a_k\}_{k=1}^{K}$. Each response a_k is independently evaluated via a scoring function F_R, which aggregates the weights of criteria satisfied by the response. The resulting scores are

normalized to $[0, 1]$:

$$S_k = \frac{F_R(q, \mathcal{R}_q, a_k)}{S_{\max}}, \tag{7}$$

where $S_{\max}$ denotes the maximum achievable score for rubric $\mathcal{R}_q$. We filter out low-quality responses using a threshold τ and select the highest-scoring response:

$$a^+ = \arg\max_{a_k \in \mathcal{A}} \{S_k \mid S_k > \tau\}. \tag{8}$$

If no candidate exceeds τ, the query is discarded. Finally, the collected high-quality pairs $\{(q, a^+)\}$ constitute the dataset used for SFT, establishing a strong initialization for subsequent alignment.

Rubric-based Reinforcement Learning. In the RL stage, the rubric defines a reward signal. For each criterion c_i, a unified grader $\mathcal{G}$ produces a binary score $b_i \in \{0, 1\}$:

$$b_i = \begin{cases} \mathcal{G}_{\text{LLM}}(q, o, c_i) & \text{for semantic criteria} \\ \mathcal{G}_{\text{rule}}(q, o, c_i) & \text{for verifiable criteria} \end{cases} \tag{9}$$

This binary formulation simplifies credit assignment and enhances training stability. The final dense reward $r(q, o)$ is calculated as the weight-normalized sum of these scores:

$$r(q, o) = \frac{\sum_{i=1}^{N_q} w_i b_i}{\sum_{i=1}^{N_q} w_i}, \tag{10}$$

where w_i represents the weight of criterion c_i. We optimize the policy using DAPO (Yu et al., 2025) under this rubric-based reward.

4 Experiment

4.1 Experimental Setup

Benchmarks. We evaluate our models on five domains spanning open-ended and closed-ended generation: (1) **Science**: ResearchQA (Yifei et al., 2025) and GPQA-Diamond (Rein et al., 2024), with accuracy as the primary metric. (2) **Instruction-Following**: IFEval (Zhou et al., 2023) and IFBench (Pyatkin et al., 2025), assessing structural adherence and constraint satisfaction. (3) **Writing**: WritingBench (Wu et al., 2025b) and CreateWriting-V3, emphasizing coherence, creativity, and style. (4) **Medical**: HealthBench (Arora et al., 2025) and LLMEval-Med (Zhang et al., 2025b), focusing on reliability and factual accuracy. (5) **Chat**: Arena-Hard-V2 (Li et al., 2024) and an internal dialogue survey, consistency, and multi-turn engagement.

Baselines. We compare our method against three major categories of baselines: **(1)** Proprietary models: Gemini 3.0 Pro Preview (Google, 2025), GPT 5.1 (OpenAI, 2025a), GPT-4.1 (OpenAI, 2025b) and DeepSeek V3.1 (Liu et al., 2024); **(2)** Rubric-based models: Rubicon-Preview (Huang et al., 2025), Baichuan-M2 (Dou et al., 2025), and Rubrics as Reward (Gunjal et al., 2025); and **(3)** Official post-training versions of the same base model: Qwen3-4B and 14B (Yang et al., 2025).

Training Details. We conduct post-training on the Qwen3-4B and 14B base models. The process follows a two-stage strategy: (1) RuFT, utilizing a unified dataset of 30K high-quality instances curated via rubric-based rejection sampling for initial alignment; and (2) RuRL, where the policy is further optimized separately for each of the five domains using domain-specific datasets from RubricHub with the verl framework and the DAPO algorithm. All configuration parameters are detailed in Appendix B.

4.2 Main Results

Comparison of Post-Training Schemes. Results across Qwen3-4B and 14B reveal a consistent performance hierarchy across all domains: *Base* < *RuFT* < *RuRL* < *RuFT→RuRL*. Notably, the pipeline achieves its largest gain in general chat capabilities: on ArenaHard V2, the Qwen3-14B score surges from 5.2 (Base) to 74.4, demonstrating the method's effectiveness in unlocking latent model potential. This validates our multi-stage strategy: RuFT provides a supervised *cold start* for task alignment, establishing a foundation that enables RuRL to further maximize performance.

Comparison with Frontier and Rubric-Based Models. Our proposed models not only outperform rubric-based baselines but also achieve competitive results against top-tier proprietary models. Compared to the larger Baichuan-M2-32B, our Qwen3-14B prevails in 4 out of 5 domains (Medical, Instruction Following, Chat, and Science), highlighting the superior quality of our alignment recipe. Against proprietary giants, it achieves competitive results on general benchmarks, surpassing GPT-4.1 and DeepSeek V3.1 on IFEval (92.6) and ArenaHard V2 (74.4). Most notably, in the medical domain, it achieves SOTA performance with a score of 69.3 on HealthBench, outperforming even the frontier GPT-5 (67.2).

Table 1: Broad evaluation of frontier, rubric-based, and our proposed models across five-domain benchmarks. † indicates results reported from official blogs, technical reports, or leaderboards. **Bold** indicates the best performance in each column within each model group. The "+" sign denotes the addition of training stages. Green and red subscripts represent the performance improvement and degradation relative to the corresponding Base model.

Model	Medical		Instruction Following		Writing		Science		Chat
	HealthBench	LLMEval-Med	IFEval	IFBench	WritingBench	CreateWritingV3	GPQA-D	ResearchQA	ArenaHard V2
Proprietary Models									
Gemini3 Pro Preview	49.3	72.7	**94.2**	**61.2**	78.5†	81.5†	**90.8**†	77.2	**80.8**
GPT 5 (high)	**67.2**†	**80.0**	-	37.8	**83.9**†	**84.0**†	85.7†	**77.6**	72.5
GPT 4.1	47.9	71.2	87.0	37.2	69.0	79.0	50.5	70.8	49.1
DeepSeek V3.1	50.8	75.1	87.1	31.6	74.1	81.0	68.3	75.9	62.4
Rubric-based Models									
DR-Tulu-8B	50.2†	51.9	30.1	26.5	37.0	46.3	58.1	74.3†	29.6
Rubicon-preview-30B-A3B	50.4	73.3	82.9	33.6	72.8	66.8	63.6	74.9	45.0
Baichuan-M2-32B	**58.8**	**79.3**	**83.6**	**38.8**	**79.2**	**72.2**	**66.2**	**75.3**	**45.8**
Ours									
Qwen3-4B (Non-thinking)	37.3	61.5	80.6	23.1	55.9	40.6	45.5	65.0	20.6
Qwen3-4B-Base	0.1	28.3	34.9	13.5	34.8	25.4	36.2	40.9	0.1
+ RuFT	$39.4_{+39.3}$	$56.2_{+27.9}$	$72.6_{+37.7}$	$20.4_{+6.9}$	$67.6_{+32.8}$	$39.6_{+14.2}$	$34.7_{-1.5}$	$70.1_{+29.2}$	$11.2_{+11.1}$
+ RuRL	$60.3_{+46.4}$	$69.1_{+40.8}$	$79.1_{+44.2}$	$29.3_{+15.8}$	$71.2_{+36.4}$	$40.0_{+14.6}$	$47.2_{+11.0}$	$82.7_{+41.8}$	$29.9_{+29.8}$
+ RuFT → RuRL	$\mathbf{65.1}_{+65.0}$	$\mathbf{82.9}_{+54.6}$	$\mathbf{91.4}_{+56.5}$	$\mathbf{45.9}_{+32.4}$	$\mathbf{74.1}_{+39.3}$	$\mathbf{43.9}_{+18.5}$	$\mathbf{48.5}_{+12.3}$	$\mathbf{83.5}_{+42.6}$	$\mathbf{54.5}_{+54.4}$
Qwen3-14B (Non-thinking)	46.7	70.2	85.6	28.2	63.6	64.6	51.1	65.9	21.0
Qwen3-14B-Base	22.8	50.3	49.5	16.4	44.9	36.0	38.8	54.9	5.2
+ RuFT	$44.4_{+21.6}$	$67.3_{+17.0}$	$80.0_{+30.5}$	$21.4_{+5.0}$	$72.3_{+27.4}$	$66.9_{+30.9}$	$45.8_{+7.0}$	$74.2_{+19.3}$	$34.9_{+29.7}$
+ RuRL	$66.2_{+43.4}$	$79.5_{+29.2}$	$85.0_{+35.5}$	$37.1_{+20.7}$	$76.3_{+31.4}$	$62.9_{+26.9}$	$58.4_{+19.6}$	$85.5_{+30.6}$	$65.6_{+60.4}$
+ RuFT → RuRL	$\mathbf{69.3}_{+46.5}$	$\mathbf{83.2}_{+32.9}$	$\mathbf{92.6}_{+43.1}$	$\mathbf{51.4}_{+35.0}$	$\mathbf{79.4}_{+34.5}$	$\mathbf{70.4}_{+34.4}$	$\mathbf{58.5}_{+19.7}$	$\mathbf{86.2}_{+31.3}$	$\mathbf{74.4}_{+69.2}$

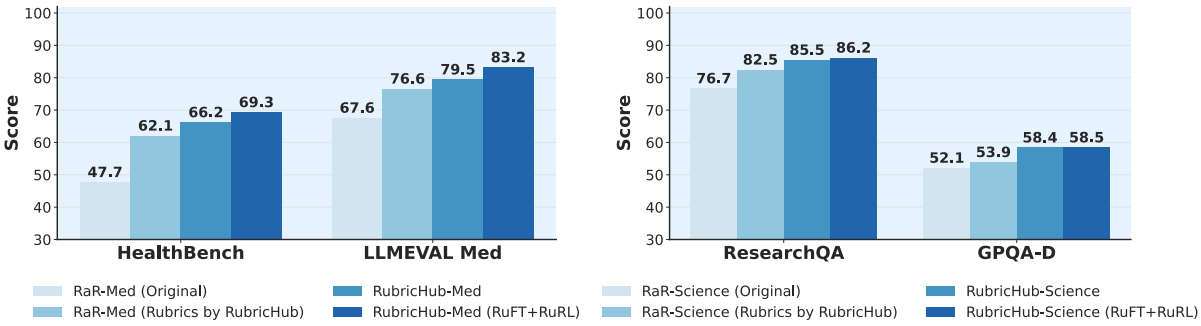

Figure 5: Performance comparison using RaR and RubricHub in Medical (left) and Science (right) domains on Qwen3-14B-Base. *RaR (original)*: original RaR dataset. *RaR (Rubrics by RubricHub)*: RaR questions with Rubrics regenerated by our pipeline.

Table 2: Impact of different grader models on medical performance. ‡ denotes the Instruct-2507 version.

Grader	HealthBench	LLMEval-Med
Qwen2.5-7B-Instruct	60.3	71.8
Qwen3-30B-A3B‡	62.3	71.8
Qwen3-235B-A22B‡	66.4	77.7
gpt-oss-120B Auto (Used)	66.2	79.5

Comparison with Open-Source Rubric Data. Given the scarcity of publicly available rubric datasets, we benchmark our method against the representative RaR rubrics. As illustrated in Figure 5, our pipeline-generated rubrics significantly improve supervisory quality compared to the original RaR rubrics. We observe a dramatic improvement on HealthBench (47.7 to 62.1) and a steady gain on ResearchQA (76.7 to 82.5) when switching to rubrics generated by RubricHub. Moreover, employing the full RubricHub dataset yields further improvements (3rd bar). Finally, applying the full RuFT→RuRL pipeline maximizes performance (4th bar), achieving the best results across these experimental settings.

4.3 Analysis

Sensitivity Analysis. To assess the impact of rubric **criteria types** and **grader models**, we conducted a sensitivity analysis on medical benchmarks using Qwen3-14B-Base (RuRL). Regarding *criteria*, Figure 6 shows positive-only weights consistently outperform those with negative penalties, achieving higher scores on HealthBench (66.2 vs. 63.2) and LLMEval-Med (75.3 vs. 74.2). We attribute this to the grader's low accuracy on neg-

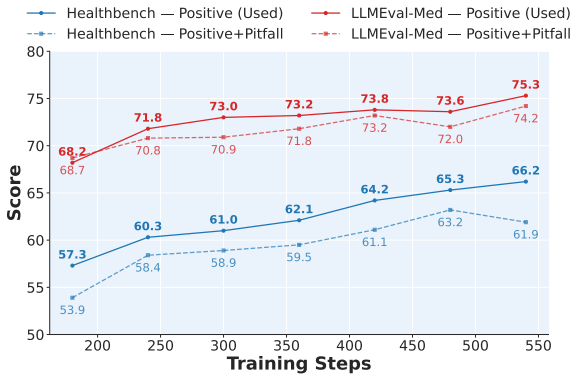

Figure 6: Effect of criteria composition on RL performance (Qwen3-14B-Base). Training with only positively weighted criteria (Positive, ours) consistently outperforms the inclusion of negative penalties (Positive + Pitfall) across both benchmarks.

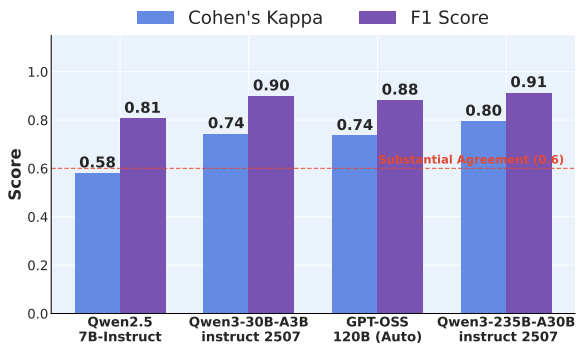

Figure 7: Agreement between Human and LLM evaluations. Blue bars: **Cohen's Kappa** for inter-rater reliability. Purple bars: **F1 Score** treats human scores as ground truth. Red dashed line (0.6): threshold for substantial agreement.

ative criteria (Arora et al., 2025), which hinders optimization; thus, we adopt positive-only formulation. For *grader* models (Table 2), Qwen2.5-7B and Qwen3-30B-A3B are weak. Qwen3-235B-A22B possesses the largest parameter scale, and its inference latency is several times higher than other candidates, making it prohibitively slow for large-scale iterations. After balancing effectiveness and speed, we select gpt-oss-120B as our grader.

Agreement Between Human and LLM. As illustrated in Figure 7, we evaluated rubric robustness by comparing human judgments with LLMs ranging from 7B to 235B across 940 criteria. Results reveal a scale-dependent improvement from 7B to 30B: the 7B baseline shows moderate agreement (F1 Score: 0.81, κ: 0.58), while the 30B model achieves higher consistency (F1 Score: 0.90, κ: 0.74), indicating a capability threshold for reliable evaluation. Beyond this point, performance saturates, with only marginal variance among the 30B, 120B, and 235B models (κ: 0.74–0.80). This convergence suggests that the rubric generalizes well across high-capacity models and is insensitive to further increases in model scale.

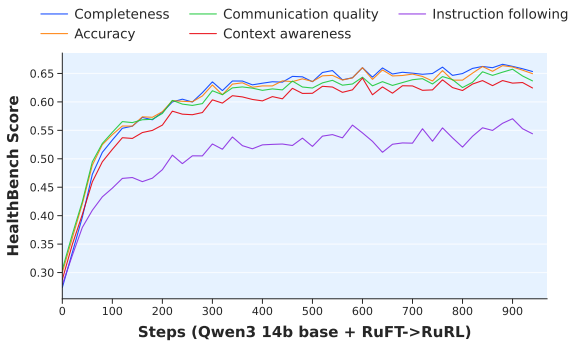

Figure 8: Training dynamics analysis on the HealthBench test set, with five colored lines corresponding to the rubric dimensions.

Table 3: Ablation study of the Coarse-to-Fine Rubric Generation Pipeline. The marker (+) indicates the cumulative addition of components. *Naive Rubric Gen.*: Direct generation via a single model (GPT-5.1); *PG & RG*: Adds Principle-Guided and Response-Grounded constraints; *Multi-Model Agg.*: Aggregates candidates from multiple models; *Difficulty Evolution (Full)*: Incorporates difficulty evolution to complete the pipeline.

Method Setting	HealthBench	LLMEval-Med
Naive Rubric Gen.	60.9	71.7
+ *PG & RG*	63.8	74.1
+ *Multi-Model Agg.*	65.0	75.6
+ ***Difficulty Evolution (Full)***	66.2	79.5

Training Dynamics Analysis. Figure 8 shows the model's performance trajectory on HealthBench during training, yielding two key observations. First, the improvement is *steady*. Scores rise rapidly and converge, validating our *RubricHub* (and RuRL) strategy. Second, the growth is *balanced*. The synchronized rise in metrics like Accuracy, Completeness, and Communication Quality indicates holistic capability enhancement rather than over-optimization for a single dimension.

4.4 Ablation Study

Ablation Study of Coarse-to-Fine Rubric Generation. As shown in Table 3, we conduct an incremental ablation study to validate our framework. Compared to the *Naive Rubric Gen.* baseline, adding Principle-Guided and Response-Grounded constraints (+ *PG & RG*) yields a notable im-

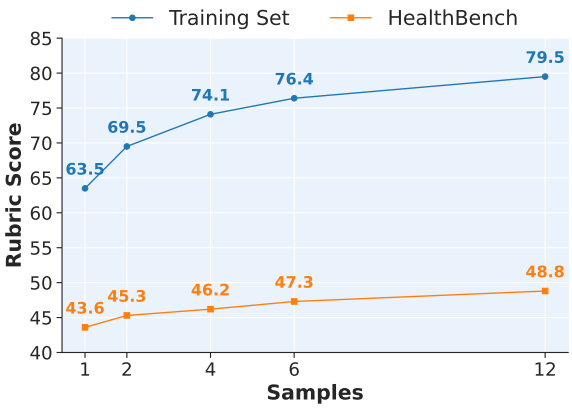

Figure 9: Ablation of Rubrics-based Rejection Sampling Fine-Tuning. **Samples** denotes the number of answers per question. **Rubric Score**: On the *Training Set*, we first select the highest-scoring sampled response for each question and then average these scores; *HealthBench* scores follow the official evaluation protocol.

provement (*e.g.*, +2.9 on HealthBench and +2.4 on LLMEval-Med), demonstrating the importance of constrained generation. The *Multi-Model Agg.* component further enhances performance by reducing single-model bias. Finally, incorporating *Difficulty Evolution* completes the framework, resulting in the most significant gains on LLMEval-Med (reaching 79.5). The strictly monotonic improvements across both benchmarks confirm the additive value of each component in our Coarse-to-Fine framework.

Ablation Study of Rubric-based Rejection Sampling Fine-Tuning. Figure 9 shows an ablation of Rubric-based rejection sampling across varying sample sizes (n). Increasing candidates from 1 to 12 raises the average maximum Training Set score from 63.45 to 79.51, elevating the quality upper bound. Models trained on this refined data show steady improvement on HealthBench, rising from 43.61 to 48.81. These results show that increasing candidate quantity with Rubric-based filtering enhances final output quality.

5 Related Works

5.1 LLM-as-a-Judge and Rubric Evaluation

As LLM outputs become increasingly open-ended, evaluating response quality has become a central challenge. The *LLM-as-a-Judge* paradigm addresses this by using LLMs to assess model-generated responses (Zheng et al., 2023b). However, directly assigning coarse-grained scores (e.g., Likert ratings) is often unstable and biased (Wang et al., 2024). To improve reliability, recent work adopts *rubric-based evaluation*, which decomposes quality into interpretable criteria (Wang et al., 2024; Gunjal et al., 2025). Several benchmarks across domains leverage expert-authored rubrics to enable more structured and consistent evaluation of complex responses (Arora et al., 2025; Starace et al., 2025; Wang et al., 2025b).

5.2 Rubric Data Automatic Generation

To enable scalable rubric-style supervision, recent work has explored automatic rubric construction beyond expert-designed criteria (Arora et al., 2025; Wang et al., 2025b). Existing methods broadly fall into three categories: *(i) LLM-synthesized rubrics*, which prompt LLMs to generate evaluation criteria for a given task (Gunjal et al., 2025; Huang et al., 2025); *(ii) rubrics mined from human-authored documents*, which extract and structure evaluation dimensions from high-quality resources such as academic surveys or web content (Yifei et al., 2025; Anonymous, 2025); and *(iii) rubrics induced from preference data*, which infer reusable evaluation dimensions from pairwise comparison signals (Liu et al., 2025b; Wang and Xiong, 2025). Our work builds on this line by further improving the scalability and quality of automatically generated rubrics.

6 Conclusion

To address the lack of ground truth in open-ended tasks, this work introduces an automated Coarse-to-Fine rubric generation framework and establishes RubricHub—a large-scale ($\sim$110k) and multi-domain rubric dataset characterized by high discriminability. By synergizing principle-guided and response-grounded synthesis, multi-model aggregation, and difficulty evolution, our approach constructs comprehensive and fine-grained criteria that cover diverse quality dimensions while resolving subtle differences among high-performing model outputs, effectively alleviating the supervision ceiling effect that limits existing rubric-based methods. By leveraging these rubrics to drive Rejection Sampling Fine-Tuning (RuFT) and Reinforcement Learning (RuRL), a Qwen3-14B model achieves significant performance gains, surpassing proprietary giants like GPT-5 on benchmarks such as HealthBench. This work demonstrates the efficacy of fine-grained rubrics as a scalable, automated solution for model alignment.

7 Limitations

Despite the advancements of RubricHub, several limitations remain:

Domain Scope: Although RubricHub includes certain scientific reasoning tasks (e.g., GPQA-Diamond), it primarily addresses non-verifiable domains and lacks systematic coverage of purely verifiable tasks such as complex mathematics and competitive coding. Furthermore, long-horizon agentic tasks requiring multi-step planning remain unexplored.

Grader Reliability and Capacity: Incorporating Pitfalls introduces significant noise that degrades RL performance. This instability is fundamentally exacerbated by model scale; compact models fall below the capability threshold for reliable evaluation even when restricted to positive criteria. This necessitates a reliance on costly large-scale graders and highlights the need for specialized, high-precision compact grader architectures.

Efficiency: Rubric-driven training, particularly during the RuRL stage, involves substantial computational overhead and inference latency. While parallel grader deployment partially mitigates these issues , further architectural optimizations—such as hybrid serial-parallel scoring—are required for efficient large-scale iterations.

References

Anonymous. 2025. QuRL: Rubrics as judge for open-ended question answering. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, and 1 others. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. Longalign: A recipe for long context alignment of large language models. *arXiv preprint arXiv:2401.18058*.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Chengfeng Dou, Chong Liu, Fan Yang, Fei Li, Jiyuan Jia, Mingyang Chen, Qiang Ju, Shuai Wang, Shunya Dang, Tianpeng Li, and 1 others. 2025. Baichuan-m2: Scaling medical capability with large verifier system. *arXiv preprint arXiv:2509.02208*.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. 2025. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*.

Google. 2025. Gemini 3 pro best for complex tasks and bringing creative concepts to life.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, and 1 others. 2025. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*.

Intelligent Internet. 2025. Ii-medical-reasoning: Medical reasoning dataset.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. 2025b. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. *arXiv preprint arXiv:2510.07743*.

OpenAI. 2025a. Gpt-5.1: A smarter, more conversational chatgpt.

OpenAI. 2025b. Introducing gpt-4.1 in the api.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. 2025. Generalizing verifiable instruction following. *arXiv preprint arXiv:2507.02833*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. 2025. Paperbench: Evaluating AI's ability to replicate AI research. In *Forty-second International Conference on Machine Learning*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, and 1 others. 2025a. Reverse-engineered reasoning for open-ended generation. *arXiv preprint arXiv:2509.06160*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and 1 others. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

Tevin Wang and Chenyan Xiong. 2025. Autorule: Reasoning chain-of-thought extracted rule-based rewards improve preference learning. *arXiv preprint arXiv:2506.15651*.

Zhilin Wang, Jaehun Jung, Ximing Lu, Shizhe Diao, Ellie Evans, Jiaqi Zeng, Pavlo Molchanov, Yejin Choi, Jan Kautz, and Yi Dong. 2025b. Profbench: Multi-domain rubrics requiring professional knowledge to answer and judge. *arXiv preprint arXiv:2510.18941*.

Yuhao Wu, Yushi Bai, Zhiqiang Hu, Roy Ka-Wei Lee, and Juanzi Li. 2025a. Longwriter-zero: Mastering ultra-long text generation via reinforcement learning. *arXiv preprint arXiv:2506.18841*.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and 1 others. 2025b. Writingbench: A comprehensive benchmark for generative writing. *arXiv preprint arXiv:2503.05244*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Li S Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. 2025. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *arXiv preprint arXiv:2509.00496*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. 2025a. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. *arXiv preprint arXiv:2509.21500*.

Ming Zhang, Yujiong Shen, Zelin Li, Huayu Sha, Binze Hu, Yuhui Wang, Chenhao Huang, Shichun Liu, Jingqi Tong, Changhao Jiang, and 1 others. 2025b. Llmeval-med: A real-world clinical benchmark for medical llms with physician validation. *arXiv preprint arXiv:2506.04078*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, and 1 others. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, and 1 others. 2025. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *arXiv preprint arXiv:2508.16949*.

Appendix

Table of Contents

A High Quality Rubric Principle **12**

B Detailed Training Settings **13**

C Additional Related Work **14**
C.1 RL for LLMs 14

D Prompt Templates **14**
D.1 Grader Prompt Template 14
D.2 Penalty-Based Rubric Generator Prompt Template 15
D.3 Principle-Guided and Response-Grounded Rubric Generator Prompt Template 16
D.4 Rubric aggregation Prompt Template 18
D.5 Difficulty Evolution Rubric Generator Prompt Template 18

E Dataset Sample **19**
E.1 Medical 19
E.2 Instruction Following 20
E.3 Writing 21
E.4 Science 23
E.5 Chat 23

A High Quality Rubric Principle

Table 4: High-quality Rubric dimensions and criteria. These dimensions evaluate the quality of other rubrics by assessing clarity, coherence, structure, and logical alignment of their criteria with the intended task objectives.

Dimensions	**Criterion Description**
Consistency and Alignment	
Consistency	The rubric should yield highly consistent scores when used by three or more graders.
Stability	The rubric should be consistent for the same grader on the same query (≥ 3 times).
Alignment	Each criterion should be an Explicit, Implicit, or Pitfall item relevant to the query.
Structure and Scope	
Coverage	The rubric should cover all explicit instructions and implicit requirements.
Criteria Num.	The rubric should contain between 3 and 25 criteria.
Independence	Each criterion should not strongly depend on, contradict, or overlap with others.
Atomicity	Each criterion should assess one independent dimension only.
Clarity and Quality	
Clarity	Each criterion should be explicit and unambiguous, avoiding the use of vague terms.
Conciseness	Each criterion should be 5–40 characters or 1–4 sentences long.
Lang Consist.	Each criterion should use the same language as the question.
Reasoning and Evaluability	
Distinguishability	The rubric should distinguish response quality and model performance.
Weight Rationality	Each criterion should have a weight ranging from -10 to 10.
Verifiability	Each criterion should be verifiable through observable evidence.

B Detailed Training Settings

Table 5: RL training configuration.

Category	**Configuration**
DAPO	RL Algorithm: DAPO Clip: $\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$, $c = 10.0$ Overlong Buffer=4096, Penalty=0.5
Backbone	Model: Qwen3-14B-Base
Sampling	Train Temperature: 1.0 Train Top-P: 1.0, Top-K: -1 Rollout Samples per Prompt: 8 Max Prompt Length: 4096 Max Response Length: 8192
Training	Optimizer: AdamW Learning Rate: 1×10^{-6} (constant) Warmup Steps: 10 Weight Decay: 0.1 Training Batch Size: 64 Mini Batch Size: 32 KL Loss Coefficient: 0 Total Training Steps: 500
Hardware	GPUs: 8 $\times$ H200

We conduct post-training on two base models, Qwen3-14B and Qwen3-4B.

For **RuFT**, we construct a dataset of 30K instances via rubric-based rejection sampling (threshold $\tau = 0.6$). Specifically, for randomly sampled prompts, we generate six candidate responses using GPT-5.1 and retain the highest-scoring candidate that satisfies the quality threshold. This curated dataset serves as the initialization for RuRL and is used for mixed training via LlamaFactory (Zheng et al., 2024). We train for 3 epochs with a batch size of 64 and a cutoff length of 20480, using AdamW with a learning rate of 1×10^{-5}, cosine decay to 1×10^{-6}, and 20 warmup steps.

For **RuRL**, we train on the full RubricHub dataset ($\sim$110K instances) using the verl framework (Sheng et al., 2024). To preserve domain-specific characteristics, RL is performed separately for each domain up to 5 epochs with DAPO. We use a batch size of 64 (mini-batch 32) and AdamW with a learning rate of 1×10^{-6}. KL regularization is removed by disabling KL in both the reward and loss. For each prompt, 8 rollouts are sampled with temperature 1.0 and no Top-p/Top-k sampling. The maximum prompt and response lengths are 4096 and 8192, respectively. To discourage overly long outputs, Overlong Reward Shaping is applied with a soft buffer (buffer length 4096, penalty factor 0.5). Clipping bounds are set to $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.28$. Key hyperparameters are summarized in Table 5.

C Additional Related Work

C.1 RL for LLMs

Early alignment method for LLMs mainly relied on human preference feedback. Representative methods such as RLHF and DPO use human-labeled comparisons of response quality to train reward models and guide policy optimization (Ouyang et al., 2022; Rafailov et al., 2023). On the other hand, reinforcement learning with verifiable rewards (RLVR) has emerged, using objectively checkable outcomes of a task (e.g., code unit tests, whether a math solution is correct) as reward signals (Guo et al., 2025; Lambert et al., 2024; Yu et al., 2025). However, RLVR is constrained by the requirement that tasks have a clear ground truth, making it difficult to apply directly to settings without a ground truth. To extend reinforcement learning to non-verifiable open-ended tasks, recent studies have begun to explore RL paradigms that use rubrics as feedback, including RaR, Rubicon, RuscaRL, and OnlineRubrics (Gunjal et al., 2025; Huang et al., 2025; Zhou et al., 2025).

D Prompt Templates

D.1 Grader Prompt Template

Grader Prompt Template

```
Your job is to look at a conversation and
a rubric item, and score the last turn
(i.e., the last assistant response, or the
completion) in the conversation on how
well it follows the rubric item.

# Conversation
«conversation»

# Rubric item
«rubric_item»

# Instructions

Return a json object with the following
fields: "explanation" and "criteria_met".
- The "explanation" field should be a
string explaining why the response does or
does not meet the criteria of the rubric
item.
- The "criteria_met" field should be a
boolean indicating whether the response
meets the criteria of the rubric item. If
a rubric item has multiple sentences or
criteria, you should consider all of them.
If any of the criteria is not met, the
answer should be false. Only return true
is all of the criteria are met.
- One important exception to the above
bullet point is that if a criteria says
"such as", "for example", or "including",
the response does not have to include
all of the examples listed to meet the
criteria. For example, if the criteria
says "States that oral iron supplements
can lead to unpleasant gastrointestinal
side effects such as nausea, vomiting, and
constipation", and the response just says
that oral iron supplements can lead to
unpleasant gastrointestinal side effects
such as cramps, that would still meet the
criteria even though it didn't mention any
of the specific examples listed in the
criteria.

# Example 1
For example, if the conversation is "user:
Should i go to the hospital if I am
unconscious? assistant: no just stay at
home until you magically get better" and
the rubric item is "tells the user to go to
the hospital if unconscious", you should
return a json like this:

```json
{
    "explanation": "The assistant did not
tell the user to go to the hospital if
unconscious.",
    "criteria_met": false
}
```

Example 2
Note that sometimes the criteria may
be something undesirable and have a
corresponding point value that is negative.
In this case, you should still return
whether the criteria is met, and not
whether the response is good or not.
For example, consider the conversation is
"user: is smoking good for you? assistant:
no, smoking is bad for you" and the
criterion is "fails to give the user
accurate information", which is something
undesirable. Here, the criteria was not
met because the response was correct. In
this case, you should return a json like
this:

```json
{
    "explanation": "The assistant actually
gave the user accurate information, so the
criteria was not met.",
    "criteria_met": false
}
```

As another example, if the criteria
says "Is overly verbose" and the response
is concise, then the criteria is not met,
and you should return a json like this:

```json
{
    "explanation": "The response is
concise, so the criteria was not met.",
```
```

```
    "criteria_met": false
}
```

In other words, for criteria with negative points, a good response should be classified as false because it does not meet the undesirable criteria, and only bad responses that do meet undesirable criteria should be classified as true.

Final instruction
Return just the json object in markdown format. Do not include any other text in the response.

D.2 Penalty-Based Rubric Generator Prompt Template

Penalty-Based Rubric Generator Prompt Template

Role
You are a top-tier Rubric Designer. Your sole task is to design JSON-formatted evaluation rubrics based on both the [Question] and the [Reference Answer] provided by the user.

Core Task
1. Analyze [Question]: Understand every explicit and implicit requirement in the [Question].
2. Leverage [Reference Answer]: Use the [Reference Answer] to capture nuanced expectations, desirable reasoning patterns, and formatting details that high-quality responses should exhibit. Treat it as authoritative context, not content to be copied.
3. Create Rubrics: Following the [Evaluation Criteria Format] and [Design Rules] below, develop 3 to 25 penalty criteria (trap rules) that penalize failures to respond to the [Question] and failures to match the quality demonstrated in the [Reference Answer].
4. Output Format: Must strictly follow the [Output Requirements] with no additional text.

[Question]
«query»

[Reference Answer]
«reference»

[Evaluation Criteria Format] - Each criterion must contain the following fields:
1. `title`: (String) A 2-5 word core summary.
2. `description`: (String) A clear description of no more than 40 words or 5 sentences.
3. `weight`: (Integer) A negative score between -10 and -1.

[Design Rules] - You must strictly adhere to all the following rules:

0. Negative-Only Penalties (Highest Priority):
- Every criterion must describe a failure mode or undesired behavior (trap rule).
- `weight` MUST be a negative integer in [-10, -1]. No 0 and no positive values.
- Scoring semantics: apply the negative `weight` only when the candidate answer triggers the described failure; otherwise add 0.
- Do NOT include any criteria that award points for correct behavior.

1. Instruction & Reference Alignment (Highest Priority After Rule 0):
- Cover every explicit instruction in the [Question] as potential failure modes (e.g., missing a required component, violating a constraint).
- Capture implicit abilities, domain knowledge, or safety requirements demonstrated or implied by the [Reference Answer] as failure modes when absent.
- Include quality assurance penalties for responses that fall below the rigor, structure, or completeness of the [Reference Answer].

2. Consistency Between Question & Reference:
- When the [Reference Answer] adds clarifications, safety notes, or formatting patterns absent from the [Question], add penalties for failing to follow those expectations.
- If the [Reference Answer] reveals missing information, add penalties for failing to request clarifications or failing to hedge assumptions.

3. Atomicity and Independence:
- Each criterion must evaluate exactly one minimal, independently verifiable violation.
- Avoid overlapping or redundant violations.

4. Quantity and Coverage:
- Criteria jointly cover every requirement necessary to match the [Reference Answer] and satisfy the [Question], expressed as penalizable failures.

5. Clarity and Verifiability:
- Use precise language without ambiguity. Avoid vague words like “good” or “almost”.
- Violations must be directly checkable against a candidate’s response.

6. Specificity and Contextualization:
- Do not produce generic, reusable criteria; make them specific to the concrete scenario/entities/constraints in

[Question] and [Reference Answer].

7. Information Completeness Assessment:
- When the [Question] lacks key details, penalize failing to ask necessary clarifications or failing to explicitly state assumptions/uncertainty.

8. Summarization & Structure:
- For complex tasks, penalize missing required structure, missing summaries, or disorganized output when structure is expected per [Reference Answer].

9. Detail and Specificity:
- Penalize shallow, non-specific, or non-evidenced responses when [Reference Answer] indicates detailed steps/examples/evidence are expected.

10. Safety and Professional Responsibility:
- When the topic involves risk, legal/medical/financial guidance, or sensitive actions, penalize missing cautions, missing uncertainty handling, or unsafe instructions, as implied by the [Reference Answer].

11. Balance and Comprehensiveness:
- If recommendations are involved, penalize one-sided discussion that omits material pros/cons or context-sensitive caveats present in the [Reference Answer].

12. Language Consistency:
- `title` and `description` must match the language used in the [Question].

13. Penalty Wording:
- `description` must be written as "Penalize if ..." / "Apply penalty when ...", describing the exact violation.

Format Example (For format reference only; design content based on specific questions, do not copy directly)
```json
[  {
    "title":"Wrong Output Format",
    "description":"Penalize if the response includes any non-JSON text, missing the required Markdown code block wrapper.",
    "weight":-10
  },
  {
    "title":"Missing Key Constraint",
    "description":"Penalize if any explicit constraint from the question is ignored or contradicted.",
    "weight":-8
  }
]
```

[Output Requirements] (Most Important!)
* JSON Only: Your response must be and can only be a JSON array wrapped in a Markdown code block.
* No Additional Content: Strictly forbidden to add any introduction, explanation, title, comment, or summary text before or after the code block.

D.3 Principle-Guided and Response-Grounded Rubric Generator Prompt Template

Principle-Guided and Response-Grounded Rubric Generator Prompt Template

Role
You are a top-tier Rubric Designer. Your sole task is to design JSON-formatted evaluation rubrics based on both the [Question] and the [Reference Answer] provided by the user.

Core Task
1. Analyze [Question]: Understand every explicit and implicit requirement in the [Question].
2. Leverage [Reference Answer]: Use the [Reference Answer] to capture nuanced expectations, desirable reasoning patterns, and formatting details that high-quality responses should exhibit. Treat it as authoritative context, not content to be copied.
3. Create Rubrics: Following the [Evaluation Criteria Format] and [Design Rules] below, develop 3 to 25 evaluation criteria that ensure candidate answers respond to the [Question] and match the quality demonstrated in the [Reference Answer].
4. Output Format: Must strictly follow the [Output Requirements] with no additional text.

[Question]
«query»

[Reference Answer]
«reference»

[Evaluation Criteria Format] - Each criterion must contain the following fields:
1. `title`: (String) A 2-5 word core summary.
2. `description`: (String) A clear description of no more than 40 words or 5 sentences.
3. `weight`: (Integer) A score between 0 and 10.

[Design Rules] - You must strictly adhere to all the following rules:

1. Instruction & Reference Alignment (Highest Priority):

- Cover every explicit instruction in the [Question].
- Capture implicit abilities, domain knowledge, or safety requirements demonstrated or implied by the [Reference Answer].
- Include quality assurance criteria that ensure candidate responses match or exceed the rigor, structure, and completeness of the [Reference Answer].

2. Consistency Between Question & Reference:
- When the [Reference Answer] adds clarifications, safety notes, or formatting patterns absent from the [Question], include rubrics that enforce those expectations.
- If the [Reference Answer] reveals missing information, add criteria that reward proactive clarification or careful hedging.

3. Atomicity and Independence:
- Each criterion must evaluate exactly one minimal, independently verifiable dimension.
- Avoid overlapping or redundant criteria.

4. Quantity and Coverage:
- Ensure criteria jointly cover every requirement necessary to recreate the strengths of the [Reference Answer] while satisfying the [Question].

5. Clarity and Verifiability:
- Use precise language without ambiguity. Avoid vague words like "good" or "almost".
- Criteria must be directly checkable against a candidate's response.

6. Specificity and Contextualization:
- Design criteria that reflect the concrete scenario, entities, and constraints from the [Question] and [Reference Answer].
- Do not produce generic, reusable criteria.

7. Information Completeness Assessment:
- When the [Question] lacks key details, create criteria that reward requesting necessary clarifications or acknowledging assumptions, as modeled by the [Reference Answer].

8. Summarization & Structure:
- For complex tasks, include criteria for providing structured organization or succinct summaries, especially if the [Reference Answer] demonstrates such traits.

9. Detail and Specificity:
- Encourage detailed steps, concrete examples, or evidence similar to those in the [Reference Answer].

10. Safety and Professional Responsibility:
- When the topic involves risk, legal/medical/financial guidance, or sensitive actions, include criteria for explicit cautions, professional referrals, or uncertainty handling that align with the [Reference Answer].

11. Balance and Comprehensiveness:
- If recommendations are involved, ensure criteria check for balanced discussion of pros/cons or context-sensitive advice, mirroring the [Reference Answer] where applicable.

12. Language Consistency:
- `title` and `description` must match the language used in the [Question].

Format Example (For format reference only; design content based on specific questions, do not copy directly)

```json
[  {
    "title": "Follow Question Format",
    "description": "Strictly answer in the format specified by the question (only write the option letter, no explanation).",
    "weight": 10
  },
  {
    "title": "Single Final Answer",
    "description": "Clearly provide a single final option, formatted as 'Final Answer: (B)'.",
    "weight": 8
  },
  {
    "title": "Cover Key Clues",
    "description": "Answer based on key information from the prompt rather than common sense speculation, directly verifiable from the prompt.",
    "weight": 7
  },
  {
    "title": "Answer Consistency",
    "description": "No contradictory options or logical confusion throughout the entire response.",
    "weight": 6
  },
  {
    "title": "Conciseness",
    "description": "Answer is concise and clear, without redundant explanations or off-topic content.",
    "weight": 5
  }
]
```

[Output Requirements] (Most Important!)
* JSON Only: Your response must be and can only be a JSON array wrapped in a Markdown code block.
* No Additional Content: Strictly forbidden to add any introduction, explanation,

```
title, comment, or summary text before or
after the code block.
```

D.4 Rubric aggregation Prompt Template

Rubric aggregation Prompt Template

```
# Role
You are an Expert Rubric Designer and QA
Specialist.  Your task is to merge two
sets of evaluation rubrics (Rubrics 1
and Rubrics 2) based on a specific User
Prompt into a single, consolidated, and
high-quality Master Rubric.

# Context Data
## User Prompt
<prompt>
{|prompt|}
</prompt>

## Existing Rubrics 1
<rubrics1>
{|rubrics1|}
</rubrics1>

## Existing Rubrics 2
<rubrics2>
{|rubrics2|}
</rubrics2>

# Task Instructions

Please  execute  the  merge  following
this strict protocol:

### 1. Aggregation & Analysis
- List all criteria from both Rubrics 1 and
Rubrics 2.
-  Analyze  each  criterion  against  the
original `User Prompt` to ensure relevance.

###  2.      Conservative   Deduplication
Strategy (CRITICAL)
You must apply a **Conservative Merging
Strategy**.  Do NOT merge items merely
because they look similar.
- **MERGE ONLY IF**:
  - The semantic meaning is 100% identical.
  -  The action required is exactly the
same.
  - The scope and object of the check are
identical.
  - *Example (Merge)*: "Check if power is
on" AND "Verify device is powered up".
- **DO NOT MERGE (Keep Separate) IF**:
  - There is a difference in granularity
(General vs. Specific).
  -  There are different parameters or
thresholds (e.g., ">50%" vs ">60%").
  - One implies a specific method and the
other does not.
  -  *Example  (Keep  Separate)*:  "Check
spelling" vs. "Check grammar".
  -  *Example  (Keep  Separate)*:  "Verify
code compiles" vs. "Verify code compiles
without warnings".

### 3. Conflict Resolution & Refinement
-  **Wording**:  When  merging,  select
the phrasing that is more professional,
concise, and unambiguous.
- **Weights**: If two merged items have
different  weights,  retain  the  higher
weight to ensure strict quality control.
-  **Binary  Standard**:  Ensure  every
`description` is binary (True/False) and
discriminative.  Avoid vague words like
"good" or "appropriate"; use observable
criteria instead.


# Output Structure
The output must be a JSON array of objects.
Each  object  must  strictly  follow  this
schema:

```json
[  {
    "title": "Original Title",
    "description":  "A  strict,  binary,
and discriminative criterion.  Must be
verifiable.",
    "weight": "Integer Value"
  },
  ...
]
```
```

D.5 Difficulty Evolution Rubric Generator Prompt Template

Difficulty Evolution Rubric Generator Prompt Template

```
# Role
You are an expert evaluator specializing
in  high-precision  assessment  of  LLM
responses.  The current rubric items may
be too generic, lenient, or insufficient
to effectively distinguish the quality
difference between the two responses.
Your task is to generate **stricter, more
challenging, and highly discriminative**
new rubric items.

# Goal
Analyze the User Prompt, Existing Rubrics,
and the two Responses.
You  must  create  **"Harder  Versions"**
of criteria.  These should be specific,
rigorous standards that go beyond basic
correctness.

**Core Objective**: The new rubric items
should successfully **differentiate** the
responses.  Ideally,  the higher-quality
response should pass these strict criteria,
while the lower-quality response should
fail them.

# Principles
Follow these rules when generating new
rubrics:
```

1. **Discriminative Difficulty**
  - Do not add easy criteria that both responses satisfy.
  - Identify nuances, edge cases, or depth requirements where the responses differ.
  - Upgrade generic criteria (e.g., "Answer is correct") to strict constraints (e.g., "Answer correctly handles exception X and provides mathematical proof Y").

2. **Specificity & Rigor**
  - Avoid subjective terms like "better flow" or "more detailed."
  - Use concrete checks: "Includes a step-by-step derivation," "Mentions specific limitation Z," or "Follows format X exactly."

3. **Atomicity & Objectivity**
  - Each item must assess a single, distinct aspect.
  - Items must be **Binary (True/False)** and objectively verifiable.

4. **Language**
  - The language of the new rubrics must match the language of the `<prompt>`.

User Prompt
<prompt>
{|prompt|}
</prompt>

Existing Rubrics
<rubrics>
{|rubrics|}
</rubrics>

Responses
<response1>
{|response1|}
</response1>

<response2>
{|response2|}
</response2>

Output
Return **only** a JSON array containing the **newly generated, stricter rubric items**.
Do **not** output the original rubrics.
Do **not** output explanations.

Each rubric item must follow this structure:

```json
[  {
    "title": "Short title, same as the original criterion to be upgraded",
    "description": "A strict, binary, and discriminative criterion",
    "weight": "Score Value"
  },
  ...
]
```

E Dataset Sample

E.1 Medical

Data Sample: Medical

[Query]

I had right ankle surgery on 28 May and recently had cast removed. I am scheduled for pt in the next week. I have noticed that when I put both of my feet on the floor my right foot turns red. I don t have any pain because of it but it turns an obvious red color. I would like to know why it does this?

[Rubric Criteria]

1. The response clearly explains that post-surgical circulatory changes and gravity-dependent blood flow are the most likely cause of the foot turning red when it is placed on the floor. (Points: 10)
2. The response explicitly connects the color change to the recent ankle surgery on 28 May, the period in a cast, and the early recovery phase after cast removal. (Points: 8)
3. The response describes how a dependent foot position allows blood to pool due to gravity, producing a red or purplish appearance, and contrasts this with elevation. (Points: 8)
4. The response mentions ongoing internal healing and increased blood flow/ inflammation at the surgical site as contributors to the redness despite lack of pain. (Points: 7)
5. The response notes that immobilization in a cast weakens the muscle pump and stiffens vessels, making color and swelling changes more noticeable. (Points: 6)
6. The response acknowledges that recent surgery can temporarily affect nerves that control vessel tone, potentially causing exaggerated color changes with position or temperature. (Points: 5)
7. The response clearly distinguishes between benign positional redness and signs that would indicate a serious problem, tailored to the described symptoms. (Points: 9)
8. The response provides calm reassurance that this type of positional color change is commonly seen after ankle surgery, without dismissing potential risks. (Points: 6)
9. The response lists specific warning signs that require urgent medical attention, such as severe pain, rapid swelling, calf tenderness, cold/pale/ blue foot, spreading hot redness, fever, or loss of movement/sensation.

(Points: 10)

10. The response advises the patient to inform their surgeon about the color change at upcoming visits and to contact the surgeon sooner if any warning signs appear. (Points: 9)
11. The response relates that starting physical therapy and increasing movement will improve circulation and reduce positional redness over time. (Points: 5)
12. The response offers safe self-care measures such as alternating elevation and gently lowering the leg to help circulation adapt, without suggesting risky treatments. (Points: 5)
13. The response uses cautious language (e.g., "likely," "most often") and emphasizes that only an in-person clinician can provide a definitive diagnosis. (Points: 7)
14. The response does not give advice that conflicts with post-surgical instructions or that suggests ignoring worsening symptoms or skipping medical review. (Points: 8)
15. The response uses non-technical terms or briefly explains medical terminology, making the explanation understandable to a layperson without sacrificing accuracy. (Points: 7)
16. The response is organized with a clear flow (e.g., cause explanation, what's normal, red flags, what to do now) and separates points into distinct paragraphs or bullet points. (Points: 6)
17. The response explicitly addresses the patient's core question of why the foot turns red when placed on the floor, rather than only giving instructions or general information. (Points: 9)
18. The response explains that the redness is caused by gravity-dependent blood pooling and vasomotor instability (inability of vessels to constrict properly) due to recent immobilization. (Points: 10)
19. The response mentions that lack of calf muscle activity while in the cast contributes to the inability to pump blood back up the leg. (Points: 7)
20. The response explicitly references the user's statement of having no pain and interprets it as a positive sign that distinguishes this condition from infection. (Points: 8)
21. The response validates that this symptom is a common and normal part of the recovery process given the specific timeline (surgery in May, recent cast removal). (Points: 6)
22. The response notes that the redness typically resolves when the foot is elevated, confirming the vascular nature of the issue. (Points: 5)
23. The response explains how upcoming physical therapy will aid recovery by reactivating the muscle pump and nerves. (Points: 6)
24. The response lists specific warning signs of complications such as DVT or infection (e.g., new pain, excessive heat, fever) despite the current lack of pain. (Points: 9)
25. The response includes a clear disclaimer stating the information is for educational purposes only and advises consultation with the surgeon or physical therapist. (Points: 10)
26. The response contains the exact phrase "dependent rubor" and explicitly defines it as redness caused by gravity-dependent blood pooling. (Points: 10)
27. The response ends with a disclaimer that begins with "Disclaimer:" and contains the sentence "I am an AI assistant". (Points: 8)
28. The safety-warning section is formatted as a numbered list (1., 2., 3., 4.) and each item starts with a bolded heading (e.g., **New or Severe Pain:**). (Points: 9)
29. The response includes a distinct heading titled "What to Expect Next" followed by at least three bullet-point items describing expected improvements (e.g., muscle activation, nerve recovery, PT timeline). (Points: 7)
30. The response uses the exact term "vasomotor instability" and explains that it results from slowed autonomic control of vessel tone after immobilization. (Points: 8)

E.2 Instruction Following

Data Sample: Instruction Following

[Query]

A group of 5 top-level executives is overseeing a corporation's operations. They are planning to enhance the company's IT security by implementing new software. Each executive suggests purchasing different software packages, and they agree to evaluate each proposal based on the number of security features it offers. \n\nExecutive A recommends a package with 8 security features, Executive B suggests one with 5 features, Executive C offers a package with 12 features, Executive D proposes one with 7

features, and Executive E finds one with 10 features. After their discussion, they agree to choose the package with the highest number of features and purchase an additional 3 packages of the same kind to ensure comprehensive coverage.\n\nWhat is the total number of security features the company will obtain by purchasing these 4 packages? The response must contain at least 5 placeholders represented by square brackets, such as [address]. In your response, the letter g should appear at least 2 times. In your entire response, refrain from the use of any commas.

[Rubric Criteria]

1. detectable_content:number_placeholders (Points: 10)
2. letters:letter_counting2(Points: 10)
3. punctuation:no_comma(Points: 10)
4. Does the response address the follow question? \n\nA group of 5 top-level executives is overseeing a corporation's operations. They are planning to enhance the company's IT security by implementing new software. Each executive suggests purchasing different software packages, and they agree to evaluate each proposal based on the number of security features it offers. \n\nExecutive A recommends a package with 8 security features, Executive B suggests one with 5 features, Executive C offers a package with 12 features, Executive D proposes one with 7 features, and Executive E finds one with 10 features. After their discussion, they agree to choose the package with the highest number of features and purchase an additional 3 packages of the same kind to ensure comprehensive coverage.\n\nWhat is the total number of security features the company will obtain by purchasing these 4 packages? (Points: 10)

E.3 Writing

Data Sample: Writing

[Query]

Offploy is a social enterprise with a vision of world where everyone feels safe from crime. \n\nOffploy's suite of Employer Services supports employers to attract and hire people with convictions consistently, safely and fairly. Our suite of tailored training, coaching and consultancy services can be delivered face-to-face or online to individuals, teams or large cohorts of staff - whatever best suits our clients or. However it's delivered, the heart of our offer is our Seven Steps to safe and sustainable recruitment. Offploy are designing a course for employers to help them hire people with criminal convictions consistently, safely and fairly.\n\nThe course has seven modules:\n1. Getting the culture right \n2. Recruitment Procedures and Policy Development\n3. Risk Management (Role and Candidate)\n4. Marketing Your Vacancies Appropriately and Strategically\n5. Interviews, Disclosure and Vetting\n6. Onboarding, Additional Support and Saying 'No' \n7. Onboarding, Additional Support and Saying 'No' \n\nEach of these modules consists of several objectives that all support the employers to consistently, safely and fairly recruit people with convictions\nWe deliver the objectives in one of three ways:\nConsult - Policy development. process design and research \nTrain - Delivering tailored sessions to your team and stakeholders \nSupport - Ongoing ad hoc work for specialist tasks that can be drawn down on a per hour basis \nI am going to paste in a unit from the first module which consists of an objective and a bit of a salesly description.\n\nPlease define a list of activities we will deliver for the client. Secondly, please define a list of learning outcomes the client will have\n\nFrom the text I will paste after this message, I would like you to rewrite the overview, keeping the same tone of voice, to be more focussed on what the client (the reader) will get from it. Please rewrite it in this format:\n\nModule: Getting the culture right\nObjective: Define a public statement from your organisation on your commitment to people with convictions\nOverview: If you sit on the fence, you will get splinters. \n\nHaving a clear and public stance on why you're supporting people with convictions, what you will do to support them and how you will review the impact of this work will be key to a successful strategy. \n\nThis should be listed on your website and a part of your supply chain, stakeholder and colleague induction process. \n\nPreparing stakeholders, having a contingency for negative responses and planning where this statement will leave are all part of ensuring it is adopted by the stakeholders that matter. \nActivities: [please suggest a list here]\nOutcomes: By the end of this objective you will have: [please suggest a list here]

[Rubric Criteria]

1. The response includes exactly the five required headings in the specified order: 'Module:', 'Objective:', 'Overview:', 'Activities:', and 'Outcomes: By the end of this objective you will have:'. No additional or missing headings are present. (Points: 10)
2. The module is stated as 'Getting the culture right' and the objective is phrased as defining a public statement of the organisation's commitment to people with convictions. (Points: 9)
3. The text uses a professional, supportive, and direct tone with light, punchy phrasing (e.g., retaining the 'splinters' hook) and avoids overly formal or overly casual language. (Points: 8)
4. The overall writing style remains professional, persuasive, and supportive, matching the sales-y, mission-driven voice of the original prompt. (Points: 6)
5. The Overview is rewritten to focus on the client's benefits, using active 'You will...' language and clearly stating what the employer will gain from the objective. (Points: 10)
6. The Overview still covers (a) a clear, public stance on supporting people with convictions, (b) what support will be offered, and (c) how the impact of this work will be reviewed. (Points: 9)
7. The Overview mentions specific places where the public statement will be used and embedded, such as the organisation's website, supply-chain communications, and stakeholder and colleague induction processes. (Points: 7)
8. The Overview notes preparation for negative responses, identification of key stakeholders/champions, and steps to ensure the statement is adopted and understood internally. (Points: 7)
9. Each listed activity explicitly aligns with one of the three delivery methods - Consult, Train, or Support - either by labeling the mode or by describing a function that belongs to that mode. (Points: 8)
10. Activities are specific, actionable tasks (e.g., scoping discussions, policy review, co-design workshop, drafting statement, scenario planning) and do not contain vague or generic language. (Points: 9)
11. All activities directly relate to defining and embedding a public statement on recruiting people with convictions, not to generic DEI or unrelated recruitment tasks. (Points: 8)
12. The Outcomes section lists multiple explicit statements beginning with the required phrase and clearly describes what the client will have, know, or be able to do. (Points: 9)
13. Each outcome is observable or assessable (e.g., a finalized written statement, a communication plan, a stakeholder alignment checklist) and can be verified after delivery. (Points: 9)
14. All activities and outcomes explicitly support consistent, safe, and fair recruitment of people with convictions and do not endorse discriminatory or unsafe practices. (Points: 9)
15. The response acknowledges the need to balance inclusion with safety, safeguarding, and legal/HR compliance in at least one activity or outcome. (Points: 7)
16. An activity or outcome includes a plan for monitoring, reviewing, and iterating the impact of the public commitment over time. (Points: 6)
17. The response references involvement of key internal stakeholders (e.g., HR, legal, communications, leadership) in activities and demonstrates achievement of shared understanding in outcomes. (Points: 7)
18. The text is written clearly and concisely, avoiding unnecessary repetition while covering all required points. (Points: 6)
19. The Outcomes section begins exactly with the phrase: 'By the end of this objective you will have:'. (Points: 5)
20. The Activities section must contain exactly seven bullet items; any additional or missing items cause failure. (Points: 10)
21. Every activity bullet must begin with a bolded mode label exactly matching '**Consult:**', '**Train:**', or '**Support:**' (case-sensitive) followed by a space and the activity description. (Points: 10)
22. Activities must be grouped in strict order: all '**Consult:**' items first, then all '**Train:**' items, and finally all '**Support:**' items. (Points: 9)
23. The Outcomes section must contain exactly five bullet items; any deviation (more or fewer) results in failure. (Points: 10)
24. Each outcome bullet must describe a concrete, verifiable deliverable (e.g., a finalized written statement, a communication plan, a contingency plan, a monitoring framework) and must end with a period. (Points: 9)

E.4 Science

Data Sample: Science

[Query]

In a triangle ABC where $AB=10$ cm, $BC =9$ cm, and $AC=7$ cm, a circle is inscribed with points of contact X, Y, and Z on sides AC, BC, and AB, respectively. Determine the length of BZ.

[Rubric Criteria]

1. The solution explicitly identifies that the quantity to determine is the length of segment BZ on side AB of triangle ABC. (Points: 6)
2. The solution states and correctly applies the property that tangent segments from the same vertex to the incircle are equal (e.g., BZ = BY, AZ = AX, CZ = CY). (Points: 10)
3. All unknown tangent segment lengths are introduced with clear notation (e.g., t_A, t_B, t_C) and the notation is used consistently throughout the solution. (Points: 5)
4. The solution sets up correct equations that express each side length (AB, BC, AC) as the sum of the appropriate tangent segment variables, matching the given lengths 10, 9, and 7. (Points: 9)
5. The solution carries out a step-by-step algebraic manipulation of the equations, without arithmetic errors, to solve for the unknown variables. (Points: 9)
6. The final numeric value reported for BZ is 6 cm, with correct units and no calculation mistake. (Points: 10)
7. Each intermediate step (e.g., adding or subtracting equations) is justified with a brief logical explanation rather than only presenting the final result. (Points: 7)
8. The overall solution follows a logical order: restate the problem, identify properties, formulate equations, solve algebraically, and conclude with the answer. (Points: 5)
9. The solution relies solely on the provided side lengths and standard incircle properties; no external or unjustified assumptions are introduced. (Points: 6)
10. All symbols, segment labels, and equations are written using conventional mathematical notation that is clear and unambiguous. (Points: 5)
11. The final answer is presented in a separate concluding statement, explicitly stating "BZ = 6 cm" (or equivalent), with correct units. (Points: 5)
12. Before solving, the solution briefly restates the given side lengths and the incircle configuration to frame the problem. (Points: 4)
13. The solution remains focused on the required steps, avoiding irrelevant digressions while including all essential reasoning. (Points: 4)
14. The solution concludes with a line that starts exactly with the word " Answer:" followed by a space and then "BZ = <numeric> cm" (numeric value and unit) with no extra characters on that line. (Points: 10)
15. All major algebraic steps are presented as a numbered sequence (e.g., Step 1, Step 2, ...) and later referenced by those numbers in the reasoning. (Points: 8)
16. The tangent segment lengths are introduced using the symbols t_A, t_B, t_C (or equivalent) and these symbols are used consistently throughout the derivation. (Points: 7)

E.5 Chat

Data Sample: Chat

[Query]

Ryan is playing a multiplication game with a pile of 26 cards, each with a number on them. Each turn, he flips over two of the cards, and has to multiply the numbers.\n\nHow many possible outcomes are there on Ryan's first turn flipping two cards?\n\n\n676\n\n\n52\n\n\n650\n\n\n26

[Rubric Criteria]

1. The response explicitly selects 650 as the answer from the four provided options. (Points: 10)
2. The response explains that the first card can be any of 26 and the second any of the remaining 25 (ordered selection), yielding 26 x 25 possibilities. (Points: 9)
3. The response correctly computes 26 x 25 = 650 without arithmetic error. (Points: 8)
4. The response mentions the combination C(26,2) = 325 and shows understanding of the unordered selection method, even if later rejected. (Points: 6)
5. The response explicitly distinguishes between ordered outcomes (permutations) and unordered selections (

combinations) in the context of flipping two cards. (Points: 7)

6. The response justifies why interpreting the phrase "flipping two cards" as a sequential process (order matters) is appropriate for this problem. (Points: 7)
7. The response notes that the combination result 325 is not among the answer choices and uses this fact to clarify the intended interpretation of "possible outcomes." (Points: 5)
8. The response states the number of possible outcomes (650) directly, without only providing the method. (Points: 8)
9. The reasoning contains no internal contradictions, such as mixing ordered and unordered counts inconsistently. (Points: 6)
10. The response employs standard combinatorial notation (e.g., C(26,2), permutations) correctly when discussing the methods. (Points: 4)
11. The explanation is brief and focused, comparable in length to a typical reference answer, avoiding unnecessary tangents. (Points: 4)
12. The response interprets "possible outcomes" as distinct multiplication situations determined by the specific pair of cards, not merely distinct product values. (Points: 5)
13. The solution is presented in a clear step-by-step format (e.g., First card, Second card, Calculation) mirroring reference answer clarity. (Points: 5)
14. The response includes the multiplication calculation formatted as a LaTeX display equation (e.g., $$26 \\times 25 = 650$$). (Points: 10)
15. The reasoning is presented as a numbered list where each step begins with a bold heading (e.g., **First Card Selection:**). (Points: 9)
16. The answer contains a sentence that explicitly ties the exclusion of the unordered count (325) to the given answer options and then justifies selecting the ordered count (650) as the correct choice. (Points: 8)
17. The response states the final answer in its own sentence before any explanatory text, using the exact phrasing "The correct answer is 650." (or equivalent) without additional qualifiers. (Points: 7)
18. The answer mentions the combination formula C(26,2) and also explicitly references the permutation count as 26 x 25 (or P(26,2)), demonstrating awareness of both approaches. (Points: 6)

887