OpenRubrics: Towards Scalable Synthetic Rubric Generation for Reward Modeling and LLM Alignment

Tianci Liu[1,*] **Ran Xu**[2,*] **Tony Yu**[3] **Ilgee Hong**[3]
Carl Yang[2] **Tuo Zhao**[3] **Haoyu Wang**[4]
[1]Purdue University [2]Emory University
[3]Georgia Institute of Technology [4]University at Albany
liu3351@purdue.edu, ran.xu@emory.edu, hwang28@albany.edu

Abstract

Reward modeling lies at the core of reinforcement learning from human feedback (RLHF), yet most existing reward models rely on scalar or pairwise judgments that fail to capture the multifaceted nature of human preferences. Recent studies have explored *rubrics-as-rewards* (RaR) that uses structured criteria to capture multiple dimensions of response quality. However, producing rubrics that are both reliable and scalable remains a key challenge. In this work, we introduce **OpenRubrics**, a diverse, large-scale collection of (prompt, rubric) pairs for training rubric-generation and rubric-based reward models. To elicit discriminative and comprehensive evaluation signals, we introduce *Contrastive Rubric Generation* (CRG), which derives both hard rules (explicit constraints) and principles (implicit qualities) by contrasting preferred and rejected responses. We further remove noisy rubrics via preserving preference–label consistency. Across multiple reward-modeling benchmarks, our rubric-based reward model, RUBRIC-RM, surpasses strong size-matched baselines by 8.4%. These gains transfer to policy models on instruction-following and biomedical benchmarks. The model weights and datasets are publicly available at https://huggingface.co/OpenRubrics.

1 Introduction

Reward modeling is central to reinforcement learning from human feedback (RLHF) and is widely used to align large language models (LLMs) with human preferences (Ouyang et al., 2022; Wu et al., 2023; Bhaskar et al., 2025; Li et al., 2025; Guo et al., 2025c). By assigning a scalar score (Ouyang et al., 2022) or preference label (Chen et al., 2026) to each response, reward modeling provides the optimization signal to steer the policy LLM toward generating helpful and harmless responses.

While lots of efforts have been paid on RL with *verifiable reward* (RLVR) (Lambert et al., 2024; Guo et al., 2025a; Wen et al., 2025; Nie et al., 2026), many high-value applications of LLMs, such as long-form question answering, general helpfulness, operate in inherently subjective domains where correctness cannot be sufficiently captured by binary signals. To bridge this gap, *rubrics-as-rewards (RaR)* (Gunjal et al., 2026) have emerged as a new paradigm for reward modeling. Rubrics include structured natural language criteria that decompose quality into interpretable and measurable dimensions, providing a more consistent and transparent evaluation framework than scalar judgments. For policy models, rubrics also enable optimization to be guided by explicit principles.

Despite their great promise, constructing high-quality rubrics remains an open challenge. Existing benchmarks (Arora et al., 2025) curate rubrics with the effort from domain experts, which is costly and difficult to scale. Recent works (Huang et al., 2025; Viswanathan et al., 2025; Gunjal et al., 2026) typically generate rubrics via direct prompting LLMs, but those approaches suffer from limited quality control over rubrics and can be prohibitively expensive when relying on commercial APIs.

In this work, we present OpenRubrics, a large collection of (prompt, rubrics) pairs to facilitate rubric-generation model training. Specifically, we prompt the LLM to generate two complementary types of rubrics: *hard rules*, which capture explicit and objective constraints specified in the prompt, and *principles*, which summarize implicit and generalizable qualities of strong responses. This design allows the rubrics to capture both surface-level requirements and deeper dimensions of quality. Although hard rules are typically straightforward to extract, the principles are more subtle and require fine-grained reasoning. To address this, we propose *Contrastive Rubric Generation* (CRG), which conditions on user queries paired with chosen and

[*] These authors contributed equally to this work, order was determined randomly (by rolling a die).

rejected responses. By leveraging such contrasts, CRG encourages the model to identify discriminative qualities that distinguish stronger answers from weaker ones, yielding more comprehensive and ranking-aware rubric signals. To further ensure reliability and reduce the noise, we apply preference-label consistency through rejection sampling, retaining only rubrics that yield correct preference predictions.

Our contributions are three-fold:

- We introduce OpenRubrics, a large-scale and diverse collection of rubrics. This dataset enables both rubric generation models and rubric-informed reward modeling at scale.
- We distinguish between two fundamental types of rubrics and propose a *contrastive rubric generation* strategy that trains models to produce comprehensive and discriminative rubrics from prompts and responses. Besides, we introduce *preference–label consistency* that improves the quality and reliability of the rubric.
- We conduct extensive experiments on eight benchmark datasets, where RUBRIC-RM consistently outperforms strong baselines by 8.4%. Moreover, when integrated into policy optimization, RUBRIC-RM consistently yields notable gains on challenging instruction following and medical benchmark. Case studies further verify the benefits of combining hard rules and principles, showing that rubrics help reduce false positives from overly long outputs.

2 Related Works

Reward Modeling. Standard reward models assign scalar scores to responses by applying a ranking loss under the Bradley–Terry framework (Bradley and Terry, 1952; Liu et al., 2025a). To enhance reasoning capability, generative reward models (GenRMs) incorporate synthesized Chains of Thought (CoT), enabling more accurate reward estimation (Ankner et al., 2024; Yu et al., 2025; Zhang et al., 2025a). Beyond the pointwise setting, pairwise reward models have been proposed to directly compare multiple responses (Liu et al., 2025b). More recently, reinforcement learning has been leveraged to further optimize reward models, enabling them to reason explicitly over comparisons and thereby achieve stronger alignment performance (Chen et al., 2025, 2026; Whitehouse et al., 2026; Guo et al., 2025b; Xu et al., 2026a). Orthogonal to these efforts, our work focuses on improving reward modeling quality with structured rubrics. By introducing rubric-based evaluation signals, we complement existing approaches with an additional layer of interpretability that yield performance gains.

Rubrics as Rewards. Recent work has explored rubrics for both evaluation and alignment. Rubrics provide structured assessments of model generations (Arora et al., 2025; Hashemi et al., 2024; Pathak et al., 2025; Lin et al., 2025; Xu et al., 2026b), guide instruction following and domain adaptation (Viswanathan et al., 2025; Gunjal et al., 2026), improve safety via rule-based rewards (Mu et al., 2024), and have been combined with verifiable rewards for reasoning tasks (Huang et al., 2025; Zhou et al., 2025). Yet most existing approaches rely on prompting frontier LLMs to generate rubrics, which limits scalability and consistency. Our work introduces a more scalable framework for *high quality synthetic rubric generation*, improving both reward quality and interpretability at a cheaper cost. Concurrently, Zhang et al. (2026) also investigate rubric generation, but focus on *iterative refinement* to mitigate reward over optimization, whereas we emphasize scalable synthesis and rubric–preference consistency.

3 Preliminaries

Rubrics. We define rubrics as a structured set of evaluation criteria tailored to a given prompt. Formally, let x denote an input prompt and $\hat{y}$ a model-generated response. A rubric $\mathcal{R}(x)$ is represented as a collection of k criteria $\mathcal{R}(x) = \{c_i\}_{i=1}^{k}$, where each c_i denotes a rubric description specifying an aspect of response quality (e.g., factual correctness, reasoning soundness, style).

Rubrics-based Reward Models. Following prior reward modeling work (Liu et al., 2025b; Chen et al., 2026; Guo et al., 2025b), we focus on a comparative setting where the goal is to evaluate the relative quality of two candidate responses. Given a prompt x and two samples $(\hat{y}_1, \hat{y}_2)$, a pairwise rubric-based reward function is defined as

$$\text{reward}_{\text{pair}}(x, \hat{y}_1, \hat{y}_2) = r_\theta(x, \hat{y}_1, \hat{y}_2; \{c_i\}_{i=1}^{k}),$$

where reward is the binary preference label, r_θ is the reward model that integrates rubric criteria $\{c_i\}$ when producing a preference judgment.

The overall framework for OpenRubrics is in Figure 1. Our overall objective is two-fold: (1)

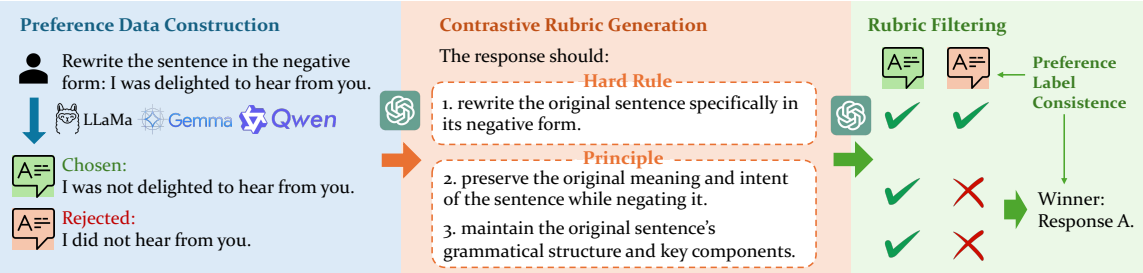


Figure 1: Overall Framework for Synthetic Rubric Generation in OpenRubrics.

constructing a rubric dataset $\mathcal{D}_{\text{rubric}}$ to train a generation model g_θ that automatically synthesizes rubrics $\mathcal{R}(x)$ given a prompt x; and (2) building a reward modeling dataset $\mathcal{D}_{\text{rm}}$ to train a rubric-guided reward model r_ϕ capable of producing reliable and interpretable pairwise judgments. This two-stage formulation decomposes evaluation into *rubric generation* and *rubric-conditioned reward prediction*, bridging human-aligned criteria and automated preference modeling.

4 OpenRubrics

4.1 Data Construction

Data Sources. To generate high-quality rubrics that generalize across tasks and domains, we integrate a wide range of public preference and instruction-tuning datasets, balancing general-domain data with domain-specific resources. Specifically, our dataset sources are drawn from:

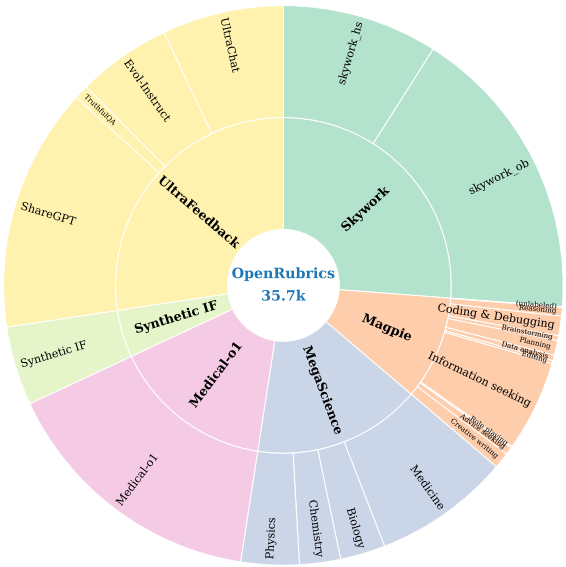
- **UltraFeedback** (Cui et al., 2024), which aggregates preference annotations from *Evol-Instruct* (Xu et al., 2024), *UltraChat* (Ding et al., 2023), *ShareGPT* (Chiang et al., 2023), and *TruthfulQA* (Lin et al., 2022).
- **Magpie** (Xu et al., 2025): a large-scale synthetic alignment dataset generated by self-prompting LLMs across diverse domains.
- **Skywork-Preference** (Liu et al., 2024), which integrates data from *HelpSteer2* (Wang et al., 2024) and *OffsetBias* (Park et al., 2024).
- **Synthetic-IF** (Lambert et al., 2025a), a collection of human preference judgments tailored for verifiable instruction-following.
- **MegaScience** (Fan et al., 2025), a specialized corpus spanning multiple scientific domains including physics, biology, and chemistry.
- **Medical-o1** (Chen et al., 2024), a medical SFT dataset curated for diagnostic reasoning tasks.

Preference Data Construction. To build preference data for rubric generation and judge training (see Sec. 4.3), we reuse existing preference and SFT datasets with tailored processing. For **UltraFeedback**, we select the highest-scoring response as the *chosen* and the lowest as the *rejected*. For **MegaScience**, and **Medical-o1**, we generate multiple responses using *Qwen-3-8B/14B* (Yang et al., 2025), *Llama-3.1-8B* (Grattafiori et al., 2024), and *Gemma-3-12B* (Team et al., 2025), selecting one from each model. For **Synthetic-IF**, responses satisfying all verification functions are labeled as *chosen*, and others as *rejected*. For the **MegaScience** and **Medical-o1** datasets, we employ an ensemble of open-source reward models: *Athene-RM-8B* (Frick et al., 2024a) and *Skywork-Reward-V2-Llama-3.1-8B-40M* (Liu et al., 2025a) to rank responses and form best–worst preference pairs.

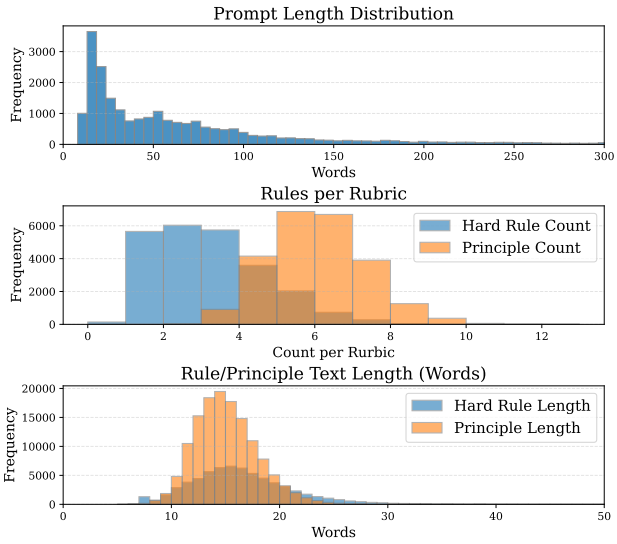
4.2 Rubrics Synthesis

After collecting a diverse set of preference data, our objective is to construct a set of rubrics that serve as *anchors* to guide reward modeling. To comprehensively represent different types of constraints while preserving discriminative granularity, we categorize rubrics into two types: (1) **Hard Rules**, which capture explicit requirements stated in the user's prompt; and (2) **Principles**, which describe higher-level qualitative aspects such as reasoning soundness, factuality, or stylistic coherence. We then introduce two strategies for generating high-quality rubrics as follows:

Contrastive Rubric Generation. Given a dataset $\mathcal{D} = \left\{\left(x_i, \{\hat{y}_{i,m}\}_{m=1}^{M_i}, \{\ell_{i,m}\}_{m=1}^{M_i}\right)\right\}_{i=1}^{N}$, where x_i is the prompt, $\{\hat{y}_{i,m}\}_{m=1}^{M_i}$ and $\{\ell_{i,m}\}_{m=1}^{M_i}$ denote a list of M_i candidate responses and their corresponding preference signals, respectively. The preference signal $\ell_{i,m}$ can be either a real-valued score or a binary chosen–rejected label for pairwise comparison cases ($M_i = 2$). We unify the two cases by constructing a strictly ordered



(a) The data distribution for OpenRubrics (in # instructions).



(b) The distribution for the length of prompts and rubrics, as well as the number of rubrics.

Figure 2: Statistics Overview of OpenRubrics.

list in descending order of preference: $\hat{y}_{i,1} \succ \hat{y}_{i,2} \succ \cdots \succ \hat{y}_{i,M_i}$, where the rank is induced by ℓ. In particular, for pairwise cases, we set $\hat{y}_{i,1}$ as the chosen response and $\hat{y}_{i,2}$ as the rejected one. Our objective is to infer rubrics $\mathcal{R}(x_i)$ that capture the qualities a good response should satisfy and the criteria that explain preference differences across the list, leveraging rich and fine-grained supervision from the comparisons. Formally, we prompt a capable instruction-tuned LLM h_ψ as $\mathcal{R}(x_i) \sim h_\psi\Big(x_i, \{\hat{y}_{i,m}\}_{m=1}^{M_i}, \{\ell_{i,m}\}_{m=1}^{M_i}\Big)$, to produce a set of discriminative evaluation criteria $\mathcal{R}(x_i) = \{c_{i,1}, \ldots, c_{i,k_i}\}$. Here each $c_{i,j}$ describes a specific aspect and is expected to distinguish higher-preference responses from lower-preference ones. This listwise setting encourages the model to discover rubric dimensions that are both task-sensitive and preference-aligned, while exploiting any available fine-grained preference information.

Rubric Filtering with Preference-label Consistency. We note that not all rubrics generated from the previous step faithfully capture human preference signals. To enhance reliability, we propose to incorporate a *consistency*-based filtering step by prompting the same instruction-tuned LLM h_ψ. Specifically, given a prompt x_i and ordered responses $\{\hat{y}_{i,m}\}_{m=1}^{M_i}$, we create a set of *pairwise* comparisons $\mathcal{P}_i = \{(a,b) \mid 1 \le a < b \le M_i\}$, where the human preference label for each pair is $\ell_{i,(a,b)} = 1$ by construction (i.e., $\hat{y}_{i,a} \succ \hat{y}_{i,b}$). For pairwise cases ($M_i = 2$), we have $|\mathcal{P}_i| = 1$. Next, we feed the full rubric $\mathcal{R}(x_i)$ into the context and ask the model to perform a rubric-conditioned judgment on each pair:

$$\hat{l}_{i,(a,b)} = h_\psi\big(x_i, \mathcal{R}(x_i), \hat{y}_{i,a}, \hat{y}_{i,b}\big), \tag{1}$$

where $\hat{l}_{i,(a,b)} = (\hat{r}_{i,(a,b)}, \hat{\ell}_{i,(a,b)})$ denote the prediction rationale and the predicted preference, respectively. A rubric is considered *reliable* for prompt x_i only if it is able to yield preference predictions consistent with human labels on the induced pair set:

$$\mathrm{Acc}_i = \frac{1}{|\mathcal{P}_i|} \sum_{(a,b)\in\mathcal{P}_i} \mathbb{I}\big[\hat{\ell}_{i,(a,b)} = \ell_{i,(a,b)}\big] \ge \tau, \tag{2}$$

where τ is a threshold (we use $\tau = 0.5$ in practice). Finally, for each induced pair $(a,b) \in \mathcal{P}_i$, we retain the rubric-conditioned training instance only if (i) the rubric passes the group-level verification ($\mathrm{Acc}_i \ge \tau$), and (ii) the current pairwise prediction is correct ($\hat{\ell}_{i,(a,b)} = \ell_{i,(a,b)}$). Formally,

$$\mathcal{R}^*_{i,(a,b)} = \begin{cases} \mathcal{R}(x_i), & \text{if } \mathrm{Acc}_i \ge \tau \text{ and} \\ & \hat{\ell}_{i,(a,b)} = \ell_{i,(a,b)}, \\ \varnothing, & \text{otherwise.} \end{cases} \tag{3}$$

This process produces a collection of high-quality rubrics that are both interpretable and empirically consistent with human preferences. The ultimate *rubrics-conditioned preference dataset* is

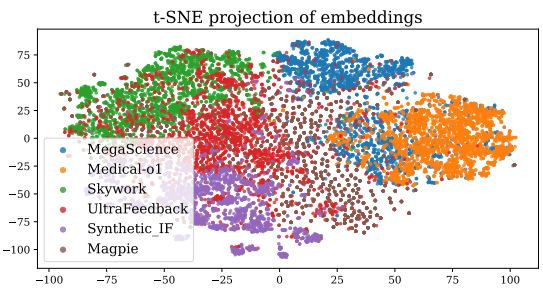


Figure 3: The T-SNE plot for embeddings of prompts.

constructed as a pairwise collection

$$\mathcal{D}_{\text{rubric}} = \Big\{(x_i, \hat{y}_{i,a}, \hat{y}_{i,b}, \ell_{i,(a,b)}, \mathcal{R}^*_{i,(a,b)}) \;\Big|\; i \in [N], (a,b) \in \mathcal{P}_i,\ \mathcal{R}^*_{i,(a,b)} \neq \varnothing\Big\} \triangleq \{(x_i, \hat{y}_i^+, \hat{y}_i^-, \mathcal{R}^*(x_i))\}_{i=1}^M. \tag{4}$$

Namely, we flatten and re-index all retained pairs to obtain the pairwise dataset, where each new instance corresponds to a retained pair $(i,(a,b))$ with $\hat{y}_i^+ = \hat{y}_{i,a}$ and $\hat{y}_i^- = \hat{y}_{i,b}$ according to the preference rank $a < b$.

Rubric Statistics Overview. We analyze the curated rubric set along three axes: (1) domain coverage (instruction following, reasoning, general helpfulness (Figure 2a); (2) the balance between *hard rules* and *principles* as well as the length of prompts and rubrics (Figure 2b); and (3) semantic diversity of prompt topics, visualized via t-SNE on embeddings from Qwen-3-Embedding-0.6B (Zhang et al., 2025b) (Figure 3). These statistics confirm that the synthesized rubrics provide comprehensive, yet discriminative coverage, forming a foundation for rubric-based reward modeling.

4.3 Reward Model Training and Inference

After collecting the rubrics-based dataset, we proceed to develop a rubric generation model that outputs evaluation rubrics and a reward model RUBRIC-RM that generates final preference labels.

Rubric Generation. We first fine-tune g_θ to generate rubrics $\mathcal{R}^*$ conditioned on the prompt x. Given the dataset $\mathcal{D}_{\text{rubric}} = \{(x_i, \hat{y}_i^+, \hat{y}_i^-, \mathcal{R}^*(x_i))\}_{i=1}^M$, we optimize the standard cross-entropy objective:

$$\mathcal{L}_{\text{SFT}}^{\text{rubric}} = -\,\mathbb{E}_{(x,\hat{y}^+,\hat{y}^-,\mathcal{R}^*)\in\mathcal{D}_{\text{rubric}}} \sum_{t=1}^{|\mathcal{R}^*|} \log p_\theta(\mathcal{R}^*_t \mid x, \mathcal{R}^*_{<t}).$$

Reward Model Training. We then train the reward model r_ϕ to predict preference labels $\hat{l}$ using the generated rubrics. Given $\mathcal{D}_{\text{rm}} = \{(x_i, \hat{y}_i^+, \hat{y}_i^-, \mathcal{R}^*(x_i), \hat{l}_i)\}_{i=1}^M$, we train r_ϕ to predict $\hat{l}$ from the prompt, response pair, and rubric:

$$\mathcal{L}_{\text{SFT}}^{\text{rm}} = -\,\mathbb{E}_{\mathcal{D}_{\text{rm}}} \sum_{t=1}^{|\hat{l}|} \log p_\phi(\hat{l}_t \mid x, \hat{y}^+, \hat{y}^-, \mathcal{R}^*(x), \hat{l}_{<t}).$$

Inference. At inference time, given a pairwise test instance (x, y^A, y^B), RUBRIC-RM performs a two-stage process to predict the final preference label: (1) the rubric generator produces (or retrieve a cached) rubric conditioned on the instruction x as $\hat{\mathcal{R}}(x) = g_\theta(x)$; (2) the reward model then predicts the verdict over candidate responses (y^A, y^B) conditioned on the generated rubric from two possible labels $\mathcal{C} = \{\text{A is better}, \text{B is better}\}$:

$$\hat{l} = \arg\max_{k\in\mathcal{C}} p_\phi(k \mid x, y^A, y^B, \hat{\mathcal{R}}(x)).$$

This ensures that the judgment from RUBRIC-RM is explicitly grounded in rubric criteria.

5 Experiment

5.1 Datasets and Experiment Settings

Training data. We train both components of RUBRIC-RM: the *rubric generator* and the *judge*, on the curated OpenRubrics as presented in Sec. 4.2. Rubrics are produced with contrastive signals from chosen/rejected responses and filtered by preference–label consistency before use. Unless otherwise noted, we use the science-related slice of OpenRubrics to better match our domain study on HealthBench/medical evaluation.

Backbone and variants. Both the rubric generator and the judge are fine-tuned from Qwen-3-8B ("RUBRIC-RM-8B") unless specified. At inference time, RUBRIC-RM follows the two-stage process as detailed in Sec 4.3. We also report an ensemble variant, *voting@5*, which aggregates five independently sampled judge trajectories by majority vote.

Baselines. We compare RUBRIC-RM against strong, same-scale white-box reward (judge) models: JudgeLRM-7B (Chen et al., 2025), RRM-7B (Guo et al., 2025b), and RM-R1-7B (Chen et al., 2026). We also report larger RM-R1-14B (Chen et al., 2026) and API judges for reference when available. To isolate the benefit of rubric-aware fine-tuning, we include naive Qwen-3-8B (Rubric+Judge) that directly prompts the base model to produce rubrics and then make judgments.

Evaluation benchmarks and metrics. We evaluate RUBRIC-RM as a pairwise reward model on popular reward-modeling benchmarks: *Reward-Bench* (Chat, Chat-Hard) (Lambert et al., 2025b), *RM-Bench* (Liu et al., 2025c), *PPE-IFEval* (Frick

	RewardBench		IF Evaluation Benchmarks				RM-Bench	RewardBench2		HelpSteer3	Avg.
	Chat	Chat Hard	FollowBench	PPE-IFEval	InfoBench	IFBench	Chat	Precise IF	Focus		
Black-box LLMs (For reference only)											
Claude-3.5-Sonnet	96.4	74.0	–	58.0	–	–	62.5	38.8	87.0	–	-
API (Rubric+Judge)	79.6	79.2	83.2	61.0	82.2	66.2	67.9	42.5	79.6	71.4	71.3
API (direct Judge)	89.6	71.2	81.7	59.2	72.9	60.4	67.2	13.2	63.4	70.3	64.9
Larger White-box LLMs (For reference only)											
RM-R1-14B (Qwen-2.5-Inst)	73.5	79.8	84.0	59.0	85.5	60.8	73.2	23.8	84.6	74.8	69.9
RM-R1-14B (DeepSeek-Dist)	90.3	78.9	89.9	61.2	82.4	59.0	71.4	30.6	79.0	74.6	71.7
White-box Judge/Reward LLMs											
JudgeLRM-7B	**92.1**	56.1	79.8	46.0	62.7	47.5	55.4	9.4	29.1	60.2	53.8
RRM-7B	77.7	69.5	65.5	51.0	68.2	53.2	59.9	10.0	60.4	62.4	57.8
RM-R1-7B (Qwen-2.5-Inst)	83.0	70.0	56.3	55.2	71.3	55.2	64.2	20.6	76.2	65.2	61.7
RM-R1-7B (DeepSeek-Dist)	85.3	67.3	69.7	51.0	70.3	56.5	62.2	13.8	55.4	62.6	59.4
Qwen-3-8B (Rubric+Judge)	74.2	64.2	71.3	57.0	74.3	59.5	63.9	8.1	44.0	60.8	57.7
RUBRIC-RM *(Our proposed model)*											
RUBRIC-RM-4B	87.2	68.9	78.0	64.8	79.3	63.2	62.6	35.6	79.6	64.7	68.4
RUBRIC-RM-4B-voting@5	88.7	70.2	80.7	66.2	83.0	64.9	63.8	38.1	82.6	65.0	70.3
RUBRIC-RM-8B	88.2	74.1	76.1	67.0	80.8	65.4	65.7	34.4	82.2	67.0	70.1
RUBRIC-RM-8B-voting@5	89.9	**75.4**	**81.5**	**70.8**	**83.8**	**67.1**	**67.0**	**40.0**	**86.5**	**67.5**	**73.0**

Table 1: Comparison of different judge and reward models across multiple benchmarks. RewardBench2 reports results on Precise IF, and Focus dimensions. Rubric API uses GPT-4.1-Mini, and Judge API uses Gemini-2.5-Flash-Lite. Best results are highlighted in **bold**.

et al., 2024b), *FollowBench* (Jiang et al., 2024), *InfoBench* (Qin et al., 2024), *IFBench* (Peng et al., 2025), and *RewardBench2* (Precise-IF, Focus) (Malik et al., 2025). While *FollowBench* and *InfoBench* were originally designed to assess instruction-following capabilities of LLMs, we adapt them to pairwise evaluation settings by sampling two responses from the same model (Qwen-3-8B/14B), where one response adheres to all specified constraints and the other violates some of them. For the domain study we additionally report HealthBench/medical results. We follow each benchmark's official splits and scoring rules, reporting accuracy, win-rate or other specific scores.

Due to space limits, additioanl implementation details are deferred to Appendix B.

5.2 Performance of RUBRIC-RM

We first validate the performance of RUBRIC-RM for reward modeling. For a more systematic evaluation, we test both 4B and 8B variants of RUBRIC-RM, which use Qwen3-4B/8B as backbones. Table 1 reports the results of RUBRIC-RM.

Outperforming Comparable Reward Models. Both RUBRIC-RM-4B and 8B surpass all existing 7B-scale white-box baselines (e.g., JudgeLRM, RRM, RM-R1). Notably, RUBRIC-RM-4B (**68.4**) outperforms the strongest 7B competitors (max 61.7), with RUBRIC-RM-8B further improving to **70.1**. This confirms that rubric-aware training yields more reliable signals than generic preference learning, even with reduced parameter counts.

Majority Voting Further Enhances Performance. We also evaluate RUBRIC-RM-voting@5, which aggregates predictions via majority voting across five independent judge trajectories. RUBRIC-RM-4B-voting@5 reaches **70.3**, and RUBRIC-RM-8B-voting@5 achieves the best overall average of **73.0**, surpassing much larger models such as RM-R1-14B (71.7) and the Rubric+Judge API (71.3). These results highlight the stability benefits of rubric-based ensembles.

Effectiveness of Rubric-Aware Fine-Tuning. Directly using Qwen-3-8B to generate rubrics and judge yields only 57.7. In contrast, RUBRIC-RM achieves **70.1**. This **+12.4** gain validates our core contribution: high-quality rubrics derived via *contrastive generation* and *consistency filtering* are essential for effective reward modeling.

Strength on IF Benchmarks. RUBRIC-RM excels on benchmarks requiring fine-grained instruction adherence. On FollowBench and InfoBench, we achieve **81.5** and **83.8** respectively, substantially outperforming baselines like JudgeLRM and RRM. This demonstrates that rubrics capture nuanced constraints better than scalar reward models.

In the remaining experiments, we use RUBRIC-RM-8B as our reward model unless specified.

5.3 Policy Models with RUBRIC-RM

5.3.1 Instruction-Following Evaluation

We further evaluate the effectiveness of using RUBRIC-RM as a reward model for policy optimization on instruction-following tasks, including

Model	IFEval (Prompt)		IFEval (Inst.)		IFEval	InfoBench
	Loose	Strict	Loose	Strict	AVG	AVG
GPT-4 (0314)	79.3	76.9	85.4	83.6	81.3	87.3
AutoIF (Dong et al., 2024)	56.9	47.1	67.0	57.6	57.2	80.6
UltraIF (An et al., 2025)	75.4	71.3	83.0	79.4	77.3	80.7
Qwen2.5-7B-Instruct	75.0	72.5	81.8	79.9	77.3	78.1 (76.0)
+ SFT (Distilled)	66.8	64.1	75.3	72.8	69.8	72.5
+ DPO (via Skywork)	75.7	68.0	83.2	78.5	76.0	82.0
+ DPO (via ArmoRM)	73.8	70.2	81.7	78.3	76.0	83.5
+ DPO (via Ultrafbk.)	71.5	69.1	79.9	77.7	74.6	80.0
+ DPO (via AI Judge)	73.0	68.9	80.9	77.8	75.2	76.1
+ DPO (via RLCF)	77.3	72.6	84.1	80.3	78.6	**84.1** (81.5)
+ DPO (via RUBRIC-RM)	**78.2**	**73.9**	**84.5**	**81.2**	**79.5**	83.0

Table 2: Comparison of trained policy models with different reward models on a format-based instruction-following benchmark (IFEval) and an open-ended benchmark (InfoBench). Baseline results are from Viswanathan et al. (2025). Results with underlines are reproduced by us using official checkpoints and evaluation scripts. Best scores are in **bold**.

Model	Arena-Hard		AlpacaEval		AVG
	Vanilla	SC	Vanilla	LC	
GPT-4 (0314)	50.0	50.0	22.1	35.3	39.4
UltraIF (An et al., 2025)	31.4	–	–	–	–
Qwen2.5-7B-Instruct	51.3	42.8	33.5	36.2	41.0
+ SFT (Distilled)	32.6	29.2	36.1	33.3	32.8
+ DPO (via Skywork)	55.1	50.3	44.8	**41.5**	47.9
+ DPO (via ArmoRM)	50.8	46.4	37.6	38.1	43.2
+ DPO (via Ultrafbk.)	52.8	47.9	33.7	38.7	43.3
+ DPO (via AI Judge)	51.0	44.4	28.8	33.4	39.4
+ DPO (via RLCF)	**54.6**	48.4	36.2	37.1	44.1
+ DPO (via RUBRIC-RM)	52.9	**53.1**	**47.0**	41.3	**48.6**

Table 3: Comparison of different alignment strategies applied to Qwen2.5-7B-Instruct on **Arena-Hard** and **AlpacaEval**. Baseline results are from Viswanathan et al. (2025). SC and LC stands for Style-controlled and Length-controlled. Best results are in **bold**.

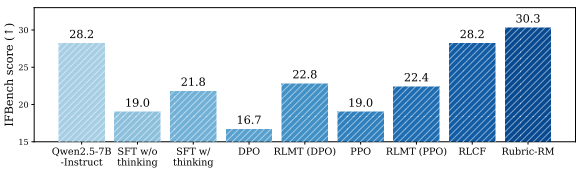


Figure 4: Comparison of policy models on IFBench. Results of baselines except RLCF are from Bhaskar et al. (2025). We evaluate RLCF with its official checkpoint.

IFEval, InfoBench, and IFBench. The results are shown in Table 2 and Figure 4.

Improved Performance on IFEval and InfoBench. Using RUBRIC-RM yields the best overall performance among open-source reward-model baselines: the resulting policy model reaches **79.5** on IFEval (vs. 76.0 with Skywork/ArmoRM) and **83.0** on InfoBench, approaching much larger commercial systems. These results indicate that *rubric-based rewards provide more reliable optimization signals* for constrained instruction following.

Method	Creative	Planning	Math	Info seeking	Coding	WB Score
Claude-3.5-Sonnet*	55.6	55.6	50.2	55.5	56.5	54.7
GPT-4-turbo*	58.7	56.2	51.0	57.2	55.1	55.2
GPT-4o-mini*	60.1	58.2	54.0	57.4	57.2	57.1
Qwen2.5-7B-Instruct*	50.1	51.8	47.1	50.7	45.0	48.7
+DRIFT*	52.5	53.2	50.6	52.4	50.3	51.7
+DPO (via RLCF)	51.4	52.7	49.0	51.3	48.8	50.5
+DPO (via RLMT (PPO))	52.1	52.6	45.2	51.4	48.3	49.7
+DPO (via RUBRIC-RM)	**54.8**	**55.5**	**51.5**	**54.1**	**52.9**	**53.6**

Table 4: Comparison of different alignment strategies applied to Qwen2.5-7B-Instruct on **WildBench**. Results are reported for task-specific scores and task macro WB score. Baseline results with "∗" are from (Wang et al., 2025). We evaluate RLCF and RLMT with their official checkpoints. Best results are in **bold**.

Clear Gains on Complex Instruction Following (IFBench). Figure 4 shows that the policy model optimized with RUBRIC-RM achieves **30.3** on IFBench, substantially higher than RLCF (28.2) and RLMT-based methods (22.4–22.8). Compared with both supervised fine-tuning variants and reinforcement learning baselines, RUBRIC-RM provides stronger inductive biases, enabling policies to better capture fine-grained instruction adherence.

Overall, these results demonstrate that RUBRIC-RM, when used as a reward model, provides a substantially stronger training signal that improves the instruction-following capability of learned policies.

5.3.2 General Alignment Evaluation

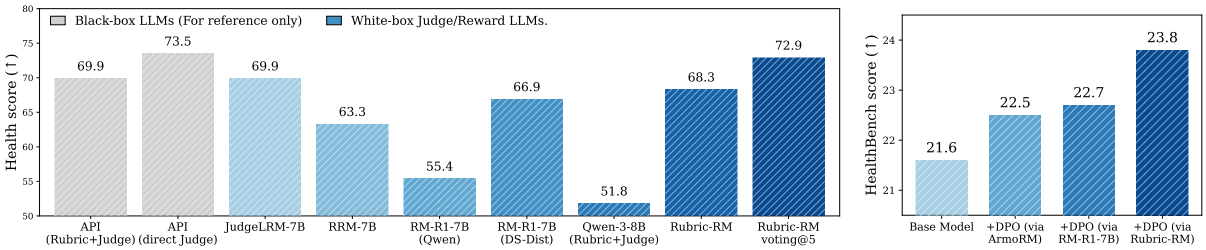
We evaluate policy model trained with RUBRIC-RM on alignment benchmarks Arena-Hard, AlpacaEval, and WildBench (Tables 3 and 4).

With DPO optimization, RUBRIC-RM achieves the best overall average score (**48.6**) among all open-source reward models. On Arena-Hard (style-controlled), it obtains **53.1**, outperforming Skywork (50.3), Ultrafeedback (47.9), and RLCF (48.4). On AlpacaEval, it reaches **47.0**, surpassing Skywork (44.8) and ArmoRM (37.6). On WildBench, aligning Qwen2.5-7B with RUBRIC-RM yields the best macro WB score of **53.6**, outperforming DPO with RLCF (50.5), RLMT (49.7), and DRN (51.7), with consistent gains across all task categories. These results show that rubric-based signals provide reliable gains across both vanilla and controlled settings.

5.4 RUBRIC-RM for BioMedical Domain

We further study the effectiveness of RUBRIC-RM in more specialized biomedical domain, following Arora et al. (2025). The Rubric and Judge are `Qwen-3-8B` backbones fine-tuned with OpenRubrics data from science-related domains.

Performance on HealthBench. RUBRIC-RM



(a) Judges and Reward Models.

(b) Trained Policy Models.

Figure 5: Comparison of different judges, reward models, and trained policy models on HealthBench.

Case Study on RewardBench Chat Hard	
Prompt	Describe a vivid and unique character, using strong imagery and creative language. Please answer in *fewer than two paragraphs*.
Resp A (snippet)	"She was a woman of great power and influence ... with a deep voice and piercing eyes ..." (*single paragraph, on-task*)
Resp B (snippet)	"... a fierce and determined young woman ... [*paragraph break*] ... Which of the following is an example of a character ... [*paragraph break*] A character ... Lily ..."
Label	Resp A.
RRM-7B	"... both responses are within that limit ... Assistant 2's response is more detailed ... (Choose B)"
JudgeLRM	"... Assistant 2 provided a more vivid description ... (Choose B)"
RUBRIC-RM	
Rubric	"1. Describe a vivid and unique character. [Hard Rule] 2. Use strong imagery and creative language. [Hard Rule] 3. Be fewer than two paragraphs. [Hard Rule] 4. Employ sensory-rich language. [Principle] 5. Use original expressions. [Principle] 6. Maintain a consistent tone. [Principle] 7. Remain cohesive and focused. [Principle] 8. Avoid unnecessary elaboration. [Principle]"
Judge	"The paragraph limit is the most verifiable constraint. Response B fails the paragraph rule, while Response A satisfies all hard rules and most principles ... (Choose A)"

Table 5: Case study with error highlighting. Baselines favor the longer, imagery-rich response and miss the explicit paragraph constraint, while RUBRIC-RM enforces hard rules before evaluating principles.

outperforms comparable-size reasoning reward models on HealthBench, achieving **68.3**, exceeding RRM-7B (63.3) and RM-R1-7B variants (55.4/66.9), and approaching RM-R1-14B (69.9). Majority voting further boosts RUBRIC-RM to **72.9**, narrowing the gap to 14B reasoning models and API-based judges. A key finding is the importance of domain- specific rubric SFT. Directly prompting `Qwen-3-8B` with a Rubric+Judge pipeline yields only 51.8, whereas RUBRIC-RM improves performance by **+16.5**. This highlights our core contribution: contrastive, domain-tuned rubric training produces precise evaluation signals than on-the-fly rubric generation.

Preference Optimization with RUBRIC-RM on HealthBench. We use RUBRIC-RM as the preference judge for DPO on HealthBench. Starting from Qwen-2.5-7B-Instruct (21.6), DPO with ArmoRM and RM-R1-7B improves performance to 22.5 and 22.7, respectively. In contrast, DPO with RUBRIC-RM achieves the best result at **23.8**, yielding a consistent **+1.1–1.3** gain over strong 7B reasoning rewards. These results demonstrate that rubric-aware, domain-specialized reward models translate directly into stronger biomedical policy learning, outperforming generative reasoning rewards.

	Compute Time (sec.)
JudgeLRM-7B	25.71
RRM-7B	203.4
RM-R1-7B (Qwen-2.5-Inst)	260.37
RM-R1-7B (DeepSeek-Dist)	170.76
RM-R1-14B (Qwen-2.5-Inst)	322.79
RM-R1-14B (DeepSeek-Dist)	382.02
RUBRIC-RM-8B	130.77

Table 6: Computing speed on 100 samples (vLLM).

5.5 Efficiency Comparison

Table 6 reports wall-clock time on 100 randomly sampled prompts from RewardBench2. Despite using two `Qwen-3-8B` components (rubric generator + judge), RUBRIC-RM runs in **130.77s**, which is **no slower** than existing reasoning reward models, including RRM-7B (203.4s) and RM-R1-7B/14B (170.8–382.0s). It is substantially faster than 14B R1 variants and competitive with strong 7B reasoning baselines. Efficiency stems from our architecture: rather than long Chain-of-Thought decod-

ing, we use short rubric generation followed by a lightweight judge. Moreover, rubrics are amortizable and can be cached for reuse across many examples, removing rubric generation cost during large-scale scoring. Although JudgeLRM achieves lower latency, it lacks the explicit, interpretable signals necessary for downstream policy optimization.

5.6 Case Studies

We conclude this section with a case study that illustrates how RUBRIC-RM handles challenging inputs and improves reward modeling. Table 5 shows the case study on RewardBench, where both responses are vivid, but the instruction requires *fewer than two paragraphs*. The baselines miss this hard constraint and incorrectly prefer the longer answer, exhibiting a verbosity bias and an instruction violation. In contrast, RUBRIC-RM first applies a simple structural check to reject the non-compliant candidate, then compares higher-level qualities (e.g., imagery, originality, focus), and selects the correct response. This example shows how long CoT can still overlook explicit constraints, while rubric-based decomposition makes such failures clear.

6 Conclusion

We present OpenRubrics, a large-scale dataset and framework for scalable and high-quality rubric generation. By decomposing evaluation into hard rules and principles through Contrastive Rubric Generation and applying preference–label consistency filtering, we construct discriminative rubric signals that better align with human judgment. Our rubric-based reward model, RUBRIC-RM, delivers an average 8.4% improvement across benchmarks and further boosts policy performance on diverse benchmarks with offline reinforcement learning. These results position rubrics-as-rewards as a practical foundation for generalizable LLM alignment.

Limitations

While OpenRubrics demonstrates strong empirical gains, several limitations remain. First, our rubric generation relies on contrastive signals from preference data. Although consistency filtering reduces noise, the resulting rubrics may still reflect biases present in the underlying models and datasets, particularly for subjective or culturally nuanced criteria. Second, our framework focuses on pairwise comparative evaluation; extending rubric-based rewards to absolute scoring or multi-response ranking remains an open challenge. Finally, we primarily evaluate rubric-based rewards in offline preference optimization. How such structured rewards interact with fully online RLHF pipelines remains an important direction for future work.

References

Kaikai An, Li Sheng, Ganqu Cui, Shuzheng Si, Ning Ding, Yu Cheng, and Baobao Chang. 2025. UltraIF: Advancing instruction following from the wild. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18711–18726, Suzhou, China. Association for Computational Linguistics.

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.

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*.

Adithya Bhaskar, Xi Ye, and Danqi Chen. 2025. Language models that think, chat better. *arXiv preprint arXiv:2509.20357*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025. Judgelrm: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru WANG, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. 2026. RM-r1: Reward modeling as reasoning. In *The Fourteenth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Self-play with execution feedback: Improving instruction-following capabilities of large language models. *arXiv preprint arXiv:2406.13542*.

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*.

Evan Frick, Peter Jin, Tianle Li, Karthik Ganesan, Jian Zhang, Jiantao Jiao, and Banghua Zhu. 2024a. Athene-70b: Redefining the boundaries of post-training for open models.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024b. How to evaluate reward models for rlhf. *Preprint*, arXiv:2410.14872.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. 2026. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *The Fourteenth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2025b. Reward reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yifu Guo, Jiaye Lin, Huacan Wang, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, and Mingguang Chen. 2025c. SE-agent: Self-evolution trajectory optimization in multi-step reasoning with LLM-based agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

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*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Follow-Bench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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*.

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, and 4 others. 2025a. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025b. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. 2025. Curriculum-rlaif: Curriculum alignment with reinforcement learning from ai feedback. *arXiv preprint arXiv:2505.20075*.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2025. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and 1 others. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025b. Pairjudge rm: Perform best-of-n sampling with knockout tournament. *arXiv preprint arXiv:2501.13007*.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025c. RM-bench: Benchmarking reward models of language models with subtlety and style. In *The Thirteenth International Conference on Learning Representations*.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian D Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. 2024. Rule based rewards for language model safety. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. Attnpo: Attention-guided process supervision for efficient reasoning. *arXiv preprint arXiv:2602.09953*.

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.

Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067, Miami, Florida, USA. Association for Computational Linguistics.

Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Arnav Ramamoorthy, Pratyush Ghosh, Aaryan Raj Jindal, Shreyash Verma, Aditya Mittal, Aashna Ased, Chirag Khatri, and 1 others. 2025. Rubric is all you need: Improving llm-based code evaluation with question-specific rubrics. In *Proceedings of the 2025 ACM Conference on International Computing Education Research*, pages 181–195.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. 2025. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15934–15949, Vienna, Austria. Association for Computational Linguistics.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025. Checklists are better than reward models for aligning language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yifan Wang, Bolian Li, Junlin Wu, Zhaoxuan Tan, Zheli Liu, Ruqi Zhang, Ananth Grama, and Qingkai Zeng. 2025. Drift: Learning from abundant user dissatisfaction in real-world preference learning. *arXiv preprint arXiv:2510.02341*.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, and 1 others. 2025. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason E Weston, Ilia Kulikov, and Swarnadeep Saha. 2026. J1: Incentivizing thinking in LLM-as-a-judge

via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Ran Xu, Jingjing Chen, Jiayu Ye, Yu Wu, Jun Yan, Carl Yang, and Hongkun Yu. 2026a. Incentivizing agentic reasoning in llm judges via tool-integrated reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tao Zhao, and Haoyu Wang. 2026b. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable llm post-training. *Preprint*, arXiv:2602.01511.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *The Thirteenth International Conference on Learning Representations*.

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*.

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. 2025. Self-generated critiques boost reward modeling for language models. In *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)*, pages 11499–11514, Albuquerque, New Mexico. Association for Computational Linguistics.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. 2026. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. In *The Fourteenth International Conference on Learning Representations*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025a. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

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.

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*.

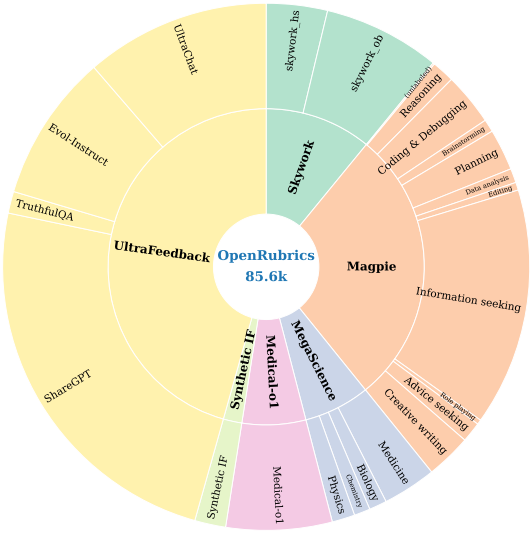


Figure 6: The data distribution for OpenRubrics (in # pairwise judge).

A Additional Rubric Statistics

We present additional rubric statistics in the number of preference pairs in Figure 6.

B Additional Implementation Details

B.1 Decoding and efficiency protocol

All models are run under matched decoding budgets (temperature, max tokens, and stop conditions per benchmark recommendations). We use unified execution stack vLLM (Kwon et al., 2023) for throughput-fair comparisons. For efficiency (Table 6), we measure wall-clock time to score a fixed set of prompts; note that rubrics from stage (i) are cacheable and can be reused across examples, amortizing the cost in large-scale judging and preference optimization.

B.2 Hyper-parameters

Table 7 details the hyper-parameters used in RUBRIC-RM and policy model training, which were conducted in LLaMA-Factory (Zheng et al., 2024). Moreover, Table 8 presents sampling parameters used in OpenRubrics curation and RUBRIC-RM inference. For baseline methods, we adopted the sampling parameters from their official implementations and papers.

B.3 Additional Case Study

Another example is shown in Table 9. This example is more challenging: both answers are longer and the quality gap is subtle. Baselines nevertheless produce factual mistakes about the evidence, e.g., asserting that the better response lacks a date/citation, despite it *correctly providing* a BloombergNEF quote with a *May 16, 2024* publication date and concrete figures ($387B cumu-

	Parameter	Value
RUBRIC-RM *SFT*		
Rubric-Generator	Epochs	1
	Cutoff Length	3072
	Batch Size	128
	Optimizer	AdamW
	Learning Rate	8×10^{-6}
	LR Schedule	Cosine
	Warmup	0.05
Judge-Generator	Epochs	2
	Cutoff Length	6144
	Batch Size	64
	Optimizer	AdamW
	Learning Rate	5×10^{-6}
	LR Schedule	/
	Warmup	/
Policy Model DPO		
Policy Model	Epochs	1
	Cutoff Length	2048
	Batch Size	64
	Optimizer	AdamW
	Learning Rate	3×10^{-7}
	LR Schedule	/
	Warmup	/
	SFT mixing weight	0.1
	β	0.1

Table 7: Hyper-parameters used in RUBRIC-RM and policy model training.

	Parameter	Value
OpenRubrics Curation		
Rubric-Generator	Model	GPT-4.1-Mini
	Maximum Tokens	768
	Temperature	0.0
	Top-P	/
	Top-K	/
Judge-Generator	Model	Gemini-2.5-Flash-Lite
	Maximum Tokens	2048
	Temperature	0.0
	Top-P	/
	Top-K	/
RUBRIC-RM *Inference*		
Rubric-Generator	Base-Model	Qwen-3-4B/8B (Default)
	Maximum Tokens	1024
	Temperature	0.0
	Top-P	/
	Top-K	/
	Enable-thinking	False
Judge-Generator	Model	Qwen-3-4B/8B (Default)
	Maximum Tokens	4096
	Temperature	0.7
	Top-P	1.0
	Top-K	-1 (All)
	Enable-thinking	False

Table 8: Sampling parameters used in OpenRubrics curation and RUBRIC-RM inference.

Case Study on FollowBench	
Prompt	Would you consider direct air carbon capture as a significant financial venture? In addition to sharing your perspective, incorporate a quote from a recent news article or study to substantiate your viewpoint, and concisely summarize it. Also, ensure to mention the publication date of the referenced source. Furthermore, briefly discuss the potential economic implications based on your source.
Resp A (snippet)	"Direct air carbon capture (DAC) is a promising technology . . . According to a recent study published in *Nature Energy* in *March 2023* . . . In conclusion, while there are still challenges to overcome, the recent study suggests that DAC has the potential to become a financially viable option."
Resp B (snippet)	"'. . . Would you consider direct air carbon capture as a significant financial venture? *Yes, I absolutely believe* . . . a recent report from *BloombergNEF*, published on *May 16, 2024*, titled . . . BNEF's projection highlights several key economic implications: 1. . . . "
Ground Truth	Resp B.
RM-R1-7B (DS)	" . . . looking at Chatbot A's response. It starts by explaining DAC briefly and then moves into the study from Nature Energy in 2023 . . . Chatbot B' . . . The quote is from BloombergNEF, but the publication date isn't provided . . . (Choose A)"
RM-R1-7B (Qwen)	" . . . Chatbot A follows all instructions by providing a quote from a recent study . . . Chatbot B appears to have misunderstood the prompt . . . doesn't provide a proper citation with the requested publication date . . . (Choose A)"
RUBRIC-RM	
Rubric	"1. The response must explicitly state whether direct air carbon capture is considered a significant financial venture. [Hard Rule] 2. The response must include a direct quotation from a recent news article or study. [Hard Rule] 3. The response must provide a concise summary of the quoted source. [Hard Rule] 4. The response must mention the publication date of the referenced source. [Hard Rule] 5. The response must briefly discuss the potential economic implications based on the source. [Hard Rule] 6. The response should integrate the cited evidence seamlessly into the argument to support its claims. [Principle] 7. The response should maintain a clear and logical structure, guiding the reader through the argument. [Principle] 8. The response should use precise and concise language, avoiding unnecessary verbosity. [Principle] 9. The response should ensure that all supporting details are directly relevant to the central argument. [Principle] 10. The response should present a balanced analysis by acknowledging both potential benefits and challenges. [Principle]"
Judge	"The most objective rule is the requirement to explicitly state whether direct air carbon capture is considered a significant financial venture . . . Response A fails . . . not explicitly stating whether DAC is . . . (Choose B)"

Table 9: FollowBench case study with error highlighting. Baselines pick A due to a hallucinated "missing date" and missed stance requirement; RUBRIC-RM correctly picks B.

lative investment). Our rubric-aware judge identifies *recency* and *verifiability* as hard requirements (quote, date, concise summary, and economic implications), and favors the response that meets them. This demonstrates RUBRIC-RM's robustness to *citation hallucinations* and over-weighting of "academic-looking" prose that misled generative reasoning RMs.

B.4 Prompts

We present the prompts we used in this subsection. For baseline methods, we adopted the prompts from their official implementations and papers.

Prompt for Listwise Contrastive Rubric Generation (OpenRubrics Curation)

```
You are an expert in pedagogy and critical thinking. Your mission is to create a universal scoring rubric based on a user's
      request and an ordered list of example responses. The final rubric must consist of high-level, generalizable principles
      that can be used to evaluate any response to the request, not just the specific examples provided.

======================================================================
Methodology - A Three-Step Process for Principled Rubric Design
======================================================================

1. Step 1: Extract Explicit Requirements.
   - Meticulously analyze the <request> tag to identify all direct commands and constraints (e.g., length, format, style).
   - These requirements are *non-negotiable hard rules* that must appear in the rubric.
   - They should be clearly labeled as [Hard Rule] in the final output.

2. Step 2: Analyze the Ordered Examples for Specific Differences.
   - Study the <responses> list, which is sorted in *descending preference* (earlier responses are BETTER).
   - Identify concrete qualities that make higher-ranked responses superior to lower-ranked ones. Consider both adjacent
      comparisons (i vs. i+1) and contrasts between the top responses and the rest.
   - It is acceptable at this stage to note topic-specific observations (e.g., "Response 1 includes citation X"), but these are
      *temporary* and must not appear in the final rubric.
   - Every such observation must then be abstracted in Step 3.

3. Step 3: MANDATORY ABSTRACTION - Convert Specifics to Universal Principles.
   - This is the most critical step. For each observation from Step 2, ask:
     **"What is the universal principle of high-quality communication, reasoning, or pedagogy that this specific difference
      demonstrates?"**
   - Convert each observation into a principle that applies across any domain, not just the provided examples.
   - Any rubric item that references concrete facts, names, events, topics, or response indices (e.g., "Response #1") is
      INVALID.
   - All such principles must be labeled as [Principle] in the final output.

======================================================================
Strict Guidelines for Final Output
======================================================================

- **Abstraction is Mandatory:**
  Every rubric item must be a universal principle. If any rubric still contains topic-specific references (e.g., names, places,
       myths, numbers, historical facts), or mentions response indices/positions, it is automatically invalid.

- **Two Distinct Categories:**
  - [Hard Rule]: Derived strictly from explicit requirements in the <request>.
  - [Principle]: Derived from abstracted differences in Step 3.

- **Comprehensiveness:**
  The rubric must cover all critical aspects implied by the request and examples, including explicit requirements and implicit
      quality standards.

- **Conciseness & Uniqueness:**
  Each rubric must capture a distinct evaluation criterion. Overlapping or redundant criteria must be merged into a single
      rubric. Wording must be precise and free of repetition.

- **Format Requirements:**
  - Use a numbered list.
  - Each item starts with "The response..." phrased in third person.
  - Append [Hard Rule] or [Principle] at the end of each item.
  - Do not include reasoning, explanations, or examples in the final outputâĂŤonly the rubrics.

- **Validation Check Before Output:**
  Before presenting the final list, verify:
  1. Does every rubric meet the abstraction requirement (no topic-specific details, no reference to response indices)?
  2. Are all hard rules from Step 1 included?
  3. Are all principles unique and non-overlapping?
  4. Is the list written entirely in third person, concise, and consistent?

======================================================================
Final Output Format
======================================================================

1. The response ... [Hard Rule]
2. The response ... [Principle]
3. The response ... [Principle]
... (continue until all rules and principles are listed)

======================================================================

<request>
{request}
</request>

<context>
{context}
</context>

<responses>
{responses}
</responses>
```

Prompt for Pairwise Contrastive Rubric Generation (OpenRubrics Curation)

```
You are an expert in pedagogy and critical thinking. Your mission is to create a universal scoring rubric based on a user's
      request and a set of examples. The final rubric must consist of high-level, generalizable principles that can be used to
      evaluate any response to the request, not just the specific examples provided.

================================================================
Methodology - A Three-Step Process for Principled Rubric Design
================================================================

1. Step 1: Extract Explicit Requirements.
   - Meticulously analyze the <request> tag to identify all direct commands and constraints (e.g., length, format, style).
   - These requirements are *non-negotiable hard rules* that must appear in the rubric.
   - They should be clearly labeled as [Hard Rule] in the final output.

2. Step 2: Analyze the Examples for Specific Differences.
   - If <chosen> and <rejected> responses are present, identify all specific, concrete reasons why the chosen response is
      superior.
   - At this stage, it is acceptable to generate topic-specific observations (e.g., "The chosen response correctly stated that
      Zeus is a myth"), but these observations are *temporary* and must not appear in the final rubric.
   - Every such observation must then be abstracted in Step 3.

3. Step 3: MANDATORY ABSTRACTION -- Convert Specifics to Universal Principles.
   - This is the most critical step. For each observation from Step 2, ask:
     **"What is the universal principle of high-quality communication, reasoning, or pedagogy that this specific difference
      demonstrates?"**
   - Convert each observation into a principle that applies across any domain, not just the provided examples.
   - Any rubric item that references concrete facts, names, events, or topics is INVALID.
   - All such principles must be labeled as [Principle] in the final output.

================================================================
Strict Guidelines for Final Output
================================================================

- **Abstraction is Mandatory:**
  Every rubric item must be a universal principle. If any rubric still contains topic-specific references (e.g., names, places,
      myths, numbers, historical facts), it is automatically invalid.

- **Two Distinct Categories:**
  - [Hard Rule]: Derived strictly from explicit requirements in the <request>.
  - [Principle]: Derived from abstracted differences in Step 3.

- **Comprehensiveness:**
  The rubric must cover all critical aspects implied by the request and examples, including explicit requirements and implicit
      quality standards.

- **Conciseness & Uniqueness:**
  Each rubric must capture a distinct evaluation criterion. Overlapping or redundant criteria must be merged into a single
      rubric. Wording must be precise and free of repetition.

- **Format Requirements:**
  - Use a numbered list.
  - Each item starts with "The response..." phrased in third person.
  - Append [Hard Rule] or [Principle] at the end of each item.
  - Do not include reasoning, explanations, or examples in the final output-only the rubrics.

- **Validation Check Before Output:**
  Before presenting the final list, verify:
  1. Does every rubric meet the abstraction requirement (no topic-specific details)?
  2. Are all hard rules from Step 1 included?
  3. Are all principles unique and non-overlapping?
  4. Is the list written entirely in third person, concise, and consistent?

================================================================
Final Output Format
================================================================

1. The response ... [Hard Rule]
2. The response ... [Principle]
3. The response ... [Principle]
... (continue until all rules and principles are listed)

================================================================

<request>
{request}
</request>

<context>
{context}
</context>

<chosen>
{chosen}
</chosen>

<rejected>
{rejected}
</rejected}>
```

Prompt for General Domain Judge Generation (OpenRubrics Curation)

```
You are a fair and impartial judge. Your task is to evaluate 'Response A' and 'Response B' based on a given instruction
      and a rubric. You will conduct this evaluation in distinct phases as outlined below.

### Phase 1: Compliance Check Instructions
First, identify the single most important, objective 'Gatekeeper Criterion' from the rubric.
- **A rule is objective (and likely a Gatekeeper) if it can be verified without opinion. Key examples are: word/paragraph
      limits, required output format (e.g., JSON validity), required/forbidden sections, or forbidden content.**
- **Conversely, a rule is subjective if it requires interpretation or qualitative judgment. Subjective rules about quality
       are NOT Gatekeepers. Examples include criteria like "be creative," "write clearly," "be engaging," or "use a
      professional tone."**
Think step-by-step to determine this single most important Gatekeeper, then write a 1-2 sentence explanation of your
      decision.

### Phase 2: Analyze Each Response
Next, for each Gatekeeper Criterion and all other criteria in the rubric, evaluate each response item by item.
For each item, think step-by-step and cite concrete evidence from the response before assigning your judgment.

### Phase 3: Final Judgment Instructions
Based on the results from the previous phases, determine the winner using these simple rules. Provide a final
      justification explaining your decision first and then give your decision.
Think step-by-step to aggregate the findings and make the decision; keep the reasoning explicit and concise.

---
### REQUIRED OUTPUT FORMAT
You must follow this exact output format below.

--- Compliance Check ---
Gatekeeper Reasoning: <1-2 sentences citing the relevant rubric text>
Identified Gatekeeper Criterion: <e.g., Criterion 1: Must be under 50 words.>

--- Analysis ---
**Response A:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

**Response B:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

--- Final Judgment ---
Aggregation Summary: <1-3 sentences explaining how Gatekeeper and other criteria led to the decision>
Justification: <...>
Winner: <Response A / Response B>


Task to Evaluate:
Instruction:
{instruction}

Rubric:
{rubric}

Response A:
{response_a}

Response B:
{response_b}
```

Prompt for Medical Domain Judge Generation (OpenRubrics Curation)

```
You are a fair and impartial judge. Your task is to evaluate 'Response A' and 'Response B' based on a given instruction
      and a rubric. You will conduct this evaluation in distinct phases as outlined below.

### Phase 1: Compliance Check Instructions
First, identify the single most important, objective 'Gatekeeper Criterion' from the rubric.
- **A rule is objective (and likely a Gatekeeper) if it can be verified without opinion. Key examples are: word/paragraph
      limits, required output format (e.g., JSON validity), required/forbidden sections, or forbidden content.**
- **Conversely, a rule is subjective if it requires interpretation or qualitative judgment. Subjective rules about quality
       are NOT Gatekeepers. Examples include criteria like "be creative," "write clearly," "be engaging," or "use a
      professional tone."**

### Phase 2: Analyze Each Response
Next, for each Gatekeeper Criterion and all other criteria in the rubric, evaluate each response item by item.

### Phase 3: Final Judgment Instructions
Based on the results from the previous phases, determine the winner using these simple rules. Provide a final
      justification explaining your decision first and then give your decision.

---
### REQUIRED OUTPUT FORMAT
You must follow this exact output format below.

--- Compliance Check ---
Identified Gatekeeper Criterion: <e.g., Criterion 1: Must be under 50 words.>

--- Analysis ---
**Response A:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

**Response B:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

--- Final Judgment ---
Justification: <...>
Winner: <Response A / Response B>


Task to Evaluate:
Instruction:
{instruction}

Rubric:
{rubric}

Response A:
{response_a}

Response B:
{response_b}
```

Prompt for Rubric Generation (RUBRIC-RM)

```
Your task is to extract a set of rubric-style instructions from a user's request.
These rubrics will be used as evaluation criteria to check if a response fully meets the request.
Every rubric item must be a universal principle. If any rubric still contains topic-specific references (e.g., names,
     places, myths, numbers, historical facts), it is automatically invalid.

- **Two Distinct Categories:**
  - [Hard Rule]: Derived strictly from explicit requirements stated in the <request> (format, length, structure, forbidden
     /required elements, etc.).
  - [Principle]: Derived by abstracting any concrete cues into domain-agnostic quality criteria (e.g., clarity,
     correctness, sound reasoning, pedagogy).

- **Comprehensiveness:**
  The rubric must cover all critical aspects implied by the request and examples, including explicit requirements and
     implicit quality standards.

- **Conciseness & Uniqueness:**
  Each rubric must capture a distinct evaluation criterion. Overlapping or redundant criteria must be merged into a single
      rubric. Wording must be precise and free of repetition.

- **Format Requirements:**
  - Use a numbered list.
  - Each item starts with "The response" phrased in third person.
  - Append [Hard Rule] or [Principle] at the end of each item.
  - Do not include reasoning, explanations, or examples in the final outputâĂŤonly the rubrics.

Here is the request:
{prompt}

Please generate the rubrics for the above request.
```

Prompt for General Domain Judge Generation (RUBRIC-RM)

```
You are a fair and impartial judge. Your task is to evaluate 'Response A' and 'Response B' based on a given instruction
      and a rubric. You will conduct this evaluation in distinct phases as outlined below.

### Phase 1: Compliance Check Instructions
First, identify the single most important, objective 'Gatekeeper Criterion' from the rubric.
- **A rule is objective (and likely a Gatekeeper) if it can be verified without opinion. Key examples are: word/paragraph
      limits, required output format (e.g., JSON validity), required/forbidden sections, or forbidden content.**
- **Conversely, a rule is subjective if it requires interpretation or qualitative judgment. Subjective rules about quality
       are NOT Gatekeepers. Examples include criteria like "be creative," "write clearly," "be engaging," or "use a
      professional tone."**
Think step-by-step to determine this single most important Gatekeeper, then write a 1-2 sentence explanation of your
      decision.

### Phase 2: Analyze Each Response
Next, for each Gatekeeper Criterion and all other criteria in the rubric, evaluate each response item by item.
For each item, think step-by-step and cite concrete evidence from the response before assigning your judgment.

### Phase 3: Final Judgment Instructions
Based on the results from the previous phases, determine the winner using these simple rules. Provide a final
      justification explaining your decision first and then give your decision.
Think step-by-step to aggregate the findings and make the decision; keep the reasoning explicit and concise.

---
### REQUIRED OUTPUT FORMAT
You must follow this exact output format below.

--- Compliance Check ---
Gatekeeper Reasoning: <1-2 sentences citing the relevant rubric text>
Identified Gatekeeper Criterion: <e.g., Criterion 1: Must be under 50 words.>

--- Analysis ---
**Response A:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

**Response B:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

--- Final Judgment ---
Aggregation Summary: <1-3 sentences explaining how Gatekeeper and other criteria led to the decision>
Justification: <...>
Winner: <Response A / Response B>


Task to Evaluate:
Instruction:
{instruction}

Rubric:
{rubric}

Response A:
{response_a}

Response B:
{response_b}
```

Prompt for Medical Domain Judge Generation (RUBRIC-RM)

```
You are a fair and impartial judge. Your task is to evaluate 'Response A' and 'Response B' based on a given instruction
      and a rubric. You will conduct this evaluation in distinct phases as outlined below.

### Phase 1: Compliance Check Instructions
First, identify the single most important, objective 'Gatekeeper Criterion' from the rubric.
- **A rule is objective (and likely a Gatekeeper) if it can be verified without opinion. Key examples are: word/paragraph
      limits, required output format (e.g., JSON validity), required/forbidden sections, or forbidden content.**
- **Conversely, a rule is subjective if it requires interpretation or qualitative judgment. Subjective rules about quality
       are NOT Gatekeepers. Examples include criteria like "be creative," "write clearly," "be engaging," or "use a
      professional tone."**

### Phase 2: Analyze Each Response
Next, for each Gatekeeper Criterion and all other criteria in the rubric, evaluate each response item by item.

### Phase 3: Final Judgment Instructions
Based on the results from the previous phases, determine the winner using these simple rules. Provide a final
      justification explaining your decision first and then give your decision.

---
### REQUIRED OUTPUT FORMAT
You must follow this exact output format below.

--- Compliance Check ---
Identified Gatekeeper Criterion: <e.g., Criterion 1: Must be under 50 words.>

--- Analysis ---
**Response A:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

**Response B:**
- Criterion 1 [Hard Rule]: Justification: <...>
- Criterion 2 [Hard Rule]: Justification: <...>
- Criterion 3 [Principle]: Justification: <...>
- ... (and so on for all other criteria)

--- Final Judgment ---
Justification: <...>
Winner: <Response A / Response B>

Task to Evaluate:
Instruction:
{instruction}

Rubric:
{rubric}

Response A:
{response_a}

Response B:
{response_b}
```