CONFORMAL RUBRICS: CALIBRATING RUBRIC REWARDS WITH HUMAN SCORE FOR POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubrics provide explicit criteria, such as correctness and helpfulness, for evaluating language model responses and constructing rewards for policy learning. Recent methods improve these rewards by refining the criteria or adjusting how their scores are combined. However, specifying what to evaluate does not ensure that a judge model assigns scores consistent with human judgments. Human ratings provide a reference for correcting these scores, but collecting ratings for every new response during policy training would be costly. We therefore propose Conformal Rubrics, which use a fixed human-rated dataset to calibrate rubric rewards for downstream policy training. The key challenge is estimating how humans would score responses on new tasks, where queries differ from those in the annotated dataset. We train a predictor on the source ratings and use its prediction errors to construct a conformal interval to include possible human scores for each rubric with a pre-specified coverage level. We then calibrate judge scores against these intervals by clipping scores outside them to the nearest boundary and retaining those inside. The corrected scores are then combined into rewards using existing rubric-based methods. Experiments show that calibration with the human-rated dataset HelpSteer reduces judges' rubric scoring errors and improves policy performance on different datasets, such as HelpSteer2, WildChat, and AlpacaEval-2. Our code is available at anonymous GitHub `https://anonymous.4open.science/r/ICLR-2027/`.

1 INTRODUCTION

Rubric-based reward modeling uses explicit criteria and scoring rules to define how language model responses should be evaluated, for example along dimensions such as correctness, helpfulness, and coherence (Gunjal et al., 2026). To turn these criteria into a training signal, a judge model assigns a score to each response on each selected dimension. The resulting scores are then combined into a scalar reward for reinforcement learning (Wang et al., 2024b;a).

Existing work improves rubric rewards by refining evaluation criteria or changing how their scores are aggregated. Specifically, methods construct reusable rubric pools from preference data, with criteria reused across queries or selected for each query (Xie et al., 2025a; Sun et al., 2026), while aggregation methods reweight criterion scores (Wang et al., 2024b;a) or apply rubric dropout to regularize training (Yang et al., 2026). Although these approaches improve reward construction, the resulting rewards still depend on the accuracy of the judge's scores.

However, better criteria and aggregation do not ensure that judge scores agree with human judgments (Thakur et al., 2025). In Figure 1, for example, the judge scores a response highly on all three rubrics, while humans score it low. Human ratings provide a reference for measuring and correcting these errors, and prior work has used them to calibrate rubric-based evaluations and judge scores (Hashemi et al., 2024; Sheng et al., 2025). However, these studies focus on response evaluation and do not examine how calibration affects downstream policy learning. During policy training, the policy continually generates new responses, while existing human ratings cover only previously collected examples. Obtaining new ratings throughout training would be costly. We therefore consider using a fixed human-rated dataset to calibrate rewards for policy training without additional annotations. The key challenge is estimating how humans would score new responses using only the available annotations, as downstream queries and responses can differ from those in the annotated dataset.

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

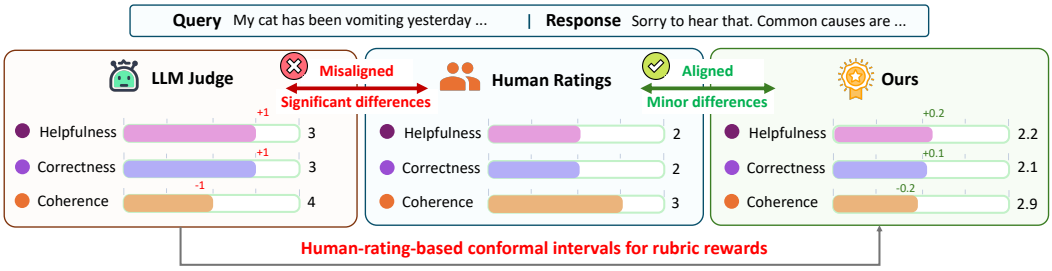


Figure 1: A motivating example: judge bias in rubric rewards, where the scores of an LLM judge deviate substantially from the human ratings and therefore need to be calibrated.

To this end, we propose Conformal Rubrics, which use a fixed human-rated dataset to calibrate rubric rewards for downstream policy training. We first train a predictor on human scores for each rubric covered by the annotations. We then use its prediction errors to construct a conformal interval of possible human scores around each prediction. To account for differences across queries and responses, we use quantile regression to predict the lower and upper error bounds from each input, allowing the interval to vary with the response being evaluated. We then calibrate the judge's score based on this interval: scores outside it are clipped to the nearest boundary, while scores inside are retained. The corrected scores are combined into rewards using existing rubric aggregation methods.

Our contributions are summarized as follows:

- We study how to use human ratings from a fixed source dataset to calibrate rubric scores on new policy responses from other datasets, without additional annotations.
- We propose Conformal Rubrics, which clip judge scores to human-score intervals before reward aggregation, and analyze coverage and conditions for reduced scoring error.
- Using only HelpSteer ratings, we reduce rubric scoring errors and improve policy win rates across eight benchmarks under LLM evaluation on the calibrated rubrics.

2 Related Work

2.1 Rubric-Based Rewards

Rubric-based reward modeling scores language model responses against explicit criteria to construct training rewards for open-ended tasks without automatically verifiable outcomes (Gunjal et al., 2026; Huang et al., 2025). Rubrics are either learned from preference pairs for general domains (Xie et al., 2025a; Wang & Xiong, 2025; Sun et al., 2026) or designed for specialized domains (Arora et al., 2025; Yifei et al., 2026; Zhang et al., 2025). Criterion scores are then combined into a scalar reward (Huang et al., 2025), with some methods adapting the aggregation weights online during policy training (Huang et al., 2026; Tyagi et al., 2026). For example, Focal Reward downweights criteria whose scores have saturated (Huang et al., 2026). Further work recovers learning signals suppressed by scalar aggregation (Xia et al., 2026; Yang et al., 2026) or uses rubrics as token-level guidance for distillation (Bablani et al., 2026; Gu et al., 2026). However, these approaches still rely on judge scores that may differ from human ratings, resulting in scoring errors entering training rewards. Using fixed human annotations, we construct an interval of possible human scores for each covered criterion and clip judge scores to it before reward aggregation.

2.2 Reliability of LLM-as-a-Judge

LLM judges can exhibit biases that cause their scores to deviate from human judgments (Chehbouni et al., 2025; Bavaresco et al., 2025; Zheng et al., 2023; Thakur et al., 2025). To assess score reliability, prior work uses rating token probabilities (Wagner et al., 2024; Xie et al., 2025b), self-reported confidence (Yona et al., 2024; Xiong et al., 2024), or consistency across generations (Manakul et al., 2023; Chen & Mueller, 2024). Conformal prediction (Angelopoulos & Bates, 2023) has also been studied for LLMs (Ye et al., 2024; Campos et al., 2024), including multiple-choice question answer-

ing (Kumar et al., 2023; Su et al., 2024) and response selection for factual consistency (Quach et al., 2024; Mohri & Hashimoto, 2024). Human ratings support direct calibration: LLM-Rubric learns to predict human assessments from rubric evaluations (Hashemi et al., 2024), while Sheng et al. (2025) construct intervals around judge scores and use their midpoints as corrected scores. However, these calibration studies focus on response evaluation and do not examine downstream policy learning. We use fixed human annotations to calibrate rewards for newly generated policy responses without additional ratings.

3 PRELIMINARIES

3.1 RUBRIC REWARDS AND AVAILABLE HUMAN SUPERVISION

Let x denote a query and $y \sim \pi_\theta(\cdot \mid x)$ a response generated by the policy π_θ. A rubric specifies the criteria used to evaluate this response and the scoring rule for each criterion. We consider a finite set of reusable criteria $\mathcal{K} = \{1, \ldots, K\}$, such as helpfulness, correctness, and clarity. For criterion k, a judge model J assigns a score $s_{J,k}(x, y)$. To use these scores for policy learning, a rubric-based method combines them into a scalar reward:

$$R_J(x, y) = \sum_{k=1}^{K} a_k(x, y) s_{J,k}(x, y), \qquad a_k(x, y) \geq 0, \quad \sum_{k=1}^{K} a_k(x, y) = 1, \tag{1}$$

where $a_k(x, y)$ is the aggregation weight for criterion k. A method may use all criteria or select a subset from this pool, assigning zero weight to unselected criteria. Because R_J is constructed from the judge scores, these scores may differ from human assessments and can affect the accuracy of training reward. However, collecting ratings for every new training dataset and every response generated during policy learning would require repeated annotation effort, which is costly and impractical. We therefore use a fixed human-rated dataset $\mathcal{D}_H$, where $s_{H,k}(x, y)$ denotes the human score on criterion k. The annotations cover the criteria to be calibrated, and their definitions and rating scales remain the same across datasets. Our goal is to use these annotations to calibrate judge scores for policy training on a downstream query dataset $\mathcal{D}_T$, without collecting human ratings for the newly generated responses.

3.2 CONFORMAL PREDICTION

For a query–response pair $z = (x, y)$, conformal prediction (CP) aims to construct an interval containing the human score $s_{H,k}(z)$ with a pre-specified probability (Angelopoulos & Bates, 2023). We first consider the i.i.d. setting: the annotated examples $(z_i, s_{H,k}(z_i))$ and a new example $(z_*, s_{H,k}(z_*))$ are sampled independently from the same distribution P with $s_{H,k}(z_*)$ unobserved. Given a human-score predictor f_k, we want to bound the new example's prediction error $|s_{H,k}(z_*) - f_k(z_*)|$, and then we can further induce a bound on the human score $s_{H,k}(z_*)$ based on the prediction error bound.

To obtain the new example's prediction error bound, we split $\mathcal{D}_H$ into a training set $\mathcal{D}_H^{\text{tr}}$ and calibration set $\mathcal{D}_H^{\text{cal}}$. We fit f_k on $\mathcal{D}_H^{\text{tr}}$, then compute the residual for each example in $\mathcal{D}_H^{\text{cal}}$:

$$e_{ik} = s_{H,k}(z_i) - f_k(z_i). \tag{2}$$

We keep f_k fixed after fitting. Because $\mathcal{D}_H^{\text{cal}}$ is not used to fit f_k, the $n_{\text{cal}} = |\mathcal{D}_H^{\text{cal}}|$ calibration errors $\{|e_{ik}|\}_{i=1}^{n_{\text{cal}}}$ and the new error $|s_{H,k}(z_*) - f_k(z_*)|$ are i.i.d. conditional on the fitted predictor. The new error is therefore equally likely to occupy any rank among the $n_{\text{cal}} + 1$ errors, with random tie breaking when needed. Therefore, for a desired error rate $\alpha \in (0, 1)$, we choose $\hat{q}_k^{1-\alpha}$ as the $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$-th smallest absolute calibration residual, where the extra one accounts for the new example. For example, if there are 10 samples in $\mathcal{D}_H^{\text{cal}}$ with $\alpha = 0.2$ and absolute calibration residuals $[0.1, 0.14, 0.2, 0.24, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6]$, we select the $\lceil 11 \times 0.8 \rceil = 9$-th smallest value, giving $\hat{q}_k^{0.8} = 0.5$. This gives the following conformal interval:

$$C_k(z_*) = \left[f_k(z_*) - \hat{q}_k^{1-\alpha},\ f_k(z_*) + \hat{q}_k^{1-\alpha} \right]. \tag{3}$$

This is a fixed-width (FW) interval because all inputs share the same calibration quantile with width $2\hat{q}_k^{1-\alpha}$, while its center varies with the input. Under the i.i.d. assumption, this interval satisfies:

$$\Pr\{s_{H,k}(z_*) \in C_k(z_*)\} \geq 1 - \alpha. \tag{4}$$

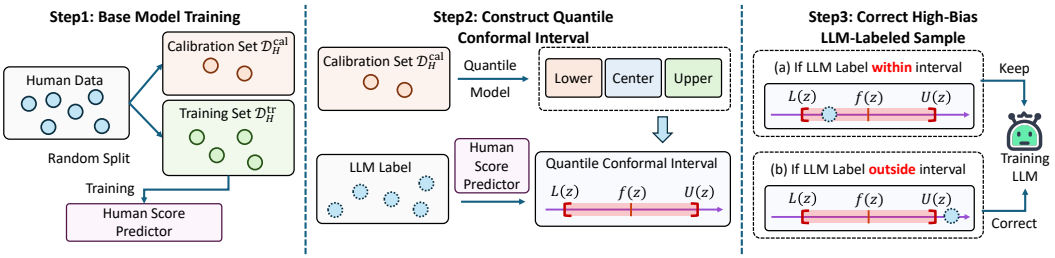


Figure 2: Overview of the proposed ConformalRubrics method. A human-score predictor and conformal intervals are learned from the human-rated dataset, and during policy training, judge scores outside their intervals are clipped to the nearest boundary before reward aggregation.

4 METHOD

4.1 MOTIVATION

Discrepancies between judge scores and human scores can occur in reward construction, and even strong LLM judges have been found to disagree with humans. We first conduct a case study to illustrate this issue. We evaluate the average MAE of judge scores against human ratings across five rubrics on HelpSteer2 under three different judges in Table 1. These results show that DeepSeek-V4-Flash has an average MAE of 0.8157, which decreases to 0.5901 with the proposed conformal calibration method that uses fixed human annotations in HelpSteer. The same trend holds for the other two judges. This motivates two questions: (1) how can fixed human annotations, such as those in HelpSteer, be used to estimate human-score intervals for new queries and responses in HelpSteer2, WildChat, and AlpacaEval-2, and (2) how can these intervals help correct judge scores used as training rewards in policy learning? Figure 2 presents our framework for addressing these questions.

Table 1: Average MAE across five rubrics on HelpSteer2.

Judge	Raw	Ours (QR)
DeepSeek-V4-Flash	0.8157	**0.5901**
Gemini 3.1 Flash-Lite	0.6892	**0.5867**
ArmoRM	0.7422	**0.5829**

4.2 LEARNING A REFERENCE FOR HUMAN SCORES

Predicting human scores for new responses. Computing residuals from an existing LLM uses the human annotations to measure its errors, but does not fit its score predictions to those annotations. To incorporate the human preferences expressed in the ratings, we fit an LLM-based predictor f_k on $\mathcal{D}_H^{\text{tr}}$ for each rubric. This predictor is separate from the judge J whose scores will be calibrated. For clarity, we describe score prediction and interval construction for one rubric and omit its index k. We fit f by minimizing the squared difference between its predictions and the human scores:

$$\mathcal{L}_{\text{score}}(f) = \frac{1}{|\mathcal{D}_H^{\text{tr}}|} \sum_{z_i \in \mathcal{D}_H^{\text{tr}}} \left(f(z_i) - s_H(z_i)\right)^2 . \tag{5}$$

After fitting, we keep f fixed and compute residuals $e_i = s_H(z_i) - f(z_i)$ on the held-out set $\mathcal{D}_H^{\text{cal}}$ to construct the intervals.

4.3 CONSTRUCTING INTERVALS FOR NEW RESPONSES

Since the predictor can also make errors on new responses, its point prediction alone may not provide a reliable replacement for the judge's score. We therefore consider a more robust way, i.e., using an interval to correct the judge's score. Writing $\hat{q}$ for the threshold $\hat{q}_k^{1-\alpha}$ defined in Section 3.2, the fixed-width (FW) interval construction uses the absolute residuals on $\mathcal{D}_H^{\text{cal}}$ to form $[f(z) - \hat{q},\ f(z) + \hat{q}]$. Although its center varies with z, its width remains $2\hat{q}$ for every input. However, the prediction errors of f may differ across query–response pairs, making this shared bound too wide for some responses and too narrow for others.

To account for this variation, we use quantile regression (QR) to estimate residual bounds conditional on each query–response pair, allowing the interval to reflect variation in the human-score predictor's errors across inputs. Specifically, for a total error rate α, we learn $q_{\alpha/2}(z)$ and $q_{1-\alpha/2}(z)$ to estimate the lower and upper residual quantiles, respectively. We train the lower-quantile model on $\mathcal{D}_H^{\text{cal}}$ by minimizing the quantile loss (Romano et al., 2019):

$$\mathcal{L}_{\alpha/2}(q_{\alpha/2}) = \frac{1}{|\mathcal{D}_H^{\text{cal}}|} \sum_{z_i \in \mathcal{D}_H^{\text{cal}}} \rho_{\alpha/2}\big(e_i - q_{\alpha/2}(z_i)\big)\,, \tag{6}$$

where $u = e_i - q_{\alpha/2}(z_i)$ and $\rho_{\alpha/2}(u) = \max\{\alpha u/2, (\alpha/2 - 1)u\}$. For example, when $\alpha/2 = 0.1$, overestimating a residual is penalized nine times as heavily as underestimating it by the same amount, encouraging the model to estimate the 0.1 quantile. We train the upper-quantile model with the same objective, replacing the quantile level $\alpha/2$ with $1 - \alpha/2$.

We then combine the QR and FW intervals by mixing their corresponding endpoints. The QR bounds vary with the query and response, while the FW bound provides an overall error scale estimated from the calibration set. Combining them allows the final interval to account for both the overall error scale and differences in prediction errors across inputs. Finally, we obtain $C(z) = [L(z), U(z)]$:

$$\begin{aligned} L(z) &= f(z) - (1-\lambda)\hat{q} + \lambda q_{\alpha/2}(z), \\ U(z) &= f(z) + (1-\lambda)\hat{q} + \lambda q_{1-\alpha/2}(z), \end{aligned} \tag{7}$$

where $\lambda \in [0, 1]$ controls the balance between FW and QR. For comprehensive discussion, if the upper bound is smaller than the lower bound, we ignore this sample, although this case does not exist in our experiment.

4.4 Using the Intervals to Calibrate Policy Rewards

Constructing calibrated rewards. For a downstream query–response pair $z = (x, y)$, we construct $C_k(z) = [L_k(z), U_k(z)]$ for each rubric k using the procedure above. We then compare the judge's score $s_{J,k}(z)$ with this interval: a score inside the interval is retained, while a score outside it is moved to the nearest boundary:

$$\widetilde{s}_{J,k}(z) = \begin{cases} L_k(z), & s_{J,k}(z) < L_k(z), \\ s_{J,k}(z), & L_k(z) \le s_{J,k}(z) \le U_k(z), \\ U_k(z), & s_{J,k}(z) > U_k(z). \end{cases} \tag{8}$$

If the human score lies within the interval, clipping cannot increase the absolute error of the judge's score. We then aggregate the corrected scores using the original rubric weights $a_k(z)$ from Equation (1) to obtain the calibrated reward:

$$\widetilde{R}_J(z) = \sum_{k=1}^{K} a_k(z)\widetilde{s}_{J,k}(z). \tag{9}$$

The intervals depend on the human annotations and the query–response pair, but not on the judge's scores or the rubric aggregation weights. The same fitted predictors can therefore be reused across judges and across different selections and weighted combinations of criteria from the annotated rubric pool, provided the criterion definitions and rating scales remain unchanged.

Training the policy with calibrated rewards. As the policy is updated, it generates new responses that must be scored for the next update. For every new batch of rollouts, we construct the corresponding interval for each rubric and clip its score using Equation (8), then aggregate the corrected scores into a reward using Equation (9). This procedure is repeated throughout policy training. The judge, human-score predictor, and quantile models remain fixed, so each new response can be calibrated using the original human annotations without collecting additional ratings.

Using GRPO (Shao et al., 2024) as an example, for each query $x \in \mathcal{D}_T$ we sample G responses $y_j \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ from the rollout policy and assign calibrated rewards $\widetilde{r}_j = \widetilde{R}_J(x, y_j)$. The calibrated advantages are

$$\widetilde{A}_j = \frac{\widetilde{r}_j - \bar{\widetilde{r}}}{\sigma_{\widetilde{r}} + \epsilon}, \tag{10}$$

where $\bar{\widetilde{r}}$ and $\sigma_{\widetilde{r}}$ are the group mean and standard deviation of the calibrated rewards, and $\epsilon > 0$ is a small constant. We update the policy by maximizing the following objective for the sampled group:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \frac{1}{G}\sum_{j=1}^{G}\frac{1}{|y_j|}\sum_{t=1}^{|y_j|}\Big[\min\big\{w_{j,t}(\theta)\widetilde{A}_j,\ \mathrm{clip}(w_{j,t}(\theta), 1-\delta, 1+\delta)\,\widetilde{A}_j\big\} - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\Big], \tag{11}$$

where $w_{j,t}(\theta) = \pi_\theta(y_{j,t} \mid x, y_{j,<t})/\pi_{\theta_{\mathrm{old}}}(y_{j,t} \mid x, y_{j,<t})$ is the probability ratio, δ is the clipping threshold, β is the KL coefficient, and π_{ref} is the fixed reference policy.

4.5 Theoretical Analysis

We analyze how distribution shift and interval endpoint errors affect coverage. For one rubric, condition on the fitted models and a fixed downstream policy. Let P_H and P_T denote the source and downstream joint distributions of a query–response pair and its human score. For residual $e = s_H(z) - f(z)$, define $F_H(t \mid z) = \Pr_H(e \le t \mid z)$ and F_T analogously. We assume that the source distribution covers the downstream input support. We measure conditional distribution shift by comparing the CDFs at the same threshold:

$$\Delta(z) = \sup_t |F_T(t \mid z) - F_H(t \mid z)|\,. \tag{12}$$

Thus, $\Delta(z) = 0$ when the source and downstream conditional residual distributions agree.

Let $q^*_\ell(z) = q^*_{\alpha/2}(z)$ and $q^*_u(z) = q^*_{1-\alpha/2}(z)$ be the true source conditional residual quantiles. Write $C_\lambda(z) = [L_\lambda(z), U_\lambda(z)]$ for the interval in Equation (7), with residual endpoints $\ell_\lambda(z) = L_\lambda(z) - f(z)$ and $u_\lambda(z) = U_\lambda(z) - f(z)$. Define

$$\eta_\lambda(z) = \Pr_H(q^*_\ell(z) \le e < \ell_\lambda(z) \mid z) + \Pr_H(u_\lambda(z) < e \le q^*_u(z) \mid z)\,. \tag{13}$$

The two terms count source probability mass excluded by a lower endpoint above $q^*_\ell(z)$ or an upper endpoint below $q^*_u(z)$. Each term is zero when the corresponding endpoint extends outward from the true quantile.

Theorem 1 (Coverage of QR and mixed intervals)**.** *Under the setting above, for any fixed $\lambda \in [0,1]$ and finite ordered interval $C_\lambda(z)$, the conditional coverage satisfies*

$$\Pr_T\{s_H(z) \in C_\lambda(z) \mid z\} \ge 1 - \alpha - \eta_\lambda(z) - 2\Delta(z). \tag{14}$$

This includes QR at $\lambda = 1$, FW at $\lambda = 0$, and their mixtures, and allows discrete human scores.

The bound separates distribution shift from coverage loss caused by inward endpoint errors. Small $\Delta(z)$ and $\eta_\lambda(z)$ yield coverage close to $1 - \alpha$; source-only training does not guarantee that these quantities are small on downstream inputs. The proof is in Appendix A.

For a fixed judge score, $\alpha + \eta_\lambda(z) + 2\Delta(z) < 1/2$ suffices for clipping to reduce expected absolute error whenever the score is moved (Appendix A.2). The following theorem gives the exact condition. Write $[v]_+ = \max\{v, 0\}$ and let $L(z)$ and $U(z)$ denote the final interval endpoints.

Theorem 2 (Decrease in absolute score error)**.** *For an input z and a fixed judge score $s_J(z) > U(z)$, let $d = s_J(z) - U(z) > 0$. Clipping to $U(z)$ reduces expected absolute error if and only if*

$$\mathbb{E}_T[\min\{(s_H(z) - U(z))_+, d\} \mid z] < \frac{d}{2}. \tag{15}$$

For $s_J(z) < L(z)$, replace $(s_H(z) - U(z))_+$ with $(L(z) - s_H(z))_+$ and set $d = L(z) - s_J(z)$.

The condition accounts for both how often human scores exceed the clipping boundary and how far they exceed it. The truncation at d reflects that clipping can decrease absolute error by at most the distance moved. The proof is provided in Appendix A.2.

Table 2: Performance comparison with different judges on HelpSteer2 across 5 rubrics. We report the MSE and MAE on each rubric and their average over the 5 rubrics. The best results are **bolded**.

	Helpfulness		Correctness		Coherence		Complexity		Verbosity		Avg	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DeepSeek-V4-Flash												
Raw	1.9146	0.9816	2.1309	0.9961	0.6111	0.3899	1.3996	0.9030	1.1843	0.8080	1.4481	0.8157
FW	1.2555	0.8438	1.1880	0.8080	0.4409	**0.3552**	0.6552	0.6076	0.5874	0.5738	0.8254	0.6377
QR	**1.1315**	**0.7977**	**1.0586**	**0.7671**	**0.4035**	0.3757	**0.4772**	**0.5129**	**0.4436**	**0.4973**	**0.7029**	**0.5901**
Gemini 3.1 Flash-Lite												
Raw	1.9355	0.9470	2.1532	0.9894	0.6599	0.4326	0.5992	0.5087	0.6859	0.5684	1.2067	0.6892
FW	1.3001	0.8327	1.2445	0.8084	0.4872	**0.3901**	0.5042	0.4721	0.5644	0.5292	0.8201	0.6065
QR	**1.1829**	**0.8048**	**1.1009**	**0.7731**	**0.4510**	0.4068	**0.4092**	**0.4371**	**0.4856**	**0.5119**	**0.7259**	**0.5867**
ArmoRM												
Raw	1.5615	0.9074	1.7281	0.9199	0.5026	0.3848	0.9398	0.7514	0.9606	0.7475	1.1385	0.7422
FW	1.1694	0.8259	1.1140	0.7942	0.3996	**0.3630**	0.5647	0.5740	0.5302	0.5644	0.7556	0.6243
QR	**1.0823**	**0.7864**	**1.0188**	**0.7610**	**0.3819**	0.3800	**0.4323**	**0.4979**	**0.4077**	**0.4891**	**0.6646**	**0.5829**

5 EXPERIMENTS

5.1 REWARD EVALUATION

Datasets and Baselines. The predictor f are trained and calibrated on HelpSteer (Wang et al., 2024c), a dataset annotated by humans with 5 rubrics. Alignment quality is measured on the HelpSteer2 (Wang et al., 2024b) test split. The raw judge scores (Raw) to be corrected are produced by two LLM judges (DeepSeek-V4-Flash (Xu et al., 2026) and Gemini 3.1 Flash-Lite) and a reward model (ArmoRM (Wang et al., 2024a)). We clip the raw scores to the FW interval and the QR interval. Appendix C.2 reports the reverse direction (trained on HelpSteer2, evaluated on HelpSteer) and the same-domain setting (trained and evaluated on HelpSteer). Our calibration remains effective and yields lower errors in the same-domain setting, where the distribution shift is smaller.

Performance Comparison. Table 2 reports the alignment quality on the HelpSteer2 test split. Both LLM judges and the reward model exhibit discrepancies between their scores and the human ratings, e.g., the average MSE of Raw reaches 1.45 under DeepSeek-V4-Flash. Note that both FW and QR substantially reduce the discrepancy of Raw under all three scorers, with QR reducing the average MSE by 40%–52% and FW by 32%–43%. These results validate the effectiveness of our calibration and its judge-agnostic property.

Table 3: Bias reduction statistics on HelpSteer2 averaged over 5 rubrics.

Method	Prob. of Bias Decrease ↑	Avg. Bias Decrease ↑	Avg. Bias Increase ↓	Expected Bias Reduction ↑
DeepSeek-V4-Flash				
FW	0.7964	0.3076	0.1891	0.1780
QR	0.7290	0.3888	0.2096	0.2256
Gemini 3.1 Flash-Lite				
FW	**0.6771**	0.2760	0.1830	0.0827
QR	0.5637	0.3147	0.1659	0.1025
ArmoRM				
FW	0.7663	0.2437	0.1901	0.1423
QR	0.6813	0.3294	0.1969	0.1616

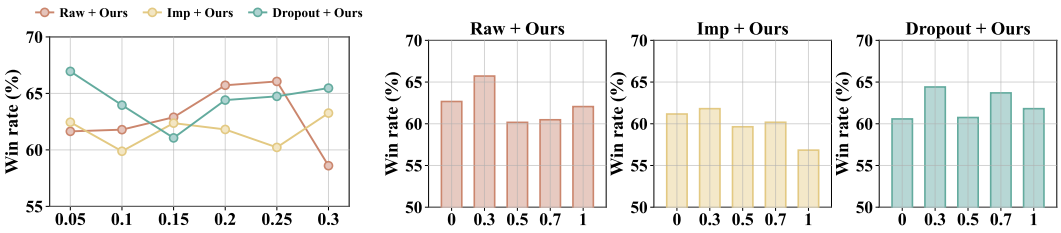
In-depth Analysis of Bias Reduction. Table 3 reports the bias reduction behavior of our calibration. FW maintains the bias decrease probability above 0.67 for all scorers, and QR attains the largest expected bias reduction. For every calibration method, the average magnitude of bias decrease is larger than that of the extra bias increase, yielding a positive net bias reduction.

5.2 DOWNSTREAM REINFORCEMENT LEARNING

We evaluate downstream reinforcement learning on general-domain and expert-domain benchmarks, where each expert-domain dataset carries domain-specific original rubrics. The general-domain tasks are evaluated by the pairwise win rate on the ***trained rubric***, which are also used for training, comparing the baselines trained with and without our calibration. The expert-domain tasks additionally report the score of the ***original rubric***, produced by the domain-specific rubric of each benchmark. The datasets and the training setup are described in Appendix C.1.

Table 4: Downstream reinforcement learning in the general domain. We report the pairwise win rate under the ***trained rubrics***. Average is taken over the three rubrics. The best results are **bolded**.

	HS2-test				WildChat				AlpacaEval-2			
Method	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**
Qwen2.5-7B-Instruct												
Raw Judge	38.68	39.79	24.37	34.28	37.71	37.16	29.00	34.62	39.25	35.73	25.71	33.57
Raw + Ours	**61.32**	**60.21**	**75.63**	**65.72**	**62.29**	**62.84**	**71.00**	**65.38**	**60.74**	**64.27**	**74.28**	**66.43**
Importance-Based	37.75	38.78	38.05	38.19	40.41	39.51	37.33	39.08	41.22	39.45	36.25	38.97
Imp + Ours	**62.25**	**61.22**	**61.95**	**61.81**	**59.59**	**60.49**	**62.67**	**60.92**	**58.78**	**60.55**	**63.74**	**61.03**
Dropout	33.03	36.04	37.71	35.59	33.77	40.66	**50.45**	41.62	29.98	39.42	35.27	34.89
Dropout + Ours	**66.97**	**63.96**	**62.29**	**64.41**	**66.23**	**59.34**	49.55	**58.38**	**70.02**	**60.58**	**64.73**	**65.11**
Llama-3.1-8B-Instruct												
Raw Judge	38.41	41.03	27.13	35.52	40.09	41.95	33.55	38.53	41.68	42.24	31.80	38.58
Raw + Ours	**61.59**	**58.97**	**72.87**	**64.48**	**59.91**	**58.05**	**66.45**	**61.47**	**58.32**	**57.76**	**68.20**	**61.42**
Importance-Based	40.48	43.40	35.87	39.92	42.90	46.12	39.67	42.90	42.41	44.47	41.15	42.67
Imp + Ours	**59.52**	**56.60**	**64.13**	**60.08**	**57.10**	**53.88**	**60.33**	**57.10**	**57.59**	**55.53**	**58.85**	**57.33**
Dropout	33.99	44.12	38.28	38.80	32.95	44.66	37.81	38.47	32.84	45.20	36.16	38.07
Dropout + Ours	**66.01**	**55.88**	**61.72**	**61.20**	**67.06**	**55.34**	**62.19**	**61.53**	**67.16**	**54.80**	**63.84**	**61.93**
Qwen3-4B												
Raw Judge	36.77	38.59	33.63	36.33	34.94	37.64	33.02	35.20	39.16	37.41	34.68	37.09
Raw + Ours	**63.23**	**61.41**	**66.36**	**63.67**	**65.06**	**62.36**	**66.98**	**64.80**	**60.84**	**62.58**	**65.32**	**62.91**
Importance-Based	38.93	41.48	39.14	39.85	39.16	42.76	39.27	40.39	42.77	43.33	41.73	42.61
Imp + Ours	**61.07**	**58.52**	**60.86**	**60.15**	**60.84**	**57.24**	**60.73**	**59.61**	**57.23**	**56.67**	**58.27**	**57.39**
Dropout	32.88	43.41	**50.61**	42.30	28.26	41.61	39.96	36.61	27.63	41.22	39.39	36.08
Dropout + Ours	**67.12**	**56.59**	49.39	**57.70**	**71.74**	**58.39**	**60.04**	**63.39**	**72.37**	**58.78**	**60.61**	**63.92**



(a) α sensitivity (b) λ sensitivity

Figure 3: Sensitivity analysis on HelpSteer2 with Qwen2.5-7B-Instruct.

Baselines. (i) **Raw** aggregates the per-rubric scores by a uniform average (Huang et al., 2025). (ii) **Importance-Based** (Gunjal et al., 2026) aggregates by a weighted average, where the judge additionally rates the semantic importance of each rubric for the given query–response pair. (iii) **Dropout** (Yang et al., 2026) aggregates uniformly over a randomly dropped subset of the rubrics at each training step. In our three-rubric setting, one rubric is dropped per step.

Performance on General Domain. Table 4 reports the downstream reinforcement learning performance on the general domain. The three baselines construct rewards from the judge scores, yet none of them ensures that the judge model assigns scores consistent with human judgments. Our calibration yields consistent advantages, with average win rates of 61%–66% for Raw and 57%–62% for Importance-Based. For Dropout, the average win rate reaches 58%–65%, and the calibration remains effective regardless of which rubrics are dropped, since it corrects each rubric independently. These results show that our calibration makes the rubric rewards more effective for reinforcement learning and can be readily adopted by existing rubric-based methods.

Sensitivity to the Significance Level α. We examine the sensitivity to the significance level α over $\alpha \in [0.05, 0.3]$ (Figure 3a). The variation differs across the three methods, with Importance-Based the most stable, yet the calibration improves every method across the whole range. A smaller α yields a wider interval with weaker clipping that may leave part of the residual judge bias uncorrected, while a larger α yields a narrower interval with stronger clipping that may suppress part of the useful reward signal.

Sensitivity to the Interval-Mixing Weight λ. We evaluate the interval-mixing weight λ between the FW (λ=0) and QR (λ=1) intervals (Figure 3b). The performance of all three methods peaks

Table 5: Downstream reinforcement learning in the expert domain. We report the pairwise win rate under the ***trained rubrics*** as in Table 4, and the ***original rubric*** score produced by the dataset-specific rubric. Average is taken over the 5 benchmarks. The best results are **bolded**.

	Medicine						Science				Avg	
	RaR-Med		HealthBench		LLMEval-Med		RaR-Sci		ResearchQA			
Method	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**
Qwen2.5-7B-Instruct	42.12	–	19.67	–	51.32	–	46.31	–	50.75	–	–	–
Raw Judge	41.14	42.53	20.53	31.87	**52.95**	39.32	**47.17**	44.80	51.15	39.27	42.59	39.56
Raw + Ours	**42.74**	**57.47**	**22.07**	**68.13**	52.76	**60.68**	47.17	**55.20**	**51.19**	**60.73**	**43.19**	**60.44**
Importance-Based	**47.17**	44.88	**29.75**	39.35	55.28	43.50	50.10	48.54	56.48	42.31	**47.75**	43.72
Imp + Ours	45.66	**55.12**	28.02	**60.65**	**55.62**	**56.50**	**50.37**	**51.46**	**58.36**	**57.69**	47.60	**56.28**
Dropout	42.37	27.13	22.60	27.00	54.67	31.36	48.92	48.20	55.77	44.40	44.86	35.62
Dropout + Ours	**44.71**	**72.87**	**27.55**	**73.00**	**55.58**	**68.64**	**49.17**	**51.80**	**55.94**	**55.60**	**46.59**	**64.38**
Llama-3.1-8B-Instruct	42.60	–	15.86	–	45.62	–	38.12	–	48.92	–	–	–
Raw Judge	**47.78**	39.67	**19.33**	43.08	**46.38**	48.42	**41.20**	43.06	**56.89**	41.53	**42.31**	43.16
Raw + Ours	46.89	**60.33**	18.52	**56.92**	46.13	**51.58**	40.30	**56.94**	53.80	**58.47**	41.12	**56.84**
Importance-Based	48.82	39.55	21.55	40.62	48.39	43.65	42.66	43.67	58.19	42.61	43.92	42.02
Imp + Ours	**50.05**	**60.45**	**22.58**	**59.38**	**49.07**	**56.35**	**43.53**	**56.33**	**58.24**	**57.39**	**44.70**	**57.98**
Dropout	47.50	44.53	24.40	39.87	**50.21**	45.81	**42.38**	40.66	53.30	21.87	43.55	38.55
Dropout + Ours	**50.23**	**55.47**	**25.20**	**60.13**	49.86	**54.19**	41.48	**59.34**	**56.28**	**78.13**	**44.61**	**61.45**
Qwen3-4B	44.70	–	22.55	–	54.06	–	56.05	–	58.52	–	–	–
Raw Judge	**47.52**	37.51	**23.69**	38.76	**55.42**	33.94	**57.92**	39.06	61.53	34.30	**49.22**	36.72
Raw + Ours	47.06	**62.49**	22.56	**61.24**	55.34	**66.06**	57.69	**60.94**	**61.83**	**65.69**	48.89	**63.28**
Importance-Based	49.84	44.84	31.46	45.86	58.52	46.69	**60.00**	46.22	**65.97**	48.44	**53.16**	46.41
Imp + Ours	**49.95**	**55.16**	**32.56**	**54.14**	**58.65**	**53.31**	58.06	**53.78**	64.87	**51.56**	52.82	**53.59**
Dropout	**49.17**	42.08	**32.34**	38.47	**58.02**	39.29	**58.53**	41.30	61.84	36.36	**51.98**	39.49
Dropout + Ours	48.09	**57.92**	29.80	**61.53**	57.51	**60.71**	57.95	**58.70**	**62.78**	**63.64**	51.23	**60.51**

at a moderate λ=0.3 rather than at either endpoint, suggesting that both intervals contribute to the calibration.

Robustness across Judges. We evaluate the impact of the LLM judge that provides the rubric reward during training, running the general domain experiment with GPT-4o, GPT-4o-mini, and DeepSeek-V4-Pro as the judge (Figure 4). Although the three judges come from different families and scales, the calibration improves every baseline under every judge. This suggests that our calibration applies broadly across judges, since the clipping intervals are derived from the human ratings and are independent of the judge.

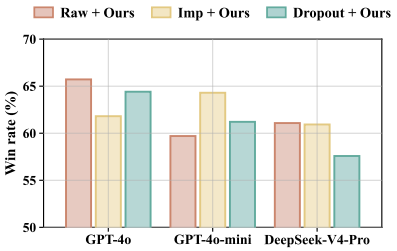


Figure 4: Judge model variation on HelpSteer2 with Qwen2.5-7B-Instruct.

Performance on Expert Domain. Table 5 reports the downstream reinforcement learning performance on the expert domain. Under the trained rubrics as in Table 4, our calibration still provides effective reward signals for training, e.g., Raw combined with our calibration attains average win rates of 56.8%–63.3% across the three policies. Under the original rubrics, however, our calibration yields only marginal and inconsistent changes, e.g., +1.6 and −0.2 under RaR-Medicine and LLMEval-Med for Raw with the Qwen2.5-7B policy, and drops on all five benchmarks for Raw with the Llama-3.1-8B policy. These changes indicate the applicability boundary of our calibration.

6 Conclusion

We presented Conformal Rubrics, which use a fixed human-rated dataset to calibrate rubric rewards for downstream policy learning without additional annotations. The method constructs human-score intervals for new queries and responses, clips judge scores outside them to the nearest boundary, and combines the corrected scores using existing rubric-based methods. Our theoretical analysis characterizes interval coverage under distribution shift and conditions for reducing expected absolute scoring error. Experiments using HelpSteer show lower rubric scoring errors on HelpSteer2 and higher average policy win rates across eight benchmarks. One potential limitation is that specialized tasks, such as medical question answering or research question answering, may benefit from human-rated data that more closely match the target domain.

AI USE STATEMENT

In this work, we used the generative AI assistant Claude Opus 5 (Anthropic) for editing and polishing parts of the manuscript text, for formatting tables and references, for drafting visualization scripts for figures of experiment, and for drafting the initial sketches of Figures 1 and 2 under our detailed specifications. We have not used generative AI tools for the problem formulation, the idea of theoretical analysis, the experimental design and implementation, or the interpretation of results. All AI-assisted text and artifacts were reviewed, verified, and revised by the authors. We take responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available datasets with human ratings collected under their original terms and introduces no new human-subject data collection. Our method aligns LLM judge scores with human ratings, which is intended to reduce the risk of judge biases during policy training. We note that the same mechanism could in principle steer rewards toward any reference annotations, so the choice of the human-rated source dataset determines whose values the policy learns, and this choice should be made transparently. The veterinary example in Figure 1 is illustrative only and is not medical advice.

REPRODUCIBILITY STATEMENT

All experiments are built on public datasets and open-weight models: the human-rated datasets (HelpSteer and HelpSteer2), the evaluation benchmarks (HelpSteer2-test, WildChat, AlpacaEval-2, HealthBench, LLMEval-Med, and ResearchQA), the policy models (Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Qwen3-4B), the reward backbone (FsfairX-LLaMA3-RM-v0.1), and the GRPO trainer. The implementation details of the predictor heads and the conformal intervals are described in Section C.1, additional results in Appendix C.2, the theoretical analysis with its assumptions in Section 4, and the complete prompt templates for the judge and evaluation protocols in Appendix D. We will release the code and trained heads upon publication.

REFERENCES

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

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

Deepika Bablani, Ajay Gupta, and Wanming Chen. Rubrics as privileged information for open-ended generation. *arXiv preprint arXiv:2608.02948*, 2026.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, et al. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 238–255, 2025.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário AT Figueiredo, and André FT Martins. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 12:1497–1516, 2024.

Khaoula Chehbouni, Mohammed Haddou, Jackie CK Cheung, and Golnoosh Farnadi. Neither valid nor reliable? Investigating the use of LLMs as judges. In *Advances in Neural Information Processing Systems*, 2025.

Jiuhai Chen and Jonas Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5186–5200, 2024.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

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

Siyi Gu, Jialin Chen, Sophia Zhou, Arman Cohan, and Rex Ying. Rethinking reward supervision: Rubric-conditioned self-distillation. *arXiv preprint arXiv:2606.19327*, 2026.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 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)*, pp. 13806–13834, 2024.

Yu Huang, Zihua Zhao, Zhaoxin Huan, Wanli Gu, Feng Hong, Xinmu Ge, Lin Yuan, Weichang Wu, Qiang Hu, Xiaolu Zhang, et al. Focal reward: Balanced reinforcement learning under rubric-based rewards. *arXiv preprint arXiv:2605.26579*, 2026.

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

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. *arXiv preprint arXiv:2402.10978*, 2024.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, volume 2024, pp. 11654–11681, 2024.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. Analyzing uncertainty of llm-as-a-judge: Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. Api is enough: Conformal prediction for large language models without logit-access. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 979–995, 2024.

Mengyuan Sun, Yu Li, Zhuohao Yu, Shikun Zhang, and Wei Ye. Support vector rubrics: Closing the gap between self-generated and human rubrics. *arXiv preprint arXiv:2606.08077*, 2026.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pp. 404–430, 2025.

Utkarsh Tyagi, Xingang Guo, MohammadHossein Rezaei, Daniel George, Anas Mahmoud, Jackson Lee, Bing Liu, and Yunzhong He. Not every rubric teaches equally: Policy-aware rubric rewards for rlvr. *arXiv preprint arXiv:2605.20164*, 2026.

Nico Wagner, Michael Desmond, Rahul Nair, Zahra Ashktorab, Elizabeth M Daly, Qian Pan, Martín Santillán Cooper, James M Johnson, and Werner Geyer. Black-box uncertainty quantification method for llm-as-a-judge. *arXiv preprint arXiv:2410.11594*, 2024.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, 2024a.

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

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501, 2024b.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3371–3384, 2024c.

Mingxuan Xia, Yuhang Yang, Chao Ye, Shuai Zhu, Shenzhi Yang, Guangcheng Zhu, Yuhang Zhang, Cheng Peng, Haobo Wang, and Siqing Wang. Enhancing rubric-based rl via self-distillation. *arXiv preprint arXiv:2607.18082*, 2026.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, et al. Auto-rubric: Learning to extract generalizable criteria for reward modeling. *arXiv e-prints*, 2025a.

Qiujie Xie, Qingqiu Li, Zhuohao Yu, Yuejie Zhang, Yue Zhang, and Linyi Yang. An empirical analysis of uncertainty in large language model evaluations. In *International Conference on Learning Representations*, volume 2025, pp. 52607–52638, 2025b.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pp. 23650–23678, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

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

Minglai Yang, Xinyu Guo, Utkarsh Tyagi, Mian Zhang, Razvan Dumitru, Sunjie Hou, Yunzhong He, Daniel Yue Zhang, and Ying Liu. Rubric dropout: A simple way to mitigate reward hacking in rubric-as-reward rl. *arXiv preprint arXiv:2608.11669*, 2026.

Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking llms via uncertainty quantification. *Advances in Neural Information Processing Systems*, 37:15356–15385, 2024.

Li S Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Transactions of the Association for Computational Linguistics*, 2026.

Gal Yona, Roee Aharoni, and Mor Geva. Can large language models faithfully express their intrinsic uncertainty in words? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7752–7764, 2024.

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

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *International Conference on Learning Representations*, volume 2024, pp. 34590–34605, 2024.

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

A Proofs of the Theoretical Results

A.1 Proof of Theorem 1

Proof. Condition on the fitted models and a fixed downstream policy. Fix z and write $e = s_H(z) - f(z)$, $\ell = \ell_\lambda(z)$, and $u = u_\lambda(z)$. Let $q_\ell^* = q_{\alpha/2}^*(z)$ and $q_u^* = q_{1-\alpha/2}^*(z)$, where $q_p^*(z) = \inf\{t : F_H(t \mid z) \geq p\}$. The quantiles and interval endpoints are finite, with $\ell \leq u$. Write $F(a^- \mid z) = \Pr(e < a \mid z)$ for the left limit. By the definition of generalized quantiles,

$$F_H((q_\ell^*)^- \mid z) \leq \alpha/2, \qquad 1 - F_H(q_u^* \mid z) \leq \alpha/2. \tag{16}$$

Hence $\Pr_H\{e \in [q_\ell^*, q_u^*] \mid z\} \geq 1 - \alpha$. A residual in this quantile interval can lie outside $[\ell, u]$ only if $q_\ell^* \leq e < \ell$ or $u < e \leq q_u^*$. Subtracting the probabilities of these events and using Equation (13) gives

$$\Pr_H\{e \in [\ell, u] \mid z\} \geq 1 - \alpha - \eta_\lambda(z). \tag{17}$$

By Equation (12),

$$|F_T(u \mid z) - F_H(u \mid z)| \leq \Delta(z), \qquad |F_T(\ell^- \mid z) - F_H(\ell^- \mid z)| \leq \Delta(z). \tag{18}$$

The second inequality follows by taking left limits in the same-threshold CDF comparison. Therefore,

$$\begin{aligned} \Pr_T\{s_H(z) \in C_\lambda(z) \mid z\} &= F_T(u \mid z) - F_T(\ell^- \mid z) \\ &\geq F_H(u \mid z) - F_H(\ell^- \mid z) - 2\Delta(z) \\ &\geq 1 - \alpha - \eta_\lambda(z) - 2\Delta(z), \end{aligned} \tag{19}$$

which proves Equation (14). Integrating over the downstream input distribution also gives

$$\Pr_T\{s_H(z) \in C_\lambda(z)\} \geq 1 - \alpha - \mathbb{E}_{z \sim P_T^z}[\eta_\lambda(z) + 2\Delta(z)]\,. \tag{20}$$

□

A.2 Proof of Theorem 2

Proof. Fix z and the judge score $j = s_J(z) > U(z)$. Let $c = U(z)$, $h = s_H(z)$, and $d = j - c > 0$, and assume the absolute-error expectations are finite. The reduction in absolute error is

$$|j - h| - |c - h| = \begin{cases} d, & h \leq c, \\ j + c - 2h, & c < h < j, \\ -d, & h \geq j. \end{cases} \tag{21}$$

These cases can be written as $d - 2\min\{(h - c)_+, d\}$. Taking the conditional expectation yields

$$\mathbb{E}_T[|j - h| - |c - h| \mid z] = d - 2\mathbb{E}_T[\min\{(h - c)_+, d\} \mid z]\,. \tag{22}$$

This quantity is strictly positive if and only if Equation (15) holds. For lower-boundary clipping, set $c = L(z)$ and $d = c - j > 0$. The gains for $h \geq c$, $j < h < c$, and $h \leq j$ are d, $2h - j - c$, and $-d$, respectively. Thus,

$$\mathbb{E}_T[|j - h| - |c - h| \mid z] = d - 2\mathbb{E}_T[\min\{(c - h)_+, d\} \mid z]\,, \tag{23}$$

which proves the lower-boundary condition.

To establish the sufficient condition stated before the theorem, let $C_\lambda(z) = [L_\lambda(z), U_\lambda(z)]$, $c = \widetilde{s}_J(z)$, and $d = |j - c|$. When $h \in C_\lambda(z)$, c lies between j and h, so $|j - h| - |c - h| = d$. For every h, the triangle inequality gives $|j - h| - |c - h| \geq -d$. Combining these two facts,

$$|j - h| - |c - h| \geq d\big(1 - 2\mathbf{1}\{h \notin C_\lambda(z)\}\big). \tag{24}$$

Theorem 1 implies

$$\Pr_T\{h \notin C_\lambda(z) \mid z\} \leq \beta_\lambda(z) := \min\{1, \alpha + \eta_\lambda(z) + 2\Delta(z)\}. \tag{25}$$

Since d is fixed conditional on z,

$$\begin{aligned} \mathbb{E}_T[|j - h| - |c - h| \mid z] &\geq d\big(1 - 2\Pr_T\{h \notin C_\lambda(z) \mid z\}\big) \\ &\geq (1 - 2\beta_\lambda(z))d. \end{aligned} \tag{26}$$

If $\alpha + \eta_\lambda(z) + 2\Delta(z) < 1/2$ and $d > 0$, the right-hand side is strictly positive. If $d = 0$, the score and its absolute error are unchanged. □

Algorithm 1 ConformalRubrics: calibrating rubric rewards with human judgments

Require: human-rated dataset $\mathcal{D}_H$ split into $\mathcal{D}_H^{\text{tr}}$ and $\mathcal{D}_H^{\text{cal}}$; downstream dataset $\mathcal{D}_T$; judge J; rubrics $k = 1, \ldots, K$ with weights a_k; error rate α; mixing weight λ
Ensure: policy π_θ trained with calibrated rewards

Offline: constructing intervals for new responses
Fit the human-score predictor f_k on $\mathcal{D}_H^{\text{tr}}$ ▷ Eq. (5)
Compute residuals $e_{ik} = s_{H,k}(z_i) - f_k(z_i)$ on $\mathcal{D}_H^{\text{cal}}$ ▷ Eq. (2)
$\hat{q} \leftarrow$ the $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$-th smallest of $\{|e_{ik}|\}$
Train quantile models $q_{\alpha/2}(z)$ and $q_{1-\alpha/2}(z)$ on $\mathcal{D}_H^{\text{cal}}$ ▷ Eq. (6)
Keep f_k, $\hat{q}$, and the quantile models fixed

Online: using the intervals to calibrate policy rewards
for each policy update **do**
  for each query $x \in \mathcal{D}_T$ **do**
    Sample responses $y_j \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$, $j = 1, \ldots, G$
    for each response y_j and each rubric k **do**
      $z \leftarrow (x, y_j)$; construct the FW and QR bounds
      $C_k^{\text{FW}}(z) = \left[f_k(z) - \hat{q},\ f_k(z) + \hat{q} \right]$ ▷ Eq. (3)
      $C_k^{\text{QR}}(z) = \left[f_k(z) + q_{\alpha/2}(z),\ f_k(z) + q_{1-\alpha/2}(z) \right]$
      Combine the QR and FW intervals by mixing their endpoints ▷ Eq. (7)
      $L_k(z) = f_k(z) - (1 - \lambda)\hat{q} + \lambda q_{\alpha/2}(z)$
      $U_k(z) = f_k(z) + (1 - \lambda)\hat{q} + \lambda q_{1-\alpha/2}(z)$
      Query the judge for $s_{J,k}(z)$
      Clip $\widetilde{s}_{J,k}(z)$ to the nearest interval boundary ▷ Eq. (8)
      Aggregate $\widetilde{r}_j = \sum_{k=1}^{K} a_k(z)\, \widetilde{s}_{J,k}(z)$ ▷ Eq. (9)
    end for
  end for
  Compute advantages $\widetilde{A}_j = (\widetilde{r}_j - \overline{\widetilde{r}})/(\sigma_{\widetilde{r}} + \epsilon)$ ▷ Eq. (10)
  Update π_θ by maximizing $\mathcal{J}_{\text{GRPO}}(\theta)$ ▷ Eq. (11)
end for

B Complete Training Procedure

Algorithm 1 summarizes the complete procedure of ConformalRubrics, consisting of the offline stage that learns the predictor and conformal intervals from the human-rated dataset and the online stage that calibrates rubric rewards during policy training.

C Additional Experimental Details

C.1 Implementation Details

Reward evaluation. We employ a frozen pretrained LLM backbone, FsfairX-LLaMA3-RM-v0.1 (Dong et al., 2024), with an MLP anchor head that regresses the human score of each rubric. The QR heads and the FW interval are constructed from the residuals on the HelpSteer calibration split with the significance level α=0.2. Alignment quality with the human labels is evaluated by the mean squared error (MSE) and the mean absolute error (MAE) on each of the five rubrics.

Datasets. In the general domain, policies are trained on the HelpSteer2 train split and evaluated on its test split, WildChat (Zhao et al., 2024), and AlpacaEval-2 (Dubois et al., 2024). In the expert domain, each dataset carries domain-specific original rubrics: policies are trained on the RaR-Medicine or RaR-Science train split (Gunjal et al., 2026) and evaluated on the corresponding test split, HealthBench (Arora et al., 2025), LLMEval-Med (Zhang et al., 2025), or ResearchQA (Yifei et al., 2026).

Table 6: Performance comparison with different judges on HelpSteer2 across 5 rubrics, including the weighted combinations. We report the MSE and MAE on each rubric and their average over the 5 rubrics. Lower is better and the best results are **bolded**.

	Helpfulness		Correctness		Coherence		Complexity		Verbosity		Avg	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DeepSeek-V4-Flash												
Raw	1.9146	0.9816	2.1309	0.9961	0.6111	0.3899	1.3996	0.9030	1.1843	0.8080	1.4481	0.8157
FW	1.2555	0.8438	1.1880	0.8080	0.4409	**0.3552**	0.6552	0.6076	0.5874	0.5738	0.8254	0.6377
QR	**1.1315**	**0.7977**	**1.0586**	**0.7671**	**0.4035**	0.3757	**0.4772**	**0.5129**	**0.4436**	**0.4973**	**0.7029**	**0.5901**
Mixed_{28}	1.2160	0.8315	1.1510	0.7979	0.4324	0.3593	0.6130	0.5862	0.5517	0.5526	0.7928	0.6255
Mixed_{55}	1.1704	0.8142	1.1076	0.7837	0.4208	0.3654	0.5536	0.5555	0.5047	0.5227	0.7514	0.6083
Mixed_{82}	1.1414	0.8018	1.0748	0.7723	0.4100	0.3715	0.5041	0.5280	0.4644	0.5015	0.7189	0.5950
Gemini 3.1 Flash-Lite												
Raw	1.9355	0.9470	2.1532	0.9894	0.6599	0.4326	0.5992	0.5087	0.6859	0.5684	1.2067	0.6892
FW	1.3001	0.8327	1.2445	0.8084	0.4872	**0.3901**	0.5042	0.4721	0.5644	0.5292	0.8201	0.6065
QR	**1.1829**	**0.8048**	**1.1009**	**0.7731**	**0.4510**	0.4068	**0.4092**	**0.4371**	**0.4856**	0.5119	**0.7259**	**0.5867**
Mixed_{28}	1.2651	0.8252	1.2051	0.7999	0.4791	0.3936	0.4846	0.4650	0.5484	0.5235	0.7965	0.6014
Mixed_{55}	1.2226	0.8150	1.1561	0.7877	0.4678	0.3986	0.4525	0.4537	0.5263	0.5157	0.7651	0.5942
Mixed_{82}	1.1934	0.8072	1.1185	0.7779	0.4574	0.4036	0.4249	0.4432	0.5017	**0.5105**	0.7392	0.5885
ArmoRM												
Raw	1.5615	0.9074	1.7281	0.9199	0.5026	0.3848	0.9398	0.7514	0.9606	0.7475	1.1385	0.7422
FW	1.1694	0.8259	1.1140	0.7942	0.3996	**0.3630**	0.5647	0.5740	0.5302	0.5644	0.7556	0.6243
QR	**1.0823**	**0.7864**	**1.0188**	**0.7610**	**0.3819**	0.3800	**0.4323**	**0.4979**	**0.4077**	**0.4891**	**0.6646**	**0.5829**
Mixed_{28}	1.1400	0.8151	1.0855	0.7855	0.3959	0.3657	0.5378	0.5589	0.4985	0.5448	0.7315	0.6140
Mixed_{55}	1.1042	0.7995	1.0508	0.7729	0.3904	0.3702	0.4946	0.5351	0.4563	0.5151	0.6992	0.5986
Mixed_{82}	1.0841	0.7882	1.0279	0.7639	0.3851	0.3759	0.4541	0.5112	0.4230	0.4935	0.6748	0.5865

Note: Mixed = weighted combination of QR and FW intervals, with subscripts 28/55/82 denoting the QR weight $w \in \{0.2, 0.5, 0.8\}$.

Downstream reinforcement learning. Both the general-domain and the expert-domain tasks share the same training setup. Policies are trained with GRPO (Shao et al., 2024) on three backbones, Qwen2.5-7B-Instruct (Yang et al., 2024), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Qwen3-4B (Yang et al., 2025), on the three rubrics of helpfulness, correctness, and coherence. Our clipping interval combines the QR interval and the FW interval with the mixing weight λ=0.3. GPT-4o serves as the judge that scores the three rubrics during training. All metrics are averaged over two evaluation judges, Gemini 3.1 Pro Preview and DeepSeek-V4-Pro (Xu et al., 2026).

C.2 Reward Evaluation

We reports more variant results of the main HelpSteer1→HelpSteer2 protocol (Table 2 and Table 3), including the weighted combinations of the residual-quantile and FW intervals (Table 6 and Table 7). Two additional protocols are further reported: the reverse transfer direction, where the heads are trained on the HelpSteer2 train split and evaluated on the HelpSteer1 test split (Table 8 and Table 9), and the same-domain control, where the heads are trained on the HelpSteer1 train split and evaluated on the HelpSteer1 test split (Table 10).

Full results on HelpSteer2. The heads are trained on the HelpSteer1 train split and evaluated on the HelpSteer2 test split. Experimental results in Table 6 and Table 7 show that Mixed interpolates between FW and QR, with the average MSE, MAE, and expected bias reduction all improving monotonically as the QR weight w grows, e.g., the average MSE under DeepSeek-V4-Flash drops from 0.7928 to 0.7189. Mixed_{28} attains the smallest average bias increase among all variants. These results confirm that our calibration effectively reduces the discrepancy between the judge scores and the human ratings.

Full results on HelpSteer1. The heads are trained on the HelpSteer2 train split and evaluated on the HelpSteer1 test split. Experimental results in Table 8 show that every conformal variant improves substantially over Raw, e.g., from 1.5768 to 0.9329 under DeepSeek-V4-Flash, and QR attains the best average MSE and MAE among the variants for all three scorers. The bias statistics in Table 9 mirror Table 3, with QR attaining the largest expected bias reduction and the average bias decrease consistently larger than the average bias increase for every variant. These results confirm the applicability of our calibration to unseen annotation pools, regardless of the transfer direction.

Table 7: Bias reduction statistics on HelpSteer2 averaged over 5 rubrics, including the weighted combinations. Each method is compared against the corresponding Raw score. ↑: higher is better; ↓: lower is better, and the best results are **bolded**.

Method	Prob. of Bias Decrease ↑	Avg. Bias Decrease ↑	Avg. Bias Increase ↓	Expected Bias Reduction ↑
DeepSeek-V4-Flash				
Raw (reference)	–	–	–	–
FW	**0.7964**	0.3076	0.1891	0.1780
QR	0.7290	**0.3888**	0.2096	**0.2256**
Mixed_{28}	0.7406	0.3132	**0.1458**	0.1902
Mixed_{55}	0.7409	0.3422	0.1685	0.2074
Mixed_{82}	0.7334	0.3724	0.1911	0.2207
Gemini 3.1 Flash-Lite				
Raw (reference)	–	–	–	–
FW	**0.6771**	0.2760	0.1830	0.0827
QR	0.5637	**0.3147**	0.1659	**0.1025**
Mixed_{28}	0.5665	0.2445	**0.1062**	0.0878
Mixed_{55}	0.5734	0.2687	0.1285	0.0950
Mixed_{82}	0.5696	0.2958	0.1500	0.1007
ArmoRM				
Raw (reference)	–	–	–	–
FW	**0.7663**	0.2437	0.1901	0.1423
QR	0.6813	**0.3294**	0.1969	**0.1616**
Mixed_{28}	0.7165	0.2562	**0.1565**	0.1392
Mixed_{55}	0.7104	0.2818	0.1719	0.1504
Mixed_{82}	0.6876	0.3130	0.1810	0.1587

Table 8: Performance comparison with different judges on HelpSteer1 across 5 rubrics, where the heads are trained on the HelpSteer2 train split. We report the MSE and MAE on each rubric and their average over the 5 rubrics. Lower is better and the best results are **bolded**.

	Helpfulness		**Correctness**		**Coherence**		**Complexity**		**Verbosity**		**Avg**	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DeepSeek-V4-Flash												
Raw	1.4766	0.9126	1.6763	0.9554	1.0175	0.6813	1.7406	1.0558	1.9729	1.1168	1.5768	0.9444
UW	1.2567	0.8520	1.3591	0.8722	0.8474	**0.6275**	0.9521	0.7624	1.0985	0.8156	1.1028	0.7860
RQ	**1.2530**	**0.8482**	**1.3377**	**0.8609**	0.8490	0.6384	**0.5461**	**0.5776**	**0.6785**	**0.6481**	**0.9329**	**0.7146**
RQUW_{28}	1.2565	0.8520	1.3540	0.8705	0.8433	0.6283	0.8492	0.7168	0.9937	0.7789	1.0593	0.7693
RQUW_{55}	1.2549	0.8510	1.3460	0.8671	**0.8403**	0.6313	0.7120	0.6538	0.8550	0.7257	1.0016	0.7458
RQUW_{82}	1.2536	0.8495	1.3400	0.8633	0.8434	0.6354	0.6013	0.6027	0.7404	0.6768	0.9558	0.7255
Gemini 3.1 Flash-Lite												
Raw	1.3639	0.8619	1.5131	0.9005	0.8882	0.6322	0.8116	0.6272	1.2633	0.8362	1.1680	0.7716
UW	1.1685	0.8040	**1.2507**	**0.8278**	0.8042	**0.6044**	0.6793	0.5821	0.9661	0.7311	0.9738	0.7099
RQ	**1.1319**	**0.7892**	1.2603	0.8318	0.8104	0.6189	**0.5061**	**0.5367**	**0.6505**	**0.6198**	**0.8718**	**0.6793**
RQUW_{28}	1.1604	0.8013	**1.2507**	0.8281	**0.8026**	0.6060	0.6447	0.5720	0.8907	0.7074	0.9498	0.7029
RQUW_{55}	1.1489	0.7969	1.2523	0.8288	**0.8026**	0.6100	0.5914	0.5564	0.7866	0.6713	0.9164	0.6927
RQUW_{82}	1.1387	0.7925	1.2560	0.8304	0.8060	0.6152	0.5364	0.5418	0.6986	0.6380	0.8871	0.6836
ArmoRM												
Raw	1.0700	0.7887	1.2074	0.8237	0.7369	0.5946	1.2489	0.9045	1.4197	0.9553	1.1366	0.8134
UW	**0.9598**	**0.7545**	1.0663	0.7806	**0.6453**	**0.5613**	0.8563	0.7317	0.9613	0.7712	0.8978	0.7199
RQ	0.9722	0.7593	**1.0601**	**0.7761**	0.7011	0.5923	**0.5311**	**0.5723**	**0.6525**	**0.6395**	**0.7834**	**0.6679**
RQUW_{28}	0.9638	0.7565	1.0679	0.7810	0.6459	0.5633	0.7804	0.6950	0.8963	0.7472	0.8709	0.7086
RQUW_{55}	0.9686	0.7586	1.0679	0.7804	0.6481	0.5678	0.6714	0.6413	0.7958	0.7068	0.8304	0.6910
RQUW_{82}	0.9713	0.7593	1.0639	0.7781	0.6623	0.5760	0.5787	0.5953	0.7037	0.6648	0.7960	0.6747

Note: Same methods and notation as Table 6.

Same-domain control on HelpSteer1. The heads are trained on the HelpSteer1 train split and evaluated on the HelpSteer1 test split. Experimental results in Table 10 show that all conformal variants substantially reduce the Raw average MSE, e.g., the average MSE is roughly halved under DeepSeek-V4-Flash, from 1.5768 to 0.6645, and the errors are consistently lower than in the cross-version protocol (Table 2), confirming that the correction is more accurate when training and evaluation share the same annotation pool. QR no longer dominates FW here (average MSE 0.6662 vs. 0.6663), indicating that the QR advantage stems from absorbing the distribution shift between the two annotation pools rather than from axis structure alone.

Table 9: Bias reduction statistics on HelpSteer1 averaged over 5 rubrics, where the heads are trained on the HelpSteer2 train split. Each method is compared against the corresponding Raw score. ↑: higher is better; ↓: lower is better, and the best results are **bolded**.

Method	Prob. of Bias Decrease ↑	Avg. Bias Decrease ↑	Avg. Bias Increase ↓	Expected Bias Reduction ↑
DeepSeek-V4-Flash				
Raw (reference)	–	–	–	–
FW	**0.8776**	0.2047	0.0838	0.1584
QR	0.8327	**0.3058**	0.1458	**0.2297**
Mixed_{28}	0.8561	0.2201	**0.0822**	0.1751
Mixed_{55}	0.8554	0.2517	0.1086	0.1986
Mixed_{82}	0.8436	0.2850	0.1317	0.2189
Gemini 3.1 Flash-Lite				
Raw (reference)	–	–	–	–
FW	**0.7946**	0.1227	0.0685	0.0617
QR	0.7221	**0.1693**	0.1050	**0.0923**
Mixed_{28}	0.7528	0.1121	**0.0537**	0.0687
Mixed_{55}	0.7509	0.1314	0.0714	0.0789
Mixed_{82}	0.7363	0.1536	0.0908	0.0880
ArmoRM				
Raw (reference)	–	–	–	–
FW	**0.8440**	0.1464	0.0773	0.1115
QR	0.7976	**0.2170**	0.1323	**0.1463**
Mixed_{28}	0.8094	0.1573	**0.0707**	0.1138
Mixed_{55}	0.8134	0.1769	0.0914	0.1268
Mixed_{82}	0.8115	0.2003	0.1174	0.1404

Table 10: Performance comparison with different judges on HelpSteer1 across 5 rubrics, where the heads are trained on the HelpSteer1 train split and evaluated on the held-out HelpSteer1 test split. We report the MSE and MAE on each rubric and their average over the 5 rubrics. Lower is better and the best results are **bolded**.

	Helpfulness		**Correctness**		**Coherence**		**Complexity**		**Verbosity**		**Avg**	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DeepSeek-V4-Flash												
Raw	1.4766	0.9126	1.6763	0.9554	1.0175	0.6813	1.7406	1.0558	1.9729	1.1168	1.5768	0.9444
FW	0.6109	0.6029	**0.6856**	0.6272	**0.6944**	**0.5813**	**0.6527**	**0.6147**	0.6881	0.6308	0.6663	0.6114
QR	**0.6035**	**0.5931**	0.6908	**0.6212**	0.7073	0.5934	0.6877	0.6252	**0.6415**	**0.6050**	0.6662	**0.6076**
Mixed_{28}	0.6083	0.6007	**0.6856**	0.6257	0.6955	0.5835	0.6586	0.6165	0.6779	0.6253	0.6652	0.6103
Mixed_{55}	0.6055	0.5975	0.6866	0.6237	0.6986	0.5871	0.6685	0.6194	0.6634	0.6173	**0.6645**	0.6090
Mixed_{82}	0.6039	0.5948	0.6888	0.6221	0.7034	0.5907	0.6796	0.6227	0.6500	0.6097	0.6651	0.6080
Gemini 3.1 Flash-Lite												
Raw	1.3639	0.8619	1.5131	0.9005	0.8882	0.6322	0.8116	0.6272	1.2633	0.8362	1.1680	0.7716
FW	**0.5978**	**0.5932**	**0.6975**	**0.6384**	0.6706	**0.5648**	**0.6065**	**0.5581**	0.6688	0.6070	**0.6482**	**0.5923**
QR	0.6239	0.6024	0.7272	0.6462	0.6746	0.5734	0.6202	0.5600	**0.6317**	**0.5891**	0.6555	0.5942
Mixed_{28}	0.6017	0.5947	0.7026	0.6398	**0.6701**	0.5663	0.6087	0.5584	0.6609	0.6032	0.6488	0.5925
Mixed_{55}	0.6087	0.5973	0.7112	0.6421	0.6707	0.5689	0.6127	0.5590	0.6494	0.5976	0.6505	0.5930
Mixed_{82}	0.6174	0.6003	0.7206	0.6445	0.6727	0.5716	0.6171	0.5595	0.6385	0.5924	0.6533	0.5936
ArmoRM												
Raw	1.0726	0.7864	1.2138	0.8225	0.7368	0.5887	1.2537	0.9032	1.4251	0.9549	1.1404	0.8111
FW	0.5682	0.5874	0.6327	0.6096	**0.5789**	**0.5379**	**0.6080**	**0.5996**	0.6393	0.6161	0.6054	0.5901
QR	**0.5610**	**0.5778**	0.6309	**0.6013**	0.6032	0.5536	0.6345	0.6070	**0.5974**	**0.5918**	0.6054	**0.5863**
Mixed_{28}	0.5662	0.5854	0.6316	0.6078	0.5831	0.5409	0.6130	0.6010	0.6302	0.6110	0.6048	0.5892
Mixed_{55}	0.5638	0.5823	0.6307	0.6052	0.5901	0.5455	0.6207	0.6032	0.6172	0.6034	**0.6045**	0.5879
Mixed_{82}	0.5619	0.5795	**0.6306**	0.6027	0.5976	0.5502	0.6289	0.6054	0.6050	0.5963	0.6048	0.5868

C.3 More results on Downstream RL Tasks

This subsection reports the per-judge breakdowns underlying the pooled main-text tables: Table 11 (DeepSeek-V4-Pro) and Table 12 (Gemini 3.1 Pro Preview) correspond to 4 (Section 5.2), and Table 13 (DeepSeek-V4-Pro) and Table 14 (Gemini 3.1 Pro Preview) correspond to 5 (Section 5.2).

Per-judge downstream results on General Domain. Experimental results in Table 11 and Table 12 show that our calibration is effective under both the DeepSeek-V4-Pro judge and the Gemini 3.1 Pro Preview judge, as every method combined with our calibration attains average win rates above 50%

Table 11: Downstream reinforcement learning in the general domain, evaluated by the DeepSeek-V4-Pro judge. We report the pairwise win rate under the trained rubrics. Average is taken over the three rubrics. The best results are **bolded**.

	HS2-test				WildChat				AlpacaEval-2			
Method	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**
Qwen2.5-7B-Instruct												
Raw Judge	34.49	37.48	21.77	31.25	33.56	34.97	24.96	31.16	35.40	30.47	22.11	29.33
Raw + Ours	**65.51**	**62.52**	**78.23**	**68.75**	**66.44**	**65.03**	**75.04**	**68.84**	**64.60**	**69.53**	**77.89**	**70.67**
Importance-Based	33.42	36.64	38.84	36.30	36.65	37.42	36.32	36.80	37.86	36.71	37.11	37.23
Imp + Ours	**66.58**	**63.36**	**61.16**	**63.70**	**63.35**	**62.58**	**63.68**	**63.20**	**62.14**	**63.29**	**62.89**	**62.77**
Dropout	24.54	32.69	38.07	31.77	28.02	38.16	**50.78**	38.99	21.27	36.77	36.96	31.67
Dropout + Ours	**75.46**	**67.31**	**61.93**	**68.23**	**71.98**	**61.84**	49.22	**61.01**	**78.73**	**63.23**	**63.04**	**68.33**
Llama-3.1-8B-Instruct												
Raw Judge	32.30	38.03	22.47	30.93	33.21	38.46	30.72	34.13	33.42	35.16	28.82	32.47
Raw + Ours	**67.70**	**61.97**	**77.53**	**69.07**	**66.79**	**61.54**	**69.28**	**65.87**	**66.58**	**64.84**	**71.18**	**67.53**
Importance-Based	35.24	41.79	32.33	36.45	37.56	44.40	40.26	40.74	35.05	39.43	42.46	38.98
Imp + Ours	**64.76**	**58.21**	**67.67**	**63.55**	**62.44**	**55.60**	**59.74**	**59.26**	**64.95**	**60.57**	**57.54**	**61.02**
Dropout	29.04	42.13	34.94	35.37	28.66	42.02	37.54	36.07	25.37	41.71	34.56	33.88
Dropout + Ours	**70.96**	**57.87**	**65.06**	**64.63**	**71.34**	**57.98**	**62.46**	**63.93**	**74.63**	**58.29**	**65.44**	**66.12**
Qwen3-4B												
Raw Judge	29.32	35.17	30.76	31.75	30.28	36.17	31.33	32.59	34.04	33.04	30.93	32.67
Raw + Ours	**70.68**	**64.83**	**69.24**	**68.25**	**69.72**	**63.83**	**68.67**	**67.41**	**65.96**	**66.96**	**69.07**	**67.33**
Importance-Based	33.23	37.23	37.68	36.05	34.65	40.65	38.26	37.85	39.38	41.15	40.58	40.37
Imp + Ours	**66.77**	**62.77**	**62.32**	**63.95**	**65.35**	**59.35**	**61.74**	**62.15**	**60.62**	**58.85**	**59.42**	**59.63**
Dropout	23.98	39.91	**50.32**	38.07	22.99	39.91	39.25	34.05	20.42	39.35	38.37	32.71
Dropout + Ours	**76.02**	**60.09**	49.68	**61.93**	**77.01**	**60.09**	**60.75**	**65.95**	**79.58**	**60.65**	**61.63**	**67.29**

Table 12: Downstream reinforcement learning in the general domain, evaluated by the Gemini 3.1 Pro Preview judge. We report the pairwise win rate under the trained rubrics. Average is taken over the three rubrics. The best results are **bolded**.

	HS2-test				WildChat				AlpacaEval-2			
Method	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**	**Hel.**	**Cor.**	**Coh.**	**Avg**
Qwen2.5-7B-Instruct												
Raw Judge	42.87	42.10	26.97	37.31	41.86	39.36	33.03	38.08	43.11	40.99	29.32	37.81
Raw + Ours	**57.13**	**57.90**	**73.03**	**62.69**	**58.14**	**60.64**	**66.97**	**61.92**	**56.89**	**59.01**	**70.68**	**62.19**
Importance-Based	42.07	40.92	37.25	40.08	44.18	41.60	38.34	41.37	44.58	42.19	35.40	40.72
Imp + Ours	**57.93**	**59.08**	**62.75**	**59.92**	**55.82**	**58.40**	**61.66**	**58.63**	**55.42**	**57.81**	**64.60**	**59.28**
Dropout	41.52	39.39	37.35	39.42	39.52	43.15	**50.12**	44.26	38.70	42.07	33.57	38.11
Dropout + Ours	**58.48**	**60.61**	**62.65**	**60.58**	**60.48**	**56.85**	49.88	**55.74**	**61.30**	**57.93**	**66.43**	**61.89**
Llama-3.1-8B-Instruct												
Raw Judge	44.51	44.03	31.79	40.11	46.97	45.43	36.38	42.93	49.94	49.32	34.78	44.68
Raw + Ours	**55.49**	**55.97**	**68.21**	**59.89**	**53.03**	**54.57**	**63.62**	**57.07**	**50.06**	**50.68**	**65.22**	**55.32**
Importance-Based	45.72	45.01	39.41	43.38	48.24	47.84	39.09	45.06	49.76	49.51	39.84	46.37
Imp + Ours	**54.28**	**54.99**	**60.59**	**56.62**	**51.76**	**52.16**	**60.91**	**54.94**	**50.24**	**50.49**	**60.16**	**53.63**
Dropout	38.94	46.12	41.62	42.23	37.23	47.29	38.09	40.87	40.32	48.70	37.76	42.26
Dropout + Ours	**61.06**	**53.88**	**58.38**	**57.77**	**62.77**	**52.71**	**61.91**	**59.13**	**59.68**	**51.30**	**62.24**	**57.74**
Qwen3-4B												
Raw Judge	44.22	42.00	36.51	40.91	39.60	39.12	34.70	37.81	44.28	41.79	38.43	41.50
Raw + Ours	**55.78**	**58.00**	**63.49**	**59.09**	**60.40**	**60.88**	**65.30**	**62.19**	**55.72**	**58.21**	**61.57**	**58.50**
Importance-Based	44.63	45.73	40.60	43.65	43.67	44.87	40.28	42.94	46.15	45.51	42.88	44.85
Imp + Ours	**55.37**	**54.27**	**59.40**	**56.35**	**56.33**	**55.13**	**59.72**	**57.06**	**53.85**	**54.49**	**57.12**	**55.15**
Dropout	41.78	46.92	**50.91**	46.54	33.53	43.31	40.67	39.17	34.84	43.08	40.42	39.45
Dropout + Ours	**58.22**	**53.08**	49.09	**53.46**	**66.47**	**56.69**	**59.33**	**60.83**	**65.16**	**56.92**	**59.58**	**60.55**

in all cells. The two judges also produce consistent rankings of the methods, which together confirms the consistency of the LLM-judge evaluation.

Per-judge downstream results on Expert Domain. Experimental results in Table 13 and Table 14 show that our calibration is effective under both judges on the expert domain, as every method combined with our calibration attains trained-rubric win rates above 50% in all cells. Meanwhile, the original rubric scores remain marginal and inconsistent under both judges, showing that the two judges consistently reveal the applicability boundary of our calibration. This confirms the consistency of the LLM-judge evaluation in the expert domain.

Table 13: Downstream reinforcement learning in the expert domain, evaluated by the DeepSeek-V4-Pro judge. We report the pairwise win rate under the trained rubrics and the original rubric score produced by the domain-specific rubric of each benchmark. Average is taken over the five benchmarks. The best results are **bolded**.

	Medicine						Science				Avg	
	RaR-Med		HealthBench		LLMEval-Med		RaR-Sci		ResearchQA			
Method	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**
Qwen2.5-7B-Instruct	45.09	–	19.38	–	54.75	–	49.18	–	46.11	–	–	–
Raw Judge	43.93	38.51	19.96	27.18	**56.31**	36.62	**50.43**	41.72	46.70	33.02	43.47	35.41
Raw + Ours	**45.20**	**61.49**	**21.76**	**72.82**	55.92	**63.38**	50.36	**58.28**	**46.81**	**66.98**	**44.01**	**64.59**
Importance-Based	**51.52**	40.99	**34.79**	31.97	**61.53**	38.93	55.50	48.68	55.43	42.33	51.75	40.58
Imp + Ours	50.10	**59.01**	33.83	**68.03**	61.38	**61.07**	**55.60**	**51.32**	**57.91**	**57.67**	**51.76**	**59.42**
Dropout	44.64	24.64	23.64	27.20	60.51	31.18	54.13	49.33	54.80	45.20	47.54	35.51
Dropout + Ours	**49.23**	**75.36**	**31.58**	**72.80**	**61.11**	**68.82**	**54.50**	**50.67**	**54.90**	**54.80**	**50.26**	**64.49**
Llama-3.1-8B-Instruct	44.66	–	14.93	–	50.10	–	39.88	–	45.12	–	–	–
Raw Judge	**49.88**	39.31	18.19	38.80	**51.12**	47.10	**43.55**	44.17	**52.17**	41.82	**42.98**	42.24
Raw + Ours	49.86	**60.69**	**18.91**	**61.20**	51.09	**52.90**	42.97	**55.83**	49.14	**58.18**	42.39	**57.76**
Importance-Based	53.79	37.96	25.80	37.44	55.65	45.13	46.85	43.69	**56.90**	43.66	47.80	41.58
Imp + Ours	**54.80**	**62.04**	**27.56**	**62.56**	**56.43**	**54.87**	**48.11**	**56.31**	56.69	**56.34**	**48.72**	**58.42**
Dropout	52.30	41.79	28.04	36.14	**56.85**	47.75	46.44	42.20	51.33	18.80	46.99	37.34
Dropout + Ours	**55.09**	**58.21**	**30.14**	**63.86**	56.55	**52.25**	**46.45**	**57.80**	**54.49**	**81.20**	**48.54**	**62.66**
Qwen3-4B	47.24	–	20.47	–	57.18	–	58.98	–	54.82	–	–	–
Raw Judge	**50.45**	39.17	**22.68**	37.04	**58.32**	32.66	61.12	37.72	57.80	34.12	**50.07**	36.14
Raw + Ours	50.04	**60.83**	21.74	**62.96**	57.75	**67.34**	**61.30**	**62.28**	**58.12**	**65.88**	49.79	**63.86**
Importance-Based	54.49	45.39	39.00	47.36	63.99	47.82	**65.94**	44.42	**65.80**	48.40	**57.84**	46.68
Imp + Ours	**54.65**	**54.61**	**39.84**	**52.64**	**64.62**	**52.18**	64.19	**55.58**	64.80	**51.60**	57.62	**53.32**
Dropout	**53.74**	42.57	**38.92**	33.33	**64.23**	38.14	**64.05**	41.53	61.36	26.65	**56.46**	36.44
Dropout + Ours	52.58	**57.43**	37.23	**66.67**	63.90	**61.86**	63.63	**58.47**	**62.25**	**73.35**	55.92	**63.56**

Table 14: Downstream reinforcement learning in the expert domain, evaluated by the Gemini 3.1 Pro Preview judge. We report the pairwise win rate under the trained rubrics and the original rubric score produced by the domain-specific rubric of each benchmark. Average is taken over the five benchmarks. The best results are **bolded**.

	Medicine						Science				Avg	
	RaR-Med		HealthBench		LLMEval-Med		RaR-Sci		ResearchQA			
Method	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**	**Original**	**Trained**
Qwen2.5-7B-Instruct	39.14	–	19.96	–	47.89	–	43.45	–	55.38	–	–	–
Raw Judge	38.35	46.56	21.10	36.56	**49.60**	42.02	43.92	47.88	**55.59**	45.52	41.71	43.71
Raw + Ours	**40.28**	**53.44**	**22.37**	**63.44**	49.60	**57.98**	**43.98**	**52.12**	55.56	**54.48**	**42.36**	**56.29**
Importance-Based	**42.83**	48.77	**24.71**	46.73	49.03	48.08	44.70	48.40	57.54	42.30	**43.76**	46.86
Imp + Ours	41.21	**51.23**	22.20	**53.27**	**49.87**	**51.92**	**45.13**	**51.60**	**58.81**	**57.70**	43.44	**53.14**
Dropout	40.10	29.62	21.56	26.80	48.84	31.54	43.71	47.08	56.73	43.60	42.19	35.73
Dropout + Ours	**40.19**	**70.38**	**23.53**	**73.20**	**50.04**	**68.46**	**43.85**	**52.92**	**56.97**	**56.40**	**42.92**	**64.27**
Llama-3.1-8B-Instruct	40.54	–	16.79	–	41.14	–	36.37	–	52.72	–	–	–
Raw Judge	**45.67**	40.02	**20.47**	47.36	**41.65**	49.75	**38.84**	41.96	**61.61**	41.25	**41.65**	44.07
Raw + Ours	43.93	**59.98**	18.12	**52.64**	41.17	**50.25**	37.63	**58.04**	58.46	**58.75**	39.86	**55.93**
Importance-Based	43.85	41.13	17.29	43.80	41.14	42.17	38.48	43.65	59.47	41.57	40.05	42.46
Imp + Ours	**45.29**	**58.87**	**17.60**	**56.20**	**41.71**	**57.83**	**38.95**	**56.35**	**59.80**	**58.43**	**40.67**	**57.54**
Dropout	42.71	47.28	**20.76**	43.60	**43.57**	43.88	**38.31**	39.11	55.27	24.93	40.12	39.76
Dropout + Ours	**45.38**	**52.72**	20.25	**56.40**	43.18	**56.12**	36.52	**60.89**	**58.07**	**75.07**	**40.68**	**60.24**
Qwen3-4B	42.15	–	24.63	–	50.94	–	53.13	–	62.21	–	–	–
Raw Judge	**44.59**	35.85	**24.69**	40.48	52.53	35.23	**54.72**	40.40	65.27	34.49	**48.36**	37.29
Raw + Ours	44.09	**64.15**	23.38	**59.52**	**52.92**	**64.77**	54.07	**59.60**	**65.54**	**65.51**	48.00	**62.71**
Importance-Based	45.20	44.29	23.93	44.37	**53.04**	45.55	**54.07**	48.01	**66.14**	48.47	**48.48**	46.14
Imp + Ours	**45.25**	**55.71**	**25.29**	**55.63**	52.68	**54.45**	51.93	**51.99**	64.94	**51.53**	48.02	**53.86**
Dropout	**44.61**	41.58	**25.76**	43.60	**51.81**	40.44	**53.02**	41.06	62.33	46.07	**47.51**	42.55
Dropout + Ours	43.59	**58.42**	22.38	**56.40**	51.12	**59.56**	52.26	**58.94**	**63.32**	**53.93**	46.53	**57.45**

D PROMPT TEMPLATES

This section shows every prompt template used in our system. We cover three parts: the scalar judge used as the RL training reward (Section D.1), the listwise judge used for post-RL policy evaluation of both general domain and expert domain (Section D.2), and the original rubric graders used for expert domain evaluation (Section D.3).

D.1 SCALAR JUDGE PROMPT (RL TRAINING REWARD)

During RL training, every rollout is scored by an external LLM judge on three rubrics (`helpfulness`, `correctness`, `coherence`), with integer scores from 0 to 4. The resulting per-rubric scores are conformally clipped against sample-conditional prediction intervals and then aggregated into a scalar reward. Each judge request comprises a fixed system message and a user message instantiated from a shared template, in which the {`rendered_contract`} placeholder is expanded into the frozen rubric definitions.

Scalar Judge Prompt — Overall Structure

```
<|system|>
You are a strict reward-modeling judge. Apply the supplied frozen
    rubric contract exactly and return only valid JSON.

<|user|>
Score the assistant response under the following frozen rubric
    contract.

{rendered_contract}

Assign one integer score from 0 to 4 for every requested rubric.
    Apply each rubric independently and use its explicit score
    anchors.

Return JSON with exactly this shape:
{example_payload}

Use the requested rubric names exactly. Do not include
    explanations.

User prompt:
{original_prompt}

Assistant response:
{original_response}
```

Scalar Judge Prompt — Complete Template

```
<|system|>
You are a strict reward-modeling judge. Apply the supplied frozen
    rubric contract exactly and return only valid JSON.

<|user|>
Score the assistant response under the following frozen rubric
    contract.

Shared judging rules:
- Evaluate only the requested rubric and keep different rubric
    scores independent.
- Use the user prompt, its task goal and restrictions, and any
    supplied reference or context as the basis for judgment; do not
    invent additional requirements.
- Response length is NOT a quality signal. A concise response that
    fully and accurately satisfies the rubric is equally good or
    better than a longer one. Longer responses may contain correct
    but irrelevant information that does not improve the answer. Do
    NOT prefer longer responses simply because they contain more
    text.
```

```
- Use the full 0-4 scale. Higher always means better performance
    on the requested rubric.

[helpfulness] Helpfulness
Definition: How useful and helpful the overall response is.
Score anchors:
  0 - Completely Unhelpful: The response is not useful or helpful
    at all. The response completely missed the essence of what the
    user wanted.
  1 - Mostly Unhelpful: The response is borderline unhelpful and
    mostly does not capture what the user was looking for, but is
    still usable and helpful in a small way.
  2 - Partially Helpful: The response is partially helpful but
    misses the overall goal of the user's query/input in some way.
    The response did not fully satisfy what the user was looking
    for.
  3 - Mostly Helpful: The response is mostly helpful and mainly
    aligned with what the user was looking for, but there is still
    some room for improvement.
  4 - Extremely Helpful: The response is extremely helpful and
    completely aligned with the spirit of what the prompt was
    asking for.

[correctness] Correctness
Definition: The response is based on facts, no hallucinations, no
    mistakes. The response covers everything required in the
    instruction.
Score anchors:
  0 - Completely Incorrect: The response is completely incorrect.
    All information provided is wrong, false or hallucinated. If
    the prompt asks the assistant to do a task, the task is not at
    all attempted, or the wrong task was attempted in the response.
    The response is completely irrelevant to the prompt.
  1 - Mostly Incorrect: The response has some correct elements but
    is mostly wrong or incomplete. The response may contain
    multiple instances of hallucinations, false information,
    misleading information, or irrelevant information. If the
    prompt asks the assistant to do a task, the task was attempted
    with a small amount of success.
  2 - Mix of Correct and Incorrect: The response contains a mix of
    correct and incorrect information. The response may miss some
    details, contain misleading information, or minor
    hallucinations, but is more or less aligned with what the
    prompt asks for. If the prompt asks the assistant to perform a
    task, the task is attempted with moderate success but still has
    clear room for improvement.
  3 - Mostly Correct: The response is mostly accurate and correct
    with a small amount of missing information. It contains no
    misleading information or hallucinations. If the prompt asks
    the assistant to do a task, the task is mostly successfully
    attempted.
  4 - Completely Correct: The response is completely correct and
    accurate to what is requested by the prompt with no necessary
    details missing and without false, misleading, or hallucinated
    information. If the prompt asks the assistant to do a task, the
    task is completely done and addressed in the response.

[coherence] Coherence
Definition: The response is self-consistent in terms of content,
    style of writing, and does not contradict itself. The response
    can be logically followed and understood by a human. The
    response does not contain redundant or repeated information.
Score anchors:
```

```
  0 - Completely Incoherent: The response is completely
    incomprehensible. No clear meaning or sensible message can be
    discerned from the response.
  1 - Mostly Incoherent: The response is mostly hard to follow
    because of inconsistencies, contradictions, confusing logic, or
    unclear language, although some parts are coherent.
  2 - A Little Incoherent: The response is generally
    understandable but contains noticeable inconsistencies,
    contradictions, confusing statements, or hard-to-follow
    sections.
  3 - Mostly Coherent: The response is mostly clear and logically
    followable, with only one or two minor wording or flow problems.
  4 - Perfectly Coherent: The response is perfectly clear,
    logical, and self-consistent throughout, with no contradictory
    assertions and no challenging breaks in the flow.

Assign one integer score from 0 to 4 for every requested rubric.
    Apply each rubric independently and use its explicit score
    anchors.

Return JSON with exactly this shape:
{"scores": {"helpfulness": 0, "correctness": 0, "coherence": 0}}

Use the requested rubric names exactly. Do not include
    explanations.

User prompt:
<user prompt>

Assistant response:
<response to be scored>
```

Importance-Based Prompt. For the importance-based methods, the template shares the same system prompt, rendered contract, and closing, and only replaces the middle instruction and the required JSON shape:

Scalar Judge — Importance-based

```
Assign one integer score from 0 to 4 for every requested rubric.
    Also assign an importance weight from 0.0 to 1.0 for each
    rubric based on the prompt and response, indicating how much
    that rubric should matter for this evaluation.

Return JSON with exactly this shape:
{"scores": {"helpfulness": 0, "correctness": 0, "coherence": 0},
    "importance": {"helpfulness": 1.0, "correctness": 1.0,
    "coherence": 1.0}}
```

D.2 Listwise Evaluation Prompts

For downtream RL evaluation, two policy responses are placed into slots and the judge selects one winner per rubric (pairwise). The per-rubric pairwise template below is instantiated with a single-rubric contract (shown for `helpfulness`).

Listwise Judge Prompt — Per-Rubric Pairwise (helpfulness)

```
<|system|>
You are a strict evaluation judge. Return only valid JSON. You
    must choose exactly one best candidate. Ties are not allowed.

<|user|>
Compare the candidate assistant responses under the following
    frozen single-rubric contract.

Shared judging rules:
- Evaluate only the requested rubric and keep different rubric
    scores independent.
- Use the user prompt, its task goal and restrictions, and any
    supplied reference or context as the basis for judgment; do not
    invent additional requirements.
- Response length is NOT a quality signal. A concise response that
    fully and accurately satisfies the rubric is equally good or
    better than a longer one. Longer responses may contain correct
    but irrelevant information that does not improve the answer. Do
    NOT prefer longer responses simply because they contain more
    text.
- Use the full 0-4 scale. Higher always means better performance
    on the requested rubric.

[helpfulness] Helpfulness
Definition: How useful and helpful the overall response is.
Score anchors:
  0 - Completely Unhelpful: The response is not useful or helpful
    at all. The response completely missed the essence of what the
    user wanted.
  1 - Mostly Unhelpful: The response is borderline unhelpful and
    mostly does not capture what the user was looking for, but is
    still usable and helpful in a small way.
  2 - Partially Helpful: The response is partially helpful but
    misses the overall goal of the user's query/input in some way.
    The response did not fully satisfy what the user was looking
    for.
  3 - Mostly Helpful: The response is mostly helpful and mainly
    aligned with what the user was looking for, but there is still
    some room for improvement.
  4 - Extremely Helpful: The response is extremely helpful and
    completely aligned with the spirit of what the prompt was
    asking for.

Assess every candidate against the same 0-4 anchors, then choose
    exactly one best response for this rubric only. If candidates
    fall in the same integer anchor band, choose the one closer to
    the next higher anchor based only on this rubric. Ties are not
    allowed.

Return JSON with exactly this shape:
{"winner":"A"}
The winner must be one of: A, B.
Do not include explanations.

User prompt:
<original user prompt>

Response A:
<response of policy A>

Response B:
```

```
<response of policy B>
```

D.3 Original Rubric Evaluation Prompts

For other-domain benchmarks that ship their own rubrics, we do not use our rubric contracts; the benchmark's original rubric grading protocol is followed verbatim.

D.3.1 Original Rubric Grader for RaR-Medicine and RaR-Science

Original Rubric Grader for RaR-Medicine and RaR-Science

```
Your job is to look at a conversation and a list of rubric items,
    and score the last turn (i.e., the last assistant response)
    against each rubric item.

# Conversation
<<conversation>>

# Rubric items
<<rubric_items>>

# Instructions
For each rubric item, determine whether the response meets the
    criteria.
Return a JSON object with a "results" field containing a list of
    objects, one per rubric item, in the same order as listed above.

Each result object must have:
- "explanation": a string explaining why the response does or does
    not meet the criteria.
- "criteria_met": a boolean (true or false).

Rules:
- If a rubric item has multiple sentences or criteria, consider
    all of them. If any is not met, the answer is false.
- If a criteria says "such as", "for example", or "including", the
    response does not have to include all examples listed.
- Some criteria may be undesirable (pitfall criteria). For those,
    criteria_met=true means the response DID exhibit the
    undesirable behavior.

# Example
```json
{
  "results": [
    {"explanation": "The response correctly identifies...",
    "criteria_met": true},
    {"explanation": "The response does not mention...",
    "criteria_met": false}
  ]
}
```

Return just the JSON object. Do not include any other text.
```

D.3.2 Original Rubric Grader for HealthBench

One call per criterion per judge, following the official HealthBench grading protocol.

Original Rubric Grader Template for HealthBench

```
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 critera 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.3.3 RESEARCHQA 5-LEVEL COVERAGE PROMPT

Official coverage scoring; ratings map to integers as `not at all`=1, `barely`=2, `moderately`=3, `mostly`=4, `completely`=5.

ResearchQA Coverage Prompt

```
Please judge the following questions based on the response below.
For each question, select one of the following ratings to indicate
    the extent to which the response addresses the question:
Not at all, Barely, Moderately, Mostly, Completely

Definitions:
- Not at all: *totally uninferable*
- Barely: *unmentioned but inferrable*
- Moderately: *mentioned but misses important details*
- Mostly: *mentioned but misses some details*
- Completely: *mentioned with sufficient details*

Only output one of the five phrases for each question, separated
    by newlines, and nothing else.

Response: {response}
Questions:
{questions}

Output:
```

D.4 ORIGINAL RUBRIC EXAMPLES

Original Rubric — RaR-Medicine (DiGeorge syndrome, 11 criteria)

```
Criterion 1  [weight +5]  Identifies DiGeorge Syndrome
  Essential Criteria: Identifies the symptoms and laboratory
    findings as consistent with DiGeorge syndrome (22q11.2 deletion
    syndrome).

Criterion 2  [weight +5]  Mentions Thymus Hypoplasia
  Essential Criteria: Mentions that the child's disorder is
    associated with aplasia or hypoplasia of the thymus.

Criterion 3  [weight +4]  Specifies Hypocalcemia Cause
  Important Criteria: Explains that hypocalcemia is due to
    impaired parathyroid function related to thymus
    underdevelopment.

Criterion 4  [weight +3]  Craniofacial Anomalies
  Important Criteria: Highlights the presence of craniofacial
    anomalies as a symptom indicative of DiGeorge syndrome.

Criterion 5  [weight +3]  Congenital Heart Defects
  Important Criteria: Discusses congenital heart defects as part
    of the symptomatology of DiGeorge syndrome.

Criterion 6  [weight +5]  States Correct Answer
  Essential Criteria: Includes a clear statement that the correct
    answer is C. Thymus.

Criterion 7  [weight +4]  Explains Condition Before Answer
  Important Criteria: Presents the explanation of the syndrome and
    associated symptoms before stating the final answer.

Criterion 8  [weight +3]  Contextual Relevance
  Important Criteria: Keeps the explanation relevant to the
    symptoms and laboratory findings described in the question.

Criterion 9  [weight -2]  Avoids Irrelevant Organs
  Pitfall Criteria: Does not mention unrelated organs like
    Ovaries, Pituitary, or Thyroid as associated with the disorder.

Criterion 10 [weight +2]  Patient-Centered Language
  Optional Criteria: Uses empathetic and patient-centered language
    when describing the child's condition.

Criterion 11 [weight +3]  Clear Symptom-Cause Link
  Important Criteria: Clearly links the described symptoms
    (convulsions and tetany) to the underlying hypocalcemia.
```

Original Rubric — ResearchQA (microplastic sorption, 6 questions)

```
Question 1  [type: Impact]
  Does the response explain how hydrophobicity influences sorption
    of organic contaminants onto microplastic polymers?

Question 2  [type: Other]
  Does the response describe the role of molecular structure
    factors, such as planarity and molecular size, on sorption
    behavior?
```

```
Question 3  [type: Other]
  Does the response discuss how specific types of microplastic
    polymers differ in their sorption affinities for contaminants?

Question 4  [type: Impact]
  Does the response explore in depth how hydrophobicity interacts
    with molecular structure factors such as planarity, size, and
    functional groups to influence sorption behavior?

Question 5  [type: Comparison]
  Does the response provide a detailed explanation of how
    different microplastic polymer properties (e.g., polarity,
    surface area, free volume) affect the sorption mechanisms?

Question 6  [type: Citation]
  Does the response cite Hüffer and Hofmann (2016) (title:
    Sorption of non-polar organic compounds by micro-sized plastic
    particles in aqueous solution) to explain the different
    sorption mechanisms on polystyrene (PS) involving π-π and
    planar molecular interactions compared to polyethylene (PE)?
```

Original Rubric — HealthBench (postpartum depression plan, 6 of 19 criteria)

```
Criterion 1  [points +10, axis: completeness]
  Advises the patient to find a mental health provider for
    diagnosis, assessment, and treatment (via Psychology Today,
    PSI, asking PCP/OB-GYN, etc.).

Criterion 2  [points +9, axis: completeness]
  Includes contacting emergency services if thoughts of wanting to
    hurt themselves or others arise.

Criterion 3  [points +7, axis: context_awareness]
  Recommends a weekly interval for therapy after care is
    established.

Criterion 4  [points -10, axis: accuracy]
  Advises patient not to attend scheduled postpartum appointments
    with a provider.

Criterion 5  [points +8, axis: completeness]
  Clearly provides a week-by-week or monthly breakdown with
    specific goals, intentions, and interventions to achieve those
    goals.

Criterion 6  [points +7, axis: completeness]
  Clearly explains how therapy works, such as cognitive therapy,
    what to expect in each session, and how it will address
    postpartum depression.
```