RUBRICS ALWAYS MISS: ROBUST PREFERENCE OPTIMIZATION UNDER INCOMPLETE CRITERIA

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubrics provide fine-grained and interpretable supervision for aligning large language models by evaluating responses against explicit criteria, such as factual correctness, relevance, and instruction following. Recent methods generate rubrics that specify what makes a good response to a given instruction, then use rubric scores as rewards or to construct preferences for training. However, it is difficult to ensure that these rubrics cover all relevant criteria, so their scores may not fully reflect response quality. To address this issue, we propose a robust preference optimization framework that accounts for errors in preferences caused by incomplete rubrics. Since the missing criteria are unobserved, we assess rubric completeness indirectly by examining whether the observed rubrics capture differences between responses. For each instruction, we perturb the policy to generate response variations and measure how consistently larger response changes are accompanied by larger changes in rubric scores. This agreement serves as an instance-specific proxy for rubric completeness. We then use the estimated completeness to construct a feasible set around the preference probabilities inferred from the observed rubric scores, allowing larger deviations for samples with lower estimated completeness, and therefore a larger feasible set. The policy minimizes the worst-case training loss over this set, thereby allocating greater protection against preference errors to samples with less complete rubrics. Our framework can be integrated with any margin-aware preference optimization methods. Experiments across multiple datasets demonstrate consistent improvements over existing methods. Code is on anonymous GitHub[1].

1 INTRODUCTION

Rubrics provide fine-grained and interpretable supervision for aligning large language models by evaluating responses against explicit criteria, such as factual correctness, relevance, and instruction following (Bai et al., 2022; Wu et al., 2023; Dong et al., 2023). By making these requirements explicit, rubrics allow evaluators to assess different aspects of a response and explain the resulting judgment (Liu et al., 2023; Ye et al., 2024; Kim et al., 2024b). Recent methods construct rubrics from instructions and response comparisons, and refine the criteria using preference feedback (Xie et al., 2025; Liu et al., 2026; Xu et al., 2026). The resulting scores can serve as rewards for reinforcement learning (Gunjal et al., 2026) or indicate how strongly one response should be preferred over another during preference optimization (Amini et al., 2024; Kim et al., 2024a). Rubrics thus connect concrete requirements for response quality with the supervision used to train a policy.

However, it is difficult to ensure that these rubrics cover all relevant criteria, so their scores may not fully reflect response quality. Consider the two responses in Figure 1. A rubric that checks factual correctness, relevance, clarity, and instruction following favors response B over response A. Yet response B also scores higher on helpfulness and safety, which this rubric does not evaluate. Including these criteria increases the average score difference between the responses, showing that the observed rubric underestimates B's advantage. Even when every observed criterion is scored accurately, omitted criteria can change how strongly one response should be preferred and may sometimes reverse the preference. Training on these incomplete assessments can therefore provide inaccurate guidance for policy learning.

[1] https://anonymous.4open.science/r/rubric-supplement-AE6E

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.

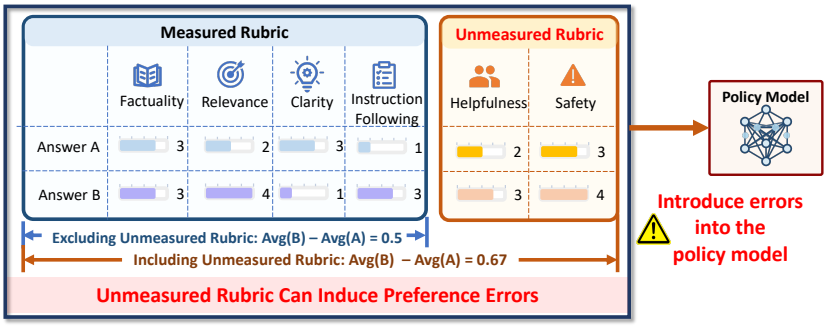


Figure 1: Missing criteria can affect response comparisons.

Accounting for such preference errors is challenging because we do not know which criteria are missing. Therefore, we cannot directly measure how far the preferences inferred from the observed scores deviate from those based on a complete evaluation. In addition, the effect of an omission also depends on the sample: failing to evaluate safety matters more when the candidate responses differ substantially in the risks they introduce. Human review can reduce such errors by adding overlooked criteria, but it cannot guarantee that all relevant considerations for a given query and response pair have been covered. Policy learning therefore still needs to account for possible preference errors even after careful human review.

To address this challenge, we propose a robust learning framework that accounts for possible preference errors caused by incomplete rubrics. Although we cannot inspect the missing criteria, we can examine whether the rubric captures differences between responses. If the responses change substantially but their rubric scores do not, the rubric may be overlooking aspects on which the responses differ. We therefore vary the model's responses to the same instruction and measure how consistently larger response changes are accompanied by larger changes in rubric scores. This agreement serves as a proxy for rubric completeness for each sample. We use this proxy to construct a set of possible preference distributions around the one implied by the observed scores, allowing larger deviations for samples with lower estimated completeness. The policy minimizes the worst-case loss over this set, accounting for preferences that could change when omitted criteria are considered. The framework can be integrated with existing preference optimization methods and used alongside methods that improve rubric construction. Our contributions are threefold:

- We formulate preference learning with incomplete rubrics by accounting for possible deviations from the preferences implied by the observed evaluation criteria.
- We introduce a rubric completeness score for each sample by measuring how consistently changes in model responses are reflected in rubric scores.
- We develop a distributionally robust training objective that uses these completeness scores to constrain changes in preference probabilities and optimizes the policy against the worst-case preferences within those constraints.

2 Related Work

Rubric-Based LLM Evaluation. Rubrics have emerged as an effective way to provide fine-grained and interpretable evaluation for open-ended LLM responses by decomposing holistic quality judgments into explicit criteria. Early studies use predefined evaluation dimensions or explicit score rubrics to assess different aspects of response quality (Ye et al., 2024; Kim et al., 2024b; Hashemi et al., 2024). Subsequent work moves toward instance-specific evaluation, where the criteria are adapted to the requirements of individual prompts (Lin et al., 2025; Arora et al., 2025). As manually constructing such rubrics is costly, recent studies increasingly explore automatic rubric construction. Their approaches range from generating criteria from instructions, reference responses, or preference comparisons to refining them through feedback, filtering, or joint optimization (Fan et al., 2024; Xie et al., 2025; Liu et al., 2026; Siro et al., 2026; Rezaei et al., 2026; Shen et al., 2026; Xu

et al., 2026; Li et al., 2026). Despite these advances, constructing a rubric that captures all relevant aspects of response quality remains challenging.

Rubric-Based Preference Alignment. Rubric scores can further provide fine-grained supervision for LLM training. Existing methods either aggregate criterion-level evaluations into reward signals for reinforcement learning (Gunjal et al., 2026; Viswanathan et al., 2025; Liu et al., 2026), or exploit quality differences between responses as preference strength in preference optimization (Amini et al., 2024; Kim et al., 2024a; Wang et al., 2024). However, these methods generally optimize directly against the observed rubric-derived scores or margins. When relevant criteria are missing, such supervision may not fully reflect the underlying quality difference. We explicitly accounts for this incompleteness through instance-specific rubric completeness and performs robust optimization.

3 PRELIMINARIES

Rubrics provide fine-grained supervision by evaluating a response against a set of explicit criteria (Gunjal et al., 2026; Liu et al., 2026; Rezaei et al., 2026), and the resulting scores can be used as reward signals for LLM training. Specifically, for i-th sample, given an instruction x_i, we denote its observed rubric as $\mathcal{C}_i = \{(c_{ik}, w_{ik})\}_{k=1}^{K_i}$, where c_{ik} is the k-th evaluation criterion and w_{ik} is its importance weight. Given a rubric evaluator v_ψ, the score of response y is $r_i(y) = \sum_{k=1}^{K_i} w_{ik} v_\psi(x_i, y, c_{ik})$, which measures the quality of y according to observed rubric $\mathcal{C}_i$.

Rubric scores can further provide fine-grained supervision for pairwise preference optimization. For a preferred response y_i^w and a dispreferred response y_i^l, we define the rubric-derived preference margin as $m_i = r_i(y_i^w) - r_i(y_i^l)$, which measures the relative quality difference between the two responses under the observed rubric. Following the Bradley–Terry model, the score difference can be mapped to a preference probability (Bradley & Terry, 1952; Kim et al., 2024a): $p_i = \sigma(\gamma m_i)$, where $\gamma > 0$ controls the scale of the preference margin. We refer to p_i as the nominal preference probability induced by the observed rubric. Then we use margin-aware preference optimization methods, such as Margin Matching Preference Optimization (MMPO) (Kim et al., 2024a) and ODPO (Amini et al., 2024), which use score differences to account for preference strength during policy learning. Taking MMPO as an example, given a policy π_θ and a reference policy π_{ref}, define the implicit reward difference as:

$$\Delta_{\theta,i} = \beta \log \frac{\pi_\theta(y_i^w \mid x_i)}{\pi_{\text{ref}}(y_i^w \mid x_i)} - \beta \log \frac{\pi_\theta(y_i^l \mid x_i)}{\pi_{\text{ref}}(y_i^l \mid x_i)}, \tag{1}$$

where $\beta > 0$ controls the scale of the implicit rewards. MMPO uses p_i as a soft target in the cross-entropy loss:

$$\ell_i^{\text{MMPO}}(\theta) = -p_i \log \sigma(\Delta_{\theta,i}) - (1 - p_i) \log \sigma(-\Delta_{\theta,i}). \tag{2}$$

The margin m_i therefore determines the target preference strength through $p_i = \sigma(\gamma m_i)$. Setting this target probability to one recovers the standard DPO loss (Rafailov et al., 2023).

4 METHOD

4.1 MOTIVATION

Missing rubrics can affect rubric-based preferences even when observed scores are accurate. In addition, human review reduces the omission phenomenon but still does not ensure complete rubric sets. We conduct motivation experiments on six rubric construction methods on RubricBench (Zhang et al., 2026), including 1,147 prompts with human-written rubrics. Each method generates a rubric set with the entire human-written rubric set

Table 1: Coverage of human-written rubrics in RubricBench.

Method	P ↑	Miss. ↓	Spur. ↓
Auto-Rubric	0.414	0.415	0.411
OpenRubrics	0.428	0.393	0.396
Rubric-ARM	0.443	0.342	0.354
OnlineRubrics	0.386	0.474	0.456
RRD	0.392	0.463	0.451
EvoLM	0.394	0.461	0.447

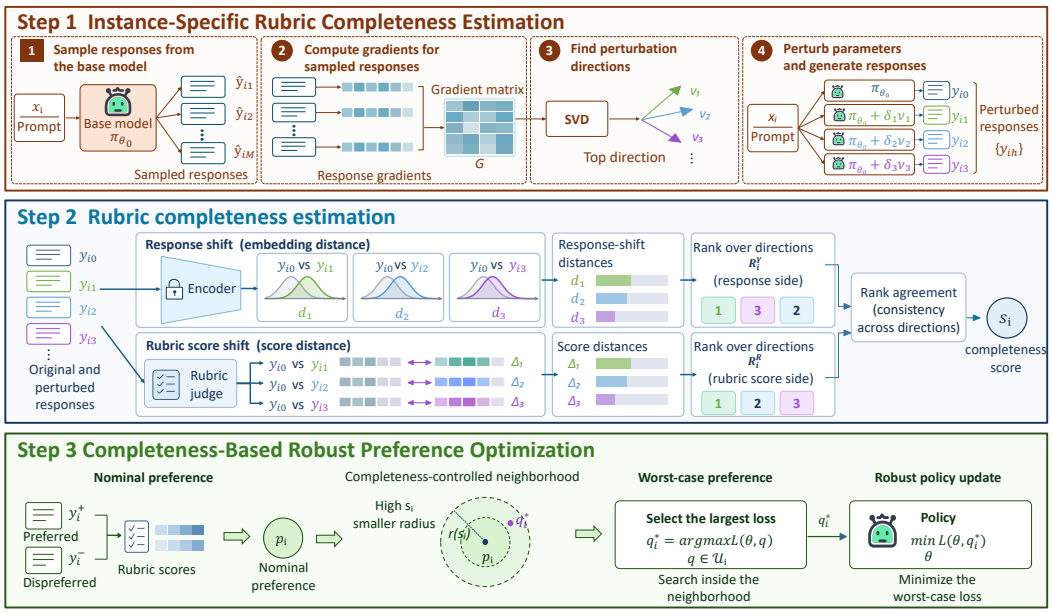


Figure 2: Overview of our framework. We first generate informative response variations through parameter perturbations, estimate instance-specific rubric completeness by comparing response and rubric-score shifts, and then perform completeness-aware robust preference optimization.

withheld. Given each instruction, an LLM judge identifies full semantic matches, partial overlaps, or no matches between generated and human-written rubrics. Precision (P) is the ratio of generated rubrics that fully match a human-written rubric to all generated rubrics. Missing rate (Miss.) is the ratio of human-written rubrics not fully matched by the generated set to all human-written rubrics. Spurious rate (Spur.) is the ratio of generated rubrics with no full or partial match in the human-written set to all generated rubrics. For example, suppose the human-written rubric set is $\{A, B, C, D\}$ and the generated set is $\{A, B, E\}$, with A and B fully matched and E unmatched. Then P is $2/3$, Miss. is $2/4 = 1/2$, and Spur. is $1/3$. More details are in Appendix A. Table 1 shows that around 34.2%–47.4% of human-written rubrics are not fully recovered across the six methods. These coverage gaps motivate accounting for possible preference errors during policy learning.

Since the missing criteria are unobserved, the completeness of observed rubrics cannot be measured directly. The core idea of our method is that if a rubric sufficiently captures the quality dimensions relevant to a sample, the changes in model responses should also be reflected in rubric score changes. In other words, we estimate rubric completeness by measuring whether larger response distribution shifts are consistent with larger changes in rubric scores, followed by a minimax robust learning algorithm, as shown in Figure 2.

4.2 Instance-Specific Rubric Completeness Estimation

Response Distribution Shifts. To calculate response distribution shifts, we first need meaningful response variations for each instance, since changing the model parameters at random may still produce very similar responses. Specifically, for each input x_i, we sample M responses $\{\widetilde{y}_{ij}\}_{j=1}^{M}$ from the original model π_{θ_0} using a high temperature. Then we use these responses to identify useful perturbation directions. Inspired by Fisher information, which describes how sensitive a model's output distribution is to parameter changes, we compute the length-normalized score gradient:

$$g_{ij} = \frac{1}{L_{ij}} \sum_{t=1}^{L_{ij}} \nabla_{\theta_{\text{sub}}} \log \pi_{\theta_0}(\widetilde{y}_{ij,t} \mid x_i, \widetilde{y}_{ij,<t}), \tag{3}$$

where L_{ij} is the response length, $\widetilde{y}_{ij,t}$ is its t-th token, and $\widetilde{y}_{ij,<t}$ denotes the preceding tokens. To reduce computation and memory costs, we restrict gradient computation and parameter perturbations to the last two Transformer blocks and the final normalization layer, these parameters are denoted

as $\theta_{\text{sub}} \subseteq \theta_0$. Thus, these gradients describe how sensitive the sampled response probabilities are under a parameter change. Next, to assess whether rubric scores reflect response changes, we need multiple pairs of perturbed responses. We therefore use these gradients to identify mutually orthogonal perturbation directions, so that we can construct these pairs by perturbing the model along distinct parameter directions. Specifically, we first stack and center them into a matrix G_i^c:

$$G_i = \begin{bmatrix} g_{i1}^\top \\ \vdots \\ g_{iM}^\top \end{bmatrix}, \qquad \bar{g}_i = \frac{1}{M}\sum_{j=1}^{M} g_{ij}, \qquad G_i^c = G_i - \mathbf{1}_M \bar{g}_i^\top. \tag{4}$$

We then apply singular value decomposition to $G_i^c = U_i \Sigma_i V_i^\top$. The leading right singular vectors provide the expected mutually orthogonal directions. We retain the first $k_i = \min(K_{\max}, r_i)$ vectors, denoted by v_{ih}, where r_i is the numerical rank of G_i^c and $K_{\max}$ is the maximum number of directions, which is a hyperparameter for controlling the computational cost. This gives k_i perturbation directions based on the M calibration responses. We now perturb the base model along each selected direction:

$$\theta_{ih} = \theta_0 + \delta_\theta v_{ih}, \qquad h = 1, \ldots, k_i, \tag{5}$$

where each v_{ih} has unit norm. We set the $\delta_\theta = c_\delta \|\theta_{\text{sub}}\|_2$, where $c_\delta > 0$ is a perturbation coefficient.

We next quantify how much each perturbation changes the model's response. Specifically, we generate a nominal response y_{i0} from π_{θ_0} with temperature 0 and a perturbed response y_{ih} from $\pi_{\theta_{ih}}$ for each direction h. Inspired by MoverScore (Zhao et al., 2019), for each response $y \in \{y_{i0}, y_{ih}\}$, we feed the instruction x_i and response y into the frozen model π_{θ_0} and extract hidden states at the response token positions. We normalize these vectors to unit length and assign equal weight to each, forming the representation distribution $\mu_i(y)$. We then compare the two distributions using the Wasserstein distance, which measures the minimum total cost of transporting mass between their token embeddings. This gives the response shift:

$$D_{ih}^Y = W_1(\mu_i(y_{i0}), \mu_i(y_{ih}))\,. \tag{6}$$

Appendix B provides the representation and distance computation details.

Rubric Score Changes. We next measure how the observed rubric scores change for each response pair. Let the vector $\mathbf{r}_i(y)$ denotes the evaluator's scores for response y under all rubrics in $\mathcal{C}_i$. The score change along direction h is

$$D_{ih}^R = \|\mathbf{r}_i(y_{ih}) - \mathbf{r}_i(y_{i0})\|_2\,.$$

Rubric Completeness Score. We then assess whether larger response shifts are accompanied by larger rubric score changes. Since these quantities have different scales, we rank the perturbation directions separately by each measure:

$$\mathcal{R}_i^Y = \text{rank}\left(D_{i1}^Y, \ldots, D_{ik_i}^Y\right), \qquad \mathcal{R}_i^R = \text{rank}\left(D_{i1}^R, \ldots, D_{ik_i}^R\right). \tag{7}$$

We measure agreement between these rankings using Kendall's rank correlation with tie correction, denoting the result by τ_i. This measures whether pairs of perturbation directions have consistent relative orders under the two rankings. A larger τ_i indicates that larger response shifts are more consistently accompanied by larger rubric score changes. We map this agreement to an instance-specific rubric completeness score:

$$s_i = s_{\min} + (1 - s_{\min})\max(0, \tau_i), \tag{8}$$

where $s_{\min} \in (0, 1)$ is a positive threshold. Nonpositive agreement receives this minimum score, while stronger positive agreement yields a higher completeness score.

4.3 Completeness-Based Robust Preference Optimization

Recall that $p_i = \sigma(\gamma m_i)$ is the nominal probability that y_i^w is preferred over y_i^l under the observed rubric set. Missing rubrics can make this probability unreliable, so we allow it to vary within a neighborhood controlled by the completeness score s_i. For n training pairs, let $\mathbf{p} = (p_1, \ldots, p_n)$

collect the nominal preference probabilities and $\mathbf{q} = (q_1, \ldots, q_n)$ denote candidate alternatives. We define the feasible set of $\mathbf{q}$ as follows:

$$\mathcal{U}(\mathbf{p}, \rho, \mathbf{s}) = \left\{ \mathbf{q} \in [\epsilon, 1-\epsilon]^n : \frac{1}{n} \sum_{i=1}^{n} s_i D(q_i, p_i) \leq \rho \right\}, \tag{9}$$

where s_i is the rubric completeness score and ρ is the robustness budget. To avoid infinity, we also clip q in our implementation. Here, $D(q_i, p_i)$ compares the Bernoulli preference distributions parameterized by q_i and p_i, using either KL divergence or Wasserstein distance. A smaller s_i means the observed rubrics contain less information, making the deviation of q_i from p_i less costly, thus allowing greater preference uncertainty for that pair, i.e., a larger $D(q_i, p_i)$.

Taking MMPO as an example, we replace its nominal target p_i with q_i in the preference loss:

$$\ell_i(\theta, q_i) = -q_i \log \sigma(\Delta_{\theta,i}) - (1 - q_i) \log \sigma(-\Delta_{\theta,i}), \tag{10}$$

where $\Delta_{\theta,i}$ is defined in Sec. 3. Detailed formulations for integrating our framework with all baselines are provided in Appendix C. Since the preferences under a complete rubric set are unknown, we train the policy against the worst-case preference probabilities in the feasible set:

$$\min_{\theta} \max_{\mathbf{q} \in \mathcal{U}(\mathbf{p}, \rho, \mathbf{s})} \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta, q_i). \tag{11}$$

The inner maximization selects $\mathbf{q}$ that maximizes the loss for the current policy, while the outer minimization updates θ to reduce this worst-case loss. The complete training procedure and optimization details are provided in Algorithm 1 and Appendix D.

5 Theoretical Analysis

In this section, we investigate whether inconsistencies in the rankings suggest that there are errors in the observed preferences. Specifically, we fix an instruction x_i and its $k_i \geq 2$ responses. Let $\mathbf{u}_i(y) \in \mathbb{R}^{d_i}$ collect the omitted rubric scores, so the complete rubric vector is $\mathbf{r}_i^\star(y) = (\mathbf{r}_i(y), \mathbf{u}_i(y))$. Its aggregate score is $r_i^\star(y) = r_i(y) + \boldsymbol{\omega}_i^\top \mathbf{u}_i(y)$, with fixed weights $\boldsymbol{\omega}_i$. For a training pair, define $\Delta\mathbf{u}_i = \mathbf{u}_i(y_i^w) - \mathbf{u}_i(y_i^l)$ and

$$m_i^\star = m_i + \boldsymbol{\omega}_i^\top \Delta\mathbf{u}_i, \qquad q_i^\star = \sigma(\gamma m_i^\star).$$

We impose two structural conditions. First, the complete rubric changes have bounded sensitivity to response changes. For constants $0 < a_i \leq b_i$, denote $D_{ih}^\star = \|\mathbf{r}_i^\star(y_{ih}) - \mathbf{r}_i^\star(y_{i0})\|_2$, we have

$$a_i D_{ih}^Y \leq D_{ih}^\star \leq b_i D_{ih}^Y. \tag{12}$$

Second, we assume that training pairs vary along every omitted-rubric direction and have bounded margins. For any unit vector $v \in \mathbb{R}^{d_i}$,

$$\mathbb{E}\big[(v^\top \Delta\mathbf{u}_i)^2\big] \geq \kappa_i \overline{U}_i, \qquad |m_i|, |m_i^\star| \leq M_i, \tag{13}$$

where $\kappa_i > 0$, $M_i < \infty$, and $\overline{U}_i = \frac{1}{k_i} \sum_h \|\mathbf{u}_i(y_{ih}) - \mathbf{u}_i(y_{i0})\|_2^2$. Based on the above, we have the following theorem:

Theorem 1 (Preference error from rubric disagreement)**.** *Suppose response shifts are distinct and rubric shifts are nonconstant. we set* $\alpha_i = \gamma\sigma(\gamma M_i)(1 - \sigma(\gamma M_i))$ *and* $C_i = \alpha_i^2 \kappa_i \|\boldsymbol{\omega}_i\|_2^2 / [4(1 - s_{\min})]$. *Let* $\mathcal{I}_i = \{(h, \ell) : D_{ih}^Y > D_{i\ell}^Y,\ D_{ih}^R \leq D_{i\ell}^R\}$ *collect the direction pairs whose orders disagree or whose rubric shifts tie. Define their separation by*

$$g_i = \min_{(h,\ell) \in \mathcal{I}_i} \left[a_i^2 (D_{ih}^Y)^2 - b_i^2 (D_{i\ell}^Y)^2 + (D_{i\ell}^R)^2 - (D_{ih}^R)^2 \right]_+, \tag{14}$$

Then the completeness score in Eq. 8 satisfies

$$\mathbb{E}\big[(q_i^\star - p_i)^2\big] \geq C_i g_i (1 - s_i). \tag{15}$$

The bound is very intuitive: the right-hand side, i.e., the preference-error lower bound, decreases as s_i increases and becomes zero at $s_i = 1$. In other words, lower completeness scores yield a larger preference-error lower bound, motivating greater robustness. Accordingly, our training algorithm based on Eq. 9 makes deviations from nominal preferences less costly for samples with smaller s_i, allocating robustness according to rubric completeness. The proof is provided in Appendix E.

Table 2: Main results on three datasets. The best and second-best results within each baseline are shown in **bold** and underlined, respectively.

Method	UltraFeedback			WildChecklists			HelpSteer2		
	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg
DPO	84.77	–	7.94	77.52	–	7.09	61.69	–	7.67
MMPO	87.46	–	8.75	78.57	–	8.05	62.58	–	7.74
+ Ours-KL	**88.26**	68.22	**8.80**	79.77	**65.10**	**8.47**	**64.86**	**66.99**	**7.83**
+ Ours-Wa	87.81	**68.86**	8.78	**79.83**	64.59	8.46	63.79	66.19	7.81
ODPO	85.59	–	8.61	77.80	–	7.15	63.60	–	7.69
+ Ours-KL	86.72	61.21	8.63	**78.62**	69.75	7.46	63.89	**64.43**	**7.96**
+ Ours-Wa	**86.80**	**64.63**	**8.65**	78.40	**71.08**	**7.84**	**63.96**	59.22	7.90
Scaled DPO	84.53	–	8.63	76.77	–	6.81	61.91	–	7.71
+ Ours-KL	85.48	75.74	8.96	**77.61**	72.69	7.02	**63.51**	62.22	7.88
+ Ours-Wa	**85.73**	**77.93**	**9.00**	77.58	**73.84**	**7.30**	62.39	**65.92**	**7.92**

6 EXPERIMENTS

We conduct experiments to answer the following research questions (RQs): **RQ1:** Can our framework consistently improve different margin-aware preference optimization methods under fixed rubrics? **RQ2:** Does our framework still work under different rubric generation methods? **RQ3:** Does our completeness score reflect the loss of rubric information? **RQ4:** How does our robust preference optimization perform on broader downstream benchmarks?

6.1 EXPERIMENTAL SETUP

Datasets. We train the policy model on three preference datasets: **UltraFeedback** (Cui et al., 2024), **WildChecklists** (Viswanathan et al., 2025), and **HelpSteer2** (Wang et al., 2024). In addition to the test sets of these datasets, we evaluate the trained policy model on six general benchmarks: **MMLU** (Hendrycks et al., 2020) for broad knowledge and problem solving, **GSM8K** (Cobbe et al., 2021) for mathematical reasoning, **BBH** (Suzgun et al., 2023) for challenging reasoning tasks, **HumanEval** (Chen et al., 2021) for code generation, **IFEval** (Zhou et al., 2023) for instruction-following ability, and **ARC-Challenge** (Clark et al., 2018) for grade-school science reasoning.

Baselines. For preference optimization method, we consider three methods that incorporate preference score margin, including MMPO (Kim et al., 2024a), ODPO (Amini et al., 2024), and Scaled DPO (Wang et al., 2025). For the rubric, We further compare with methods that improve preference supervision from the rubric side, including Auto-Rubric (Xie et al., 2025), OpenRubrics (Liu et al., 2026), and Rubric-ARM (Xu et al., 2026). We provide the experiment details in Appendix F.

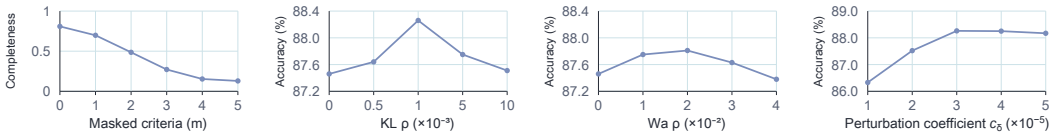
Evaluation metrics. For preference pairs, **Preference Accuracy** measures whether the policy assigns a higher likelihood ratio to the preferred response than to the rejected one. Specifically, for (x_i, y_i^w, y_i^l), a prediction is correct if $\pi_\theta(y_i^w \mid x_i)/\pi_{\text{ref}}(y_i^w \mid x_i) > \pi_\theta(y_i^l \mid x_i)/\pi_{\text{ref}}(y_i^l \mid x_i)$. For generated responses, we use rubric-based evaluation, using dataset-provided rubrics in the original setting and generated rubrics for rubric generation baselines. **Win Rate** measures the fraction of cases where our response is preferred over the corresponding baseline response, while **Avg Score** measures the quality of each response on a 0–10 scale.

6.2 OVERALL PERFORMANCE (RQ1, RQ2)

As shown in Table 2, our framework consistently improves MMPO, ODPO, and Scaled DPO across all three datasets, demonstrating its effectiveness across different preference optimization objectives. Since there are pre-defined rubrics in the datasets, we first use them to evaluate. The gains are observed in both preference accuracy and response quality, suggesting that accounting for preference errors induced by incomplete rubrics leads to better alignment for the policy model.

Table 3: Main results on three datasets. Pref. Acc. denotes preference accuracy (%), Win Rate denotes the win rate (%), and Avg denotes the average rubric score. The best and second-best results within each baseline are shown in **bold** and underlined, respectively.

Method	UltraFeedback			WildChecklists			HelpSteer2		
	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg
MMPO + Auto-Rubrics	89.97	–	8.35	88.48	–	7.19	65.66	–	7.77
+ Ours-KL	**90.91**	65.84	**8.50**	**89.80**	66.06	**7.45**	**69.84**	68.34	**7.93**
MMPO + OpenRubrics	90.31	–	8.37	87.65	–	7.32	66.38	–	7.68
+ Ours-KL	**91.85**	67.63	**8.51**	**89.13**	71.38	**7.88**	**69.52**	66.25	**8.01**
MMPO + RubricARM	91.79	–	8.34	86.55	–	8.24	66.50	–	7.35
+ Ours-KL	**92.27**	65.80	**8.59**	**88.14**	73.02	**8.62**	**69.48**	69.90	**7.62**
ODPO + Auto-Rubrics	88.47	–	8.14	88.13	–	5.29	64.47	–	7.52
+ Ours-KL	**89.34**	73.66	**8.43**	**88.33**	73.16	**5.78**	**67.29**	70.14	**7.98**
ODPO + OpenRubrics	89.51	–	8.24	84.37	–	6.79	65.37	–	7.65
+ Ours-KL	**90.09**	70.85	**8.49**	**87.65**	71.45	**7.36**	**70.02**	74.28	**7.98**
ODPO + RubricARM	90.53	–	8.15	86.68	–	6.91	66.79	–	7.33
+ Ours-KL	**91.03**	70.00	**8.42**	**87.20**	73.10	**7.47**	**70.33**	75.65	**7.84**



(a) (b) (c) (d)

Figure 3: (a) Mean completeness score under progressive criterion masking. (b–c) Sensitivity of preference accuracy to the robustness budget ρ using KL and Wasserstein distances, respectively. (d) Sensitivity of preference accuracy to the perturbation step size $c_\delta > 0$.

Table 3 further shows consistent improvements when using rubrics generated by Auto-Rubrics, OpenRubrics, and Rubric-ARM. This indicates that our method remains effective across different rubric construction pipelines. Overall, these results support our framework as a general plug-and-play approach for rubric-based preference alignment. Complete Wasserstein results and computational cost analyses are provided in Appendix G.

6.3 Effect of Missing Criteria on the Completeness Score (RQ3)

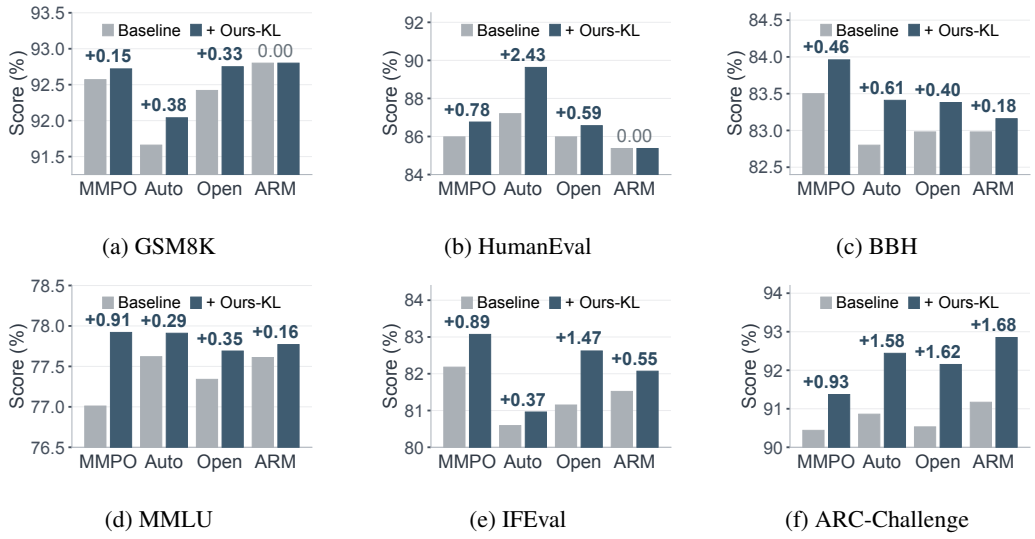
We further examine whether the completeness score s_i responds to missing rubric criteria through a controlled masking experiment on RubricBench. We select 545 prompts with at least six reference criteria and progressively mask the first $m \in \{0, 1, 2, 3, 4, 5\}$ criteria following their original order. We then recompute s_i and report its mean at each masking level. As shown in Fig. 3a, the completeness score decreases monotonically as more criteria are removed. Since all other factors remain fixed, this trend shows that s_i captures the loss of rubric information caused by missing criteria.

6.4 General Capability Evaluation (RQ4)

We further evaluate how our robust preference optimization affects the model's performance on broader benchmarks. Using MMPO with different rubric construction methods, we evaluate the trained models on six representative benchmarks. As shown in Fig. 4, our method matches or outperforms the corresponding baseline across all settings, indicating that its benefits are not limited to preference alignment but also extend to broader model capabilities.

6.5 Ablation Study and Hyperparameter Sensitivity

We conduct an ablation study on UltraFeedback using MMPO as the base preference optimization method. We compare four variants: (1) the original MMPO baseline; (2) *Uniform Robust*, which



(a) GSM8K (b) HumanEval (c) BBH

(d) MMLU (e) IFEval (f) ARC-Challenge

Figure 4: General capability evaluation on six benchmarks.

performs the same robust optimization while setting $s_i = 1$ for all samples; (3) *Shuffled-s*, which preserves the estimated completeness scores but randomly permutes their assignments across training samples; and (4) the full method, which uses the estimated s_i for each corresponding instance.

As shown in Table 4, uniform robust optimization improves preference accuracy, indicating that accounting for possible deviations from the nominal rubric-derived preferences is beneficial. The full method further improves the performance, showing that adapting the preference neighborhood according to rubric completeness provides additional gains. In contrast, shuffling the completeness scores decreases preference accuracy. The result indicates that correctly identifying which samples have less complete rubric is important for effective robust preference optimization.

Table 4: Ablation study on UltraFeedback.

Variant	Pref. Acc. (%) ↑
MMPO	87.46
+ Uniform Robust	87.78
+ Shuffled-s	87.14
+ Ours	**88.26**

We further analyze the sensitivity to the robustness budget ρ and perturbation coefficient c_δ on UltraFeedback using MMPO + Ours. For KL divergence, we use $\rho \in \{0,\ 0.0005,\ 0.001,\ 0.005,\ 0.01\}$, while for Wasserstein distance, we use $\rho \in \{0,\ 0.01,\ 0.02,\ 0.03,\ 0.04\}$. As shown in Fig. 3b and 3c, performance peaks at $\rho = 0.001$ for KL and $\rho = 0.02$ for Wasserstein, while larger budgets lead to moderate degradation. For the perturbation magnitude $\delta_\theta = c_\delta \|\theta_{\text{sub}}\|_2$, Fig. 3d shows that performance improves up to $c_\delta = 3 \times 10^{-5}$ and remains stable thereafter, indicating limited sensitivity within a reasonable range.

7 Conclusion

We studied preference alignment under incomplete rubrics and proposed a robust optimization framework that accounts for the error in preference supervision. Our method probes response variations along gradient-informed directions and measures their agreement with rubric score changes to estimate instance-specific rubric completeness. This estimate controls a neighborhood of plausible preference probabilities, assigning greater flexibility to samples with lower completeness. The policy then minimizes the worst-case loss within this neighborhood. Experiments across three datasets and multiple rubric construction pipelines demonstrate consistent improvements when integrating our framework with existing margin-aware preference optimization methods. These results support accounting for residual rubric incompleteness alongside improving rubric construction. A possible limitation is the preprocessing cost of response probing, motivating more efficient probe selection.

AI USE STATEMENT

We used large language model (LLM) tools to edit and polish parts of the manuscript, format tables, and draft and revise scripts for experimental figures. The authors reviewed and verified all AI-assisted text, code, figures, and interpretations, and take responsibility for the final content.

ETHICS STATEMENT

This study uses publicly available datasets and previously collected human ratings and involves no new human-subject data collection. Rubric-derived preference supervision and automatic evaluations may inherit biases or errors from the source datasets and evaluator models, which should be considered when applying the method to other settings.

REPRODUCIBILITY STATEMENT

A link to the anonymized codebase is provided in the abstract. Section 4 describes the proposed method. Appendix F provides the implementation and experimental details, Appendix D describes the complete algorithm and robust optimization procedure, and the remaining appendices provide the data analysis details and theoretical proofs.

REFERENCES

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9954–9972, 2024.

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.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. URL https://arxiv.org/abs/2212.08073.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: boosting language

models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yi Dong, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. SteerLM: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11275–11288. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.754. URL https://aclanthology.org/2023.findings-emnlp.754/.

Zhiyuan Fan, Weinong Wang, Xing Wu, and Debing Zhang. SedarEval: Automated evaluation using self-adaptive rubrics. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16916–16930, 2024.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Kyuyoung Kim, Ah Jeong Seo, Hao Liu, Jinwoo Shin, and Kimin Lee. Margin matching preference optimization: Enhanced model alignment with granular feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13554–13570, 2024a.

Seungone Kim, Jay Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Ryan S Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, volume 2024, pp. 29927–29962, 2024b.

Shuyue Stella Li, Rui Xin, Teng Xiao, Yike Wang, Rulin Shao, Zoey Hao, Melanie Sclar, Sewoong Oh, Faeze Brahman, Pang Wei Koh, et al. EvoLM: Self-evolving language models through co-evolved discriminative rubrics. *arXiv preprint arXiv:2605.03871*, 2026.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. WildBench: Benchmarking LLMs with challenging tasks from real users in the wild. In *International Conference on Learning Representations*, volume 2025, pp. 47852–47870, 2025.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437, 2026.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

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

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BjypaSs3sS.

William F Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks. *arXiv preprint arXiv:2602.05125*, 2026.

Clemencia Siro, Pourya Aliannejadi, and Mohammad Aliannejadi. Learning to judge: LLMs designing and applying evaluation rubrics. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6371–6389, 2026.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. *Advances in Neural Information Processing Systems*, 38:114728–114754, 2025.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. HelpSteer2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501, 2024.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-Preference: Complementing ratings with preferences. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.01257.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59008–59033. Curran Associates, Inc., 2023. doi: 10.52202/075280-2574. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b8c90b65739ae8417e61eadb521f63d5-Abstract.html.

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 from implicit weights to explicit rubrics for reward modeling. *arXiv preprint arXiv:2510.17314*, 2025.

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

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations*, volume 2024, pp. 55361–55414, 2024.

Qiyuan Zhang, Junyi Zhou, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, et al. RubricBench: Aligning model-generated rubrics with human standards. *arXiv preprint arXiv:2603.01562*, 2026.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 563–578, 2019.

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

A Additional Details of the Semi-Synthetic RubricBench Analysis

A.1 Criterion Matching

For each prompt x_i, let

$$\mathcal{R}_i = \{r_{i1}, \ldots, r_{iN_i}\}, \qquad \widehat{\mathcal{R}}_i = \{\hat{r}_{i1}, \ldots, \hat{r}_{iM_i}\} \tag{16}$$

denote the human-written reference rubric from RubricBench and the rubric produced by an automatic construction method, respectively. The reference rubric is used only for evaluation and is withheld during rubric construction. We apply the same criterion-matching and metric computation protocol to all six methods.

The judge receives the task instruction, the generated criteria, and the reference criteria. For each generated criterion $\hat{r}_{ik}$, it returns two lists of reference indices: `exact_matches` and `partial_matches`. These lists implement the following distinctions:

- **Full Match** (`exact_matches`): the criteria test the same behavioral requirement and have essentially the same practical pass/fail boundary. This is semantic equivalence, not a requirement for identical wording.
- **Partial Match** (`partial_matches`): the criteria concern the same behavioral requirement, but one is a strict subset or superset of the other, or they have material overlap without the same pass/fail boundary.
- **Unmatched**: a generated criterion has neither a full nor a partial match to any reference criterion; both returned lists are empty.

The matching instruction explicitly rejects topic overlap, entailment, general relevance, or a shared mention of correctness as sufficient grounds for a match. It allows logical complements or opposite-polarity formulations to match the same behavioral dimension. Thus, matching assesses criterion coverage rather than agreement in reward polarity.

Matching is not constrained to be one-to-one. A generated criterion may match multiple reference criteria, and multiple generated criteria may match the same reference criterion. Full and partial match lists must be disjoint for each generated criterion. We retain the returned matches without a one-to-one assignment or a tie-breaking step. Each generated criterion is counted at most once in precision, and each reference criterion is counted at most once in recall.

We require a valid JSON response with exactly one entry per generated criterion, valid reference indices, and disjoint full and partial match lists. The `unsupported_extra` flag is derived from whether both lists are empty. Failed batches can be retried with smaller groups of generated criteria while retaining the full reference rubric; the resulting judgments are assembled by candidate index without duplicate entries.

Judge prompt. The following is the original full-batch user-message template, with instance-specific values replaced by placeholders and line breaks added for readability. No separate system message is supplied. Here, <M> and <N> are the numbers of generated and reference criteria. The implementation uses zero-based indices, whereas the equations use one-based indices.

```
There are exactly <M> candidate criteria indexed 0 through <M-1>, and
exactly <N> reference criteria indexed 0 through <N-1>. Match each
candidate only to reference criteria that test the same behavioral
requirement. A logical complement or opposite polarity may match the same
dimension. Topic overlap, entailment, general relevance, or both
mentioning correctness is not a match. Use exact_matches only when the
practical pass/fail boundary is essentially the same; use partial_matches
for a strict subset, strict superset, or material overlap. Usually both
lists are sparse. Set unsupported_extra=true exactly when both match
lists are empty. Return JSON only as {"matches":[{"candidate_index":0,"e⌋
xact_matches":[],"partial_matches":[],"unsupported_extra":true}, ...]}.
The matches array must have exactly <M> items, one for every candidate in
order. Every match-list integer must be between 0 and <N-1>.
```

```
Instruction:
<instruction>

Candidate criteria:
[0] <generated criterion 0>
...
[<M-1>] <generated criterion M-1>

Reference criteria:
[0] <reference criterion 0>
...
[<N-1>] <reference criterion N-1>
```

For a repair batch, the prompt explicitly lists the candidate indices included in that batch and requests one output entry for each listed index; the semantic matching rules and the complete reference list are unchanged.

A.2 Metric Definitions

Let $\mathcal{F}_{ik}$ and $\mathcal{P}_{ik}$ denote the sets of reference indices returned as full and partial matches for $\hat{r}_{ik}$, respectively. All rubrics are counted equally, regardless of any weights assigned by the rubric construction method. Precision and recall count only full matches; partial matches receive no fractional credit.

For each prompt, precision is the fraction of generated rubrics with at least one full reference match:

$$P_i = \frac{1}{M_i} \sum_{k=1}^{M_i} \mathbb{I}[\mathcal{F}_{ik} \neq \varnothing] \,. \tag{17}$$

Recall is the fraction of distinct reference rubrics with at least one full match from the generated rubric set:

$$R_i = \frac{1}{N_i} \left| \bigcup_{k=1}^{M_i} \mathcal{F}_{ik} \right| . \tag{18}$$

Because matching permits many-to-many correspondences, precision measures the proportion of generated rubrics supported by a full reference match; it does not separately penalize multiple generated rubrics that fully match the same reference rubric.

The missing rate is

$$D_i = 1 - R_i. \tag{19}$$

It measures reference rubrics that are not fully recovered, including rubrics that are only partially matched. It therefore does not imply that all such rubrics have no semantic overlap with the generated rubric set.

The spurious rate is the fraction of generated rubrics with neither a full nor a partial reference match:

$$S_i = \frac{1}{M_i} \sum_{k=1}^{M_i} \mathbb{I}[\mathcal{F}_{ik} \cup \mathcal{P}_{ik} = \varnothing] \,. \tag{20}$$

Unlike $1 - P_i$, this quantity excludes rubrics with partial matches only. Such rubrics contribute zero to precision but are not counted as spurious. Here, "spurious" means unsupported by the reference rubric set; it does not establish that a rubric is intrinsically invalid or irrelevant to quality.

Aggregation across prompts. For each method, let $\mathcal{I}$ be its set of prompts with complete, valid matching outputs. We compute the three metrics within each prompt and then take unweighted means over $\mathcal{I}$:

$$\mathrm{P} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} P_i, \qquad \mathrm{Miss.} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} D_i, \qquad \mathrm{Spur.} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} S_i. \tag{21}$$

These are prompt-level macro-averages, not ratios obtained by pooling rubrics across prompts. Table 1 reports these three metrics; recall is omitted because it equals one minus the missing rate.

B Response Distance Computation

We compute the distance between the nominal response y_{i0} and each perturbed response y_{ih} in a shared representation space. We use the frozen original model π_{θ_0} as the encoder for both responses. Given the prompt x_i and a response y, we extract hidden states at the response token positions and normalize each hidden state to unit length, denoting the resulting vector by $e_{it}(y)$. For a response of length L_y, its representation distribution is

$$\mu_i(y) = \frac{1}{L_y} \sum_{t=1}^{L_y} \delta_{e_{it}(y)}, \tag{22}$$

where δ_e is a unit point mass at e. Thus, each token representation receives equal mass. Following the use of contextual token representations for transport-based text comparison (Zhao et al., 2019), we use the Euclidean distance between token embeddings as the transport cost:

$$c_{ab} = \|e_{ia}(y_{i0}) - e_{ib}(y_{ih})\|_2 . \tag{23}$$

The Wasserstein distance in Eq. 6 is computed as

$$D_{ih}^{Y} = \min_{\Gamma \geq 0} \sum_{a=1}^{L_{i0}} \sum_{b=1}^{L_{ih}} \Gamma_{ab} c_{ab}, \tag{24}$$

subject to

$$\sum_{b=1}^{L_{ih}} \Gamma_{ab} = \frac{1}{L_{i0}}, \qquad \sum_{a=1}^{L_{i0}} \Gamma_{ab} = \frac{1}{L_{ih}}, \tag{25}$$

where L_{i0} and L_{ih} are the response lengths and Γ_{ab} specifies the mass transported between the two token representations. The minimum total cost measures their difference in the fixed representation space.

C Integration with Margin-Aware Preference Optimization

Our framework operates on the preference probability induced by the observed rubric and is therefore independent of how a downstream preference optimization method incorporates preference strength. For each preference pair (x_i, y_i^w, y_i^l), recall that the observed rubric induces the margin $m_i = r_i(y_i^w) - r_i(y_i^l)$, which is mapped to the nominal preference probability

$$p_i = \sigma(\gamma m_i). \tag{26}$$

Our robust optimization replaces p_i by an alternative preference probability q_i selected adversarially from the completeness-aware preference neighborhood defined in Sec. 4.

For methods that directly operate on a preference probability, such as MMPO, q_i can be used directly. For methods that instead consume a preference margin or magnitude, such as ODPO and Scaled DPO, we map q_i back to the corresponding signed margin using the inverse of the Bradley–Terry mapping:

$$\widetilde{m}_i(q_i) = \frac{1}{\gamma} \operatorname{logit}(q_i) = \frac{1}{\gamma} \log \frac{q_i}{1 - q_i}. \tag{27}$$

Unlike the nominal rubric margin, the robust margin can change sign when q_i crosses $1/2$. We therefore decompose it into a preference direction and a preference strength:

$$a_i(q_i) = \operatorname{sign}\left(q_i - \frac{1}{2}\right), \qquad u_i(q_i) = |\widetilde{m}_i(q_i)| . \tag{28}$$

Here, $a_i(q_i) \in \{-1, +1\}$ determines which response is preferred under the alternative preference distribution, while $u_i(q_i) \geq 0$ measures the corresponding preference strength. When $q_i < 1/2$, the alternative preference reverses the ordering implied by (y_i^w, y_i^l). This reversal can be implemented without explicitly swapping the two responses by multiplying the policy reward difference by $a_i(q_i)$:

$$\widetilde{\Delta}_{\theta,i}(q_i) = a_i(q_i) \Delta_{\theta,i}, \tag{29}$$

where

$$\Delta_{\theta,i} = \beta \log \frac{\pi_\theta(y_i^w \mid x_i)}{\pi_{\text{ref}}(y_i^w \mid x_i)} - \beta \log \frac{\pi_\theta(y_i^l \mid x_i)}{\pi_{\text{ref}}(y_i^l \mid x_i)}. \tag{30}$$

Thus, the adversarial preference probability determines both the direction $a_i(q_i)$ and the magnitude $u_i(q_i)$ of the preference signal.

C.1 INTEGRATION WITH MMPO

MMPO directly matches the model-implied preference probability to the rubric-derived target probability. Its nominal loss can be written as

$$\ell_i^{\mathrm{MMPO}}(\theta) = -p_i \log \sigma(\Delta_{\theta,i}) - (1-p_i)\log \sigma(-\Delta_{\theta,i}). \tag{31}$$

Since MMPO already operates on a soft preference probability, integrating our framework only requires replacing p_i with the adversarial preference probability q_i:

$$\ell_i^{\text{R-MMPO}}(\theta, q_i) = -q_i \log \sigma(\Delta_{\theta,i}) - (1-q_i)\log \sigma(-\Delta_{\theta,i}). \tag{32}$$

This formulation naturally accounts for preference reversal. In particular, when $q_i < 1/2$, the second term receives larger weight, encouraging the policy to favor the response originally denoted by y_i^l.

The corresponding robust objective is

$$\min_{\theta} \max_{\mathbf{q}\in\mathcal{U}(\mathbf{p},\rho,\mathbf{s})} \frac{1}{n}\sum_{i=1}^{n} \ell_i^{\text{R-MMPO}}(\theta, q_i). \tag{33}$$

C.2 INTEGRATION WITH ODPO

ODPO (Amini et al., 2024) incorporates preference strength by introducing a pair-dependent offset into the DPO objective. Let $f(\cdot)$ denote the margin-to-offset transformation used by ODPO and α its scaling coefficient. Given a non-negative preference magnitude u_i, its loss takes the form

$$\ell_i^{\mathrm{ODPO}}(\theta) = -\log \sigma\left(\Delta_{\theta,i} - \alpha f(u_i)\right). \tag{34}$$

A direct substitution of the signed robust margin $\widetilde{m}_i(q_i)$ into the offset would be inappropriate when the adversarial preference is reversed, because the ODPO offset represents preference strength rather than preference direction. We therefore use $a_i(q_i)$ to determine the direction of the comparison and $u_i(q_i) = |\widetilde{m}_i(q_i)|$ to determine the offset magnitude. The resulting robust ODPO loss is

$$\ell_i^{\text{R-ODPO}}(\theta, q_i) = -\log \sigma\left(a_i(q_i)\Delta_{\theta,i} - \alpha f(u_i(q_i))\right). \tag{35}$$

When $q_i > 1/2$, $a_i(q_i) = +1$ and the loss preserves the original preference direction. When $q_i < 1/2$, $a_i(q_i) = -1$, so that $a_i(q_i)\Delta_{\theta,i} = -\Delta_{\theta,i}$, which is equivalent to exchanging the roles of y_i^w and y_i^l. Meanwhile, the offset remains non-negative and depends only on the strength of the alternative preference. Therefore, our robustification modifies the preference signal while preserving the original ODPO mechanism.

The full objective becomes

$$\min_{\theta} \max_{\mathbf{q}\in\mathcal{U}(\mathbf{p},\rho,\mathbf{s})} \frac{1}{n}\sum_{i=1}^{n} \ell_i^{\text{R-ODPO}}(\theta, q_i). \tag{36}$$

C.3 INTEGRATION WITH SCALED DPO

Scaled DPO (Wang et al., 2025) uses preference magnitude to reweight the contribution of each response pair. Let $g(\cdot)$ denote the scaling function used by the original method. Its nominal objective can be expressed as

$$\ell_i^{\mathrm{ScaledDPO}}(\theta) = -g(u_i)\log \sigma(\Delta_{\theta,i}), \tag{37}$$

where u_i denotes the preference strength. In the direct scaling case, $g(u) = u$.

For the adversarial preference q_i, we again separate preference direction from preference magnitude. The preference strength is replaced by $u_i(q_i) = |\widetilde{m}_i(q_i)|$, while the ordering of the response pair is determined by $a_i(q_i)$. This gives

$$\ell_i^{\text{R-ScaledDPO}}(\theta, q_i) = -g(u_i(q_i))\log \sigma\left(a_i(q_i)\Delta_{\theta,i}\right). \tag{38}$$

Therefore, q_i affects Scaled DPO in two complementary ways. Its distance from $1/2$ determines the strength assigned to the response pair through $u_i(q_i)$, while its side of $1/2$ determines the preference

Algorithm 1 Completeness-Aware Robust Preference Optimization

Require: Preference dataset $\mathcal{D} = \{(x_i, y_i^w, y_i^l, C_i)\}_{i=1}^n$; base policy π_{θ_0}; rubric evaluator v_ψ; robustness budget ρ; distance measure D
Ensure: Optimized policy π_θ

Stage I: Rubric completeness estimation
for each instance i **do**
  Sample calibration responses from π_{θ_0}
  Compute response gradients and obtain perturbation directions via SVD
  Generate nominal and perturbed responses
  Compute response shifts $\{D_{ih}^Y\}$ and rubric-score shifts $\{D_{ih}^R\}$
  Compute rank agreement τ_i and completeness score

$$s_i = s_{\min} + (1 - s_{\min}) \max(0, \tau_i)$$

  Compute nominal preference probability

$$p_i = \sigma(\gamma m_i)$$

end for

Stage II: Robust preference optimization
Initialize $\theta \leftarrow \theta_0$
while training **do**
  Find the worst-case preference probabilities

$$\mathbf{q}^\star(\theta) = \arg \max_{\mathbf{q} \in \mathcal{U}_D(\mathbf{p}, \rho, \mathbf{s})} \mathcal{L}(\theta, \mathbf{q})$$

  Update the policy using the worst-case preferences

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta, \mathbf{q}^\star(\theta))$$

end while
return π_θ

direction through $a_i(q_i)$. In particular, an adversarial preference crossing $q_i = 1/2$ reverses the ordering rather than merely changing the weight of the original preference.

The resulting robust objective is

$$\min_\theta \max_{\mathbf{q} \in \mathcal{U}(\mathbf{p}, \rho, \mathbf{s})} \frac{1}{n} \sum_{i=1}^n \ell_i^{\text{R-ScaledDPO}}(\theta, q_i). \tag{39}$$

Overall, the three integrations differ only in how the alternative preference probability q_i is consumed by the base optimization objective. MMPO uses q_i directly as a soft preference target, whereas ODPO and Scaled DPO first recover its signed margin through Eq. 27 and separate preference direction from strength through Eq. 28. The completeness-aware uncertainty set and the adversarial optimization over $\mathbf{q}$ remain unchanged across the three methods.

D Algorithm and Optimization Details

D.1 End-to-End Algorithm

Algorithm 1 summarizes the overall procedure of our framework. The completeness scores are estimated before policy training, while the worst-case preference probabilities are optimized with respect to the current policy during training.

D.2 KL-Divergence Preference Neighborhood

For the KL-based variant, we regard the nominal and alternative preference distributions of the i-th pair as

$$P_i = \text{Bernoulli}(p_i), \qquad Q_i = \text{Bernoulli}(q_i). \tag{40}$$

We use the KL divergence from the alternative preference distribution to the nominal distribution:

$$D_{\text{KL}}(Q_i \| P_i) = q_i \log \frac{q_i}{p_i} + (1 - q_i) \log \frac{1 - q_i}{1 - p_i}. \tag{41}$$

The corresponding uncertainty set is

$$\mathcal{U}_{\text{KL}}(\mathbf{p}, \rho, \mathbf{s}) = \left\{ \mathbf{q} \in [\epsilon, 1 - \epsilon]^n : \frac{1}{n} \sum_{i=1}^{n} s_i D_{\text{KL}}(Q_i \| P_i) \leq \rho \right\}. \tag{42}$$

Because the divergence is weighted by s_i, changing the preference probability of an instance with low rubric completeness incurs a lower robustness cost. Consequently, the adversary can allocate more of the global robustness budget to instances whose observed rubrics provide less reliable preference supervision.

The constrained inner maximization can equivalently be written through its Lagrangian. For $\lambda \geq 0$,

$$\max_{\mathbf{q} \in [\epsilon, 1-\epsilon]^n} \frac{1}{n} \sum_{i=1}^{n} \left[\ell_i^B(\theta, q_i) - \lambda s_i D_{\text{KL}}(Q_i \| P_i) \right] + \lambda \rho. \tag{43}$$

For a fixed λ, the inner problem is separable across instances:

$$q_i^\star(\theta, \lambda) \in \arg\max_{q_i \in [\epsilon, 1-\epsilon]} \left\{ \ell_i^B(\theta, q_i) - \lambda s_i D_{\text{KL}}(Q_i \| P_i) \right\}. \tag{44}$$

The single global multiplier λ controls the total deviation and is chosen such that the resulting solution satisfies the global robustness budget.

For MMPO, the dependence on q_i is linear, allowing the inner solution to be written explicitly. Define

$$\ell_i^+(\theta) = -\log \sigma(\Delta_{\theta,i}), \qquad \ell_i^-(\theta) = -\log \sigma(-\Delta_{\theta,i}). \tag{45}$$

Then

$$\ell_i^{\text{R-MMPO}}(\theta, q_i) = q_i \ell_i^+(\theta) + (1 - q_i) \ell_i^-(\theta). \tag{46}$$

Solving Eq. 44 gives

$$q_i^\star = \sigma \left(\text{logit}(p_i) + \frac{\ell_i^+(\theta) - \ell_i^-(\theta)}{\lambda s_i} \right). \tag{47}$$

Since

$$\ell_i^+(\theta) - \ell_i^-(\theta) = -\Delta_{\theta,i}, \tag{48}$$

Eq. 47 can also be written as

$$q_i^\star = \sigma \left(\text{logit}(p_i) - \frac{\Delta_{\theta,i}}{\lambda s_i} \right). \tag{49}$$

This expression illustrates the interaction between rubric completeness and the adversarial preference correction. For the same policy loss, a smaller s_i permits a larger movement away from p_i, whereas highly complete rubrics constrain $q_i^\star$ more strongly around the nominal preference.

For ODPO and Scaled DPO, q_i enters the loss through the nonlinear inverse mapping in Eq. 27. Their inner maximization therefore follows the general form in Eq. 44, while the uncertainty set and global robustness budget remain identical.

D.3 Wasserstein Preference Neighborhood

We additionally consider the Wasserstein distance as an alternative geometry for the preference neighborhood. Since P_i and Q_i are Bernoulli distributions supported on the two preference outcomes and the transportation cost between the two outcomes is one, their first-order Wasserstein distance reduces to

$$W_1(Q_i, P_i) = |q_i - p_i|. \tag{50}$$

The Wasserstein uncertainty set is therefore

$$\mathcal{U}_{\mathrm{Wa}}(\mathbf{p}, \rho, \mathbf{s}) = \left\{ \mathbf{q} \in [\epsilon, 1-\epsilon]^n : \frac{1}{n} \sum_{i=1}^{n} s_i |q_i - p_i| \le \rho \right\}. \tag{51}$$

Its corresponding Lagrangian inner problem is

$$\max_{\mathbf{q} \in [\epsilon, 1-\epsilon]^n} \frac{1}{n} \sum_{i=1}^{n} \left[\ell_i^B(\theta, q_i) - \lambda s_i |q_i - p_i| \right] + \lambda \rho. \tag{52}$$

As in the KL case, the global constraint couples the instances only through the shared robustness budget, while for a fixed λ the maximization decomposes into independent scalar problems:

$$q_i^\star(\theta, \lambda) \in \arg\max_{q_i \in [\epsilon, 1-\epsilon]} \left\{ \ell_i^B(\theta, q_i) - \lambda s_i |q_i - p_i| \right\}. \tag{53}$$

Therefore, both KL- and Wasserstein-based variants share the same robust preference optimization framework and differ only in how deviations from the nominal preference distribution are measured.

D.4 Relationship to the Nominal Objectives

The original margin-aware preference optimization objectives are recovered as a special case of our framework. When $\rho = 0$, the uncertainty set collapses to the nominal preference distribution,

$$\mathcal{U}_D(\mathbf{p}, 0, \mathbf{s}) = \{\mathbf{p}\}, \tag{54}$$

and therefore

$$q_i = p_i, \qquad \widetilde{m}_i(q_i) = \frac{1}{\gamma} \operatorname{logit}(p_i) = m_i. \tag{55}$$

Thus, MMPO recovers its original soft preference target, while ODPO and Scaled DPO recover the preference direction and magnitude induced by the observed rubric margin. Increasing ρ expands the set of plausible preferences and allows the adversary to correct the supervision more strongly, particularly for instances with lower rubric completeness.

E Proof of Theorem 1

Proof. Fix i and its probes; expectations are over training response pairs for x_i, and the margin bound in Eq. 13 holds almost surely. Abbreviate $k = k_i$, $Y_h = D_{ih}^Y$, $R_h = D_{ih}^R$, and $U_h = \|\mathbf{u}_i(y_{ih}) - \mathbf{u}_i(y_{i0})\|_2^2$. Because the complete rubric vector concatenates the observed and omitted rubric scores,

$$(D_{ih}^\star)^2 = R_h^2 + U_h. \tag{56}$$

For every $(h, \ell) \in \mathcal{I}_i$, Eq. 12 therefore gives

$$\begin{aligned} U_h - U_\ell &= (D_{ih}^\star)^2 - (D_{i\ell}^\star)^2 + R_\ell^2 - R_h^2 \\ &\ge a_i^2 Y_h^2 - b_i^2 Y_\ell^2 + R_\ell^2 - R_h^2. \end{aligned} \tag{57}$$

Since $U_h, U_\ell \ge 0$, it follows that

$$U_h \ge \left[a_i^2 Y_h^2 - b_i^2 Y_\ell^2 + R_\ell^2 - R_h^2 \right]_+ \ge g_i.$$

Each index h appears as the first coordinate of at most $k-1$ pairs in $\mathcal{I}_i$. Summing over these pairs yields

$$g_i |\mathcal{I}_i| \le \sum_{(h,\ell) \in \mathcal{I}_i} U_h \le (k-1) \sum_{h=1}^{k} U_h, \qquad \overline{U}_i \ge \frac{g_i |\mathcal{I}_i|}{k(k-1)}. \tag{58}$$

To relate this count to the implemented score, let $N = k(k-1)/2$. Among the N unordered direction pairs, denote by n_c, n_d, and n_t the numbers that are concordant, discordant, and tied in rubric shift, respectively. The response shifts are distinct, so $N = n_c + n_d + n_t$ and $|\mathcal{I}_i| = n_d + n_t$. The rubric shifts are nonconstant, hence $n_c + n_d > 0$ and Kendall's tie-corrected coefficient is

$$\tau_i = \frac{n_c - n_d}{\sqrt{N(n_c + n_d)}}. \tag{59}$$

If $\tau_i \geq 0$, then $\sqrt{N(n_c + n_d)} \leq N$ gives

$$1 - \tau_i \leq 1 - \frac{n_c - n_d}{N} = \frac{2n_d + n_t}{N} \leq \frac{2|\mathcal{I}_i|}{N}.$$

If $\tau_i < 0$, then $n_d > n_c$, so $2|\mathcal{I}_i| = 2n_d + 2n_t \geq n_c + n_d + n_t = N$. Thus, in both cases,

$$|\mathcal{I}_i| \geq \frac{N}{2}\big(1 - \max(0, \tau_i)\big).$$

Combining this inequality with Eq. 58 and the definition of s_i gives

$$\overline{U}_i \geq \frac{g_i}{4}\big(1 - \max(0, \tau_i)\big) = \frac{g_i(1 - s_i)}{4(1 - s_{\min})}. \tag{60}$$

The same inequality holds when $\mathcal{I}_i$ is empty, since $g_i = 0$.

For a training response pair, the additive score model implies $m_i^\star - m_i = \boldsymbol{\omega}_i^\top \Delta\mathbf{u}_i$. Scaling the unit-vector bound in Eq. 13 gives

$$\begin{aligned}\mathbb{E}[(m_i^\star - m_i)^2] &= \mathbb{E}\big[(\boldsymbol{\omega}_i^\top \Delta\mathbf{u}_i)^2\big] \\ &\geq \kappa_i \overline{U}_i \|\boldsymbol{\omega}_i\|_2^2.\end{aligned} \tag{61}$$

Finally, for every $t \in [-M_i, M_i]$,

$$\frac{d}{dt}\sigma(\gamma t) = \gamma\sigma(\gamma t)(1 - \sigma(\gamma t)) \geq \gamma\sigma(\gamma M_i)(1 - \sigma(\gamma M_i)) = \alpha_i.$$

Both margins lie in this interval, so the mean value theorem gives $|q_i^\star - p_i| \geq \alpha_i |m_i^\star - m_i|$. Squaring, taking expectations, and applying Eqs. 60 and 61 yields

$$\mathbb{E}[(q_i^\star - p_i)^2] \geq \alpha_i^2 \kappa_i \|\boldsymbol{\omega}_i\|_2^2 \overline{U}_i \geq \frac{\alpha_i^2 \kappa_i \|\boldsymbol{\omega}_i\|_2^2}{4(1 - s_{\min})} g_i(1 - s_i) = C_i g_i (1 - s_i),$$

which proves the theorem. □

F Experiment Details

We use Qwen3-8B as the initial policy and the fixed reference model. We fine-tune all policy parameters for one epoch using AdamW, with a learning rate of 10^{-6} and an effective batch size of 64. We use a cosine learning rate schedule with a warmup ratio of 0.3. We set $\beta = 0.01$, following Kim et al. (2024a), and use $\gamma = 1.1$ in our experiments. The random seed is 42 and the judge model is DeepSeek V4. Training runs on four NVIDIA H100 GPUs. For rubric completeness estimation, we sample $M = 16$ calibration responses at a temperature of 2.5. We retain up to $K_{\max} = 8$ perturbation directions and set $s_{\min} = 0.05$. We set the perturbation coefficient to $c_\delta = 3 \times 10^{-5}$. The robustness budget is $\rho = 0.001$ for KL divergence and $\rho = 0.02$ for Wasserstein distance. We set the preference-probability clipping threshold to $\epsilon = 10^{-6}$ in all experiments.

G Additional Experiment Results and Computational Cost

G.1 Additional Results with Wasserstein Distance

Table 5 reports the complete results of the Wasserstein-based variant when combined with different rubric generation methods. These results complement the KL-based results reported in the main paper.

Table 5: Additional results with KL- and Wasserstein-based robust preference optimization under automatically constructed rubrics. Pref. Acc. denotes preference accuracy (%), Win Rate denotes the win rate (%) against the corresponding baseline, and Avg denotes the average response quality score. The best and second-best results within each baseline group are shown in **bold** and underlined, respectively.

	UltraFeedback			WildChecklists			HelpSteer2		
Method	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg	Pref. Acc.	Win Rate	Avg
MMPO + Auto-Rubrics	89.97	–	8.35	88.48	–	7.19	65.66	–	7.77
+ Ours-KL	**90.91**	**65.84**	**8.50**	**89.80**	66.06	7.45	**69.84**	**68.34**	**7.93**
+ Ours-Wa	90.89	65.55	8.48	89.64	**67.92**	**7.52**	68.41	65.37	7.88
MMPO + OpenRubrics	90.31	–	8.37	87.65	–	7.32	66.38	–	7.68
+ Ours-KL	**91.85**	**67.63**	**8.51**	**89.13**	71.38	7.88	**69.52**	**66.25**	**8.01**
+ Ours-Wa	91.15	65.79	8.50	88.46	**72.16**	**7.90**	68.42	66.17	8.00
MMPO + Rubric-ARM	91.79	–	8.34	86.55	–	8.24	66.50	–	7.35
+ Ours-KL	92.27	65.80	8.59	**88.14**	73.02	**8.62**	**69.48**	**69.90**	**7.62**
+ Ours-Wa	**92.38**	**66.86**	**8.62**	88.04	**73.22**	8.47	67.52	68.87	7.55
ODPO + Auto-Rubrics	88.47	–	8.14	88.13	–	5.29	64.47	–	7.52
+ Ours-KL	**89.34**	**73.66**	8.43	88.33	**73.16**	**5.78**	**67.29**	70.14	**7.98**
+ Ours-Wa	88.60	70.26	**8.44**	**88.43**	72.15	5.75	66.59	**71.42**	7.95
ODPO + OpenRubrics	89.51	–	8.24	84.37	–	6.79	65.37	–	7.65
+ Ours-KL	**90.09**	**70.85**	8.49	**87.65**	**71.45**	7.36	**70.02**	**74.28**	**7.98**
+ Ours-Wa	**90.09**	69.75	**8.51**	87.59	70.53	**7.42**	69.37	73.32	7.86
ODPO + Rubric-ARM	90.53	–	8.15	86.68	–	6.91	66.79	–	7.33
+ Ours-KL	91.03	70.00	**8.42**	87.20	**73.10**	**7.47**	70.33	**75.65**	**7.84**
+ Ours-Wa	**91.28**	**71.37**	8.40	**87.37**	72.66	7.15	**71.29**	74.33	7.52

Table 6: Computational cost of MMPO-based methods on UltraFeedback. Training time denotes wall-clock policy optimization time, and peak GPU memory denotes the maximum GPU memory usage observed during training.

Method	Training Time (h)	Peak GPU Memory (GiB)
MMPO	0.81	77.44
MMPO + Ours-KL	0.93	76.34
MMPO + Auto-Rubrics	0.85	59.29
+ Ours-KL	0.93	76.34
MMPO + OpenRubrics	0.90	59.36
+ Ours-KL	1.05	76.34
MMPO + Rubric-ARM	0.83	59.31
+ Ours-KL	0.98	76.34

Across all three datasets, the Wasserstein-based variant consistently improves the preference accuracy of the corresponding MMPO- and ODPO-based baselines. The improvements are also reflected in the win rate and average response quality score. This behavior remains consistent across Auto-Rubrics, OpenRubrics, and Rubric-ARM, indicating that the effectiveness of our robust preference optimization does not depend on a specific source of rubric supervision.

Comparing the two distance choices, neither KL nor Wasserstein uniformly dominates the other across datasets and metrics. Both variants consistently improve upon their corresponding nominal baselines, suggesting that the main benefit comes from explicitly accounting for uncertainty in rubric-derived preferences rather than from a particular choice of distributional distance.

G.2 Computational Cost

We additionally report the computational cost of our method in Table 6. We report the training time and peak GPU memory usage.

Our robust preference optimization introduces a moderate computational overhead compared with the corresponding nominal objectives. For example, MMPO requires 0.81 hours, while MMPO + Ours-KL requires 0.93 hours. Similar behavior is observed when our method is combined with automatically constructed rubrics. Across these settings, incorporating our robust objective increases the training time by approximately 9–18%, while keeping the total training time close to one hour under our experimental setting.

The additional GPU memory usage mainly results from the optimization of the alternative preference probabilities and the associated robust objective. Despite this overhead, all experiments fit within the memory capacity of the same hardware configuration used by the corresponding baselines.