CRIPO: ENHANCING RUBRIC-BASED RL VIA SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubric-based Reinforcement Learning (RL) has recently shown promise in improving Large Language Models (LLMs) on open-ended tasks. A widely recognized limitation of rubric-based RL is limited exploration: criteria that no rollout manages to satisfy (*Unexplored Criteria*) receive no optimization signal. Recent methods address this by incorporating rubric information as external guidance during rollout generation, yet they introduce a *train-inference mismatch*: the policy is optimized on rollouts produced under external guidance while this guidance is absent at inference time, causing error accumulation through autoregressive decoding. Moreover, these exploration-focused approaches overlook a fundamentally different failure mode that we term *Suppressed Criteria*—criteria that are satisfied by some rollouts yet whose learning signals might be lost during optimization because scalar reward aggregation assigns them non-positive aggregate advantages. Our analysis reveals that suppressed criteria constitute a persistent and non-negligible failure mode—over 25% of samples exhibit this issue throughout training. To simultaneously address both unexplored and suppressed criteria without introducing training-inference mismatch, we propose **Criterion-Distilled Policy Optimization (CriPO)**, which enhances rubric-based RL via on-policy self-distillation. For unexplored criteria, CriPO constructs a behavior-injection teacher and computes a filtered forward-KL loss to inject missing behaviors into the policy. For suppressed criteria, CriPO uses a counterfactual teacher to locate criterion-relevant tokens in negative-advantage rollouts, and corrects their advantages in GRPO to preserve useful patterns. Experiments on medicine and science benchmarks demonstrate that CriPO outperforms existing rubric-based RL methods, e.g., achieving an average gain of **3.3** points over GRPO on Qwen3-4B.

1 INTRODUCTION

In recent years, Reinforcement Learning from Verifiable Rewards (RLVR), instantiated by methods such as GRPO (Shao et al., 2024), has emerged as a prominent paradigm for improving the reasoning capabilities of large language models (DeepSeek-AI, 2025; Cui et al., 2026). RLVR has achieved notable success in verifiable domains such as mathematical reasoning and code generation, where exact answers or executable test cases provide stable and reliable supervision (He et al., 2024; Wei et al., 2025). However, in open-ended scenarios such as medical consultation and long-form writing, model outputs typically do not admit a unique gold answer and are difficult to verify with deterministic rules. To extend RLVR to such domains, recent work has explored rubric-based rewards, which decompose response quality into interpretable evaluation criteria, use LLM-as-a-Judge (Gu et al., 2024a) to assign multi-dimensional scores, and aggregate them into a scalar reward for RL training (Gunjal et al., 2025; Viswanathan et al., 2025; Dineen et al., 2025; Liu et al., 2026a).

While rubric-based RL has shown promising results in these open-ended domains, it still faces a fundamental challenge of limited exploration: since GRPO only optimizes over behaviors present in the sampled rollout group (Yue et al., 2025; Nguyen et al., 2025), criteria that no rollout manages to satisfy, termed ***Unexplored Criteria***, receive no effective optimization signal. Recent methods address this issue by using rubrics as external guidance during rollout generation. For example, RuscaRL (Zhou et al., 2025) uses rubric criteria as explicit scaffolding to elicit more diverse and higher-quality rollouts, while HeRL (Zhang et al., 2026b) leverages failed trajectories and their unmet criteria as hindsight guidance for generating revised rollouts. However, these exploration-

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.

Rubric-based Reward & GRPO Advantage

Rollout	C1 (3 pts)	C2 (3 pts)	C3 (1 pts)	C4 (1 pts)	Reward	Adv.
y_1	✓	✓	✗	✗	6	+1.61
y_2	✓	✗	✓	✗	4	-0.23
y_3	✗	✓	✓	✗	4	-0.23
y_4	✓	✗	✗	✗	3	-1.15

✓Satisfy ✗Dissatisfy Unexplored Criteria Suppressed Criteria

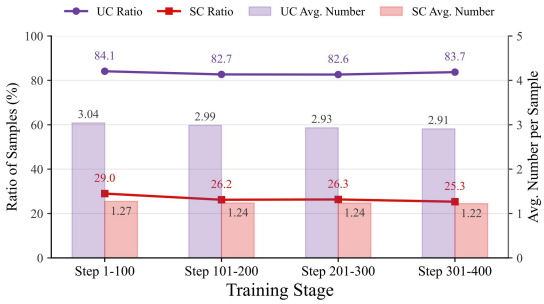

Figure 1: **Left:** Two failure modes in GRPO: *Unexplored Criteria* (UC), which are never satisfied by any rollout in the group (C4), and *Suppressed Criteria* (SC), for which criterion-satisfying rollouts receive non-positive aggregate advantages (C3). **Right:** Ratio of samples with UC/SC and number of UC/SC on average among samples containing UC/SC when training Qwen3-4B on RaR-Medicine.

enhancement methods still have two important drawbacks. First, they introduce a ***training-inference mismatch***: during training, rollouts are generated with privileged information, so each token is optimized under externally guided prefixes; at inference time, however, the model must generate solely from its own prefixes. This discrepancy constitutes exposure bias (Agarwal et al., 2024; Song & Zheng, 2026; Li et al., 2026b), where early generation error that could be avoided during guided training may propagate through subsequent autoregressive decoding. Also, these mixed-policy optimization pipelines further lead to longer training times; see Section C.2 for more details.

Second, these methods overlook another fundamentally different failure mode beyond unexplored criteria: ***Suppressed Criteria***—already explored criteria whose learning signals might be systematically lost during GRPO optimization. This occurs because GRPO aggregates criterion-wise scores into a single scalar reward and broadcasts it uniformly to the entire response. Consequently, rollouts that satisfy certain criteria but underperform on others receive negative or negligible aggregate advantages, causing useful criterion-satisfying behaviors to be penalized or ignored rather than reinforced. For example, in Figure 1 (left), y_2 and y_3 satisfy criterion C3, but their advantage sum is negative (-0.46). Our statistical analysis in Figure 1 (right) reveals that suppressed criteria remain a non-negligible issue in practice: when training Qwen3-4B on RaR-Medicine, over 25% of samples contain suppressed criteria throughout training, with more than 1.2 suppressed criteria on average among samples containing SC. While some prior works (Chen et al., 2026; Lan, 2026; Xu et al., 2026) also recognize that scalar reward aggregation can obscure individual criterion contributions, they do not provide a systematic analysis of when and why criterion-level signals are lost, and they also fail to address the complementary challenge of unexplored criteria.

To address both unexplored and suppressed criteria without introducing training-inference mismatch, in this paper, we propose **Criterion-Distilled Policy Optimization (CriPO)**, which enhances rubric-based RL via On-Policy Self-Distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026; Ye et al., 2026). Unlike the approaches that inject knowledge or behaviors by producing rollouts under privileged information, OPSD constructs a privilege-conditioned self-teacher and provides token-level supervision on the model's own on-policy rollouts, thereby obtaining learning signals without a distributional gap between training and inference. Specifically, CriPO adopts GRPO as the stable, reward-grounded optimization backbone and uses the following two OPSD-derived signals for targeted criterion-level correction: for unexplored criteria, CriPO constructs a *behavior-injection self-teacher* and computes forward-KL loss with contribution-based token filtering as an auxiliary loss to inject missing criterion-specific behaviors into the policy; for suppressed criteria, CriPO constructs a *counterfactual self-teacher* to locate suppressed-criterion-relevant tokens in negative-advantage rollouts, and flip or mask their GRPO advantages to preserve useful patterns. Experiments on five medicine and science benchmarks demonstrate that CriPO outperforms five baselines in both in-domain and cross-domain settings (e.g., achieving an average gain of **3.3** points over GRPO on Qwen3-4B), and further analysis shows that CriPO effectively reduces the occurrence rates of both unexplored and suppressed criteria. Note that CriPO requires at most one additional forward pass per rollout, resulting in about a 7% increase on average in training time compared with GRPO, and CriPO also improves optimization efficiency, reaching GRPO's converged performance in no more than half the optimization steps and less than half the wall-clock training time.

2 PRELIMINARIES

Rubric-based Reinforcement Learning (RL). We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the core RL algorithm for training LLMs with rubric-based rewards. Given a prompt x, the policy π_θ samples a group of on-policy rollouts $\{y_i\}_{i=1}^{G}$, where $y_i \sim \pi_\theta(\cdot \mid x)$. Each rollout is evaluated with a set of criteria $\mathcal{C} = \{(c_j, \omega_j)\}_{j=1}^{M}$, where c_j denotes an evaluation criterion and ω_j its importance weight. A judge model assigns a criterion-wise score $r_{ij} = c_j(x, y_i)$, indicating how well rollout y_i satisfies criterion c_j. These criterion-wise scores are then aggregated into a scalar reward: $R_i = \sum_{j=1}^{M} \omega_j r_{ij} / \sum_{j=1}^{M} \omega_j$. GRPO then computes a group-relative advantage by normalizing the reward within the group: $A_i = (R_i - \mu_R)/(\sigma_R + \epsilon)$, where μ_R and σ_R are the mean and standard deviation of the group's reward, and $\epsilon > 0$ avoids division by zero. The policy model is optimized with the PPO-style (Schulman et al., 2017) clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{\sum_i |y_i|} \sum_{i=1}^{G} \sum_{t=1}^{|y_i|} \min(\rho_{i,t} A_i, \text{clip}\left(\rho_{i,t}, 1-\epsilon_{\text{c}}, 1+\epsilon_{\text{c}}\right) A_i) \right], \tag{1}$$

where $\rho_{i,t} = \pi_\theta(y_{i,t} \mid x, y_{i,<t}) / \pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})$ is the importance ratio and $\pi_{\theta_{\text{old}}}$ is the frozen rollout policy and π_θ is the current policy with gradient. Note that we use token-mean aggregation following the default implementation of GRPO in the verl codebase (Sheng et al., 2025).

Failure in Rubric-based RL: Unexplored and Suppressed Criteria. Although rubric-based rewards provide criterion-level supervision, two inherent limitations of GRPO prevent this supervision from being fully exploited: (i) *limited exploration*: GRPO only optimizes over behaviors present in the sampled rollouts, so that criteria with no rollout managed to satisfy receive no optimization signal; and (ii) *reward ambiguity*: aggregating multiple criteria into a single scalar reward obscures criterion-specific token contributions and may penalize useful behaviors. We show that these limitations lead to two types of failure criteria:

• ***Unexplored Criteria*** correspond to behaviors absent from the current rollout group due to limited exploration (e.g., C4 in Figure 1). Consider the setting where each criterion-wise score is binary, i.e., $r_{ij} = c_j(x, y_i) \in \{0, 1\}$, indicating whether the corresponding criterion is satisfied or not. The unexplored criteria are defined as criteria that are not satisfied by any rollout in the current group:

$$\mathcal{C}_u = \{c_j \in \mathcal{C} \mid \forall y_i,\ r_{ij} = 0\}. \tag{2}$$

• ***Suppressed Criteria*** correspond to behaviors that the model has already discovered, but the criterion-satisfying rollouts receive non-positive aggregate advantages due to reward ambiguity (e.g., C3 in Figure 1). Specifically, we consider the following two suppression scenarios: (i) *negative-advantage suppression*, where the advantage sum across all satisfying rollouts is negative, causing the criterion-satisfying behavior to be penalized, and (ii) *zero-advantage suppression*, where the advantage sum is zero, and the criterion is rarely satisfied (e.g., by not more than half of the group), so that the criterion receives negligible optimization pressure despite being insufficiently learned. Let $\mathcal{S}_j = \{i \mid r_{ij} = 1\}$ denote the rollout set that satisfies criterion c_j, the suppressed criteria are defined as:

$$\mathcal{C}_s = \left\{ c_j \in \mathcal{C} \;\middle|\; \mathcal{S}_j \neq \emptyset \ \wedge \ \left(\sum_{i \in \mathcal{S}_j} A_i < 0 \ \vee \ \left(\sum_{i \in \mathcal{S}_j} A_i = 0 \ \wedge \ |\mathcal{S}_j| \leq G/2 \right) \right) \right\}. \tag{3}$$

Figure 1 (right) shows that both unexplored criteria and suppressed criteria are persistent and non-negligible in practice. More statistics of these two failure modes on other settings and using different LLM judges can be found in Appendix C.1.

On-Policy Self-Distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026; Ye et al., 2026) has recently emerged as a promising paradigm for converting privileged information into dense token-level supervision while avoiding the training-inference distribution mismatch (Agarwal et al., 2024; Gu et al., 2024b). Given an on-policy rollout $y \sim \pi_\theta(\cdot \mid x)$, OPSD compares two next-token distributions produced by the same model: a student distribution conditioned only on the original context, and a self-teacher distribution conditioned additionally on privileged information ξ.

Specifically, the policy is trained to match the privilege-conditioned prediction through a token-level divergence at each token position t (reverse-KL instantiation):

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{x,y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{|y|}\sum_{t=1}^{|y|} D_{\text{KL}}\left(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \text{sg}(\pi_\theta(\cdot \mid x, \xi, y_{<t}))\right)\right], \tag{4}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation. Compared with standard RLVR methods that optimize sparse outcome rewards, OPSD provides fine-grained supervision at each token position and can reveal how the privileged information changes the model's local generation preference.

In rubric-based RL, rubric criteria can naturally serve as the privileged information ξ for OPSD. By conditioning the self-teacher on rubrics, OPSD can expose *criterion-level supervision*, making it well suited to address the two limitations discussed above. However, directly relying on OPSD as the primary optimization objective may introduce unreliable token-level supervision due to issues such as privileged-information leakage or entropy collapse (Yang et al., 2026; Kim et al., 2026b; Zhu et al., 2026; Li et al., 2026a; Kim et al., 2026a). To this end, **we retain GRPO as the stable, reward-grounded optimization backbone and use OPSD-derived signals only for targeted criterion-level correction**, which follows the emerging paradigm of combining OPSD with RLVR rather than relying on standalone OPSD (Li et al., 2026a; Yang et al., 2026; Kim et al., 2026a; Wang et al., 2026a; Liu et al., 2026b).

3 CriPO: Criterion-Distilled Policy Optimization

To simultaneously address the two failures in rubric-based RL without introducing training-inference mismatch, we propose **Criterion-Distilled Policy Optimization** (**CriPO**). The overall framework of CriPO is shown in Figure 3 and the pseudo-code of CriPO is shown in Algorithm 1.

3.1 Behavior Injection for Unexplored Criteria

To inject the missing behaviors into the policy, we construct a *behavior-injection self-teacher* to compute the auxiliary OPSD loss, which aims to revise the previous response given the unexplored criteria. The prompt template is shown in the prompt box in Figure 3 at the top. Formally, let $p_t^S = \pi_\theta(\cdot \mid x, y_{<t})$ and $p_t^T = \pi_\theta(\cdot \mid x, y, \mathcal{C}_u, y_{<t})$ denote the student's and the teacher's distributions at token position t, we compute the per-token **forward-KL** divergence $d_t = D_{\text{KL}}\left(\text{sg}(p_t^T) \,\|\, p_t^S\right)$ to distill the teacher's criterion-conditioned behavior to the student. We adopt forward KL since it enables the student to cover teacher-preferred modes, which encourages exploration.

Contribution-based Token Filtering. In open-ended generations, only a few positions are actually needed to be modified to satisfy the missing criteria—the teacher distribution shifts substantially only at some tokens that affect criterion-specific signals, while the remaining tokens exhibit small KL values that primarily reflect perturbations from prompt modification. As shown in Figure 2, when training Qwen3-4B on the first data batch of RaR-Medicine, only 34.6% of tokens account for 95% of the total divergence. We therefore apply distillation only on the smallest set of tokens whose cumulative KL contribution reaches a threshold γ:

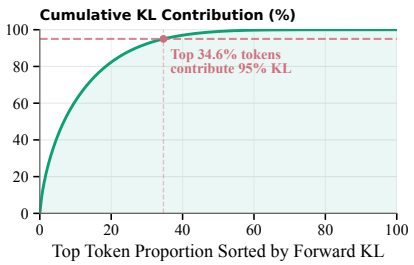

Figure 2: Cumulative KL contribution across tokens sorted by magnitude.

$$\mathcal{T}^u = \text{TopCum}\left(\{d_t\}_{t=1}^{|y|}; \gamma\right), \tag{5}$$

where we set $\gamma = 0.95$ by default. Moreover, for each rollout group, we only compute the distillation loss on the **highest-advantage rollout**. Since this rollout already satisfies the largest proportion of criteria, the self-teacher only needs to make minimal revisions to incorporate the remaining unexplored criteria, yielding more reliable and less noisy corrections than revising an arbitrary or low-quality response. Let y^* be the highest-advantage rollout in the group; the overall loss of behavior

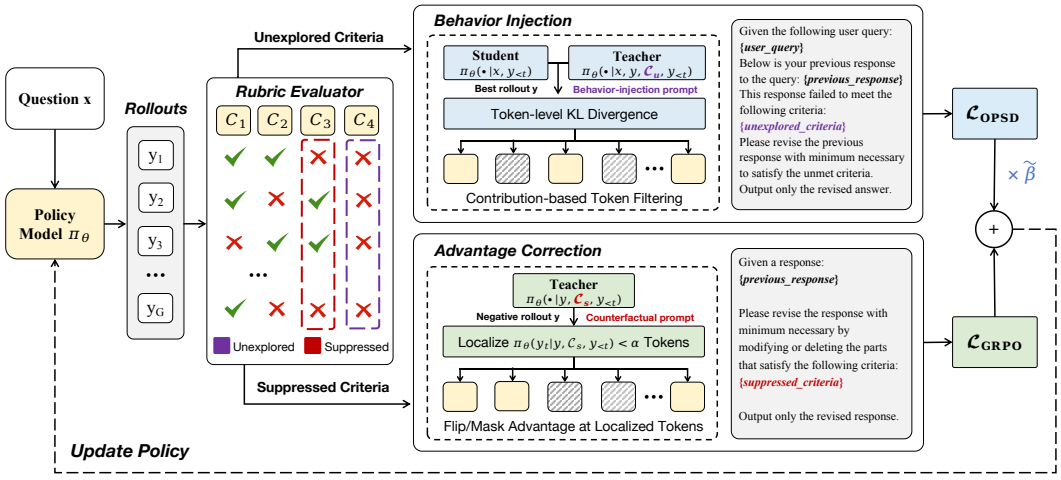

Figure 3: **Overview of CriPO.** CriPO first identifies unexplored and suppressed criteria from rubric-evaluated rollouts. For unexplored criteria, CriPO injects the missing criterion-specific behavior through **OPSD** Loss. For suppressed criteria, CriPO locates criterion-relevant tokens and masks or flips their advantages to modify **GRPO** Loss. The two objectives are combined to update the policy.

injection is as follows:

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{x,\{y_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{1}{|\mathcal{T}^u|}\sum_{t\in\mathcal{T}^u} D_{\text{KL}}\left(\text{sg}(\pi_\theta(\cdot \mid x, y^*, \mathcal{C}_u, y^*_{<t})) \,\|\, \pi_\theta(\cdot \mid x, y^*_{<t})\right)\right], \quad (6)$$

3.2 Advantage Correction for Suppressed Criteria

For suppressed criteria, we apply GRPO advantage correction for negative-advantage rollouts in the group, preserving the criterion-satisfying parts in them without reinforcing the entire trajectory.

Suppressed-Criterion Token Localization. For each negative-advantage rollout y_i and its satisfying suppressed criteria $\mathcal{C}_s^i = \{c_j \in \mathcal{C}_s \mid r_{ij} = 1\}$, we construct a *counterfactual self-teacher* that aims to revise the rollout by modifying or deleting the parts that satisfy these suppressed criteria; see the prompt template shown at the bottom of Figure 3. Because the counterfactual teacher is explicitly instructed to remove all suppressed-criterion behaviors present in y_i, it assigns low probability to tokens that express those behaviors. Let $p^T_{i,t} = \pi_\theta(\cdot \mid x, y_i, \mathcal{C}_s^i, y_{i,<t})$ denote the corresponding teacher distribution at position t. We identify candidate criterion-relevant tokens in rollout y_i using a low-probability threshold α:

$$\mathcal{T}_i^s = \left\{t \mid p^T_{i,t}(y_{i,t}) < \alpha\right\}, \quad (7)$$

The polarized distribution in Figure 11 shows that the teacher usually retains the original token with high confidence or strongly disfavors it, providing an empirical basis for using low teacher probability as a localization signal. In practice, we simply set $\alpha = 0.1$ to select the low-probability subset. We also apply two cutoff strategies to these candidate tokens, as detailed in Appendix B.

Selective Advantage Flipping and Masking. Because the localized tokens inherit the negative rollout-level advantage $A_i < 0$, GRPO penalizes them despite their contribution to satisfying suppressed criteria. A direct correction is to flip all these tokens' advantages to a small positive value τ_{flip}. However, unconditionally flipping every localized token may alter GRPO's original advantage structure more than necessary. To this end, we use masking when the group already provides positive support for these patterns. Specifically, let $\mathcal{P}_j = \{k \mid r_{kj} = 1, A_k > 0\}$ denote the positive-advantage rollouts satisfying c_j. If $\mathcal{P}_j$ is nonempty for every $c_j \in \mathcal{C}_s^i$, we mask the localized tokens to prevent negative updates; otherwise, we flip them to provide a positive signal for criteria without such support. Let $z_i = \mathbb{I}\left[\exists c_j \in \mathcal{C}_s^i : \mathcal{P}_j = \emptyset\right]$ be the factor that captures this decision; the corrected token-level advantage is

$$\tilde{A}_{i,t} = A_i\, \mathbb{I}[t \notin \mathcal{T}_i^s] + z_i \tau_{\text{flip}}\, \mathbb{I}[t \in \mathcal{T}_i^s], \quad (8)$$

where we set $\tau_{\text{flip}} = 0.1$ by default. The corrected GRPO loss is computed using $\tilde{A}_{i,t}$ in place of the rollout-level advantage:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{\sum_i |y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big(\rho_{i,t}\tilde{A}_{i,t}, \text{clip}\left(\rho_{i,t}, 1-\epsilon_{\text{c}}, 1+\epsilon_{\text{c}}\right)\tilde{A}_{i,t}\Big)\right], \tag{9}$$

3.3 Training Objective

To coordinate the optimization strengths of GRPO loss and OPSD loss, we adapt the OPSD coefficient to the relative-quality composition of each rollout group. Specifically, when only a few rollouts have non-negative advantages, many rollouts require direct reward-based correction, and we therefore let GRPO dominate. Conversely, when most rollouts have non-negative advantages, only a small subset requires strong negative correction from GRPO, leaving more optimization capacity for OPSD to target missing behaviors. The overall training objective is

$$\begin{gathered}\mathcal{L}_{\text{CriPO}} = \mathcal{L}_{\text{GRPO}} + \tilde{\beta}\cdot\mathcal{L}_{\text{OPSD}}, \\ \text{where}\quad \tilde{\beta} = \Big(1+\text{sign}\big(s-\tfrac{1}{2}\big)\,|2s-1|^{t}\Big)\,\beta, \quad s = \frac{1}{G}\sum\nolimits_{i=1}^{G}\mathbb{I}[A_i \geq 0]\,.\end{gathered} \tag{10}$$

Here, s is the proportion of rollouts with non-negative advantage in the group, β is the base OPSD coefficient, and t is a temperature parameter that controls how sensitively the coefficient responds to s. We set $t = 1$ in all experiments, in which case $\tilde{\beta} = 2s\beta$, and during the final 25% of training steps, we linearly decay the base coefficient β to zero to stabilize late-stage optimization.

4 Experiment

4.1 Experimental Setup

Datasets and Evaluation. We evaluate CriPO on both medicine and science QA tasks, where RaR-Medicine and RaR-Science (Gunjal et al., 2025) are adopted as the training datasets for medicine and science tasks, respectively. We use their own test split for in-domain evaluation, and for cross-domain evaluation, we use HealthBench (Arora et al., 2025) and LLMEval-Med (Zhang et al., 2025) for medicine tasks, and ResearchQA (Yifei et al., 2025) for science tasks. We use Qwen3-32B (Team, 2025) as the rubric judge during training and GPT-4o-mini (OpenAI, 2023) as the judge for evaluation. For every evaluated response, the evaluation score is computed using the scalar rubric-reward aggregation defined in Section 2. More details about the datasets and evaluation settings are provided in Appendix B.

Baselines and Models. Our main experiments are conducted on Qwen3-1.7B and Qwen3-4B, and the baselines include: **SFT**, which fine-tunes the model to fit synthesized targets generated by rubric guidance; **OPSD** (Zhao et al., 2026), an on-policy self-distillation baseline that distills the full rubric into the model; **GRPO** (Shao et al., 2024), the standard group-relative policy optimization baseline; **RuscaRL** (Zhou et al., 2025), a rubric-based RL method that uses rubric scaffolding during rollout generation and policy optimization; and **HeRL** (Zhang et al., 2026b), a rubric-based RL method that leverages hindsight feedback from failed trajectories and unmet rubrics to enhance exploration. All methods are trained on the same data and evaluated under the same protocol for fair comparison. More implementation details of baselines and CriPO are provided in Appendix B.

4.2 Main Results

CriPO outperforms existing baselines. As shown in Table 1, SFT underperforms the base model on most benchmarks, suggesting that fitting rubric-guided targets does not generalize reliably. Standalone OPSD is also unsatisfactory without a stable reward-grounded backbone. GRPO consistently improves the base model, yielding average gains of roughly 4–5 points across the two model scales, confirming the benefit of reward-grounded optimization. RuscaRL and HeRL further strengthen GRPO through rubric-guided exploration. Built on GRPO, CriPO raises the average score from 61.2

Table 1: Experimental results (%) across medicine and science benchmarks using GPT-4o-mini as judge. The best results are marked in **bold**, and the second-best are underlined. Arrows indicate changes over the base model. All results are evaluated on models trained after 400 steps.

Method	Medicine			Science		Avg.
	RaR-Medicine	HealthBench	LLMEval-Med	RaR-Science	ResearchQA	
Qwen3-1.7B	48.6 ↑ 0.0	61.6 ↑ 0.0	56.5 ↑ 0.0	63.3 ↑ 0.0	57.2 ↑ 0.0	57.4 ↑ 0.0
+ SFT	45.8 ↓ 2.8	50.1 ↓ 11.5	46.5 ↓ 10.0	62.1 ↓ 1.2	47.0 ↓ 10.2	50.3 ↓ 7.1
+ OPSD	48.5 ↓ 0.1	51.9 ↓ 9.7	54.3 ↓ 2.2	64.8 ↑ 1.5	52.5 ↓ 4.7	54.4 ↓ 3.0
+ GRPO	50.7 ↑ 2.1	63.0 ↑ 1.4	60.6 ↑ 4.1	68.3 ↑ 5.0	63.6 ↑ 6.4	61.2 ↑ 3.8
+ RuscaRL	51.5 ↑ 2.9	63.8 ↑ 2.2	62.1 ↑ 5.6	70.7 ↑ 7.4	64.8 ↑ 7.6	62.6 ↑ 5.2
+ HeRL	52.6 ↑ 4.0	65.8 ↑ 4.2	**64.2** ↑ 7.7	70.1 ↑ 6.8	62.3 ↑ 5.1	63.0 ↑ 5.6
+ CriPO	**52.8** ↑ 4.2	**67.3** ↑ 5.7	**64.2** ↑ 7.7	**71.0** ↑ 7.7	**65.0** ↑ 7.8	**64.1** ↑ 6.7
Qwen3-4B	55.8 ↑ 0.0	65.7 ↑ 0.0	68.4 ↑ 0.0	69.2 ↑ 0.0	62.1 ↑ 0.0	64.2 ↑ 0.0
+ SFT	54.6 ↓ 1.2	55.2 ↓ 10.5	59.2 ↓ 9.2	71.1 ↑ 1.9	54.0 ↓ 8.1	58.8 ↓ 5.4
+ OPSD	56.6 ↑ 0.8	55.8 ↓ 9.9	58.5 ↓ 9.9	71.1 ↑ 1.9	58.5 ↓ 3.6	60.1 ↓ 4.1
+ GRPO	60.0 ↑ 4.2	69.8 ↑ 4.1	69.9 ↑ 1.5	76.0 ↑ 6.8	70.0 ↑ 7.9	69.1 ↑ 4.9
+ RuscaRL	63.2 ↑ 7.4	72.1 ↑ 6.4	71.4 ↑ 3.0	77.1 ↑ 7.9	72.4 ↑ 10.3	71.2 ↑ 7.0
+ HeRL	62.4 ↑ 6.6	**73.2** ↑ 7.5	71.7 ↑ 3.3	74.5 ↑ 5.3	71.7 ↑ 9.6	70.7 ↑ 6.5
+ CriPO	**63.5** ↑ 7.7	72.6 ↑ 6.9	**73.9** ↑ 5.5	**78.6** ↑ 9.4	**73.6** ↑ 11.5	**72.4** ↑ 8.2

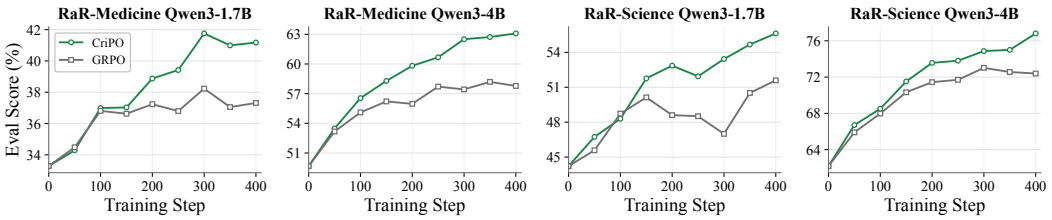

Figure 4: **Comparisons of evaluation scores during training.** Scores are judged by Qwen3-32B.

to **64.1** on Qwen3-1.7B (+**2.9**) and from 69.1 to **72.4** on Qwen3-4B (+**3.3**). It also achieves the best overall result, outperforming the strongest baseline by 1.1 on Qwen3-1.7B and 1.2 on Qwen3-4B. These results demonstrate the effectiveness of our on-policy framework that simultaneously addresses the two failure modes in standard rubric-based RL.

Efficiency and Convergence Analysis. Figure 4 compares the optimization trajectories of CriPO and GRPO over 400 training steps. GRPO improves rapidly in the early stage, but its subsequent gains become slower and less consistent, whereas CriPO exhibits more sustained improvement. Across all model-domain settings, CriPO reaches or exceeds the best score attained by GRPO within the first 150 to 200 steps and continues toward a substantially higher final score, demonstrating more efficient use of training samples and policy updates. In terms of the training time, CriPO requires at most one additional forward pass per rollout, resulting in about a 7% increase on average in training time compared with GRPO. Detailed comparisons of wall-clock training time as well as peak GPU memory are provided in Appendix C.2.

4.3 Analysis

Training Dynamics. To examine how the two modules in CriPO affect optimization, Figure 5 compares GRPO, CriPO-U (behavior injection only), and CriPO over 400 steps of training Qwen3-4B on RaR-Medicine and RaR-Science. As shown in the **left** column, CriPO-U consistently follows a higher reward trajectory than GRPO, while adding advantage correction generally yields the highest reward, demonstrating the complementary benefits of the two modules. The **middle** column shows that GRPO undergoes rapid entropy collapse, whereas behavior injection substantially slows this

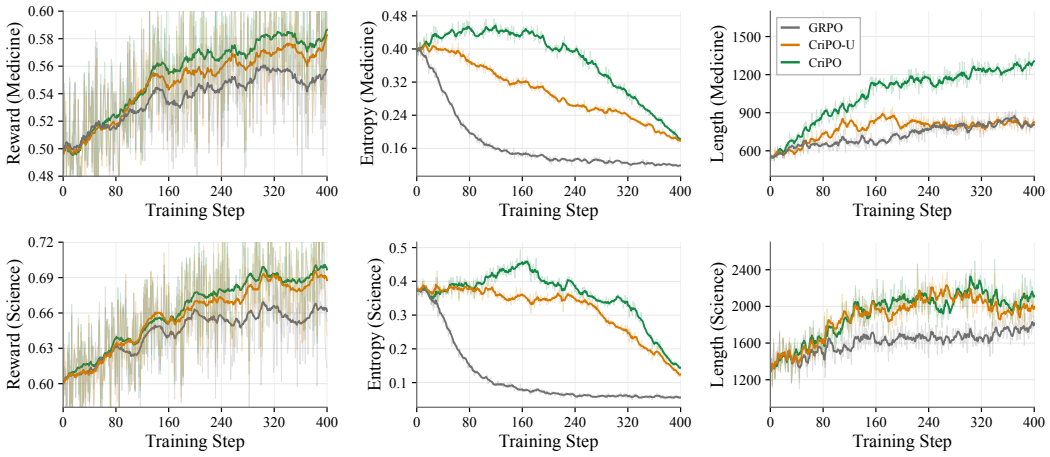

Figure 5: **Training dynamics of CriPO.** We compare CriPO and GRPO across reward, entropy, and response length over 400 training steps when training Qwen3-4B on RaR-Medicine and RaR-Science.

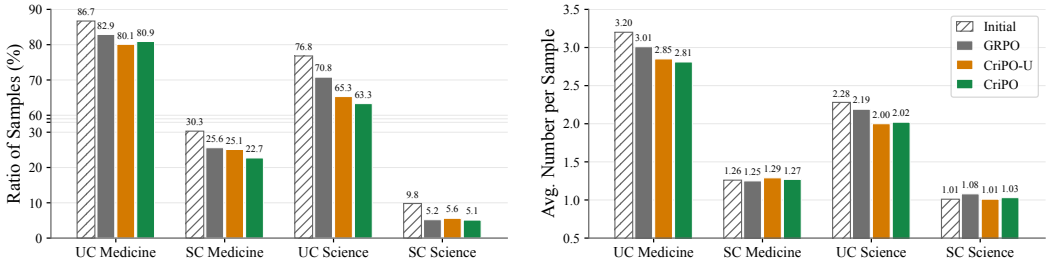

Figure 6: **Statistics of unexplored and suppressed criteria on the testing set.** We report the occurrence ratio and average count across methods trained with Qwen3-4B.

decline and thereby preserves exploration. The full CriPO maintains even higher entropy through most of training, suggesting that advantage correction further protects diverse criterion-relevant patterns from negative updates. As the base OPSD coefficient β is linearly decayed toward zero in the final training stage, the entropy of both CriPO variants also decreases, allowing the policies to settle into more stable output distributions. The **right** column further shows that CriPO produces longer responses than GRPO in both domains. Behavior injection encourages the model to cover missing criterion-specific behaviors, while advantage correction preserves content associated with suppressed criteria; the latter produces a particularly clear additional length increase on RaR-Medicine. Together with the reward gains, these trends suggest that the increased length reflects richer rubric-relevant content rather than mere verbosity. Appendix D.1 provides an illustrative example in which CriPO gives more complete clinical reasoning and satisfies more rubric criteria than GRPO.

Statistics of Unexplored and Suppressed Criteria. To examine whether the two modules alleviate their targeted failure modes, Figure 6 reports the statistics on the RaR-Medicine and RaR-Science testing sets for methods trained with Qwen3-4B. We use the training-time sampling configuration rather than that of evaluation to better approximate the training rollout distribution. The following two observations emerge. First, **behavior injection substantially reduces UC**. Compared with the initial model, GRPO lowers the UC ratio by 3.8 and 6.0 percentage points on medicine and science. Adding behavior injection (CriPO-U) yields further reductions of 2.8 and 5.5 points over GRPO, while decreasing the average UC count from 3.01 to 2.85 on medicine and from 2.19 to 2.00 on science. These improvements show that behavior injection helps the policy acquire criterion-specific behaviors absent from its rollout groups. Second, **advantage correction further mitigates SC**. Relative to CriPO-U, the full CriPO reduces the SC ratio from 25.1% to 22.7% on medicine and from 5.6% to 5.1% on science, and the average SC count remains nearly unchanged on both domains.

Table 3: **Types of tokens localized by the counterfactual teacher.** Statistics are computed over 8,497 selected tokens from 1,641 suppressed-criterion rollout assignments.

Token Category	Representative Tokens	Proportion
Formatting & Structural Boundaries	`\n`, `\n\n`, `.`, `:`, `(`, `**`, `###`, `--`, `-`	44.3%
Knowledge & Semantic Content	`clinical`, `blood`, `therapy`, `liver`, `Fever`, `virus`	21.5%
Logical & Discourse Connectors	`and`, `that`, `which`, `However`, `Let`, `Other`, `Explanation`, `If`, `but`	17.1%
Grammatical Function Words	`the`, `a`, `in`, `of`, `to`, `is`, `this`	15.4%
Numbers & Option Labels	`1`, `2`, `3`, `4`, `A`, `B`, `C`, `D`	1.6%

Also, adding advantage correction largely preserves the UC gains and even reduces the UC ratio on science, consistent with the intuition that preserving suppressed behaviors prevents explored criteria from becoming unexplored in subsequent rollouts. Overall, each module primarily improves its intended failure mode, while their combination provides the most balanced reduction of UC and SC.

Reliability of Behavior-Injection Teacher. To assess whether the behavior-injection teacher provides reliable supervision, we directly sample four teacher rollouts for samples and use the highest-reward rollout as a proxy for the teacher's ability to elicit the missing behaviors. Table 2 compares the rewards of student and teacher rollouts and reports the student's average UC count. "Recovered" denotes the number of the student's UC that are satisfied by the selected teacher rollout. Across all model-domain settings, the teacher consistently produces higher-reward responses than the student, with gains of 14.4–20.0 points. It also recovers 1.25–1.98 UC per sample, corresponding to approximately 43–64% of the criteria missed by the student. These consistent improvements support distilling the teacher's token-level signal into the student for behavior injection.

Table 2: Comparisons between the behavior-injection teacher rollouts and the student rollouts on the testing set.

Setting	Rollout Reward		UC Number	
	Student	Teacher	Student	Recovered
Qwen3-1.7B Medicine	53.0	73.0 ↑ 20.0	3.86	1.93
Qwen3-4B Medicine	60.7	75.1 ↑ 14.4	3.20	1.98
Qwen3-1.7B Science	58.5	74.5 ↑ 16.0	2.92	1.25
Qwen3-4B Science	67.0	82.3 ↑ 15.3	2.29	1.46

What Tokens Does the Counterfactual Teacher Locate? We analyze the tokens localized by the counterfactual teacher using Qwen3-4B on the RaR-Medicine testing set, and their major categories and distributions are summarized in Table 3. Advantage correction on formatting and structural boundaries can preserve the organization and extent of criterion-bearing spans, correction on logical and discourse connectors can protect useful reasoning relations and points of elaboration, correction on knowledge and semantic content can retain criterion-relevant facts and entities, and correction on numbers and option labels can preserve quantitative information and final-answer choices. In contrast, grammatical function words carry little meaning in isolation and may provide limited value for advantage correction. Appendix D.2 provides a criterion-wise breakdown and representative success and failure cases.

Please refer to more results and analysis in the Appendix, including results using Qwen3-32B as judge (C.3), other LLMs (C.4), multiple seeds (C.5), different OPSD variants (C.6), OOD evaluation (C.7), using DAPO as the optimization backbone (C.8), and ablation studies (C.9).

5 Conclusion

In this work, we study rubric-based RL for improving LLMs on open-ended tasks and identify two prevalent criterion-level failure modes: unexplored criteria, where no rollout satisfies a criterion, and suppressed criteria, where criterion-satisfying behaviors are penalized or ignored due to non-positive aggregate advantages. To address both issues, we propose Criterion-Distilled Policy Optimization (CriPO), an on-policy framework that retains GRPO as a reward-grounded backbone while using OPSD for targeted criterion-level correction. Specifically, CriPO injects unexplored behavior through a localized forward-KL distillation loss and preserves useful suppressed behaviors by locating relevant tokens and correcting their advantages. Experiments on medicine and science benchmarks demonstrate the effectiveness and efficiency of CriPO against existing rubric-based RL methods. We hope our work can inspire the community to further investigate how to jointly address both limited exploration and reward ambiguity in standard rubric-based RL.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and implement methods. We did not use generative AI tools to design or provide feedback on the research methodology or experiments, support qualitative and thematic data analysis, or interpret results. Generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for mathematical proofs, assisting in writing proofs, proposing or refining hypotheses, and cleaning or reformatting datasets were not applicable to this work. Additionally, we used generative AI tools to create or edit software code, search for information, and edit the paper to improve readability. We reviewed all AI-assisted work by manually checking the AI-generated text, code, and search results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health. *CoRR*, abs/2505.08775, 2025.

Yukun Chen, Jiaming Li, Longze Chen, Ze Gong, Jingpeng Li, Zhen Qin, Hengyu Chang, Ancheng Xu, Zhihao Yang, Hamid Alinejad-Rokny, Qiang Qu, Bo Zheng, and Min Yang. Rucl: Stratified rubric-based curriculum learning for multimodal large language model reasoning. *CoRR*, abs/2602.21628, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Trans. Mach. Learn. Res.*, 2026, 2026.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Jacob Dineen, Aswin RRV, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta Baral, Muhao Chen, and Ben Zhou. QA-LIGN: aligning llms through constitutionally decomposed QA. In *EMNLP Findings*, pp. 20619–20642, 2025.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *CoRR*, abs/2506.11763, 2025.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594, 2024a.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, 2024b.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *CoRR*, abs/2507.17746, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *ACL*, pp. 3828–3850, 2024.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *CoRR*, abs/2601.20802, 2026.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled RLVR. *CoRR*, abs/2605.10781, 2026a.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *CoRR*, abs/2603.24472, 2026b.

Guangchen Lan. Alternating reinforcement learning with contextual rubric rewards. *CoRR*, abs/2603.15646, 2026.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *CoRR*, abs/2604.02288, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *CoRR*, abs/2604.13016, 2026b.

Wenhan Liu, Jiajie Jin, Zhaoheng Huang, Tongyu Wen, Guanting Dong, Ziliang Zhao, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. The rules of the game: A survey of rubrics for large language models. 2026a.

Yifeng Liu, Shiyuan Zhang, Yifan Zhang, and Quanquan Gu. Self-distilled policy gradient. *arXiv preprint arXiv:2606.04036*, 2026b.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026.

Phuc Minh Nguyen, Chinh D. La, Duy M. H. Nguyen, Nitesh V. Chawla, Binh T. Nguyen, and Khoa D. Doan. The reasoning boundary paradox: How reinforcement learning constrains language models. *CoRR*, abs/2510.02230, 2025.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. Rlcsd: Reinforcement learning with contrastive on-policy self-distillation. *arXiv preprint arXiv:2606.11709*, 2026.

MohammadHossein Rezaei, Anas Mahmoud, Zihao Wang, Utkarsh Tyagi, Advait Gosai, Razvan-Gabriel Dumitru, Aakash Sabharwal, Bing Liu, and Yunzhong He. Rubric-guided self-distillation: Post-training without rubric verifiers. *arXiv preprint arXiv:2606.12507*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297, 2025.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *CoRR*, abs/2604.00626, 2026.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

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. *CoRR*, abs/2507.18624, 2025.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. TRACE: distilling where it matters via token-routed self on-policy alignment. *CoRR*, abs/2605.10194, 2026a.

Yuhan Wang, Zhengxi Lu, Yuchen Yan, Kaitao Song, Wenqi Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Papergym: Rubric-centered evolution for research-plan generation. *CoRR*, abs/2608.31119, 2026b. doi: 10.48550/ARXIV.2608.31119. URL https://doi.org/10.48550/arXiv.2608.31119.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. SWE-RL: advancing LLM reasoning via reinforcement learning on open software evolution. *CoRR*, abs/2502.18449, 2025.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. *CoRR*, abs/2503.05244, 2025.

Tianze Xu, Yanzhao Zheng, Pengrui Lu, Lyumanshan Ye, Yong Wu, ZhenTao Zhang, YuanQiang Yu, Chao Ma, JiHuai Zhu, Pengfei Liu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, and Gang Yu. Rubrics to tokens: Bridging response-level rubrics and token-level rewards in instruction following tasks. *CoRR*, abs/2604.02795, 2026.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *CoRR*, abs/2604.03128, 2026.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *CoRR*, abs/2602.12275, 2026.

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. *CoRR*, abs/2509.00496, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837, 2025.

Bo-Wen Zhang, Junwei He, Wen Wang, Song-Lin Lv, Wentao Ma, Rongyi Lin, Shuhan Zhong, and Lan-Zhe Guo. Cort: Counterfactual replay for token-level rubric-guided policy optimization. *CoRR*, abs/2607.25659, 2026a. doi: 10.48550/ARXIV.2607.25659. URL https://doi.org/10.48550/arXiv.2607.25659.

Ming Zhang, Yujiong Shen, Zelin Li, Huayu Sha, Binze Hu, Yuhui Wang, Chenhao Huang, Shichun Liu, Jingqi Tong, Changhao Jiang, Mingxu Chai, Zhiheng Xi, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. Llmeval-med: A real-world clinical benchmark for medical llms with physician validation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *EMNLP Findings*, pp. 4888–4914, 2025.

Wenjian Zhang, Kongcheng Zhang, Jiaxin Qi, Baisheng Lai, and Jianqiang Huang. Experience is the best teacher: Motivating effective exploration in reinforcement learning for llms. *CoRR*, abs/2603.20046, 2026b.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *CoRR*, abs/2601.18734, 2026.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Jiale Zhao, Jingwen Yang, Jianwei Lv, Kongcheng Zhang, Yihe Zhou, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general LLM reasoning. *CoRR*, abs/2508.16949, 2025.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *CoRR*, abs/2605.11182, 2026.

A Related Work

Rubric-based RL. Rubric-based RL extends RLVR to open-ended tasks whose quality cannot be verified by exact answers or executable tests. By decomposing response quality into explicit dimensions such as factuality, completeness, safety, evidence grounding, and task utility, they provide more interpretable and controllable supervision than holistic judgments (Arora et al., 2025; Du et al., 2025). Existing rubric-based RL methods typically use an LLM judge to score individual criteria and aggregate these scores into a scalar reward for GRPO (Gunjal et al., 2025; Viswanathan et al., 2025; Wu et al., 2025; Dineen et al., 2025). Subsequent work improves this paradigm through better reward aggregation, criterion reweighting, and fine-grained credit assignment, making rubric signal more discriminative and robust (Chen et al., 2026; Lan, 2026; Xu et al., 2026; Zhang et al., 2026a). A parallel line of work uses rubric feedback to enhance exploration. RuscaRL (Zhou et al., 2025) conditions rollout generation on rubric guidance, while HeRL (Zhang et al., 2026b) uses hindsight feedback from failed trajectories to construct revised rollouts. Although these methods help discover behaviors associated with previously unmet criteria, they rely on privileged information during rollout generation, introducing a training-inference mismatch, and also overlook another crucial failure of suppressed criteria. In contrast, CriPO recovers criterion-level supervision for both failure modes through an on-policy framework, without relying on externally guided rollouts.

On-Policy Self-Distillation. On-Policy Self-Distillation (OPSD) uses the current policy under privileged or feedback-augmented contexts as a self-teacher, converting auxiliary information into dense token-level supervision while avoiding off-policy teacher mismatch (Agarwal et al., 2024; Zhao et al., 2026; Hübotter et al., 2026). Despite its promise, dense OPSD is often unstable due to noisy token-level gradients, teacher–student inconsistency, and privileged-information leakage (Yang et al., 2026; Kim et al., 2026b; Zhu et al., 2026; Li et al., 2026a; Kim et al., 2026a). Recent work therefore increasingly combines OPSD with RLVR, using reference solutions, verified traces, or environment feedback to improve sparse-reward optimization (Yang et al., 2026; Lu et al., 2026; Pan et al., 2026). Most of these studies focus on verifiable domains such as mathematical reasoning, where privileged supervision is relatively well aligned with the objective.

Recent works have also begun exploring OPSD for rubric-based post-training in open-ended tasks (Rezaei et al., 2026; Gu et al., 2026), while they treat self-distillation as the primary optimization objective rather than retaining a reward-grounded RL backbone. Concurrent work (Wang et al., 2026b) combines OPSD and GRPO sequentially, using OPSD before RL to obtain a more knowledgeable initialization. In contrast, CriPO integrates self-distillation directly into GRPO training and uses it only for targeted criterion-level correction.

B Detailed Experimental Settings

Dataset Details. For RaR-Medicine and RaR-Science (Gunjal et al., 2025), we first filter out overly easy samples to improve data efficiency. Specifically, we remove samples on which Qwen3-4B-Instruct achieves a reward higher than 0.9. This results in 15,658 training samples and 1,936 testing samples for RaR-Medicine, and 10,874 training samples and 1,365 testing samples for RaR-Science. The weights of criteria are all positive for the two training datasets where pitfall criteria are phrased in positive form following Gunjal et al. (2025). For HealthBench (Arora et al., 2025) and ResearchQA (Yifei et al., 2025), we randomly select a subset of 500 samples from each dataset for cross-domain evaluation. For LLMEval-Med, we follow the evaluation protocol of the original project (Zhang et al., 2025), which contains 667 medical questions across five categories.

Evaluation Details. During training, we use Qwen3-32B as the rubric judge, and we use GPT-4o-mini as the rubric judge in our main evaluation. The sampling temperature for the judge model is set as 0.1 for low uncertainty, and we disable the thinking mode for Qwen3-32B. The judge prompt template is shown in Figure 14. For evaluation, we set the policy model's generation parameters to rollout_n$=1$, top_k$=-1$, top_p$=0.8$, and temperature$=0.7$ across all benchmarks.

Implementation Details. The training is implemented based on the verl (Sheng et al., 2025) framework, and the OPSD implementation follows SDPO (Hübotter et al., 2026). Hardware configuration details are withheld due to organizational confidentiality restrictions. Detailed training configurations

of CriPO and all baselines are provided in Table 4. We use Qwen3-32B to synthesize the training target for SFT, which is guided by the full rubric. The target generation prompt for SFT and the teacher prompt for OPSD are shown in Figure 13. The KL divergence in the OPSD loss is implemented as a top-k vocabulary approximation, where we use $k = 64$.

Table 4: Training configurations across different methods and model backbones.

Settings	Hyperparameters
SFT	n_gpus $= 8$, train_batch_size $= 64$, micro_batch_size_per_gpu $= 1$ learning_rate $= 1e{-5}$, lr_scheduler_type $=$ cosine max_length $= 8192$, warmup_steps $= 40$, total_training_steps $= 400$
RL Sampling	rollout_n $= 8$, top_k $= -1$, top_p $= 1.0$, temperature $= 1.0$ max_prompt_length $= 1024$, max_response_length $= 5120$
RL Training	n_gpus $= 8$, train_batch_size $= 64$, ppo_mini_batch_size $= 32$ ppo_micro_batch_size_per_gpu $= 2/1$ for RaR-Medicine/RaR-Science learning_rate $= 1e{-6}$, kl_loss_coef $= 1e{-3}$, total_training_steps $= 400$
RuscaRL/HeRL	The hyperparameters in RuscaRL/HeRL follow their default settings.
OPSD	rollout_n$= 1$, top_k$= -1$, top_p$= 0.8$, temperature$= 0.7$ loss type: forward KL, privileged information: full rubric
CriPO	$\gamma = 0.95$, $\alpha = 0.1$, $\tau_{\text{flip}} = 0.1$, $\beta = 0.05$, $\kappa = 2048$, $\zeta = 10$

More Implementation Details for CriPO. To keep advantage correction targeted, we use two cutoffs. Let $\kappa = 2048$ be the maximum response length eligible for advantage correction and $\zeta = 10$ the maximum number of key tokens selected per rollout. If $|y_i| > \kappa$, we leave its token advantages unchanged. Otherwise, among the tokens satisfying the teacher-probability threshold in Eq.(7), we retain at most the ζ tokens with the lowest teacher-assigned probabilities. Advantage correction can increase response length by preserving or reinforcing selected tokens; correcting too many tokens could amplify this effect and destabilize training. Longer responses also make criterion-specific token localization less reliable, motivating the response-length cutoff. Together, these cutoffs constrain the scope of advantage correction while retaining its most salient token-level signals.

We apply further safeguards to the behavior-injection self-teacher. We retain at most the three highest-weighted unexplored criteria when constructing the teacher prompt. We also clamp each token's KL value to a maximum of 10 before contribution-based filtering and gradient updates to limit outliers. Finally, we observe larger KL values near the beginning and end of responses (Figure 7). Early tokens may reflect the teacher instruction, such as a preamble before revising the response. At the end, the teacher is conditioned on a suboptimal student prefix and may favor post-hoc corrections rather than reliable continuation. We therefore mask the first and last 1% of response tokens before contribution-based filtering. Overall, **we use the same settings for the above parameters and γ, α, τ_{flip}, β, κ, and ζ across all model and dataset settings, without per-setting tuning**; see Appendix C.9 for a detailed hyperparameter analysis.

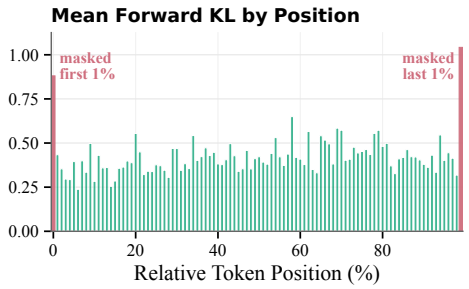

Figure 7: **Token-wise forward KL across positions.** Divergence is higher near response boundaries.

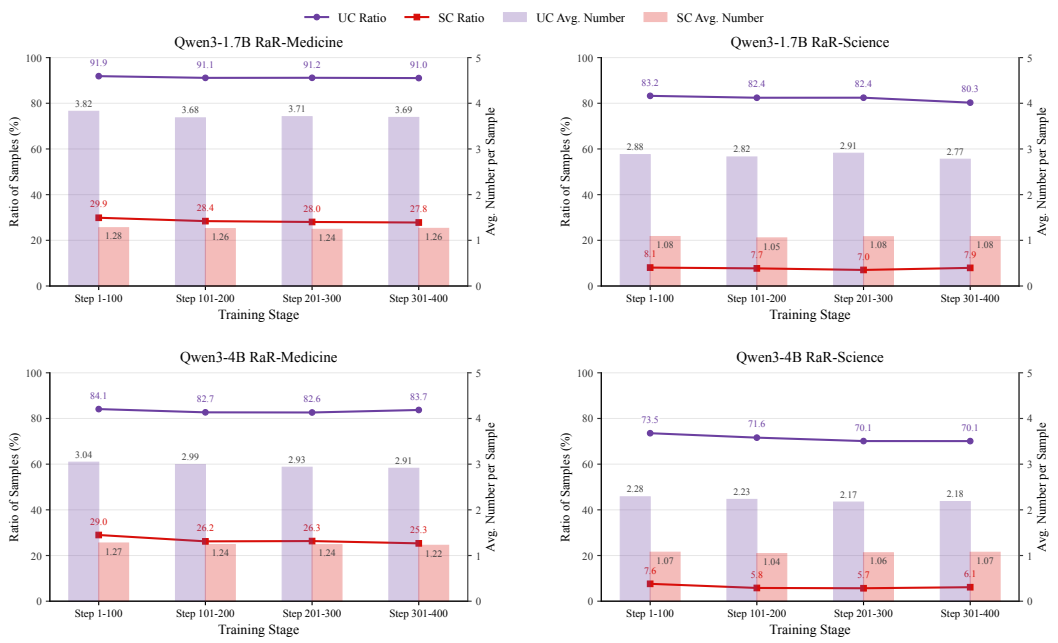

Figure 8: Statistics of unexplored and suppressed criteria of GRPO on different settings.

Table 5: Statistics of unexplored and suppressed criteria obtained with different judge models.

Judge Model	UC Ratio	SC Ratio	UC Count	SC Count
Qwen3-32B	0.81	0.31	3.10	1.40
GPT-4o-mini	0.75	0.25	2.90	1.19
GPT-5.6-luna	0.88	0.34	3.39	1.32

C Supplementary Experiments

C.1 Statistics of Unexplored and Suppressed Criteria

Statistics on Different Settings. We further examine UC and SC throughout GRPO training across two model scales and two domains. Figure 8 groups the 400 training steps into four stages and reports both the occurrence ratio and average count of each failure mode. UC remains widespread in every setting: its ratio ranges from 70.1% to 91.9%, with an average count of 2.17–3.82. Scaling from Qwen3-1.7B to Qwen3-4B generally lowers UC, particularly on RaR-Science, but does not eliminate it. SC is less frequent and more domain-dependent, with ratios of 25.3–29.9% on RaR-Medicine and 5.7–8.1% on RaR-Science, while its average count remains between 1.04 and 1.28. Both failure modes decrease only modestly as training proceeds. For example, on Qwen3-4B, the SC ratio on RaR-Medicine decreases from 29.0% in steps 1–100 to 25.3% in steps 301–400, while the UC ratio on RaR-Science decreases from 73.5% to 70.1%. In other settings, the statistics are nearly flat; the UC ratio of Qwen3-1.7B on RaR-Medicine remains around 91% throughout training. These results show that standard GRPO alone does not resolve either failure mode: UC and SC persist across model scales, domains, and training stages, motivating explicit criterion-level correction.

Statistics using Different LLM Judges. To determine whether these observations depend on the choice of LLM judge, we use GPT-4o-mini and GPT-5.6-luna, in addition to the Qwen3-32B judge used during training, to independently evaluate the same Qwen3-4B rollouts for the first 64 samples of the RaR-Medicine testing set. As shown in Table 5, all three judges identify UC in the majority of samples: the UC ratio ranges from 75% to 88%, with an average count of 2.90–3.39. They also consistently detect SC, with occurrence ratios of 25%–34% and average counts of 1.19–1.40.

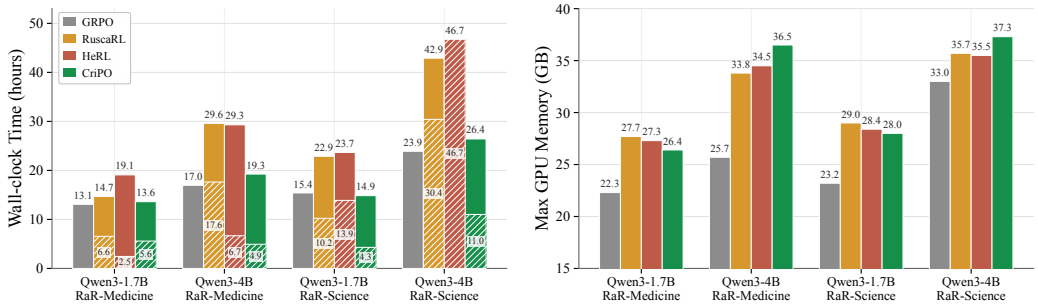

Figure 9: **Computation cost comparison. Left:** Total wall-clock training time of GRPO, RuscaRL, HeRL, and CriPO across model scales and training domains. Numbers above the bars denote total training time, while the hatched regions and their labels indicate the time required to reach the final performance achieved by GRPO; GRPO itself serves as the reference. **Right:** Maximum per-GPU memory usage under the same settings.

Although the absolute estimates vary—GPT-4o-mini is the most conservative, whereas GPT-5.6-luna reports the highest UC and SC ratios—the qualitative conclusion is consistent across judges. Thus, the prevalence of both failure modes is not an artifact of the particular judge used during training.

C.2 Computation Cost Analysis

Comparisons on Training Time. Figure 9 compares the wall-clock training time and maximum GPU memory of CriPO with GRPO and rubric-guided baselines. With at most one additional forward pass per rollout, the total training time of CriPO remains close to that of GRPO: Averaged over the four settings, CriPO requires 18.6 hours, only about 7% more than GRPO (17.4 hours), while being substantially faster than RuscaRL (27.5 hours) and HeRL (29.7 hours). The runtime differences arise from the distinct training pipelines: RuscaRL assigns a large weight to rubric-guided generation during the early training stage, which substantially increases both rollout entropy and response length. Because autoregressive generation cost grows with sequence length, these longer rollouts lead to higher wall-clock time. HeRL incurs a different source of overhead: each sample passes through two rollout stages and requires two calls to the LLM judge. Switching between rollout modes and performing the additional judge evaluation both add substantial latency. In contrast, CriPO reuses the original on-policy rollout and primarily adds one extra actor forward pass to obtain the self-teacher signal, explaining why its total runtime remains much closer to GRPO.

More importantly, the faster convergence of CriPO offsets its additional per-step cost. As indicated by the hatched regions in Figure 9 (left), CriPO reaches GRPO's final performance in only 5.6, 4.9, 4.3, and 11.0 hours across the four settings. These values correspond to a 2.2×–3.6× wall-clock speedup over the time required to fully train GRPO. CriPO is also the fastest non-GRPO method to reach the GRPO reference performance in three of the four settings; the exception is Qwen3-1.7B on RaR-Medicine, where HeRL reaches it in 2.5 hours. The above computation cost comparison shows that CriPO provides a favorable time-to-quality tradeoff rather than merely improving final performance through a larger computation budget.

Comparisons on GPU Memory. Figure 9 (right) shows that this efficiency comes with a moderate memory overhead. The maximum per-GPU memory of CriPO is 26.4, 36.5, 28.0, and 37.3 GB, compared with 22.3, 25.7, 23.2, and 33.0 GB for GRPO. This increase mainly arises from the top-k vocabulary approximation used to compute forward KL, which requires retaining probabilities or logits for the selected vocabulary entries during the actor forward pass. Nevertheless, peak memory remains below 40 GB in every setting and is generally comparable to RuscaRL and HeRL. Overall, CriPO introduces manageable memory overhead while delivering substantially faster convergence and stronger final performance.

Table 6: Experimental results (%) across medicine and science benchmarks using Qwen3-32B as judge. The best results are marked in **bold**, and the second-best are underlined. Arrows indicate changes over the base model. All results are evaluated on models trained after 400 steps.

Method	Medicine			Science		Avg.
	RaR-Medicine	HealthBench	LLMEval-Med	RaR-Science	ResearchQA	
Qwen3-1.7B	33.3 ↑ 0.0	52.0 ↑ 0.0	24.4 ↑ 0.0	44.2 ↑ 0.0	51.7 ↑ 0.0	41.1 ↑ 0.0
+ SFT	34.1 ↑ 0.8	36.1 ↓ 15.9	13.5 ↓ 10.9	42.7 ↓ 1.5	39.0 ↓ 12.7	33.1 ↓ 8.0
+ OPSD	35.5 ↑ 2.2	45.8 ↓ 6.2	23.5 ↓ 0.9	46.7 ↑ 2.5	47.4 ↓ 4.3	39.8 ↓ 1.3
+ GRPO	37.3 ↑ 4.0	54.3 ↑ 2.3	24.1 ↓ 0.3	51.6 ↑ 7.4	59.1 ↑ 7.4	45.3 ↑ 4.2
+ RuscaRL	39.5 ↑ 6.2	54.2 ↑ 2.2	23.7 ↓ 0.7	55.2 ↑ 11.0	**61.5** ↑ 9.8	46.8 ↑ 5.7
+ HeRL	**41.6** ↑ 8.3	55.8 ↑ 3.8	**27.3** ↑ 2.9	52.9 ↑ 8.7	57.3 ↑ 5.6	47.0 ↑ 5.9
+ CriPO	41.2 ↑ 7.9	**57.8** ↑ 5.8	25.6 ↑ 1.2	**55.6** ↑ 11.4	61.3 ↑ 9.6	**48.3** ↑ 7.2
Qwen3-4B	49.7 ↑ 0.0	59.5 ↑ 0.0	38.1 ↑ 0.0	62.2 ↑ 0.0	57.5 ↑ 0.0	53.4 ↑ 0.0
+ SFT	47.0 ↓ 2.7	48.1 ↓ 11.4	26.5 ↓ 11.6	59.1 ↓ 3.1	48.3 ↓ 9.2	45.8 ↓ 7.6
+ OPSD	53.7 ↑ 4.0	51.4 ↓ 8.1	33.1 ↓ 5.0	65.2 ↑ 3.0	52.5 ↓ 5.0	51.2 ↓ 2.2
+ GRPO	57.8 ↑ 8.1	62.7 ↑ 3.2	39.1 ↑ 1.0	72.4 ↑ 10.2	67.9 ↑ 10.4	60.0 ↑ 6.6
+ RuscaRL	62.6 ↑ 12.9	64.7 ↑ 5.2	39.3 ↑ 1.2	75.5 ↑ 13.3	70.6 ↑ 13.1	62.5 ↑ 9.1
+ HeRL	60.1 ↑ 10.4	64.6 ↑ 5.1	39.3 ↑ 1.2	71.0 ↑ 8.8	69.2 ↑ 11.7	60.8 ↑ 7.4
+ CriPO	**63.1** ↑ 13.4	**65.6** ↑ 6.1	**39.7** ↑ 1.6	**76.8** ↑ 14.6	**72.6** ↑ 15.1	**63.6** ↑ 10.2

C.3 Results using Different LLM Judges

The main results in Table 1 use GPT-4o-mini as the evaluation judge. To examine whether the conclusions depend on this evaluator, we further evaluate the same set of trained models using Qwen3-32B. As shown in Table 6, the results lead to the same overall conclusions as those obtained with GPT-4o-mini: SFT and standalone OPSD are less reliable, reward-grounded RL improves the base models, and CriPO achieves the strongest average performance at both model scales.

Specifically, SFT and standalone OPSD underperform the corresponding base models on average. GRPO raises the average score from 41.1 to 45.3 on Qwen3-1.7B (+4.2) and from 53.4 to 60.0 on Qwen3-4B (+6.6), confirming the benefit of reward-grounded optimization. RuscaRL and HeRL provide further improvements through rubric-guided exploration. Built on GRPO, CriPO further increases the average score to **48.3** on Qwen3-1.7B (+**3.0** over GRPO) and **63.6** on Qwen3-4B (+**3.6**). It also outperforms the strongest competing baseline by 1.3 and 1.1 points, respectively. The consistent advantage of CriPO under both GPT-4o-mini and Qwen3-32B indicates that the benefits of targeted criterion-level correction are robust to the choice of evaluation judge.

C.4 Results on Other Models

To assess whether the effectiveness of CriPO generalizes beyond the base Qwen3-1.7B and Qwen3-4B models, we further evaluate it on an instruction-tuned model (**Qwen3-4B-Instruct**), a larger model (**Qwen3-8B**), and a model from a different family (**Llama3.2-3B-Instruct**) (Team, 2024). As shown in Table 7, CriPO consistently outperforms both the initial model and GRPO across all model backbones and benchmarks. On RaR-Medicine, CriPO improves over the initial models by 11.1, 11.9, and 22.1 points, exceeding GRPO by 2.8, 2.1, and 10.3 points on Qwen3-4B-Instruct, Qwen3-8B, and Llama3.2-3B-Instruct, respectively. On HealthBench, CriPO yields gains of 5.3, 8.5, and 10.5 points over the corresponding initial models and surpasses GRPO by 1.2, 2.6, and 3.3 points. The particularly large improvement on Llama3.2-3B-Instruct demonstrates that the benefits of criterion-level correction transfer beyond the Qwen model family.

C.5 Multi-Seed Results

Since a complete training run requires substantial computation time (see Appendix C.2), the main tables report one run for each model–domain configuration. As a sanity check for training variability, we repeat both CriPO and GRPO with three seeds (40, 41, and 42) using Qwen3-4B on RaR-Medicine

Table 7: Experimental results (%) on additional model backbones using Qwen3-32B as judge.

Model	Method	RaR-Medicine	HealthBench
Qwen3-4B-Instruct	Initial	56.6 ↑ 0.0	62.8 ↑ 0.0
	GRPO	64.9 ↑ 8.3	66.9 ↑ 4.1
	CriPO	**67.7** ↑ 11.1	**68.1** ↑ 5.3
Qwen3-8B	Initial	61.3 ↑ 0.0	62.6 ↑ 0.0
	GRPO	71.1 ↑ 9.8	68.5 ↑ 5.9
	CriPO	**73.2** ↑ 11.9	**71.1** ↑ 8.5
Llama3.2-3B-Instruct	Initial	39.9 ↑ 0.0	44.1 ↑ 0.0
	GRPO	51.7 ↑ 11.8	51.3 ↑ 7.2
	CriPO	**62.0** ↑ 22.1	**54.6** ↑ 10.5

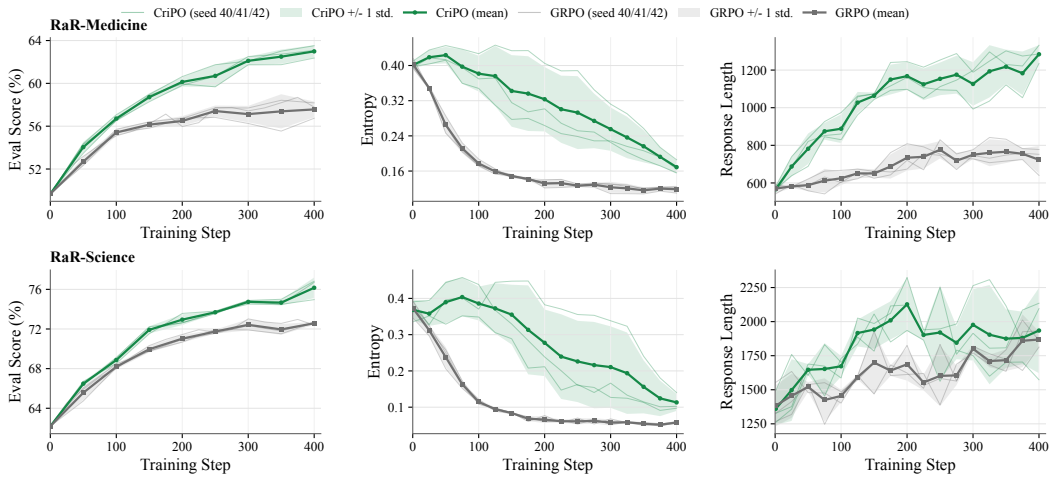

Figure 10: **Three-seed training dynamics on Qwen3-4B.** We compare CriPO and GRPO on RaR-Medicine (top) and RaR-Science (bottom) using seeds 40, 41, and 42. Thin lines denote individual runs, thick lines denote the seed mean, and shaded regions show mean ± one standard deviation. Columns report evaluation score, policy entropy, and response length.

and RaR-Science. Figure 10 presents the individual trajectories together with their seed mean and ± 1 standard-deviation band. The evaluation-score trajectories are consistent across seeds: on both domains, every CriPO run converges to a higher score than the corresponding GRPO runs, and the uncertainty bands are narrow relative to the gap between the two methods in the later stages of training. Moreover, the three-seed means confirm the performance advantage of CriPO over GRPO observed in the single-seed main results, indicating that the reported gains are not driven by an anomalously favorable run. The other training statistics exhibit the same qualitative behavior observed in the main analysis: CriPO maintains higher entropy for most of training and generally produces longer responses than GRPO. Overall, these experiments serve as a sanity check that the performance advantage of CriPO over GRPO is robust to seed variation.

C.6 Comparison of OPSD Loss Objectives

We compare four loss objectives for the standalone OPSD baseline, using the full rubric as the privileged information ξ in Eq.(4). Let $p_t = \pi_\theta(\cdot \mid x, y_{<t})$ be the student distribution, $q_t = \mathrm{sg}(\pi_\theta(\cdot \mid x, \xi, y_{<t}))$ the self-teacher distribution, and $m_t = (p_t + q_t)/2$. Three objectives use

full-distribution, KL-style losses at each token position:

$$\ell_t^{\mathrm{KLFKL}} = D_{\mathrm{KL}}(q_t \| p_t)\,, \tag{11}$$

$$\ell_t^{\mathrm{KLRKL}} = D_{\mathrm{KL}}(p_t \| q_t)\,, \tag{12}$$

$$\ell_t^{\mathrm{KLJSD}} = \tfrac{1}{2} D_{\mathrm{KL}}(p_t \| m_t) + \tfrac{1}{2} D_{\mathrm{KL}}(q_t \| m_t)\,. \tag{13}$$

Here KLFKL is forward KL, KLRKL is the reverse KL shown in Eq.(4), and KLJSD is Jensen–Shannon divergence. We also test policy-gradient-style reverse KL (PGRKL), whose surrogate uses only the log probabilities of the sampled token y_t:

$$\ell_t^{\mathrm{PGRKL}} = \mathrm{sg}[\log p_t(y_t) - \log q_t(y_t)] \log p_t(y_t). \tag{14}$$

For each variant, the OPSD objective is the expectation of $|y|^{-1} \sum_{t=1}^{|y|} \ell_t$ over on-policy rollouts, as in Eq.(4). At a fixed prefix, with $y_t \sim p_t$ and the sampled token held fixed during backpropagation, the expected gradient of the PGRKL surrogate equals that of reverse KL. Unlike the three full-distribution losses, PGRKL requires only student and teacher log probabilities for the sampled tokens, without obtaining probabilities over the vocabulary.

Table 8 reports results for Qwen3-4B trained on RaR-Medicine. KLFKL achieves the highest RaR-Medicine score, 53.7, improving over the initial model by 4.0 points and exceeding KLRKL, KLJSD, and PGRKL by 1.7, 2.2, and 5.4 points, respectively. On the cross-domain HealthBench evaluation, KLFKL ties with KLRKL for the highest OPSD score of 51.4, but all four variants fall 8.1–8.6 points below the initial model's 59.5. This drop suggests that standalone OPSD fits the training-domain rubrics too closely and transfers poorly to rubrics from a different domain. We therefore implement KLFKL for the OPSD baseline in both main-result tables (Tables 1 and 6). We also ablate the loss type within CriPO; see Appendix C.9 for details.

Table 8: Comparison of OPSD loss objectives using Qwen3-32B as judge.

Model	Method	RaR-Medicine	HealthBench
Qwen3-4B	Initial	49.7 ↑ 0.0	59.5 ↑ 0.0
	OPSD-KLFKL	**53.7** ↑ 4.0	**51.4** ↓ 8.1
	OPSD-KLRKL	<u>52.0</u> ↑ 2.3	**51.4** ↓ 8.1
	OPSD-KLJSD	51.5 ↑ 1.8	50.9 ↓ 8.6
	OPSD-PGRKL	48.3 ↓ 1.4	<u>51.2</u> ↓ 8.3

C.7 OOD Generalization on Instruction Following.

To evaluate whether CriPO preserves general instruction-following ability beyond its rubric-based training domains, we conduct out-of-domain evaluation on IFEval and IFBench. As shown in Table 9, both CriPO variants improve Qwen3-1.7B on the two benchmarks: medicine training yields gains of 0.4 and 2.0 points, while science training yields larger gains of 1.3 and 3.3 points on IFEval and IFBench, respectively. On Qwen3-4B, the changes are mixed but remain modest: medicine training decreases IFEval and IFBench by 0.6 and 0.4 points, whereas science training decreases IFEval by 1.3 points but improves IFBench by 2.0 points. Overall, these results suggest that CriPO largely preserves instruction-following performance on these two benchmarks.

C.8 Adaptation to DAPO

We further examine whether CriPO can be incorporated into a stronger RL backbone that explicitly encourages exploration. Specifically, we replace GRPO with DAPO and apply the same criterion-level behavior injection and token-level advantage correction on top of it. For DAPO, we set the upper importance-ratio clipping bound to 0.28 and disable KL regularization. To compute the length-penalty score, we set the maximum length to the rollout response-length limit and the cache length to one-quarter of that limit. We retain the token-mean loss aggregation already used in our GRPO implementation. As shown in Table 10, DAPO outperforms GRPO on RaR-Medicine and the two cross-domain medicine benchmarks, improving RaR-Medicine, HealthBench, and LLMEval-Med by

Table 9: **OOD evaluation (%) on instruction-following benchmarks.** CriPO (Medicine) and CriPO (Science) are trained on RaR-Medicine and RaR-Science, respectively. Arrows indicate changes over the corresponding initial model.

Model	Method	IFEval	IFBench
Qwen3-1.7B	Initial	69.7	18.7
	CriPO (Medicine)	70.1 ↑ 0.4	20.7 ↑ 2.0
	CriPO (Science)	71.0 ↑ 1.3	22.0 ↑ 3.3
Qwen3-4B	Initial	80.6	25.7
	CriPO (Medicine)	80.0 ↓ 0.6	25.3 ↓ 0.4
	CriPO (Science)	79.3 ↓ 1.3	27.7 ↑ 2.0

Table 10: **Adaptation of CriPO to DAPO on Qwen3-4B.** We report performance on medicine benchmarks (judged by Qwen3-32B) and out-of-domain instruction-following (IF) benchmarks. All values are on the percentage scale, and changes are measured against the initial model.

Method	Medicine			IF	
	RaR-Medicine	HealthBench	LLMEval-Med	IFEval	IFBench
Initial	49.7 ↑ 0.0	59.5 ↑ 0.0	38.1 ↑ 0.0	80.6 ↑ 0.0	25.7 ↑ 0.0
GRPO	57.8 ↑ 8.1	62.7 ↑ 3.2	39.1 ↑ 1.0	79.1 ↓ 1.5	26.0 ↑ 0.3
CriPO	63.1 ↑ 13.4	65.6 ↑ 6.1	39.7 ↑ 1.6	80.0 ↓ 0.6	25.3 ↓ 0.4
DAPO	66.8 ↑ 17.1	68.9 ↑ 9.4	44.1 ↑ 6.0	78.4 ↓ 2.2	24.3 ↓ 1.4
DAPO+CriPO	68.4 ↑ 18.7	69.2 ↑ 9.7	45.7 ↑ 7.6	79.3 ↓ 1.3	27.3 ↑ 1.6

9.0, 6.2, and 5.0 points, respectively. Adding CriPO to DAPO yields further gains of 1.6, 0.3, and 1.6 points, showing that CriPO can also benefit a more exploration-oriented RL backbone.

Because DAPO uses a larger upper clipping bound and disables KL regularization, its stronger policy updates may also increase the risk of drifting from the initial model's general capabilities. We therefore evaluate out-of-domain instruction following on IFEval and IFBench. DAPO decreases performance from 80.6 to 78.4 on IFEval and from 25.7 to 24.3 on IFBench. In contrast, DAPO+CriPO retains 79.3 on IFEval and improves IFBench to 27.3, outperforming DAPO by 0.9 and 3.0 points, respectively. We conjecture that the improved generalization comes from the fine-grained nature of CriPO: its corrections are concentrated on localized, criterion-relevant tokens instead of broadly altering the entire response distribution. These results highlight the benefit of token-level credit assignment and show that CriPO can improve both task performance and generalization when adapted to a more exploration-oriented RL objective.

C.9 Ablation Studies

This section examines CriPO at three levels: the contributions of behavior injection (BI) and advantage correction (AC), design choices within each module, and hyperparameter sensitivity. All results use Qwen3-32B as the evaluation judge. The overall ablation includes Qwen3-1.7B and Qwen3-4B trained on RaR-Medicine and RaR-Science for the corresponding benchmark groups. The sub-module and hyperparameter analyses use Qwen3-4B trained on RaR-Medicine.

Overall Ablation. Table 11 compares GRPO, GRPO with behavior injection (CriPO-U), and the full CriPO. Adding behavior injection raises the average score from 45.3 to 47.7 on Qwen3-1.7B (+2.4) and from 60.0 to 62.1 on Qwen3-4B (+2.1). Adding advantage correction on top of behavior injection further increases the averages to 48.3 (+0.6) and 63.6 (+1.5), respectively. The full method achieves the highest average at both model scales, although individual benchmark scores do not improve uniformly. These results show that both components contribute to the gains over GRPO.

Table 11: **Overall ablation of CriPO.** BI denotes behavior injection and AC advantage correction; GRPO+BI is CriPO-U, and GRPO+BI+AC is the full CriPO.

Model	GRPO	BI	AC	Medicine			Science		Avg.
				RaR-Medicine	HealthBench	LLMEval-Med	RaR-Science	ResearchQA	
Qwen3-1.7B	✓			37.3	54.3	24.1	51.6	59.1	45.3
	✓	✓		39.8	55.9	**27.0**	54.6	**61.4**	47.7
	✓	✓	✓	**41.2**	**57.8**	25.6	**55.6**	61.3	**48.3**
Qwen3-4B	✓			57.8	62.7	39.1	72.4	67.9	60.0
	✓	✓		61.4	64.2	37.5	75.8	71.5	62.1
	✓	✓	✓	**63.1**	**65.6**	**39.7**	**76.8**	**72.6**	**63.6**

Table 12: **Ablation of sub-module designs in CriPO.** We report evaluation scores together with the last step's rollout entropy and average response length in thousands of tokens. A dash indicates that the training encounters CUDA OOM due to exploded length.

Module	Sub-Module	Method	Eval Performance		Training Dynamics	
			RaR-Medicine	HealthBench	Entropy	Lengths (k)
–	–	CriPO	63.1	65.6	0.19	1.29
Behavior Injection	KL Loss Type	KLJSD	60.1	63.1	0.11	1.26
		KLRKL	52.5	62.1	0.08	0.72
		PGRKL	64.9	69.2	0.73	2.63
	γ	CriPO w/o selection	58.3	63.5	0.11	1.02
	Dynamic β	CriPO w/o dynamic	65.4	68.4	0.56	1.91
		CriPO-U	61.4	64.2	0.18	0.83
		CriPO-U w/o dynamic	60.7	64.3	0.18	0.85
	Other Designs	Full-rubric teacher	62.1	60.2	0.65	1.87
		Rubric-feedback teacher	62.1	65.3	0.22	1.14
		Worst-1 rollout	–	–	0.81	1.48
Advantage Correction	τ_{flip}	All flip	62.0	65.9	0.20	0.97
		All mask	62.6	66.4	0.21	1.10
	Selection Strategy	Random selection	61.6	65.1	0.20	0.96

Ablation on Sub-Modules. Table 12 examines the main design choices within behavior injection and advantage correction. For behavior injection, the divergence choice substantially changes both evaluation performance and training dynamics. KLJSD and KLRKL underperform the default forward-KL (KLFKL) design on both benchmarks and exhibit lower entropy and shorter responses. PGRKL obtains higher evaluation scores, but it also sharply increases entropy from 0.19 to 0.73 and response length from 1.29k to 2.63k, indicating a substantially more aggressive exploration regime. Removing contribution-based token filtering decreases RaR-Medicine and HealthBench by 4.8 and 2.1 points, respectively, supporting the need to filter diffuse teacher signals. Removing the dynamic β schedule similarly raises the evaluation scores but produces much higher entropy and longer responses; also, CriPO-U without the schedule suffers from a performance drop on average compared to CriPO-U, which suggests that dynamic weighting effectively regulates the behavior-injection signal. Among teacher designs, conditioning on the full rubric notably hurts cross-domain generalization on HealthBench and also leads to substantially higher entropy, while the rubric-feedback teacher (i.e., the teacher is conditioned on the criteria that the best rollout does not present) remains closer to CriPO but is still weaker on RaR-Medicine. These suggest that adding the least necessary criteria information for GRPO is the most stable way. Using the worst rollout instead of the best one suffers from exploded entropy and response length, indicating that distilling on the bad rollout is not reliable.

For advantage correction, applying the same operation to every localized token is less balanced than the selective masking-and-flipping rule: all-flip and all-mask each improve one metric relative to the other, but neither consistently dominates CriPO across the two benchmarks. More importantly,

Table 13: **Hyperparameter sensitivity of CriPO.** We report evaluation scores together with the last step's rollout entropy and average response length in thousands of tokens. A dash indicates that the training encounters CUDA OOM due to exploded length.

Module	Sub-Module	Method	Eval Performance		Training Dynamics	
			RaR-Medicine	HealthBench	Entropy	Lengths (k)
Behavior Injection	γ	$\gamma = 1.0$ (CriPO w/o selection)	58.3	63.5	0.11	1.02
		$\gamma = 1.0,\ \beta = 0.1$	58.0	63.3	0.10	0.99
		$\gamma = 0.95$ (CriPO)	63.1	65.6	0.19	1.29
		$\gamma = 0.9$	–	–	1.65	2.01
		$\gamma = 0.9,\ \beta = 0.025$	62.8	64.9	0.18	1.07
		$\gamma = 0.5$	–	–	3.11	1.90
		$\gamma = 0.5,\ \beta = 0.01$	–	–	1.37	1.93
		$\gamma = 0.5,\ \beta = 0.005$	60.0	64.3	0.15	1.01
	β	$\beta = 0.1$	65.5	68.9	1.38	3.18
		$\beta = 0.05$ (CriPO)	63.1	65.6	0.19	1.29
		$\beta = 0.01$	58.6	63.7	0.12	1.07
Advantage Correction	τ_{flip}	$\tau_{\text{flip}} = 1.0$	61.8	65.3	0.21	1.28
		$\tau_{\text{flip}} = 0.1$ (CriPO)	63.1	65.6	0.19	1.29
	Cutoff	$\kappa = 2048, \zeta = 10$ (CriPO)	63.1	65.6	0.19	1.29
		$\kappa = 5120$	–	–	0.74	1.77
		$\zeta = 5120$	62.8	65.1	0.21	1.21

replacing counterfactual-teacher localization with random token selection (ensuring the same number of selected tokens) lowers RaR-Medicine from 63.1 to 61.6 and HealthBench from 65.6 to 65.1. This result shows that the benefit does not arise merely from modifying additional token advantages; it depends on locating criterion-relevant tokens. Moreover, since we apply advantage correction only to negative-advantage rollouts, there exists a theoretical corner case that our method may not cover: a suppressed criterion receives no direct advantage correction when all of its satisfying rollouts have zero advantage. However, we did not observe such a case in the training runs examined, and also, restricting correction to negative-advantage rollouts keeps the counterfactual teacher disjoint from behavior injection on the highest-advantage rollout, preserving at most one additional teacher evaluation per rollout. Overall, the ablations show that the sub-module choices jointly control the trade-off among task performance, exploration, and response length.

Hyperparameter Sensitivity. Table 13 shows that the behavior-injection hyperparameters control a clear trade-off between evaluation performance and training dynamics. Setting $\gamma = 1.0$, which removes contribution-based token filtering, lowers performance on both benchmarks. Increasing the strength for CriPO w/o selection with a larger β of 0.1 does not improve its performance, indicating the effectiveness of contribution-based token filtering does not simply stem from increasing the magnitude of the loss. Reducing γ concentrates supervision on fewer high-contribution tokens, but using $\gamma = 0.9$ or 0.5 with a comparatively large distillation coefficient produces substantially higher entropy and longer responses and causes CUDA OOM during training. Reducing the behavior injection loss coefficient β stabilizes these settings: $\gamma = 0.9, \beta = 0.025$ approaches the default performance with similar entropy, whereas the more aggressive $\gamma = 0.5, \beta = 0.005$ remains weaker. The β sweep exhibits the same pattern. A small value of 0.01 provides insufficient improvement, while $\beta = 0.1$ achieves higher evaluation scores at the cost of a large increase in entropy and response length. The default $\gamma = 0.95$ and $\beta = 0.05$ therefore provide a more balanced operating point.

The advantage-correction hyperparameters are less sensitive around the default setting. Increasing τ_{flip} from 0.1 to 1.0 reduces RaR-Medicine by 1.3 points and HealthBench by 0.3 points, supporting a mild positive correction rather than a large flipped advantage. Increasing the selected-token cutoff to $\zeta = 5120$ causes a small performance drop, while extending the eligible response length to $\kappa = 5120$ substantially raises entropy and response length, resulting in CUDA OOM. Overall, the default $\tau_{\text{flip}} = 0.1$, $\kappa = 2048$, and $\zeta = 10$ is robust and avoids overly aggressive corrections on long responses or large token sets. For the teacher-probability threshold α, Figure 11 shows a highly polarized distribution of $p^T_{i,t}(y_{i,t})$ during Qwen3-4B training on RaR-Medicine: most tokens receive

probabilities near one, whereas a much smaller subset lies near zero. We therefore set $\alpha = 0.1$ to select this low-probability tail, where the counterfactual teacher disfavors the observed token, while excluding the large mass of high-probability tokens.

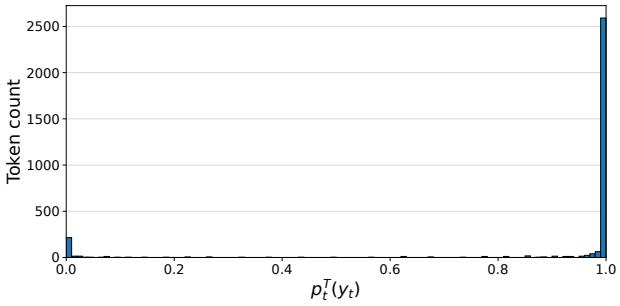

Figure 11: **Counterfactual-teacher probabilities assigned to sampled tokens.** Histogram of $p_{i,t}^T(y_{i,t})$ during Qwen3-4B training on RaR-Medicine. The 3,344 tokens with probability exactly one are omitted for readability. The concentration near one and smaller low-probability tail motivate the threshold $\alpha = 0.1$ for key-token selection.

D Case Studies

D.1 Case Study on Medical Task

Figure 12 compares GRPO and CriPO on a question about a 7-day-old infant born at 26 weeks of gestation who has grossly bloody stool, abdominal distention, and increasing oxygen requirements. The best initial diagnostic step is option A, an abdominal series, given the concern for necrotizing enterocolitis (NEC). GRPO recognizes NEC as a possibility and even notes that imaging is often preferred before endoscopy in preterm infants; accordingly, it satisfies the rubric criterion for recognizing the clinical presentation. Nevertheless, it repeatedly selects option B, fiberoptic endoscopy, because it treats direct visualization as the more definitive way to evaluate gastrointestinal bleeding. This final choice conflicts with its own caution about the risks of endoscopy and leaves the other four criteria unmet: GRPO neither selects and clearly states the correct answer nor provides NEC-specific radiographic findings or a consistent rationale for starting with a fast, non-invasive test.

By contrast, CriPO connects the infant's extreme prematurity and clinical signs to suspected NEC, compares the diagnostic options, and selects the abdominal series as the first step. It explains that abdominal radiography can reveal findings such as portal venous gas and bowel abnormalities, while providing a rapid, non-invasive assessment. It also explains why fiberoptic endoscopy is inappropriate as an initial test in this premature neonate because of its procedural risks. The response ends with the explicit answer "A. Abdominal series," satisfying all five criteria shown in the figure. This case illustrates how the additional response length can contain criterion-relevant diagnostic reasoning that supports the final decision.

D.2 Case Study of Token Selection for Suppressed Criteria.

To characterize the suppressed criteria encountered in practice, Table 14 groups them into seven categories and reports a representative criterion and descriptive token-selection statistics for each. The analysis covers 1,641 suppressed-criterion rollout assignments on the testing set of RaR-Medicine using Qwen3-4B. A rollout can be associated with suppressed criteria from multiple categories, so the category rows do not represent disjoint sets of rollouts. These criteria extend beyond medical facts to conciseness, avoidance, reasoning, safety, answer presentation, and style; the table reports their counts and descriptive token-selection statistics. Note that avoidance criteria defined by the absence of particular content pose a challenge for token localization: a response satisfying such a criterion may contain no token that directly expresses the omission; see Bad Case 2 in Table 15.

Table 15 gives concrete examples of what the selected tokens capture. In Good Cases 1–2, structural and discourse markers (`###`, `Explanation`, and `Other`) point to an explanatory span or list, while

Table 14: **Statistics for different types of suppressed criteria.** "#Criteria" reports the number of criteria that occurred, and "#Tokens" reports the mean number of selected tokens per rollout–criterion-category pair. The five abbreviated columns report the proportions of formatting/structural boundaries (F), knowledge/semantic content (K), logical/discourse connectors (L), grammatical function words (G), and numbers/option labels (N).

Criterion Category	Representative Criterion	#Criteria	#Tokens	F	K	L	G	N
Medical Facts & Task Content	"Highlights the use of corticosteroids, specifically mentioning intravenous methylprednisolone and oral prednisone."	327	4.0	45%	23%	18%	13%	2%
Conciseness & Redundancy Control	"Remains concise and avoids unnecessary detail in the explanation."	178	8.4	43%	20%	16%	19%	2%
Negative & Avoidance Criteria	"Does not mention distinguishing between fine detail and gross movements, which confuses the parvocellular with the magnocellular pathway."	116	3.7	44%	25%	18%	12%	1%
Explanation & Reasoning	"Explains the correlation between upper abdominal pain and bloody diarrhea with the diagnosis."	55	4.3	45%	20%	19%	15%	2%
Safety & Uncertainty	"Mentions that certain conditions may require heightened caution and monitoring during ECT."	15	4.1	51%	19%	18%	11%	2%
Answer Presentation & Ordering	"Includes a clear statement 'The final answer is (D)'."	11	5.7	58%	10%	20%	10%	1%
Style & Formatting	"Uses an empathetic tone in discussing sexually transmitted infections to ensure sensitivity."	6	7.4	32%	47%	10%	11%	0%

Table 15: **Representative good and bad cases of suppressed-criterion token localization.** Selected key tokens are highlighted in their local response spans. ∅ indicates that no key token is selected.

Case Type	User Query	Suppressed Criterion	Key Token(s)	Rollout Span with Key Token(s)
Good Case 1	What drug used in the treatment of osteoporosis acts by both decreasing bone resorption and inducing new bone formation?	Provides a concise explanation without unnecessary details.	`###`, `Explanation`	RANKL inhibitor—specifically Denosumab. **`### Explanation`**: Denosumab is a monoclonal antibody that targets RANKL.
Good Case 2	According to the modified Jones criteria, what is one of the minor criteria for the diagnosis of Rheumatic Fever?	Mentions other minor criteria like fever and certain inflammatory blood test results.	`Other`, `Fever`	**`Other`** minor criteria include: – **`Fever`** – Macrocyclic erythema (a type of skin rash) – Subcutaneous nodules – Choreiform movements – Elevated ESR or CRP
Good Case 3	What is the appropriate treatment for a child with pauci-immune crescentic glomerulonephritis?	Highlights the use of corticosteroids, specifically mentioning intravenous methylprednisolone and oral prednisone.	`Pred`	**`Pred`**nisone (oral or intravenous) is often used to reduce inflammation.
Good Case 4	A 35-year-old sexually active male presents with a painless penile vesicle and inguinal lymphadenopathy. The infection has been definitively diagnosed, involving an organism known to exist in both extracellular and intracellular forms. What is the causative organism?	References the patient's age and sexual activity as context for the infection.	`3`	Clues in the question: – **3**5-year-old sexually active male
Bad Case 1	A 32-year-old man has progressive severe headaches, seizures, projectile vomiting, and low-grade fever after otitis media and mastoiditis. What is it?	Points out that the patient's age can be relevant to the differential diagnosis.	`are`	Otitis media and mastoiditis **`are`** ear infections that can ascend into the middle cranial fossa and spread to the meninges.
Bad Case 2	A 23-year-old man experienced unexpected blood loss during a donor nephrectomy and was subsequently transfused with O negative packed red blood cells. He then developed generalized pruritus, expiratory wheezing, and multiple pink, edematous wheals. Given his history of mild recurrent infections as a child, what underlying immunological condition is most likely responsible for these symptoms?	Does not mention Selective IgA Deficiency as the likely immunological condition.	∅	The most likely underlying immunological condition is C3 deficiency. (No key token is selected)

`Fever` marks relevant medical content. In Good Case 3, the wordpiece `Pred` in `Prednisone` identifies more specific content associated with the respective criteria, and in Good Case 4, the selected digit 3 in "35-year-old" marks the patient-age information explicitly referenced by the criterion, illustrating localization of a numerical detail within a relevant response span. We also demonstrate two types of bad cases. The generic word `are` is selected even though it does not convey

the criterion about the patient's age; for the absence-based criterion, no token is selected from a response that omits the specified diagnosis, so that it cannot be repaired by advantage correction. These examples suggest that counterfactual-teacher probabilities can highlight criterion-related cues, but the selected tokens do not always faithfully represent the criterion.

E PSEUDO-CODE OF CRIPO

Algorithm 1 shows the pseudo-code of CriPO.

Algorithm 1 CriPO: Criterion-Distilled Policy Optimization

Require: Policy π_θ; rubric criteria $\mathcal{C}$; rollout number G; hyperparameters $\gamma, \alpha, \tau_{\text{flip}}, \beta, \kappa, \zeta$

repeat
  Sample rollouts $\{y_i\}_{i=1}^{G}$ for prompt x, evaluate them with rubric criteria $\mathcal{C}$
  Compute rewards $\{R_i\}$, group advantages $\{A_i\}$ and identify Criteria $\mathcal{C}_u, \mathcal{C}_s$
  Initialize token advantages $\tilde{A}_{i,t} \leftarrow A_i$ and OPSD loss $\mathcal{L}_{\text{OPSD}} \leftarrow 0$
  ▷ **Behavior Injection for Unexplored Criteria**
  if $\mathcal{C}_u \neq \emptyset$ **then**
    Select the highest-advantage rollout y^*, build teacher and compute forward-KL d_t
    Select tokens $\mathcal{T}^u$ following Eq.(5) and compute $\mathcal{L}_{\text{OPSD}}$ following Eq.(6)
  ▷ **Advantage Correction for Suppressed Criteria**
  if $\mathcal{C}_s \neq \emptyset$ **then**
    for each rollout y_i with $A_i < 0$, $|y_i| \leq \kappa$, and $\mathcal{C}_s^i \neq \emptyset$ **do**
      Build one counterfactual teacher using all criteria in $\mathcal{C}_s^i$
      Select at most ζ lowest-teacher-probability tokens from Eq.(7) as $\mathcal{T}_i^s$
      Compute the corrected token-level advantage $\tilde{A}_{i,t}$ following Eq.(8)
  Compute $\mathcal{L}_{\text{GRPO}}$ following Eq.(9)
  Compute $\mathcal{L}_{\text{CriPO}}$ following Eq.(10) and update θ
until converged

F LIMITATIONS

Despite its gains, CriPO inherits several constraints of rubric-based RL. Its criterion-level signals depend on the quality and coverage of the rubrics and their judgments: missing, ambiguous, or incorrectly scored criteria can distort the identification of unexplored and suppressed criteria, and consequently misdirect behavior injection or advantage correction. Our experiments use curated rubrics in medicine and science, so the robustness of CriPO to lower-quality rubric settings remains to be established. Furthermore, the rubrics are fixed throughout training, although the policy's strengths and failure modes change. Criteria that were informative early on may become less useful, while newly emerging weaknesses may remain uncovered. We leave the study of CriPO under low-quality rubrics and the development of adaptive rubrics that evolve with the policy to provide more targeted supervision to future works.

G PROMPT TEMPLATES

The prompt of the teacher in the OPSD baseline, as well as for target generation in SFT, is shown in Figure 13, and the LLM judge prompt is shown in Figure 14.

Question: A 7-day-old premature infant born at 26 weeks of gestation now has a grossly bloody stool, abdominal distention, and increasing oxygen requirements. choose the best initial diagnostic step in the evaluation of the patient's apparent GI hemorrhage. A. Abdominal series B. Fiberoptic endoscopy C. Apt test D. Routine stool culture

Rubric Criteria Status (GRPO / CriPO)

- C1: Identifies an abdominal series (X-ray) as the best initial diagnostic step. (✗/✓)
- C2: Clearly states that the final answer is A. Abdominal series. (✗/✓)
- C3: Recognizes suspected necrotizing enterocolitis (NEC) in a premature infant with bloody stool and abdominal distention. (✓/✓)
- C4: Explains that abdominal X-ray can reveal NEC findings such as pneumatosis intestinalis or portal venous gas. (✗/✓)
- C5: Indicates that abdominal series is a fast, non-invasive initial test and avoids invasive procedures such as fiberoptic endoscopy. (✗/✓)

Response of GRPO ✗

Best Initial Diagnostic Step:
The best initial step is fiberoptic endoscopy if it is safe and feasible. However, in preterm infants, endoscopy is often avoided due to risks. Instead, the initial step is often imaging or lab tests that are less invasive and more informative.
But in the context of suspected GI bleeding (especially in a preterm infant), the most appropriate initial step is:
B. Fiberoptic endoscopy
- However, in preterm infants, endoscopy is not typically performed as the initial step. Instead, imaging (like abdominal X-ray or ultrasound) is used first to evaluate for structural abnormalities (e.g., NEC, duodenal atresia).
But if endoscopy is performed, it is the gold standard for diagnosing GI bleeding. However, due to risks, it is not the first step.
Correct Answer:
B. Fiberoptic endoscopy is the best initial diagnostic step if it is performed (despite risks), because it is direct and definitive. However, in preterm infants, endoscopy is often avoided and imaging (like abdominal X-ray or ultrasound) is preferred as the initial step.
But the question asks for the best initial diagnostic step in the evaluation of the patient's apparent GI hemorrhage. Among the options given, fiberoptic endoscopy is the most specific and direct method, even though it is not always used first in preterm infants.
Final Answer:
B. Fiberoptic endoscopy
Reason: It is the most specific and definitive method for diagnosing GI bleeding, even though it is not always the first step in preterm infants. It allows direct visualization and identification of the source of bleeding. However, in practice, endoscopy is often avoided in neonates due to risks, and imaging (like abdominal X-ray or ultrasound) is preferred as the initial step. But among the options given, B is the best.
Answer: B

Response of CriPO ✓

Let's analyze the case and determine the best initial diagnostic step for this 7-day-old premature infant with symptoms of gross bloody stool, abdominal distention, and increasing oxygen requirements.
Patient Profile:
- Age: 7 days
- Gestational age: 26 weeks (very preterm)
- Symptoms: Grossly bloody stool (possible GI bleeding), Abdominal distention, Increasing oxygen requirements (suggests possible respiratory distress or underlying systemic illness)
- Concern: GI hemorrhage
Key Considerations:
1. Prematurity: Very preterm infants (≤32 weeks) are at higher risk for gastrointestinal (GI) complications, including: Intestinal ischemia, Hemorrhage, Neonatal necrotizing enterocolitis (NEC)
2. Gross bloody stool is a red flag for NEC, which is a common and serious complication in preterm infants. NEC is characterized by: Blood in the stool, Abdominal distension, Fever (though not always present), Poor feeding, Hypotension or hypoxia
3. Increasing oxygen requirements could indicate: Systemic infection (e.g., sepsis), Hemodynamic instability due to bleeding, Respiratory distress from underlying illness
Initial Diagnostic Steps:
The goal is to confirm the presence of GI bleeding and rule out NEC, while avoiding interventions that could worsen the condition.
Option A: Abdominal series (X-ray)
- What it shows: Air in the intestines (pneumoperitoneum), which can indicate perforation.
- Use in NEC: A non-contrast abdominal X-ray is the initial imaging study for suspected NEC. It can show: Air in the bowel wall (perforation), Bowel distension, Portal venous gas (a specific finding in NEC)
- Advantages: Fast, non-invasive, and readily available.
- Disadvantages: It may not always detect early NEC, and it can be misleading (e.g., false positives for pneumatosis intestinalis).
- Conclusion: This is a reasonable initial step to evaluate for perforation or other structural abnormalities.
Option B: Fiberoptic endoscopy
- What it shows: Direct visualization of the GI tract.
- Use in NEC: Endoscopy is not typically used in neonates because of the risk of trauma, difficulty in intubation, and the potential to worsen bleeding.
- Conclusion: Not recommended as the initial step in a neonate with suspected NEC. It is usually reserved for confirmed cases or when other investigations are inconclusive.
Option C: Apt test (Immunofixation of fecal occult blood)
- What it shows: Detects hidden blood in the stool.
- Use in NEC: This is a sensitive test for detecting blood loss, but it is not specific for NEC. It can be positive in other conditions (e.g., infection, meconium staining).
- Conclusion: While it can help confirm blood in the stool, it is not the initial step. It is more of a confirmatory test after other evaluations.
Option D: Routine stool culture
- What it shows: Identifies bacterial pathogens in the stool.
- Use in NEC: While infection can contribute to NEC, stool culture is not the initial step. It is used to rule out sepsis after other investigations.
- Conclusion: Not the initial diagnostic step for GI bleeding.
Final Answer:
A. Abdominal series (X-ray)
Rationale:
- The initial step in evaluating a preterm infant with suspected NEC is a non-contrast abdominal X-ray to look for pneumoperitoneum or portal venous gas, which are specific findings of NEC.
- This is non-invasive, quick, and critical for early diagnosis and guiding treatment (e.g., stopping feeds, starting antibiotics, etc.).
- Endoscopy is not used initially due to risks.
- Apt test and stool culture are secondary steps.
Correct Answer: A. Abdominal series

Figure 12: **Case study on medical reasoning.** CriPO produces a more complete reasoning process than GRPO by exploring rubric-relevant clinical cues and diagnostic rationales.

Please respond to the following user query:

{user_query}

Evaluation criteria that a good response should satisfy:

{all_criteria}

After understanding these evaluation criteria, provide your own thorough response to the problem. Address the criteria naturally without explicitly referencing them.

Figure 13: Prompt of the teacher in OPSD baseline and target generation in SFT.

Given a user prompt, a generated response, and a list of quality rubrics, please evaluate the response against EACH rubric.
For each rubric,
- Mark "PRESENT" if the criterion is satisfied, or "NOT_PRESENT" if it is not. For example, given the response "Apples are red", the rubric "Mentions apples" is PRESENT, "Does not mention strawberries" is also PRESENT since the response doesn't mention strawberries and "Mentions oranges" is NOT_PRESENT. Also, "Avoids mentioning strawberries" is PRESENT because the response doesn't mention strawberries. However, "Avoids mentioning apples" is NOT_PRESENT because the response mentions apples.
- 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 NOT PRESENT. Only return PRESENT if all of the criteria are met.
- One important exception to the above bullet point is that if a rubric 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. That is, there are no partial credit for any of the criteria.
Start your response with a valid JSON object that starts with "```json" and ends with "```".
The keys must be the numbers of the rubrics provided and the values must be either "PRESENT" or "NOT_PRESENT" based on your evaluation. Ensure the JSON is valid and contains no extra text or explanations.
Example response:

```json
{{
 "1": "PRESENT",
 "2": "NOT_PRESENT",
 "3": "PRESENT"
}}
```

<Prompt>
{prompt_str}
</Prompt>

<Response>
{response}
</Response>

<Rubrics>
{rubrics_str}
</Rubrics>

Figure 14: The LLM judge prompt. The judge evaluates model responses according to a predefined rubric and produces criterion-wise scores for reward computation.