RVPO: RISK-SENSITIVE ALIGNMENT VIA VARIANCE REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current critic-less RLHF methods aggregate multi-objective rewards via an arithmetic mean, leaving them vulnerable to constraint neglect: high-magnitude success in one objective can numerically offset critical failures in others (e.g., safety or formatting), masking low-performing "bottleneck" rewards vital for reliable multi-objective alignment. We propose Reward-Variance Policy Optimization (RVPO), a risk-sensitive framework that penalizes inter-reward variance during advantage aggregation, shifting the objective from "maximize sum" to "maximize consistency." Its LogSumExp (SoftMin) objective can be locally approximated by the mean reward minus an inter-objective variance penalty. We evaluate RVPO on rubric-based medical and scientific reasoning with up to 17 concurrent LLM-judged reward signals (Qwen2.5-3B/7B/14B) and on tool-calling with rule-based constraints (Qwen2.5-1.5B/3B). By discouraging the model from neglecting difficult constraints to exploit easier objectives, RVPO improves overall scores on HealthBench (0.254 vs. 0.207 for GDPO at 14B, averaged across five training seeds) and maintains competitive accuracy on GPQA-Diamond while reducing the late-stage degradation observed in other multi-reward methods, demonstrating that variance regularization mitigates constraint neglect across model scales without sacrificing general capabilities.

1 INTRODUCTION

Multi-objective reinforcement learning is essential for balancing competing goals in LLM alignment, yet current methods struggle to prioritize strict constraints alongside general performance. Recent critic-less RLHF methods like Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and Group Decoupled Policy Optimization (GDPO) (Liu et al., 2026) reduce memory overhead by eliminating the Value Network, but rely on arithmetic mean aggregation. This inherently assumes "more is better," leaving optimization vulnerable to **constraint neglect**: models can exploit high-variance metrics (e.g., verbosity) to mask failures in strict, low-variance constraints (e.g., safety or formatting) (Figure 1). While constrained optimization methods (Achiam et al., 2017; Dai et al., 2023) and post-hoc model merging (Rame et al., 2023) offer alternatives, they require explicit constraint specification or separate policies, undermining the efficiency of critic-less methods. Constraint neglect can arise in any multi-objective RL setting. In group-relative methods, the linear aggregation used to compute advantages cannot distinguish a generation that satisfies all constraints from one that offsets critical failures with greater performance on easier objectives.

To address this, we introduce **Reward-Variance Policy Optimization (RVPO)**, a risk-sensitive framework that penalizes inter-reward variance.

Our core contributions are as follows:

1. We identify and empirically demonstrate the constraint neglect vulnerability inherent in mean-aggregated, critic-less multi-objective RL.
2. We introduce RVPO and show via Taylor expansion that the LogSumExp (SoftMin) objective can be locally approximated by the mean reward minus an inter-objective variance penalty, with a risk coefficient k smoothly interpolating between mean and min aggregation.

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.

3. We validate RVPO across two multi-objective paradigms: LLM-judged rubric criteria (5–17 rewards), where RVPO improves HealthBench scores relative to GDPO across five seeds per scale and reduces late-training degradation, and rule-based tool-calling (2 rewards), where RVPO accelerates convergence on the bottleneck format constraint while preserving execution accuracy.

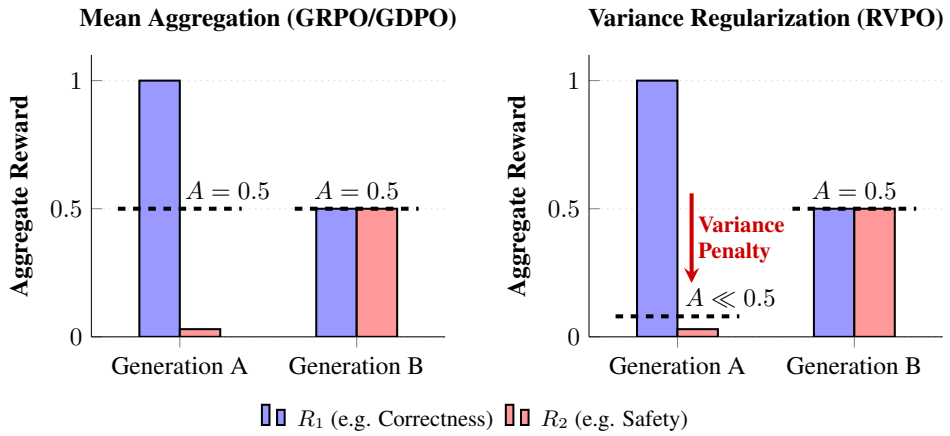

Figure 1: **Constraint neglect in multi-objective RLHF. (Left)** Mean aggregation (GRPO/GDPO) treats outputs with critical constraint failures (Gen A) as mathematically identical to balanced outputs (Gen B), blinding the optimizer to critical failures. **(Right)** RVPO applies a soft-min operator to penalize inter-reward variance, heavily discounting Gen A to enforce bottleneck constraints.

2 Related Work

Reinforcement Learning from Human Feedback (RLHF) and Reward Hacking. The alignment of large language models (LLMs) relies heavily on RLHF to fine-tune models toward human preferences (Ouyang et al., 2022; Rafailov et al., 2023). However, standard approaches are susceptible to reward hacking, where the policy exploits misspecifications in the reward model to achieve high scores while degrading generation quality (Chen et al., 2024b; Dai et al., 2025; Gao et al., 2023). A well-documented instance is length gaming, where models exploit verbosity. While recent works address this via length-normalized rewards, explicit penalties, or disentangled reward modeling (Meng et al., 2024; Park et al., 2024; Chen et al., 2024a), RVPO applies a complementary aggregation-level approach: its SoftMin objective reduces compensation across normalized reward channels. It does not identify unreliable reward signals or solve reward misspecification.

Multi-Objective Alignment and Structured Evaluation. In practice, LLMs must balance multiple competing objectives. Moving beyond monolithic scalar rewards, recent work decomposes feedback into structured rubric criteria (Gunjal et al., 2025; Hashemi et al., 2024), multi-attribute reward models (Wang et al., 2024), checklists (Viswanathan et al., 2025), and explicit safety rules (Mu et al., 2024). A common approach to optimizing these decomposed signals is linear scalarization (Li et al., 2025; Hayes et al., 2022), which permits gains in one objective to compensate for failures in another. Alternative paradigms treat this as a constrained optimization problem requiring explicit constraint thresholds and separate cost models (Achiam et al., 2017; Paternain et al., 2019; Dai et al., 2023), utilize post-hoc model merging (Rame et al., 2023), or resolve gradient conflicts across objectives (Chen et al., 2026). Inference-time approaches steer frozen models via targeted intervention (Nguyen et al., 2025) or unified preference-aware reward models (Lin et al., 2025). Unlike these, RVPO modifies the training objective itself: the variance penalty implicitly elevates bottleneck objectives within a single training run, smoothly relaxing a min-max objective that optimizes the worst-performing reward channel.

Critic-Less and Group-Relative Optimization. Standard PPO requires maintaining a Value Network, incurring substantial memory overhead. To alleviate this, critic-less alternatives utilize group-relative advantage estimation (Ahmadian et al., 2024), scaling successfully to emergent reasoning

(Guo et al., 2025; Yu et al., 2025). However, when extended to multi-objective settings, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) suffers from scale dominance. Group Decoupled Policy Optimization (GDPO) (Liu et al., 2026) addresses this by normalizing individual reward models independently before summation. While GDPO prevents scale dominance, its reliance on the arithmetic mean leaves it vulnerable to loss compensation—where a critical failure in one constraint is offset by success in another. RVPO directly builds upon the GDPO framework, replacing its arithmetic mean with a variance-penalized objective to mitigate this constraint neglect.

Risk-Sensitive and Variance-Penalized RL. The theoretical foundation of RVPO is rooted in risk-sensitive Markov Decision Processes (Howard & Matheson, 1972), where agents optimize exponential-utility or variance-constrained objectives rather than expected returns (Hu & Leung, 2023; Zhong et al., 2026). In particular, mellowmax (Asadi & Littman, 2017) applies a LogSumExp soft-max over actions for smoother value estimation within a single-objective MDP. RVPO operates in a fundamentally different setting: the soft-min is applied across concurrent reward channels within the advantage computation. Where mellowmax smooths the policy's action selection, RVPO reshapes which reward signals drive the gradient. RVPO connects to risk-sensitive/exponential-utility objectives, isoelastic social-welfare aggregation (Atkinson, 1970), population-proportional preference aggregation (Kim et al., 2025), non-linear multi-objective scalarization (Roijers et al., 2013; Hayes et al., 2022), and maximin-style alignment (Chakraborty et al., 2024). Our formulation applies SoftMin to per-channel Z-normalized rewards during advantage computation.

3 Background: Reward Aggregation and Constraint Neglect

Critic-less RLHF methods eliminate the Value Network's memory burden by estimating advantages based on the intra-group relative performance of G sampled responses. However, the mathematical mechanism used to aggregate these multi-objective rewards fundamentally dictates the policy's vulnerability to constraint neglect.

GRPO and Scale Dominance. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) aggregates M distinct objectives by summing the raw rewards for each generation, and then normalizing this total score across the group:

$$A_{GRPO} = \frac{\sum_{j=1}^{M} R_j - \mu_{total}}{\sigma_{total}},$$

where R_j is the raw reward for the j-th objective, and μ_{total} and σ_{total} are the mean and standard deviation of the summed rewards computed across the G generations in the group. Because raw rewards are summed directly, metrics with naturally large variances or unbounded scales (e.g., generation length or raw helpfulness) numerically dominate the advantage calculation. Small, sparse, or binary rewards (e.g., a strict penalty for a JSON schema violation) can be overwhelmed, hindering the model's learning of rigid constraints.

GDPO and Constraint Neglect. Group Decoupled Policy Optimization (GDPO) (Liu et al., 2026) resolves scale dominance by normalizing the rewards independently. For each reward model j, GDPO computes a standard score (Z_j) across the G generations, forcing all objectives into a scale-free distribution with zero mean and unit variance. These Z-scores are then aggregated via an arithmetic mean:

$$A_{GDPO} = \frac{1}{M} \sum_{j=1}^{M} Z_j = \mu_Z.$$

While GDPO ensures scale parity, its reliance on arithmetic summation introduces a subtle flaw: *loss compensation*. Because the objective strictly maximizes the mean, a catastrophic failure on a bottleneck constraint ($Z_{format} \ll 0$) can be perfectly offset by over-performance on an easily exploitable metric ($Z_{length} \gg 0$). This aggregation implicitly signals to the policy that a generation with extreme flaws and extreme peaks is mathematically equivalent to a safely balanced generation. Mean aggregation therefore permits compensation between training reward channels even after Z-normalization. Consistent with the constraint-neglect hypothesis, GDPO reaches mean

best-checkpoint Communication Quality of 43.1% versus Completeness of 12.3% at 7B across five training seeds on HealthBench (Section 6.1). To resolve this, the aggregation objective must shift from simply maximizing the mean to penalizing inter-objective disagreement.

4 REWARD-VARIANCE POLICY OPTIMIZATION

We propose a risk-sensitive aggregation framework that penalizes disagreement between reward models, encouraging the policy to respect bottleneck constraints. For a given rollout g from a group of G generations, let $Z_j^{(g)} = (R_j^{(g)} - \mu_j)/\sigma_j$ denote the standardized reward on objective j, where μ_j and σ_j are computed across the group. We choose the negative LogSumExp (SoftMin) aggregation objective:

$$A_{RVPO}^{(g)} = -\frac{1}{k} \ln \left(\frac{1}{M} \sum_{j=1}^{M} e^{-kZ_j^{(g)}} \right),$$

where $k > 0$ is the inverse temperature, which we term the **Risk Coefficient**. It smoothly interpolates between the mean and the minimum reward:

$$\lim_{k \to 0} A_{RVPO}^{(g)} = \mu_Z^{(g)} = A_{GDPO}^{(g)}, \qquad \lim_{k \to +\infty} A_{RVPO}^{(g)} = \min_j Z_j^{(g)}.$$

Here $\mu_Z^{(g)} = \frac{1}{M} \sum_j Z_j^{(g)}$ and $(\sigma_Z^{(g)})^2 = \frac{1}{M} \sum_j (Z_j^{(g)} - \mu_Z^{(g)})^2$ are the mean and variance across objectives, not across rollouts. For small $|k(Z_j^{(g)} - \mu_Z^{(g)})|$, the local approximation is

$$A_{RVPO}^{(g)} \approx \mu_Z^{(g)} - \frac{k}{2} (\sigma_Z^{(g)})^2.$$

The detailed Taylor expansion is in Appendix A.4. We also evaluate the explicit quadratic ablation $A_{RVPO\text{-}explicit}^{(g)} = \mu_Z^{(g)} - \beta(\sigma_Z^{(g)})^2$, with $\beta > 0$. Unlike this unbounded quadratic penalty, SoftMin approaches hard-min as k increases. Both formulations shift optimization from maximizing average performance to maximizing consistent performance across objectives.

Power-mean / Nash connection. Writing $\mathcal{M}_p(u) = (\frac{1}{M} \sum_j u_j^p)^{1/p}$ for the power mean of order $p \neq 0$, the exponential change of variables $u_j = e^{Z_j} > 0$ gives the exact identities

$$\log \mathcal{M}_p(e^Z) = \frac{1}{p} \log \left(\frac{1}{M} \sum_j e^{pZ_j} \right),$$

$$\log \mathcal{M}_{-k}(e^Z) = A_{RVPO}.$$

The geometric/Nash limit is

$$\mathcal{M}_0(u) = \lim_{p \to 0} \mathcal{M}_p(u) = \left(\prod_j u_j \right)^{1/M}, \qquad \log \mathcal{M}_0(e^Z) = \mu_Z = A_{GDPO}.$$

Thus GDPO and RVPO lie on one interpolation family: $p \to 0$ gives the geometric/Nash point on u, $p = -k < 0$ gives RVPO, and $p \to -\infty$ gives hard-min. The exponential transformation is essential because standardized rewards can be signed.

The sensitivity of the SoftMin advantage to each normalized reward is $\partial A_{RVPO}/\partial Z_j = w_j = e^{-kZ_j} / \sum_\ell e^{-kZ_\ell}$. For each generation, channels with lower normalized reward scores receive more weight, and that extra weight decreases as their scores improve relative to the other channels. Finite k retains contributions from several weak channels rather than assigning all pressure to one current minimum. This bottleneck-aware feedback makes compensation less profitable. The full procedure is summarized in Algorithm 1 (Appendix).

SoftMin is always at most the arithmetic mean across channels, and the gap grows with inter-objective disagreement; this is the intended penalty on uneven generations. The resulting scalar advantages are then whitened across the batch before the PPO update, which re-centers and rescales them.

Aggregating raw reward components before group normalization is a related alternative, as in GRPO-explicit. We normalize first because channels differ in scale and sparsity, making them comparable before SoftMin determines their relative influence.

5 Experimental Setup

To evaluate the efficacy of RVPO in mitigating constraint neglect, we benchmark our approach against GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026) across two multi-objective paradigms: LLM-judged constraints (Rubrics-as-Rewards) and deterministic, rule-based constraints (Tool Calling).

5.1 Environments and Reward Formulation

Rubrics-as-Rewards (LLM-Judged Constraints): We evaluate RVPO using the Rubrics-as-Rewards (RaR) framework (Gunjal et al., 2025), where the number of reward signals varies dynamically per prompt (5–17 criteria). We evaluate on two domains from Gunjal et al. (2025): RaR-Medicine (20k clinical reasoning prompts) and RaR-Science (20k graduate-level science prompts). Unlike prior RaR implementations that collapse evaluations into a single scalar, we treat each of the M criteria as an independent reward channel. This high-dimensional decomposition explicitly exposes the constraint neglect vulnerability (§3). For each criterion, `gpt-4o-mini` acts as the judge, outputting a binary satisfaction score. The RaR framework assigns categorical priority weights to each criterion, which we incorporate pre-normalization; we evaluate post-normalization weighting in Appendix A.6.

Tool Calling (Rule-Based Constraints): We utilize RLLA-4k (Qian et al., 2025), a curated subset of 4,000 tool-calling trajectories, to evaluate multi-step reasoning alongside rigid structural adherence. We define two competing reward signals: Execution Correctness, a continuous scalar evaluating tool and parameter matching against the ground truth, and Format Adherence, a strictly binary constraint verifying XML schema compliance.

5.2 Training Implementation and Baselines

All experiments were conducted on a single node with 8 NVIDIA H100 GPUs. We train Qwen2.5 models (Yang et al., 2024) using the `verl` (Sheng et al., 2024) and `TRL` frameworks: 1.5B and 3B for tool-calling; 3B, 7B, and 14B for rubrics-as-rewards. For rubrics, k and β were selected separately within Medicine and Science at 7B, favoring final-checkpoint performance over the highest early peak, then transferred unchanged to 3B and 14B. Medicine uses $k = 0.5 \rightarrow 2.0$, $\beta = 1.0$; Science uses $k = 1.0 \rightarrow 2.0$, $\beta = 0.5$. This was in-domain selection, without a separate held-out development set. Full training hyperparameters (e.g., learning rates, batch sizes, group sizes) are detailed in Appendix A.2.

Across all domains, we compare RVPO against standard GRPO (which sums raw rewards) and GDPO (which independently Z-normalizes channels before summing). For rubric-based RVPO, the risk coefficient k is linearly annealed from k_{start} to k_{end} (denoted $k = k_{\text{start}} \rightarrow k_{\text{end}}$). This shifts aggregation from a near-mean objective toward greater emphasis on bottleneck channels. In the rubrics setting, we include two single-reward baselines from the RaR framework (Gunjal et al., 2025): GRPO (Explicit), which aggregates per-criterion scores into a single weighted scalar; and GRPO (Implicit), the strongest baseline in the original work, where the LLM judge outputs a single holistic score.

5.3 Evaluation Methodology

For models trained on the RLLA-4k tool-calling dataset, we evaluate using the AST-based metrics of the Berkeley Function Call Leaderboard v3 (BFCL-v3) (Patil et al., 2025), averaging results across five independent runs per method. For models trained on RaR-Medicine, we evaluate multi-objective alignment using the full 5,000-example HealthBench framework (Arora et al., 2025), whose rubric scoring was validated against physician preferences. HealthBench explicitly scores models across five independent rubric axes: *Communication Quality, Instruction Following, Accuracy, Context*

Awareness, and *Completeness*. Finally, for models trained on RaR-Science, we evaluate loglikelihood accuracy on the GPQA-Diamond benchmark (Rein et al., 2024). For Medicine, we report mean and standard deviation over five distinct training seeds per method and scale. Checkpoints are evaluated every 50 steps over 300 steps (six evaluations); best is selected separately within each run and final is step 300. Science results remain single-run and are reported separately in Appendix A.3. GPQA-Diamond and BFCL-v3 serve as capability parity checks in the evaluated settings.

6 RESULTS

6.1 RUBRICS-AS-REWARDS

Table 1: **Five-seed RaR-Medicine results on HealthBench.** Mean ± standard deviation over five distinct training seeds per method and scale, evaluated on all 5,000 questions using the micro-average of per-question rubric scores. Best is selected within each run; final is step 300. Bold marks the highest displayed mean in each size/checkpoint column, including ties.

Size	Method	Best checkpoint	Final (300)
3B	GRPO (Implicit)	0.167 ± 0.008	0.072 ± 0.013
	GRPO (Explicit)	0.179 ± 0.007	0.064 ± 0.043
	GDPO	0.178 ± 0.011	0.111 ± 0.036
	RVPO-min ($k = \infty$)	$\mathbf{0.180 \pm 0.008}$	0.136 ± 0.012
	RVPO-explicit ($\beta = 1.0$)	0.172 ± 0.009	0.023 ± 0.025
	RVPO ($k = 0.5 \to 2.0$)	$\mathbf{0.180 \pm 0.005}$	$\mathbf{0.154 \pm 0.020}$
7B	GRPO (Implicit)	0.196 ± 0.022	$\mathbf{0.160 \pm 0.036}$
	GRPO (Explicit)	0.213 ± 0.011	0.035 ± 0.044
	GDPO	0.192 ± 0.006	0.022 ± 0.026
	RVPO-min ($k = \infty$)	0.190 ± 0.013	0.041 ± 0.070
	RVPO-explicit ($\beta = 1.0$)	$\mathbf{0.215 \pm 0.007}$	$\mathbf{0.160 \pm 0.035}$
	RVPO ($k = 0.5 \to 2.0$)	0.214 ± 0.010	0.152 ± 0.040
14B	GRPO (Implicit)	0.242 ± 0.011	0.143 ± 0.117
	GRPO (Explicit)	0.232 ± 0.006	0.039 ± 0.078
	GDPO	0.207 ± 0.032	0.042 ± 0.084
	RVPO-min ($k = \infty$)	0.243 ± 0.005	0.157 ± 0.077
	RVPO-explicit ($\beta = 1.0$)	0.222 ± 0.018	0.153 ± 0.080
	RVPO ($k = 0.5 \to 2.0$)	$\mathbf{0.254 \pm 0.017}$	$\mathbf{0.217 \pm 0.044}$

Table 1 reports the five-seed HealthBench comparison. RVPO has higher mean best- and final-checkpoint scores than GDPO at all three scales, reaching a mean best score of 0.254 ± 0.017 versus 0.207 ± 0.032 at 14B. The clearest pattern is improved end-of-training retention: final scores improve over GDPO by $0.043/0.130/0.175$ at 3B/7B/14B. RVPO also has higher mean scores than GRPO-explicit at both checkpoints across scales. At 7B, GRPO-implicit and RVPO-explicit remain competitive, with mean final scores of 0.160 versus RVPO's 0.152. Across repeated runs, RVPO reduces late-training degradation while remaining competitive at the best checkpoint.

Training diagnostics show persistent inter-rubric disagreement under GDPO: the standard deviation across active rubric Z-scores, averaged over generations and training, is $0.635 \pm 0.003/0.599 \pm 0.015/0.602 \pm 0.006$ at 3B/7B/14B (mean ± seed standard deviation). At the best checkpoint, Completeness is the weakest mean HealthBench axis at all scales, with Communication Quality $4.0/3.5/2.9\times$ higher. These summaries show persistent disagreement during training and imbalanced downstream performance; they do not establish a sample-level correlation between the two.

Figure 2 illustrates the per-axis trade-off in single-run 7B experiments. In these runs, GDPO scores highly on Communication Quality but poorly on Completeness and Context Awareness. RVPO improves Accuracy, Context Awareness, and Completeness while giving up some Communication Quality and Instruction Following performance. This illustrates how bottleneck improvements can accompany higher overall performance; Table 1 provides the five-seed overall comparison.

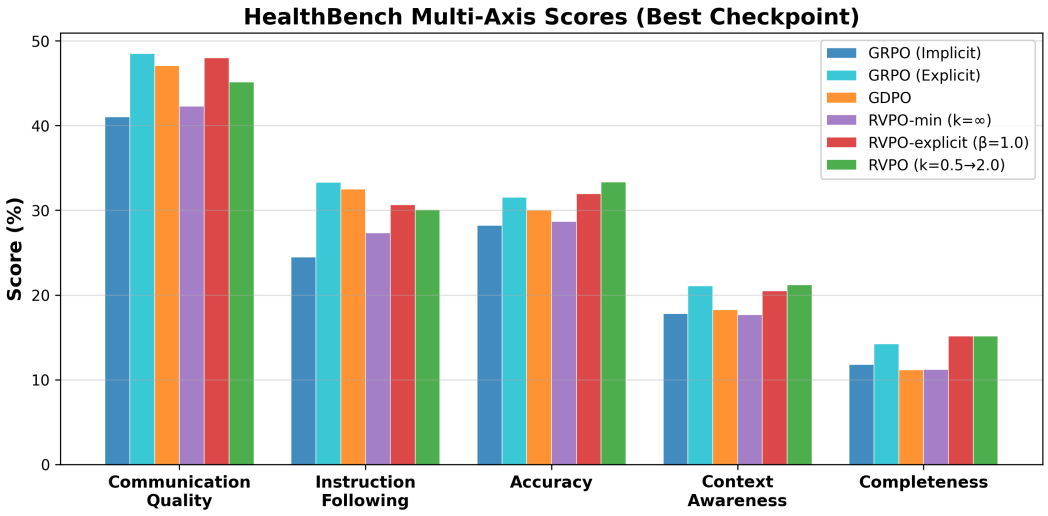

Figure 2: **Illustrative single-run per-axis comparison on HealthBench (Arora et al., 2025) (Medicine, Qwen2.5-7B).** Results at each method's best overall-HealthBench checkpoint, using the original single-run experiments. Five-seed overall results are reported in Table 1.

GPQA-Diamond evaluates whether balanced rubric optimization generalizes to a held-out reasoning benchmark. While RaR-Science covers graduate-level topics, GPQA tests specialized expert knowledge requiring different reasoning patterns. The single-run results on this small benchmark ($N = 198$; Appendix A.3) show broadly comparable accuracy, providing a parity check.

6.2 Tool Calling Dynamics

Table 2: **BFCL-v3 evaluation (Patil et al., 2025).** Qwen2.5-1.5B and 3B. Avg is the mean of the three reported subcategories; AST Sum evaluates single and simple function calls; Parallel and Par. Multiple evaluate the ability to invoke multiple functions simultaneously. All methods achieve comparable accuracies on the evaluated tool-calling tasks. Results averaged across five runs per method (run-to-run std $\approx 0.4\%$).

Size	Method	Avg (%)	AST Sum (%)	Parallel (%)	Par. Multiple (%)
1.5B	GRPO	70.4	71.7	70.3	69.2
	GDPO	**71.9**	**73.4**	71.7	**70.7**
	RVPO-explicit ($\beta = 1.0$)	71.3	72.5	71.0	70.5
	RVPO ($k = 1.0$)	**71.9**	72.9	**72.1**	70.6
3B	GRPO	80.3	81.3	77.8	81.9
	GDPO	80.7	81.6	78.8	81.7
	RVPO-explicit ($\beta = 1.0$)	**81.4**	**82.1**	**80.1**	**81.9**
	RVPO ($k = 1.0$)	80.4	81.5	78.4	81.3

In the tool-calling setting, execution correctness is a continuous signal and format adherence is a sparse, binary constraint. As shown in Figure 3, GRPO is highly sensitive to this disparity: the larger magnitude of the correctness reward overshadows the binary format reward, delaying formatting improvements until step 25. GDPO mitigates this scale disparity by normalizing the reward distributions (Z_j), leading to earlier formatting improvements.

However, GDPO plateaus before reaching full format compliance. This behavior illustrates the loss compensation effect of arithmetic mean aggregation: the policy learns that high execution scores can mathematically offset missing format constraints. In contrast, RVPO's variance penalty discounts

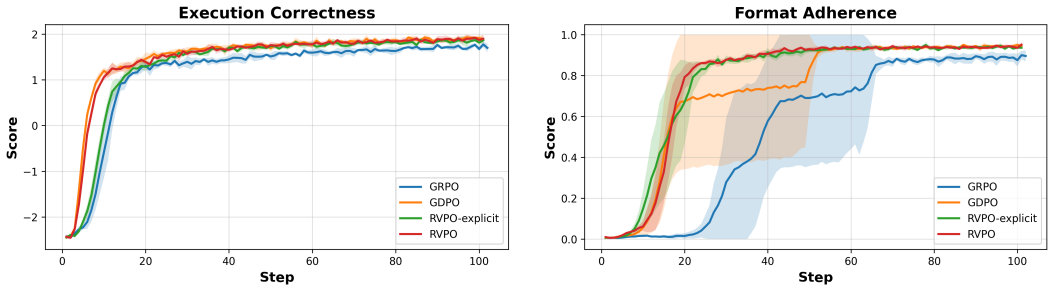

Figure 3: **Tool calling (RLLA) training dynamics (Qian et al., 2025).** Qwen2.5-1.5B training progression across five independent runs; solid lines show the mean and shaded regions ± 1 standard deviation. *(Left)* While mean-based baselines (GDPO and GRPO) successfully maximize execution correctness, they struggle to satisfy the strict format adherence constraint *(Right)*. In contrast, RVPO and RVPO-explicit enforce this bottleneck constraint via a variance penalty, achieving simultaneous convergence across both objectives.

outputs with high execution correctness but failed syntax checks, encouraging the policy to satisfy both objectives simultaneously. Consequently, RVPO shows improved convergence on the format metric while reducing the inter-run variance observed in the baselines.

Table 2 shows that all methods achieve comparable downstream accuracy on BFCL-v3 across 1.5B and 3B scales. This indicates that RVPO's primary benefit in this low-dimensional setting (M=2) is faster convergence on the bottleneck constraint during training.

6.3 Risk Coefficient Selection

The risk coefficient k serves as an explicit knob along the mean-to-min aggregation spectrum: low k optimizes average performance, high k prioritizes worst-case satisfaction. Unlike methods that train separate policies per objective and merge post-hoc (Rame et al., 2023), k parameterizes this trade-off within a single training run. The 7B Medicine selection favored the annealed $k = 0.5 \rightarrow 2.0$ schedule over the higher-peak but less stable static $k = 5.0$ setting. Appendix A.5 reports the single-run selection sweep and gives a training-statistics rule of thumb for the scale of k. The selected settings transfer unchanged to 3B and 14B; tool-calling uses static $k = 1.0$. We retain RVPO-explicit as an ablation; the single-run β sweep in Appendix A.8 illustrates its sensitivity to the penalty coefficient.

6.4 Controlled Corruption

We test a fixed weak rubric channel on 3B RaR-Medicine with five seeds per cell. Under symmetric label noise at $p = 0.5$, the target receives 1.194 ± 0.010 times equal-share aggregation weight under RVPO, versus 2.095 ± 0.042 under hard-min. RVPO assigns less weight to the corrupted channel than hard-min. Post-normalization weighting can separately encode a declared priority. Appendix A.7 reports the complete conditions and results.

7 Limitations and Future Work

The primary limitation of RVPO is the sensitivity of the risk coefficient k. The choice of k remains sensitive to reward space dimensionality, group size G (our experiments use G=4 for tool-calling and G=16 for rubrics), and inter-objective conflict, motivating future work on adaptive scheduling. The settings selected at 7B transfer across scales, but broader validation of the training-statistics heuristic remains open (Appendix A.5). Additionally, as demonstrated in Appendix A.6, RVPO's bottleneck prioritization is driven by per-channel normalized scores rather than declared priority; post-normalization weighting can encode priorities (Appendix A.7). RVPO may also amplify noise from unreliable reward channels, as the soft-min focuses optimization on whichever objective produces the lowest Z-scores regardless of whether this reflects policy weakness or reward model noise. Our corruption tests cover independent per-generation errors at 3B; Appendix A.7 details their scope

and downstream effects. Constrained-optimization and Pareto-style comparisons remain useful future directions.

8 Conclusion

We introduced Reward-Variance Policy Optimization (RVPO), which addresses constraint neglect in mean-aggregated multi-objective RL via a LogSumExp variance penalty. RVPO scales from two rule-based rewards to 17 concurrent LLM-judged criteria across multiple model sizes, improving mean best- and final-checkpoint HealthBench scores over GDPO across five seeds per scale while reducing late-training degradation. The risk coefficient k provides an explicitly tunable parameter for navigating multi-objective trade-offs within a single training run. As alignment increasingly relies on decomposed reward signals—from rubric criteria to safety constraints—robust aggregation that discourages individual objectives from being silently neglected becomes essential for reliable deployment.

AI Use Statement

We used LLMs to help launch and monitor experiments and summarize and interpret their results. We also used them to help find related work, discuss research ideas, clarify mathematical explanations, and draft and edit parts of the paper. The rubric experiments use an LLM judge, as described in the experimental setup. The authors reviewed the AI-assisted work and take responsibility for the paper's content.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

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

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pp. 243–252. PMLR, 2017.

Anthony B. Atkinson. On the measurement of inequality. *Journal of Economic Theory*, 2(3):244–263, 1970. doi: 10.1016/0022-0531(70)90039-6.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Bedi, and Mengdi Wang. MaxMin-RLHF: Alignment with diverse human preferences. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6116–6135. PMLR, 2024. URL `https://proceedings.mlr.press/v235/chakraborty24b.html`.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. ODIN: Disentangled reward mitigates hacking in RLHF. *arXiv preprint arXiv:2402.07319*, 2024a.

Peter L Chen, Xiaopeng Li, Xi Chen, and Tianyi Lin. Reward-free alignment for conflicting objectives. *arXiv preprint arXiv:2602.02495*, 2026.

Yanjun Chen, Dawei Zhu, Yirong Sun, Xinghao Chen, Wei Zhang, and Xiaoyu Shen. The accuracy paradox in RLHF: When better reward models don't yield better language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2980–2989, 2024b.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Juntao Dai, Taiye Chen, Yaodong Yang, Qian Zheng, and Gang Pan. Mitigating reward over-optimization in RLHF via behavior-supported regularization. *arXiv preprint arXiv:2503.18130*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

Xiaoyan Hu and Ho-fung Leung. A tighter problem-dependent regret bound for risk-sensitive reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 5411–5437. PMLR, 2023.

Kihyun Kim, Jiawei Zhang, Asuman Ozdaglar, and Pablo A Parrilo. Beyond RLHF and NLHF: Population-proportional alignment under an axiomatic framework. *arXiv preprint arXiv:2506.05619*, 2025.

Xuying Li, Zhuo Li, Yuji Kosuga, and Victor Bian. Optimizing safe and aligned language generation: A multi-objective GRPO approach. *arXiv preprint arXiv:2503.21819*, 2025.

Baijiong Lin, Weisen Jiang, Yuancheng Xu, Hao Chen, and Ying-Cong Chen. PARM: Multi-objective test-time alignment via preference-aware autoregressive reward model. *arXiv preprint arXiv:2505.06274*, 2025.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety. *Advances in Neural Information Processing Systems*, 37:108877–108901, 2024.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20619–20634, 2025.

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

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4998–5017, 2024.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

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

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First conference on language modeling*, 2024.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013. doi: 10.1613/jair.3987.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv: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. *arXiv preprint arXiv: 2409.19256*, 2024.

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. *arXiv preprint arXiv:2507.18624*, 2025.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Han Zhong, Xun Deng, Ethan X Fang, Zhuoran Yang, Zhaoran Wang, and Runze Li. Risk-sensitive deep RL: Variance-constrained actor-critic provably finds globally optimal policy. *Journal of the American Statistical Association*, pp. 1077–1089, 2026. doi: 10.1080/01621459.2025.2583501.

A APPENDIX

A.1 RVPO ALGORITHM

Algorithm 1 summarizes the full RVPO training procedure, including per-channel Z-normalization, inactive reward masking, and SoftMin aggregation. In the RaR setting, the number of active reward channels varies per prompt ($M_{\text{active}} \in [5, 17]$), as each prompt defines a distinct rubric; channels corresponding to criteria not applicable to the current prompt are excluded from the aggregation. The risk coefficient $k(t)$ may be held constant or annealed (e.g., linearly) over training steps. As $k \to 0$, RVPO recovers standard GDPO (mean aggregation); as $k \to \infty$, it reduces to hard-min.

Algorithm 1 Reward-Variance Policy Optimization (RVPO).

Require: Policy π_θ, reference policy π_{ref}, M reward functions $\{R_j\}_{j=1}^{M}$, group size G, risk coefficient schedule $k(t)$

for each training step t **do**
  Sample batch of prompts $\{x_i\}$
  for each prompt x_i **do**
    Generate G responses: $\{y_i^{(g)}\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x_i)$
    Compute rewards: $R_j^{(g)} = R_j(x_i, y_i^{(g)})$ for $j = 1, \ldots, M$
    for each reward channel $j = 1, \ldots, M$ **do**
      $\mu_j = \frac{1}{G} \sum_g R_j^{(g)}, \quad \sigma_j = \text{std}(\{R_j^{(g)}\}_g)$ ▷ *Z-normalize*
      $Z_j^{(g)} = (R_j^{(g)} - \mu_j)/(\sigma_j + \epsilon)$
    end for
    $A^{(g)} = -\frac{1}{k(t)} \ln\Big(\frac{1}{M_{\text{active}}} \sum_{j \in \text{active}} e^{-k(t) \cdot Z_j^{(g)}}\Big)$ ▷ *SoftMin over active channels*
  end for
  Whiten advantages across batch: $\hat{A}^{(g)} = (A^{(g)} - \mu_A)/\sigma_A$
  Update θ via clipped policy gradient with advantages $\hat{A}^{(g)}$ and KL penalty against π_{ref}
end for
Note: For RVPO-explicit, replace the SoftMin aggregation with $A^{(g)} = \mu_Z^{(g)} - \beta \cdot \Big(\sigma_Z^{(g)}\Big)^2$ for active channels only.

A.2 TRAINING HYPERPARAMETERS

This section details the hyperparameter configurations used for the experiments. All training was conducted on a single node with 8 NVIDIA H100 GPUs. On this hardware, tool-calling training runs require ≈1.5 hours for 1.5B models and ≈2.5 hours for 3B models. In the rubrics setting, 300-step training runs require ≈2 hours (3B), ≈4.5 hours (7B), and ≈8 hours (14B) per experiment. RVPO's aggregation step (LogSumExp over M channels) adds negligible wall-clock overhead (<1%) relative to GDPO.

Tool Calling (RLLA-4k): To ensure a rigorous baseline comparison, we utilize the exact training recipe established by GDPO (Liu et al., 2026) and ToolRL (Qian et al., 2025). We use the `verl` framework (Sheng et al., 2024) to fine-tune Qwen2.5-Instruct models (1.5B and 3B) for 15 epochs (≈117 steps) with a PPO mini-batch size of 128, a learning rate of 1×10^{-6}, a group size of $G = 4$ rollouts per prompt, and an adaptive KL penalty ($\beta_{KL} = 0.001$). We use a fixed risk coefficient $k = 1.0$.

Rubrics-as-Rewards (RaR-Medicine and RaR-Science): Similarly, we adopt the training hyperparameters established by the Rubrics-as-Rewards framework (Gunjal et al., 2025). We use the `TRL` framework to fine-tune Qwen2.5 base policies (3B, 7B, 14B) for 300 steps with an effective batch size of 96 prompts per step. We apply a learning rate of 5×10^{-6} with a 10% linear warmup schedule, sampling $G = 16$ responses per prompt at a temperature of 1.0, and a KL penalty coefficient of $\beta_{KL} = 0.04$ against the frozen reference policy. Checkpoints are saved and evaluated every 50 steps; we report the best-performing checkpoint alongside the final checkpoint (step 300).

Asset Licenses: Qwen2.5 models (Apache 2.0), RLLA-4k (Apache 2.0), RaR-Medicine/Science (CC-BY 4.0), HealthBench (MIT), GPQA-Diamond (CC-BY 4.0), BFCL-v3 (Apache 2.0), TRL (Apache 2.0), verl (Apache 2.0).

A.3 Science Evaluation

Table 3 reports the GPQA-Diamond results, with one training run per configuration and separate best and final checkpoints.

Table 3: **RaR-Science GPQA-Diamond evaluation.** Loglikelihood accuracy on 198 questions, one training run per configuration. Best and final checkpoints are reported as a descriptive parity check.

Size	Method	Best checkpoint	Final (300)
3B	GRPO (Implicit)	0.338	0.313
	GRPO (Explicit)	0.343	0.343
	GDPO	0.308	0.293
	RVPO-min ($k = \infty$)	0.348	0.283
	RVPO-explicit ($\beta = 0.5$)	0.313	0.308
	RVPO ($k = 1.0 \to 2.0$)	0.313	0.303
7B	GRPO (Implicit)	0.318	0.283
	GRPO (Explicit)	0.343	0.343
	GDPO	0.323	0.318
	RVPO-min ($k = \infty$)	0.323	0.308
	RVPO-explicit ($\beta = 0.5$)	0.318	0.318
	RVPO ($k = 1.0 \to 2.0$)	0.338	0.338
14B	GRPO (Implicit)	0.394	0.359
	GRPO (Explicit)	0.404	0.338
	GDPO	0.394	0.374
	RVPO-min ($k = \infty$)	0.369	0.293
	RVPO-explicit ($\beta = 0.5$)	0.414	0.359
	RVPO ($k = 1.0 \to 2.0$)	0.384	0.384

A.4 Local Mean–Variance Approximation

To build intuition for the local mean–variance interpretation of SoftMin, we perform a second-order Taylor expansion around the mean $\mu_Z^{(g)}$. Let $Z_j^{(g)} = \mu_Z^{(g)} + \delta_j$, such that the mean deviation $\frac{1}{M}\sum \delta_j = 0$ and the variance $\frac{1}{M}\sum \delta_j^2 = \left(\sigma_Z^{(g)}\right)^2$. By factoring out $e^{-k\mu_Z^{(g)}}$ and applying the approximations $e^y \approx 1 + y + \frac{y^2}{2}$ (for small y) alongside $\ln(1+y) \approx y$, the objective simplifies as follows:

$$
\begin{aligned}
A_{RVPO}^{(g)} &= \mu_Z^{(g)} - \frac{1}{k}\ln\left(\frac{1}{M}\sum_{j=1}^{M} e^{-k\delta_j}\right) \\
&\approx \mu_Z^{(g)} - \frac{1}{k}\ln\left(1 + \frac{k^2}{2}\left(\sigma_Z^{(g)}\right)^2\right) \\
&\approx \mu_Z^{(g)} - \frac{k}{2}\left(\sigma_Z^{(g)}\right)^2 = A_{RVPO\text{-}explicit}^{(g)}(k/2).
\end{aligned}
$$

This is a local approximation for small $|k\delta_j|$, giving an effective variance coefficient $\beta = k/2$ in that regime.

A.5 HYPERPARAMETER SELECTION DETAILS

Table 4 reports the single-run 7B Medicine selection sweep. Settings were selected separately for Medicine and Science, favoring final-checkpoint performance, and then transferred unchanged to 3B and 14B. The selected explicit-variance coefficients are $\beta = 1.0$ for Medicine and $\beta = 0.5$ for Science. Table 1 reports five-seed results for the selected settings.

Table 4: **Risk coefficient selection on HealthBench (Medicine, Qwen2.5-7B).** Original exploratory results, one training run per configuration. The selected schedule favors final-checkpoint retention over the highest early peak; repeated-run method comparisons are reported separately in Table 1.

Schedule (k)	**Best Ckpt**	**Final (300)**
Baselines		
GRPO (Implicit)	0.193	0.193
GRPO (Explicit)	0.221	0.102
GDPO ($k = 0$)	0.198	0.026
RVPO-min ($k = \infty$)	0.191	0.000
Constant Schedules		
$k = 0.25$	0.207	0.129
$k = 0.5$	0.171	0.095
$k = 1.0$	0.194	0.015
$k = 2.0$	0.225	0.083
$k = 5.0$	**0.241**	0.095
$k = 10.0$	0.202	0.010
Annealed Curricula		
$0.1 \to 2.0$	0.214	0.132
$0.25 \to 1.0$	0.209	0.082
$0.5 \to 5.0$	0.198	0.139
$0.5 \to 2.0$	0.230	**0.204**
$1.0 \to 2.0$	0.143	0.000

Beyond the selection protocol, equating the small-k penalty scale $\frac{k}{2}(\sigma_Z^{(g)})^2$ with the worst-channel scale $|\min_j Z_j^{(g)}|$ gives the rule of thumb

$$k^* = \frac{2\,\mathbb{E}_g[|\min_j Z_j^{(g)}|]}{\mathbb{E}_g[(\sigma_Z^{(g)})^2]}.$$

In the analyzed rubric runs, it converged near 4.4 within the first few steps and remained there. The Medicine schedule ends below this crossover at 2.0, where SoftMin still combines information from several channels. This gives a rule of thumb for choosing the scale of k from a short pilot or the opening training steps; it is not an exact optimum or a separately validated selection protocol. Validation across additional group sizes, channel counts, and conflict levels remains future work.

A.6 PRE- VS. POST-NORMALIZATION WEIGHTING

Table 5 evaluates methods under a weighted regime on the HealthBench (Medicine) domain. Standard RVPO incorporates the AI-generated categorical priority weights from the RaR framework (Gunjal et al., 2025) (e.g., Essential=1.0, Optional=0.3) directly into the raw reward computation ($R_j = c_j \cdot w_j$, where $c_j \in \{0, 1\}$ is the binary criterion score). However, for binary criteria, these scalar weights are naturally absorbed by the per-channel Z-normalization (pre-normalization weighting). Table 5 evaluates an alternative regime where these weights are explicitly applied post-normalization ($Z'_j = w_j \cdot Z_j$) to force the optimizer to recognize static priorities.

This exploratory single-run comparison shows that standard RVPO has the higher peak (0.230), while post-normalization weighted GDPO has the higher final score (0.213 versus 0.204). The

tested post-normalization RVPO settings score lower than standard RVPO in these runs. This 7B comparison of categorical weighting is distinct from the controlled 3B single-channel priority intervention in Appendix A.7.

Table 5: **Pre- vs. post-normalization weighting on HealthBench (Medicine, Qwen2.5-7B).** Original exploratory results, one training run per configuration. Standard RVPO has the highest peak; post-normalization weighted GDPO has the highest final score. GRPO-explicit is the raw weighted-scalar baseline.

Method	Best Checkpoint	Final (Step 300)
Pre-normalization Weighting (Standard)		
GDPO (Mean)	0.198	0.026
RVPO ($k = 0.5 \to 2.0$)	**0.230**	0.204
Weighted Baselines (GRPO raw; GDPO post-normalization)		
GRPO (Explicit)	0.221	0.102
GDPO	0.213	**0.213**
Post-normalization Weighting (RVPO)		
RVPO ($k = 0.5 \to 2.0$)	0.192	0.078
RVPO ($k = 1.0 \to 2.0$)	0.201	0.000
RVPO ($k = 2.0$ const)	0.160	0.106

A.7 Controlled Corruption and Priority Mismatch

We target a fixed rubric position that is reproducibly the weakest channel in clean 3B RaR-Medicine runs. Each cell uses five seeds. We compare GDPO, RVPO ($k = 0.5 \to 2.0$), and hard-min. The conditions are symmetric label noise at $p = 0.25$ and $p = 0.5$, a harsh judge with one-sided $1 \to 0$ flips at $p = 0.25$, and a priority mismatch intervention that down-weights the weak channel to 0.2 after normalization. Because the corruption is controlled, we record both the observed training signal and the underlying pre-corruption signal.

Table 6: **Controlled corruption and priority intervention at 3B.** Target-channel aggregation weight divided by equal share $1/M$ (five seeds per cell, mean $\pm$ standard deviation). The priority intervention is distinct from label corruption.

Condition	**RVPO** ($k = 0.5 \to 2$)	**Hard-min**
Reference ($p = 0$)	1.009 ± 0.003	0.922 ± 0.021
Symmetric noise ($p = 0.25$)	1.122 ± 0.008	1.703 ± 0.034
Symmetric noise ($p = 0.5$)	1.194 ± 0.010	2.095 ± 0.042
Harsh judge ($p = 0.25$)	1.060 ± 0.010	1.174 ± 0.024
Priority weight 0.2	0.826 ± 0.003	0.304 ± 0.011

Table 6 reports target-channel aggregation weight normalized by its equal share $1/M$. GDPO has weight 1.0 by construction in the unweighted reference and corruption conditions. Under symmetric noise at $p = 0.5$, hard-min rises from 0.922 to 2.095 times equal share, while finite-k RVPO rises from 1.009 to 1.194. The harsh-judge condition gives the same qualitative result. Neither formulation identifies the channel as unreliable; the finite-k configuration concentrates less weight on it than hard-min in these corruption tests.

The difference between early-to-late drift in the observed and underlying signals remains within approximately ± 0.015 across arms and conditions. Corrupted-minus-reference HealthBench differences range from -0.007 to $+0.008$ across arms. We interpret this as no detectable downstream degradation at this sample size rather than proof of no effect.

Applying a post-normalization weight of 0.2 reduces the target share to 0.826 for RVPO and 0.304 for hard-min. This offers a way to encode known priorities separately from difficulty. Positive

multiplicative weights applied before per-channel normalization are absorbed by that normalization, whereas post-normalization weights change the aggregation inputs.

The tests are limited to 3B, and corruption is independent per generation. They probe unreliable judges rather than exploitability of judges whose errors correlate with an output feature.

A.8 Explicit Variance Penalty (β) Ablation

Figure 4 shows the 7B Medicine sweep over constant values of the explicit variance coefficient β, with one training run per setting. Performance varies non-monotonically across the sweep, illustrating sensitivity to the penalty coefficient. Unlike SoftMin, the quadratic penalty grows without bound as inter-objective disagreement increases. Table 1 separately reports five-seed results for the selected $\beta = 1.0$ setting.

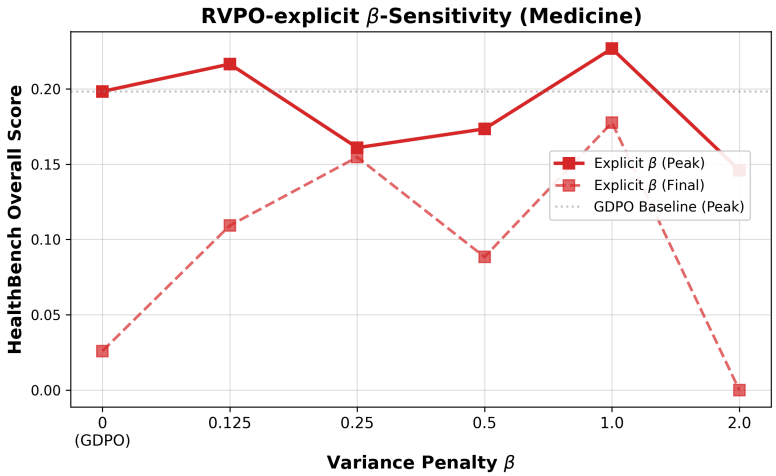

Figure 4: **Explicit variance penalty (β) sweep on HealthBench (Medicine, Qwen2.5-7B).** Best- and final-checkpoint scores for constant penalty coefficients, with one training run per setting. The selected $\beta = 1.0$ configuration is evaluated across five seeds in Table 1.

A.9 Broader Impacts

This work improves the ability of large language models to balance competing objectives and satisfy bottleneck constraints. By mitigating constraint neglect, RVPO provides a computationally efficient mechanism for encouraging more consistent behavior while maintaining competitive performance on the evaluated capability benchmarks. However, as a reward aggregation strategy, variance penalization is agnostic to the semantic nature of the constraints. While we apply it to enforce structural correctness and clinical accuracy, the same explicit constraint enforcement could theoretically be misused if malicious or biased reward models are intentionally provided to the optimizer.