DISENTANGLED VALUE ESTIMATION: RETHINKING SCALAR CRITICS IN RUBRIC-BASED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Value estimation is crucial for many reinforcement learning algorithms used to optimize large language models (LLMs). In critic-based methods such as Proximal Policy Optimization (PPO), an additional critic model predicts expected future outcomes for each generated token based on intermediate generation states, and this value is used to compute the advantages that guide policy optimization. Such value estimation is relatively well defined when the reward specification is fixed and consistent, but becomes more challenging in open-ended tasks where evaluation criteria vary across queries. We study this problem in RL with rubric rewards, where each query is associated with multiple query-specific criteria whose scores are collapsed into a scalar reward. A standard critic observes only the query and response prefix and is supervised by this aggregated reward, requiring it to implicitly recover the underlying reward specification before estimating values. To address this, we propose $\mathcal{D}$isentangled **V**alue **E**stimation ($\mathcal{D}$VE), which uses a shared critic conditioned on one rubric at a time to estimate rubric-specific values, computes token-level advantages for each rubric, and aggregates them using the original rubric weights. Across five benchmarks spanning instruction following, chat, and scientific question answering, $\mathcal{D}$VE outperforms the vanilla PPO and GRPO in 13 of 15 model–benchmark combinations. Further analysis shows that $\mathcal{D}$VE achieves lower value prediction error, higher explained variance, stronger response-level ranking consistency, and more targeted token-level credit assignment. Code is available at Anonymous Repo.

1 INTRODUCTION

Value estimation is a fundamental component of many reinforcement learning (RL) algorithms for training large language models (LLMs) (Lambert et al., 2024; Shao et al., 2024). It predicts the expected outcome the LLM policy would achieve from the current generation prefix (referred to as the *value*), which allows us to measure the each relative improvement for each action. To estimate fine-grained token-level values, algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) train a learnable critic model to predict values that fit the target outcomes from the corresponding reward function. These predicted values are then used to compute token-level advantages, typically via generalized advantage estimation (GAE) algorithms (Schulman et al., 2016).

Recent studies reveal that the end-to-end training performance may benefit by an effective and stable critic learning (Yuan et al., 2025; Yue et al., 2025; Kazemnejad et al., 2024). The core idea is that the critic model, which by definition estimates under a given reward function, can implicitly infer the evaluation criteria, thereby allowing us to more accurately estimate the advantage for each action. For example, in verifiable task such as mathematical reasoning (Hu et al., 2026; Lambert et al., 2024) and conventional game playing task (Mnih et al., 2015; Schrittwieser et al., 2020), the rewards are typically computed by measuring whether the final answer matches an objectively verifiable target. Under such a fixed and stable specification, the critic can achieve lower value prediction error, leading to more reliable advantage estimates and more effective end-to-end policy optimization (Yue et al., 2025; Huang et al., 2026).

However, value estimation becomes increasingly challenging in open-ended tasks, such as long-form generation (Li et al., 2026; Yifei et al., 2025). The main reason is that their reward functions depend on complex evaluation criteria that can vary across queries, making critic learning more difficult.

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.

Recent work introduces *rubrics-as-reward* (Gunjal et al., 2026; Viswanathan et al., 2026). These methods pair each query with a set of *query-specific* rubrics that a desirable response should satisfy, and use a judge LLM to evaluate the generated response against each rubric independently (Cook et al., 2024). The rubric-level scores are then aggregated into a scalar outcome reward for policy optimization, so that a single reward value encodes multiple query-specific rubric. This creates an information mismatch for critic learning: *a standard critic typically receives only the query and response prefix, while learning from a collapsed scalar reward.* As a result, the critic model implicitly recovers the underlying reward specification from the query and scalar supervision alone, making value prediction more difficult and ultimately degrading advantage estimation.

In this work, we propose $\mathcal{D}$isentangled **V**alue **E**stimation ($\mathcal{D}$VE), a rubric-conditioned value estimation framework that explicitly preserves multi-aspect rubric throughout critic learning and the latter advantage estimation stage. Instead of requiring the critic to implicitly recover the reward specification from the query and aggregated scalar supervision, $\mathcal{D}$VE uses a shared critic conditioned on the query, response prefix, and one rubric at a time to estimate the expected outcome under each individual rubric. We optimize the critic model using return targets derived from the corresponding rubric-level rewards, thereby preserving specific supervision throughout critic learning. Based on these rubric-specific value estimates, $\mathcal{D}$VE computes token-level advantages separately for each rubric through the standard GAE procedure and then aggregates them using the original rubric weights. Finally, the aggregated token-level advantages are used to optimize the policy with the standard clipped policy-gradient objective. $\mathcal{D}$VE retains the original rewards before collapsed into the final scalar and can be readily integrated into existing RL post-training frameworks.

We conduct experiments on five benchmarks spanning three open-ended domains (i.e., instruction following, chat, and scientific question answering) using Qwen3 models with 0.6B, 1.7B, and 4B parameters. $\mathcal{D}$VE outperforms both vanilla PPO and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) in 13 of 15 model–benchmark combinations. For example, on the instruction-following benchmark MulDimIF, $\mathcal{D}$VE improves performance over vanilla PPO by 8.00 percentage points at the 1.7B scale. To better understand the improvements of $\mathcal{D}$VE, we systematically analyze the quality of value estimation from both token-level prediction accuracy and response-level ranking consistency. At the token level, we measure mean absolute error (MAE) and explained variance (EV), which assess absolute prediction error and the extent to which the critic captures variation in observed rewards, respectively. $\mathcal{D}$VE generally achieves lower MAE and higher EV, indicating more accurate token-level value estimates. At the response level, we further compute Spearman's rank correlation between the value prediction associated with the EOS token and the observed outcome reward. $\mathcal{D}$VE achieves stronger ranking consistency, indicating that its critic more consistently assigns higher values to responses with higher rewards.

Our contributions are summarized as follows: (i) We propose $\mathcal{D}$VE, a $\mathcal{D}$isentangled **V**alue **E**stimation framework that explicitly incorporates multi-aspect rubrics into critic learning and computes rubric-specific advantages for policy optimization. (ii) We systematically evaluate the value estimation quality of $\mathcal{D}$VE, demonstrating improvements in prediction accuracy and response-level ranking consistency. (iii) We conduct extensive experiments across three domains, five benchmarks, and three model scales, demonstrating the effectiveness of $\mathcal{D}$VE in improving policy performance.

2 METHOD

2.1 PRELIMINARIES

Rubric as Reward. In conventional RLVR settings, rewards are typically based on the correctness of the final answer. By contrast, RL with rubric rewards pairs each query x with a checklist $\mathcal{C} = \{(c_k, w_k)\}_{k=1}^{|\mathcal{C}|}$, specifying the criteria that an ideal response should satisfy. Here, c_k denotes the k-th rubric and w_k its corresponding weight, which we normalize as $\hat{w}_k = w_k / \sum_{i=1}^{|\mathcal{C}|} w_i$. Given a response $y = (y_1, \ldots, y_{|y|})$[1] sampled from the policy $\pi_\phi(\cdot \mid x)$, a judge model evaluates the response y against each rubric c_k separately, producing the corresponding rubric-level reward $R(x, y, c_k)$. These rubric-level rewards are then aggregated into a scalar outcome reward using the normalized rubric weights $R(x, y, \mathcal{C}) = \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k R(x, y, c_k)$.

[1] We index generated response tokens from $t = 1$, with $y_{|y|} = \langle \text{EOS} \rangle$.

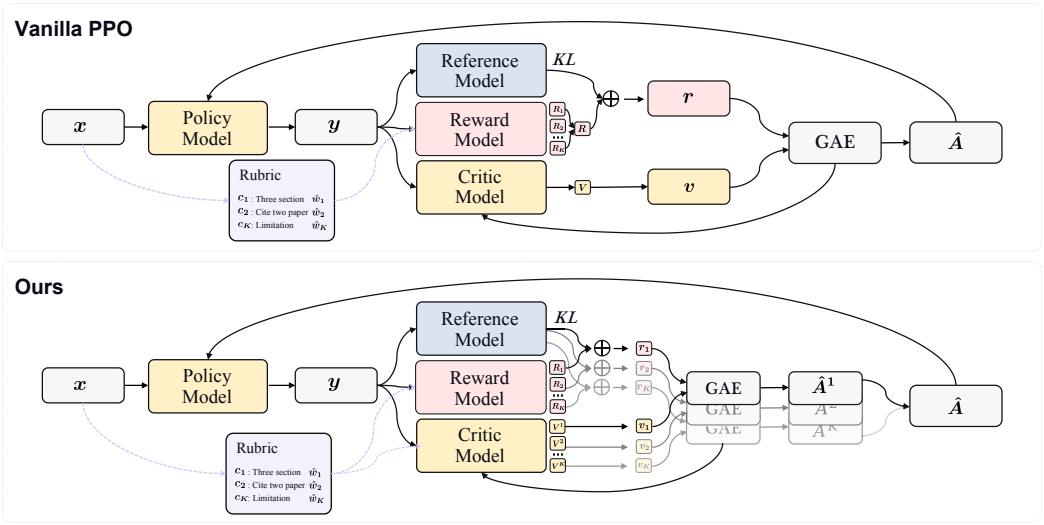


Figure 1: Comparisons between PPO and $\mathcal{D}$VE. In PPO, rubrics are only seen by the reward model, and the rewards from multiple rubrics are collapsed into a scalar for training. Alternatively, $\mathcal{D}$VE conditions the critic model explicitly on the same rubrics, estimating value and advantage per rubric.

Generalized Advantage Estimation (GAE). GAE estimates action advantages as an exponentially weighted sum of temporal-difference (TD) residuals (Sutton, 1988; Schulman et al., 2016), enabling a trade-off between bias and variance in advantage estimation. In LLM training, rewards are typically assigned only at the end of each response, so we define the token-level reward as

$$r_t = \begin{cases} R(x, y, \mathcal{C}), & t = |y|, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, the token-level reward for rubric c_k is $r_t^k = R(x, y, c_k)$ when $t = |y|$, and $r_t^k = 0$ otherwise. Standard PPO uses the aggregated reward $R(x, y, \mathcal{C})$ to train a critic model that predicts the expected return at each response prefix. Let θ denote the parameters of the critic model. We formulate the value prediction as $V_{\theta,t} = V_\theta(x, y_{<t})$ and the TD residual as

$$\delta_t = r_t + \gamma V_{\theta,t+1} - V_{\theta,t}, \tag{2}$$

where $\gamma \in [0, 1]$ is the discount factor, and the value after the final token is set to zero, i.e., $V_{\theta,|y|+1} = 0$. The token-level advantage is then estimated as $\hat{A}_t = \sum_{\ell=0}^{|y|-t} (\gamma\lambda)^\ell \delta_{t+\ell}$, where $\lambda \in [0, 1]$ controls the bias–variance trade-off in GAE. In contrast, $\mathcal{D}$VE uses the individual rubric rewards $R(x, y, c_k)$ to train a shared, rubric-conditioned critic and computes advantages separately for each rubric.

2.2 Integrating RL with $\mathcal{D}$isentangled Value Estimation

The central idea of $\mathcal{D}$VE is to use a shared rubric-conditioned critic to estimate values separately for each rubric, thereby computing the corresponding rubric-specific advantages before aggregation.

Rubric-disentangled Value Estimation. Figure 1 shows the overall process. Given a query x, a response prefix $y_{<t}$, and a rubric c_k, the critic predicts the value at this prefix, formulated as $V_{\theta,t}^k = V_\theta(x, y_{<t}, c_k)$. Conditioning on c_k makes the rubric explicit, allowing the critic to estimate the value under that rubric without having to infer it from the query alone. Moreover, supervising the critic with returns derived from the individual rubric rewards $R(x, y, c_k)$ preserves distinctions that are collapsed in the aggregated reward $R(x, y, \mathcal{C})$: responses with the same aggregate reward may satisfy different subsets of rubrics and thus provide distinct rubric-level supervision signals.

Using these rubric-conditioned value estimates, we compute token-level advantages separately for each rubric. Specifically, for rubric c_k, the temporal-difference residual is

$$\delta_t^k = r_t^k + \gamma V_{\theta,t+1}^k - V_{\theta,t}^k, \tag{3}$$

where the terminal value is set to zero, i.e., $V^k_{\theta,|y|+1} = 0$. The corresponding advantage is computed using GAE algorithm $\hat{A}^k_t = \sum_{\ell=0}^{|y|-t} (\gamma\lambda)^\ell \delta^k_{t+\ell}$. We then combine these rubric-level advantages using the same normalized weights as in the reward aggregation, formulated as:

$$\hat{A}_t = \sum\nolimits_{k=1}^{|\mathcal{C}|} \hat{w}_k \hat{A}^k_t. \tag{4}$$

Overall Training Objective. After obtaining the rubric-specific advantages $\{\hat{A}^k_t\}_{k=1}^{|\mathcal{C}|}$ and aggregating them into $\hat{A}_t$ in Eq 4, we use the resulting token-level advantage to optimize the policy. Specifically, we follow the standard clipped policy-gradient objective:

$$\mathcal{L}_{\text{policy}}(\phi) = \mathbb{E}_t \left[\min \left(\rho_t(\phi)\hat{A}_t,\ \text{clip}\big(\rho_t(\phi), 1-\epsilon, 1+\epsilon\big)\hat{A}_t \right) \right], \tag{5}$$

where $\rho_t(\phi) = \pi_\phi(y_t \mid x, y_{<t}) / \pi_{\phi_{\text{old}}}(y_t \mid x, y_{<t})$ is the importance sampling ratio between the current and behavior policies, and ϵ controls the PPO clipping range. We also jointly train the shared rubric-conditioned critic to support this rubric-specific advantage estimation. For each rubric c_k, we construct the return target as $\hat{V}^k_t = \hat{A}^k_t + V^k_{\theta,t}$, using the rubric-level GAE estimate and the value prediction computed before the critic update. We train the critic model θ with loss $\mathcal{L}_V(\theta)$:

$$\mathcal{L}_V(\theta) = \frac{1}{|\mathcal{C}|} \sum\nolimits_{k=1}^{|\mathcal{C}|} \frac{1}{|y|} \sum\nolimits_{t=1}^{|y|} \left(V_\theta(x, y_{<t}, c_k) - \hat{V}^k_t \right)^2. \tag{6}$$

This preserves rubric-disentangled supervision throughout critic learning: rather than fitting a single value target derived from the aggregated scalar reward, the shared critic learns a separate value function explicitly conditioned on rubrics. We give each rubric equal weight in Eq. 6 so that the critic model learns all rubric independently. Consequently, $\mathcal{D}$VE preserves the scalar reward training objective while retaining the fine-grained rubric structure during both value and advantage estimation.

2.3 Analysis of Rubric-Conditioned Value Estimation

We analyze the scalar prediction error induced by rubric-level value decomposition. Both critics are evaluated against the same scalar outcome reward $R(x, y, \mathcal{C})$. Formally, suppose the shared critic exactly recovers the conditional mean of every rubric-level reward, i.e., $V^k_{\theta,t} = \mathbb{E}[R(x, y, c_k) \mid x, y_{<t}]$. With the checklist and rubric weights fixed for each query, linearity of conditional expectation implies that their weighted aggregation recovers the conditional mean of $R(x, y, \mathcal{C})$. Consequently, the scalar prediction errors satisfy

$$\text{MSE}_{\text{vanilla}} - \text{MSE}_{\text{DVE}} = \mathbb{E}\left[\left(V_\theta(x, y_{<t}) - \sum\nolimits_{k=1}^{|\mathcal{C}|} \hat{w}_k V^k_{\theta,t} \right)^2 \right] \geq 0, \tag{7}$$

where $V_\theta(x, y_{<t})$ denotes the vanilla critic's prediction, and both MSEs are computed against the same scalar outcome reward $\mathcal{R}(x, y, \mathcal{C})$ under the same evaluation distribution.

This result establishes that exact rubric-level value estimates reconstruct an MSE-optimal scalar baseline. Under simultaneous exact recovery of the full-return conditional means, uniform weighting of the criterion-level squared loss is compatible with this result, while $\hat{w}_k$ determines how the predictions are aggregated. The analysis does not guarantee that finite-data training achieves exact recovery or yields lower prediction error than vanilla. We evaluate the learned critics empirically in § 4.4. The formal proposition and proof are provided in Appendix A.8.

2.4 Evaluating the Value Estimation of Critic Model

Evaluating the quality of value estimation provides insight into credit assignment during RL training and helps interpret the learning signals used to optimize LLM policies. A straightforward metric is the token-level MSE loss in Eq. 6 with lower values indicating more accurate estimation. However, MSE depends on the reward scale and amplifies large prediction errors, making it sensitive to outliers (Hodson, 2022). Thus, we introduce the following three complementary metrics to reliably measure both prediction errors and response-level ranking consistency of critic models.

Mean Absolute Error (MAE). MAE measures the average absolute discrepancy between the critic's predictions and observed outcome rewards over all valid response-token positions:

$$\text{MAE} = \mathbb{E}_{x,y,t} \left[|R(x, y, \mathcal{C}) - V_{\theta,t}| \right]. \tag{8}$$

Lower MAE indicates more accurate predictions on the original reward scale. Compared with MSE, MAE penalizes errors linearly rather than quadratically, making it less sensitive to extreme prediction errors. It also captures constant prediction offsets that EV ignores, providing a complementary assessment of numerical accuracy.

Explained Variance (EV). EV provides a normalized assessment of mean squared prediction error after removing the mean residual (Pan et al., 2026). Formally, we denote $V_{\theta,t}$ as the scalar value prediction and compute EV over all valid response-token positions as:

$$\mathrm{EV} = 1 - \frac{\mathrm{Var}(R(x, y, \mathcal{C}) - V_{\theta,t})}{\mathrm{Var}(R(x, y, \mathcal{C}))}. \tag{9}$$

Vanilla PPO directly predicts $V_{\theta,t} = V_\theta(x, y_{<t})$, while $\mathcal{D}$VE uses $V_{\theta,t} = \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k V_{\theta,t}^k$. Intuitively, EV measures how much of the variation in observed rewards is captured by the critic's predictions. At $\mathrm{EV} = 1$, the residual variance is zero, meaning that the predictions match the observed rewards up to a constant offset. As EV increases from 0 to 1, a smaller fraction of reward variation remains in the prediction residuals.

Response-level Ranking Correlation. To evaluate whether the critic model assigns higher values to responses with higher outcome rewards, we compute Spearman's rank correlation (ρ). For each response, we use the value prediction associated with the EOS token, $V_{\theta,|y|}$, where $y_{|y|} = \langle \mathrm{EOS} \rangle$. This prediction is conditioned on the response prefix immediately preceding EOS. We compute

$$\rho = \mathrm{Corr}\left(\mathrm{rank}(R(x, y, \mathcal{C})), \mathrm{rank}\big(V_{\theta,|y|}\big)\right), \tag{10}$$

where $\mathrm{rank}(\cdot)$ are computed across evaluation responses, with average ranks assigned to ties, and Corr denotes Pearson correlation. A ρ closer to 1 indicates stronger agreement between predicted and observed response rankings, reflecting the critic's ability to distinguish relative response quality independently of prediction magnitudes.

3 Experimental Setup

Training Datasets. We train on three rubric-based domains: Instruction Following (IF) with HiR-16K (Zhang et al., 2025), and Chat and Science with the corresponding subsets of RubricHub (Li et al., 2026). Each instance pairs a query x with a weighted rubrics $\mathcal{C}(x) = \{(c_k, w_k)\}_{k=1}^{|\mathcal{C}|}$; the median $|\mathcal{C}|$ is 7 for IF and 27 for Chat and Science. Full dataset statistics appear in Appendix A.4.1.

Evaluation Metric. IF uses IFEval (Zhou et al., 2023), IFBench (Pyatkin et al., 2025), and MulDimIF (Ye et al., 2026): we report strict prompt-level and instruction-level *accuracy* on IFEval and IFBench, and instruction-level accuracy on MulDimIF. Chat uses Arena-Hard-V2 (Li et al., 2024), rated by the style-controlled Bradley-Terry (BT) win rate against Gemini-2.0-Flash-001. Science uses ResearchQA (Yifei et al., 2025), scored by *accuracy*. Since RubricHub partially draws on ResearchQA, we remove all test questions that in our science training set to avoid contamination. For each checkpoint, we conduct three independent evaluation runs, regenerating responses and evaluating them in each run. We report the mean; evaluation details are in Appendix A.4.2. We also evaluate the critic model using the metrics from § 2.4.

Baselines. We compare $\mathcal{D}$VE with two widely used baselines: PPO (Schulman et al., 2017), which uses a critic and GAE to form token-level advantages, and GRPO (Shao et al., 2024), which drops the critic and estimates advantages by normalizing rewards within a group of responses per query. Implementation details for both baselines are given in Appendix A.7.

Implementation Details. We implement $\mathcal{D}$VE with Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B-Instruct. Policy and critic models share the same initial checkpoint at each scale. Training uses four H100 GPUs. Hyperparameters appear in Appendix A.7. Every rubric is graded to a binary reward $R(x, y, c_k) \in \{0, 1\}$ and aggregated into the scalar $R(x, y, \mathcal{C})$ with the normalized weights $\hat{w}_k$. Following official setup, for Chat and Science domains, an LLM judge verifies every rubric. For Instruction Following, most rubrics are verified by rule, with the same LLM judge as fallback, and all rubrics carry equal weight, so $\hat{w}_k = 1/|\mathcal{C}|$ and $R(x, y, \mathcal{C})$ reduces to the constraint satisfaction rate. All judging uses DeepSeek-V3.2, for both training and evaluation.

Model	ArenaHard V2	ResearchQA	IFBench		IFEval		MulDimIF
	Overall	Overall	Prompt	Instruct	Prompt	Instruct	Acc.
Qwen3-0.6B	5.33	38.29	17.35	17.91	56.38	67.27	24.58
+ GRPO	6.15 ↑0.82	40.09 ↑1.80	17.48 ↑0.13	19.22 ↑1.31	61.37 ↑4.99	70.62 ↑3.35	32.68 ↑8.10
+ PPO	6.67 ↑1.34	40.97 ↑2.68	17.35 ↑0.00	18.51 ↑0.60	**61.55** ↑5.17	**71.58** ↑4.31	30.75 ↑6.17
+ $\mathcal{D}$VE (Ours)	**7.28** ↑1.95	**43.12** ↑4.83	**19.05** ↑1.70	**19.40** ↑1.49	59.70 ↑3.32	70.38 ↑3.11	**34.50** ↑9.92
Qwen3-1.7B	12.78	53.63	17.69	19.70	68.39	76.38	32.17
+ GRPO	16.22 ↑3.44	**58.61** ↑4.98	20.41 ↑2.72	22.09 ↑2.39	70.61 ↑2.22	78.78 ↑2.40	38.08 ↑5.91
+ PPO	16.22 ↑3.44	57.37 ↑3.74	19.73 ↑2.04	21.19 ↑1.49	71.35 ↑2.96	79.14 ↑2.76	47.25 ↑15.08
+ $\mathcal{D}$VE (Ours)	**16.26** ↑3.48	57.91 ↑4.28	**22.11** ↑4.42	**23.88** ↑4.18	**72.46** ↑4.07	**79.98** ↑3.60	**55.25** ↑23.08
Qwen3-4B-Instruct	58.69	66.84	30.61	32.84	82.99	88.13	57.00
+ GRPO	61.59 ↑2.90	74.21 ↑7.37	31.29 ↑0.68	33.73 ↑0.89	84.84 ↑1.85	89.81 ↑1.68	70.00 ↑13.00
+ PPO	60.07 ↑1.38	74.00 ↑7.16	30.95 ↑0.34	32.84 ↑0.00	84.66 ↑1.67	89.45 ↑1.32	67.58 ↑10.58
+ $\mathcal{D}$VE (Ours)	**62.98** ↑4.29	**74.44** ↑7.60	**34.01** ↑3.40	**35.52** ↑2.68	**85.21** ↑2.22	**89.93** ↑1.80	**70.33** ↑13.33

Table 1: Main results across the three domains. Scores are shown with the delta (↑/↓) over the corresponding base model; $\mathcal{D}$VE rows are highlighted, best per group **bold**, second-best underlined. Scores are averaged over three independent evaluation runs, each with newly generated responses.

4 Experiments

4.1 End-to-end Evaluation

$\mathcal{D}$VE improves policy performance. As shown in Table 1, $\mathcal{D}$VE outperforms both vanilla PPO and GRPO in 13 of 15 model-benchmark combinations. Improvements are particularly pronounced on IF tasks: $\mathcal{D}$VE consistently leads on IFBench and MulDimIF across model scales, with the largest gains on MulDimIF at 1.7B, surpassing PPO and GRPO by 8.00 and 17.17 percentage points, respectively. Similar improvements are observed in Chat and Science domains. $\mathcal{D}$VE achieves the highest scores on Arena-Hard-V2 across all three scales. These results support the effectiveness of $\mathcal{D}$VE in improving downstream policy performance across diverse rubric-based tasks.

4.2 Ablation Study

$\mathcal{D}$VE outperforms a variant that merely conditions on the rubrics. We compare $\mathcal{D}$VE with a variant, denoted *w/o disentangle*, that concatenates all rubrics into a single critic input and predicts one aggregate value. Table 2 shows that $\mathcal{D}$VE outperforms it on all five metrics at every model size. The largest gains occur on MulDimIF, where $\mathcal{D}$VE exceeds this variant by 8.33 and 19.08 percentage points at 0.6B and 1.7B, respectively. At 4B, the variant is more competitive, but $\mathcal{D}$VE still improves every metric, including a 1.75-point gain on MulDimIF. These results suggest that separately modeling rubric-conditioned values and aggregating their advantages provides benefits beyond jointly exposing the critic to all evaluation criteria.

Model / Method	IFBench		IFEval		MulDimIF
	Prompt	Instruct	Prompt	Instruct	Overall
Qwen3-0.6B					
vanilla $\mathcal{D}$VE	**19.05**	**19.40**	**59.70**	**70.38**	**34.50**
- w/o disentangle	17.35 ↓1.70	18.51 ↓0.89	58.96 ↓0.74	68.94 ↓1.44	26.17 ↓8.33
Qwen3-1.7B					
vanilla $\mathcal{D}$VE	**22.11**	**23.88**	**72.46**	**79.98**	**55.25**
- w/o disentangle	19.39 ↓2.72	20.60 ↓3.28	69.50 ↓2.96	77.10 ↓2.88	36.17 ↓19.08
Qwen3-4B-Instruct					
vanilla $\mathcal{D}$VE	**34.01**	**35.52**	**85.21**	**89.93**	**70.33**
- w/o disentangle	33.33 ↓0.68	34.63 ↓0.89	84.47 ↓0.74	89.57 ↓0.36	68.58 ↓1.75

Table 2: Ablation of $\mathcal{D}$VE. Red arrows show drops in percentage points from $\mathcal{D}$VE.

4.3 Training Dynamics

$\mathcal{D}$VE holds its advantage across checkpoints. Figure 2 shows training dynamics by reporting downstream policy performance across all training checkpoints. $\mathcal{D}$VE leads vanilla PPO at every checkpoint in 9 of 15 model-benchmark combinations, and at most checkpoints in the remaining six. For example, on MulDimIF at 1.7B, $\mathcal{D}$VE continues to improve during training and reaches approximately 55 percentage points, compared with around 47 percentage points for vanilla PPO.

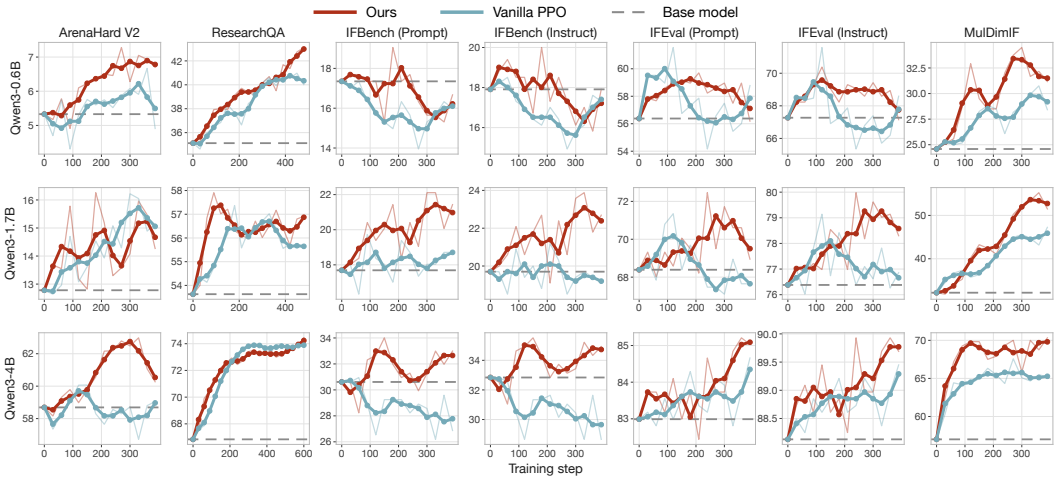


Figure 2: Benchmark performance of $\mathcal{D}$VE and vanilla PPO across training checkpoints. Rows represent 0.6B, 1.7B, and 4B Qwen3 models. Scores are percentages (higher is better); dashed lines mark base-model performance. THIN lines show raw scores; **THICK** lines show smoothed trends.

$\mathcal{D}$VE maintains higher EV during training. Training EV depends on the policy that generated the rollouts as well as on the critic, but it remains the standard way to monitor critic learning during PPO, measuring how much of the reward variation in a batch the value predictions capture. Figure 3 shows the Chat domain as an example: $\mathcal{D}$VE stays above vanilla PPO over most of training at all three model sizes, and at 1.7B its smoothed EV reaches approximately 0.7 late in training against around 0.6 for vanilla PPO. Section 4.4 removes the policy dependence by scoring both critics on a fixed set of responses; complete training EV results appear in Appendix A.2.

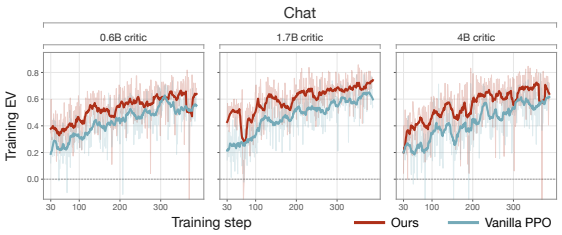


Figure 3: Training EV of $\mathcal{D}$VE and vanilla PPO on Chat domain at three model sizes. Thin lines show raw values; thick lines show smoothed trends.

4.4 EVALUATING VALUE ESTIMATION

So far, we have illustrated that the $\mathcal{D}$VE achieve higher EV during the training process. However, training rollouts are generated by each method's own updated policy, so the observed EV reflect both critic quality and differences in the rollout distributions. We therefore conduct a more controlled comparison by evaluating all critics on the same fixed sets of responses. Specifically, we generate responses using two off-the-shelf policies, Qwen3-1.7B and DeepSeek-V3.2, to assess value estimation across different OOD response. We designate Qwen3-1.7B as the *near-OOD* policy, given its closer model scale to our training policies, and DeepSeek-V3.2 as the *far-OOD* policy. Figure 4 reports the results across two benchmarks and three model sizes under both settings.

$\mathcal{D}$VE reduces absolute prediction errors. Figure 4 shows that $\mathcal{D}$VE generally achieves lower MAE under near-OOD policies, indicating that its token-level value predictions are closer to observed returns. Improvements are particularly clear at 1.7B, where MAE falls to approximately 0.45 in both domains, compared with around 0.52 for vanilla PPO. Under far-OOD policies, the gains are most evident at 4B, where $\mathcal{D}$VE maintains lower MAE over many checkpoints. In Chat domain, its MAE is approximately 0.56, compared with around 0.74 for vanilla PPO. These results complement the EV findings with smaller absolute prediction errors, including offsets that EV does not capture.

$\mathcal{D}$VE better captures return variation. EV is a normalized MSE computed after removing the mean residual, measuring how well the critic's predictions capture variation in observed rewards. EV closer to 1 indicates better capture, while EV at or below 0 indicates no improvement over a constant prediction. As shown in Figure 4, $\mathcal{D}$VE generally achieves higher EV under policy shift. Under near-OOD policies, $\mathcal{D}$VE achieves the highest EV across all model sizes and both domains.

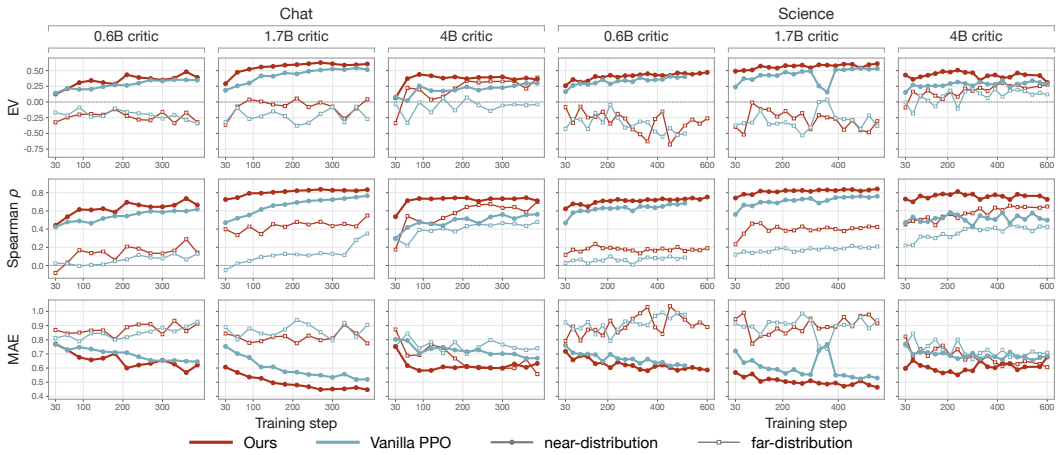


Figure 4: Training dynamics of critic quality for $\mathcal{D}$VE and vanilla PPO in Chat and Science at three scales. Rows show EV (↑), Spearman's rank correlation (↑), and MAE (↓). Filled and hollow markers denote near-OOD (Qwen3-1.7B) and far-OOD (DeepSeek-V3.2) evaluation.

Case 1: Email Editing - Chat

Query
I hope this email find you well

Rubrics
1. Corrects the sentence.
2. Greets or offers help.
3. Offers a formal alternative.
4. Notes the incomplete start and requests details.

GT annotation *Advantage*

Vanilla PPO ✗

I hope this email finds you well.
I wanted to follow up on [briefly mention the topic or context of the email, e.g., "the discussion we had last week"] and see if there's anything further you'd like to discuss.
Please let me know if there's anything else I can assist with.
Let me know if you'd like to adjust the tone or add specific details!

Ours ✓

I hope this email finds you well.
I wanted to follow up on [briefly mention the topic or context of the email, e.g., "the discussion we had last week"] and see if there's anything further you'd like to discuss.
Please let me know if there's anything else I can assist with.
Let me know if you'd like to adjust the tone or add specific details!

Table 3: Token-level attribution for vanilla PPO (left) and $\mathcal{D}$VE (right) on a Chat example. Gray marks the ground-truth rubric-relevant spans; color intensity gives the token-level advantage. $\mathcal{D}$VE concentrates advantage on the annotated spans, while vanilla PPO spreads it across the response.

For example, at 1.7B, EV reaches approximately 0.6, compared with around 0.5 for vanilla PPO. Under far-OOD policies, the larger policy shift makes returns harder to predict, with relatively low EV. Nevertheless, $\mathcal{D}$VE still improves EV, with gains particularly evident at 4B. In Chat domain, its far-OOD EV reaches approximately 0.3, compared with around 0.0 for vanilla PPO.

$\mathcal{D}$VE improves response-level ranking consistency. Besides prediction errors, we evaluate whether the critic can distinguish relative response quality by computing the correlation between value predictions at the EOS token and observed rewards. As shown in Figure 4, $\mathcal{D}$VE achieves higher Spearman's rank correlation over most checkpoints across both domains and all three model sizes. Under near-OOD policies, the advantage persists across model sizes. For example, at 4B in the Chat domain, correlation reaches 0.8, compared with around 0.6 for vanilla PPO. Under far-OOD policies, $\mathcal{D}$VE maintains stronger ranking consistency. For example, at 1.7B in the Science domain, correlation reaches approximately 0.4, compared with around 0.2 for vanilla PPO.

4.5 Token-level Attribution Analysis

$\mathcal{D}$VE concentrates advantage on rubric-relevant tokens, while vanilla PPO spreads it. To explain the strength of $\mathcal{D}$VE more intuitively, we provide a case study and compare the token-level attribution patterns of vanilla PPO and $\mathcal{D}$VE. Table 3 presents an email editing example from the Chat training set, with the response generated by Qwen-3-1.7B. We compute token-level advantages using the critic trained by vanilla PPO and $\mathcal{D}$VE, respectively, and visualize them as heatmaps, where **darker** indicates **larger** advantage; underlines mark the ground-truth rubric-relevant spans, with annotation details in Appendix A.6. As shown in the Table 3, $\mathcal{D}$VE concentrates positive advantage on the phrases that satisfy the rubrics, such as the corrected opening, the offer of assistance, and the request for additional details, whereas vanilla PPO spreads advantage diffusely across the response. Additional cases appear in Appendix A.1.

5 Related Work

RL with Rubrics. Recent work introduces *Rubrics-as-Rewards* (Gunjal et al., 2026) to extend RL from objective and verifiable tasks (Guo et al., 2025; Jin et al., 2025) to broader open-ended tasks (Yifei et al., 2025; Arora et al., 2025), where response quality goes beyond correctness alone. Rather than relying on a small set of fixed reward functions (Xie et al., 2025), rubric-based RL associates each query with a multidimensional, query-specific set of rubrics, typically uses an LLM-as-judge to evaluate the response against each criterion independently, and aggregates the resulting scores into a scalar reward (Gunjal et al., 2026; Viswanathan et al., 2026). Recent work scales this construction further, with many more criteria per query and with automatically generated, dynamically updated rubrics (Shao et al., 2025; Li et al., 2026). This paradigm has shown promising performance across various open-ended tasks, including chat (Xu et al., 2026a; Chen et al., 2026), text generation (Huang et al., 2025), and complex instruction following (Zhang et al., 2025).

Estimating Advantage from Values. Advantage estimation can be divided into critic-free and critic-based categories. Critic-free methods such as GRPO (Shao et al., 2024) compare a response's reward with those of other responses to the same query and distribute the response-level advantage uniformly across tokens (Ahmadian et al., 2024; Yu et al., 2025). As tasks get more complex (Huang et al., 2026), several studies revisit actor–critic algorithms, particularly PPO, where a critic supplies token-level estimates (Schulman et al., 2015; Hou et al., 2026; Liu et al., 2026; Yue et al., 2025), and report that a more reliable critic yields more effective policy optimization (Yuan et al., 2025). We investigate an *information mismatch* in value estimation under rubric rewards: the reward is determined by query-specific criteria, but a standard critic model receives only the query and generation prefix, making it difficult to infer the underlying reward specification and estimate values accurately. $\mathcal{D}$VE addresses this by keeping the critic conditioned on rubrics throughout value estimation and also the latter advantage estimation stage. Concurrent work shares the idea of giving the critic prior knowledge (Venkatraman et al., 2026), but our ablation (§ 4.2) shows that this prior alone recovers only part of the gain.

Evaluating Value Estimation. Value estimation is a foundational component of critic-based RL, as its quality directly affects advantage estimation and, consequently, the learning signals used for policy optimization (Schulman et al., 2016; Kazemnejad et al., 2024). A common metric for evaluating value estimation is explained variance (EV), which measures how well critic predictions capture variation in observed returns (Good & Fletcher, 1981). VAPO (Yue et al., 2025) monitors EV during value pretraining as an indicator of critic learning quality. However, EV measures the variation captured by value predictions and provides limited insight into response-level consistency with observed rewards. To fill this gap, we systematically evaluate the critic model from both perspectives: token-level prediction quality and response-level ranking consistency with observed rewards.

6 Conclusion

In this work, we revisit value estimation for reinforcement learning with rubric rewards, where the outcome reward is derived from multiple query-specific criteria. Widely used algorithms such as PPO use a critic model to estimate the value of each generation state and derive token-level advantages from these value estimates. However, we identify an information mismatch in rubric-based RL: while rewards are determined by query-specific rubrics, a standard critic receives only the query and generation prefix, making it difficult to infer the underlying reward specification from scalar supervision. To address this issue, we propose $\mathcal{D}$isentangled **V**alue **E**stimation ($\mathcal{D}$VE), a rubric-conditioned value estimation framework that preserves fine-grained rubric information throughout critic learning and the latter advantage estimation stage. $\mathcal{D}$VE estimates rubric-specific values and token-level advantages separately with a shared critic model and aggregates them using the original rubric weights for policy optimization. Across five benchmarks spanning instruction following, chat, and scientific question answering, $\mathcal{D}$VE outperforms vanilla PPO and GRPO in 13 of 15 model–benchmark combinations. Further analyses show that $\mathcal{D}$VE improves value estimation, with lower prediction errors and stronger correlations between value predictions and observed rewards. These results highlight the importance of preserving fine-grained reward specifications throughout critic learning and advantage estimation.

AI USE STATEMENT

In writing this paper, large language models were used solely to check spelling and grammar during the preparation of this manuscript. No AI tools were used for theory or proof assistance, hypothesis generation and experiment design. All research ideas, methods, experiments, analyses were developed by the authors, who take full responsibility for the content of this paper.

ETHICAL CONSIDERATIONS

This work does not involve human subjects research. All datasets, and pretrained models used in this work are publicly available and used in accordance with their intended research use and applicable licenses. We provide dataset descriptions, evaluation settings in the main body of the paper and detailed explanation in Appendix. No part of this work includes deceptive practices or intentional misuse of information. We are committed to conducting and presenting this research with integrity and social responsibility. We do not foresee any direct ethical risks or misuse beyond those already present in large language models.

REPRODUCIBILITY STATEMENT

We provide the source code and instructions for reproducing our experiments at Anonymous. All training and evaluation settings and hyperparameters are listed in § 3 and the Appendix.

REFERENCES

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.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 88313–88342, 2026.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. Ticking all the boxes: Generated checklists improve llm evaluation and generation. *arXiv preprint arXiv:2410.03608*, 2024.

Ron Good and Harold J Fletcher. Reporting explained variance. *Journal of Research in Science Teaching*, 18(1):1–7, 1981.

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

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.

Timothy O Hodson. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, 2022:1–10, 2022.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.07508*, 2026.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Chenghua Huang, Lu Wang, Fangkai Yang, Pu Zhao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Pretrain value, not reward: Decoupled value policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 115045–115066, 2026.

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

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Sunzhu Li, Jiale Zhao, Huimin Ren, Zhenlin Wei, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Chen Wei. Rubrichub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31320–31344, 2026.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline, 2024. URL https://arxiv.org/abs/2406.11939.

Jiashun Liu, Johan S Obando Ceron, Han Lu, Yancheng He, Weixun Wang, Bo Zheng, Pablo Samuel Castro, Aaron Courville, Ling Pan, et al. Asymmetric proximal policy optimization: Mini-critics boost llm reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 12574–12593, 2026.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Chengjun Pan, Shichun Liu, Jiahang Lin, Dingwei Zhu, Jiazheng Zhang, Shihan Dou, Songyang Gao, Zhenhua Han, Binghai Wang, Rui Zheng, et al. Evpo: Explained variance policy optimization for adaptive critic utilization in llm post-training. *arXiv preprint arXiv:2604.19485*, 2026.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025. URL https://arxiv.org/abs/2507.02833.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, et al. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, 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. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Siddarth Venkatraman, Matthieu Dinot, and Laurence Aitchison. Le critique: Privileged value functions for llm reinforcement learning. *arXiv preprint arXiv:2608.16739*, 2026.

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Sherry Wu. Checklists are better than reward models for aligning language models. *Advances in Neural Information Processing Systems*, 38:114728–114754, 2026.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

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

Tianze Xu, Yanzhao Zheng, Pengrui Lu, Lyumanshan Ye, Yong Wu, Zhentao Zhang, Yuanqiang Yu, Chao Ma, Jihuai Zhu, Pengfei Liu, et al. Rubrics to tokens: Bridging response-level rubrics and token-level rewards in instruction following tasks. *arXiv preprint arXiv:2604.02795*, 2026b.

Junjie Ye, Caishuang Huang, Zhuohan Chen, Wenjie Fu, Chenyuan Yang, Leyi Yang, Yilong Wu, Peng Wang, Meng Zhou, Xiaolong Yang, Tao Gui, Qi Zhang, Zhongchao Shi, Jianping Fan, and Xuanjing Huang. Muldimif: A multi-dimensional constraint framework for evaluating and improving instruction following in large language models, 2026. URL https://arxiv.org/abs/2505.07591.

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, 2025. URL https://arxiv.org/abs/2509.00496.

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, 2025. *URL https://arxiv. org/abs/2503.14476*, 1(2), 2025.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Wenjian Zhang, Min Cen, Yang Zhou, Wenkai Fang, Yiru Zhao, Baisheng Lai, and Mingli Song. Replay failures as successes: Sample-efficient reinforcement learning for instruction following, 2025. URL https://arxiv.org/abs/2512.23457.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A Appendix

A.1 Token Attribution Case Studies

Tables 4–11 extend the case study of Section 4.5 from the Chat and Science training sets, spanning critic sizes from 0.6B to 4B and response lengths from 63 to 1,119 tokens. Each table gives the query and its annotated rubrics, followed by the token-level advantages computed with the critic trained by vanilla PPO (left) and by $\mathcal{D}$VE (right), where **darker** indicates **larger** advantage and underlines mark the ground-truth rubric-relevant spans. Across all eight cases, $\mathcal{D}$VE concentrates advantage on the underlined spans, whereas vanilla PPO spreads it across the response.

Query

I hope this email find you well

Annotated rubrics

1. Supplies the corrected sentence.
2. Courteously acknowledges the greeting or offers help.
3. Gives a more formal alternative phrasing.
4. Notes the incomplete opening and asks for the rest.

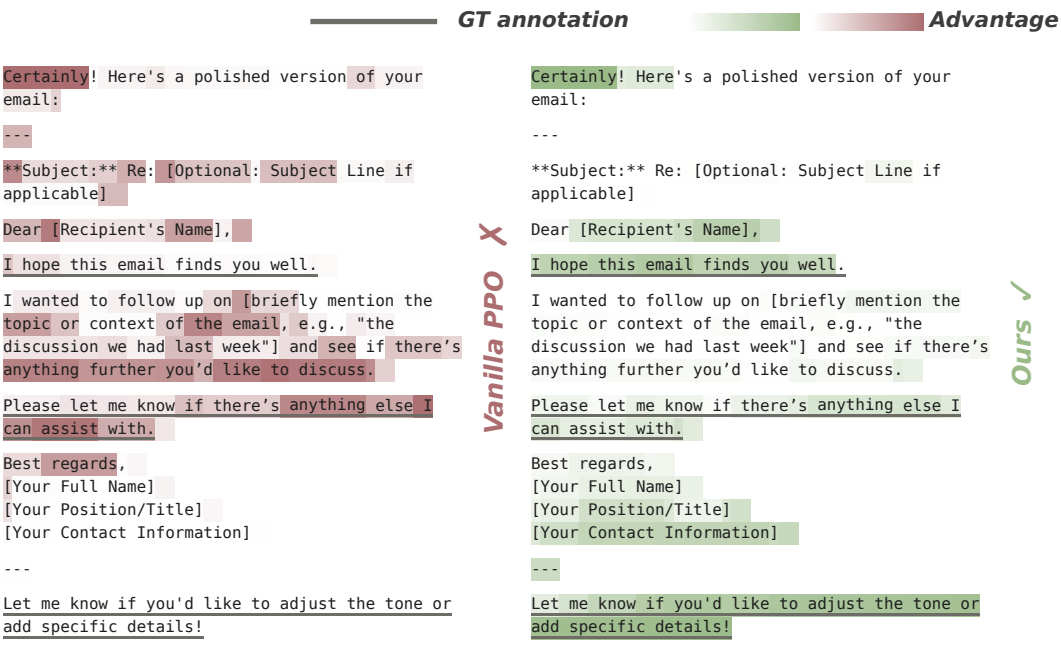


Table 4: **Case 1: Email Editing** (Chat, 1.7B critic). The user sends only an ungrammatical greeting, with no instruction. The rubrics ask for the corrected sentence, an explanation of the subject–verb agreement, and a request for the rest of the email.

Query
A priest, a rabbi, and a nun walk into a bar…

Annotated rubrics
1. Delivers a complete punchline for the setup.
2. Avoids clichéd or caricatured religious humor.
3. Adds a follow-up analysis of the joke's mechanics.
4. The bartender delivers the punchline.
5. Includes a twist explainable in 15 words or fewer.

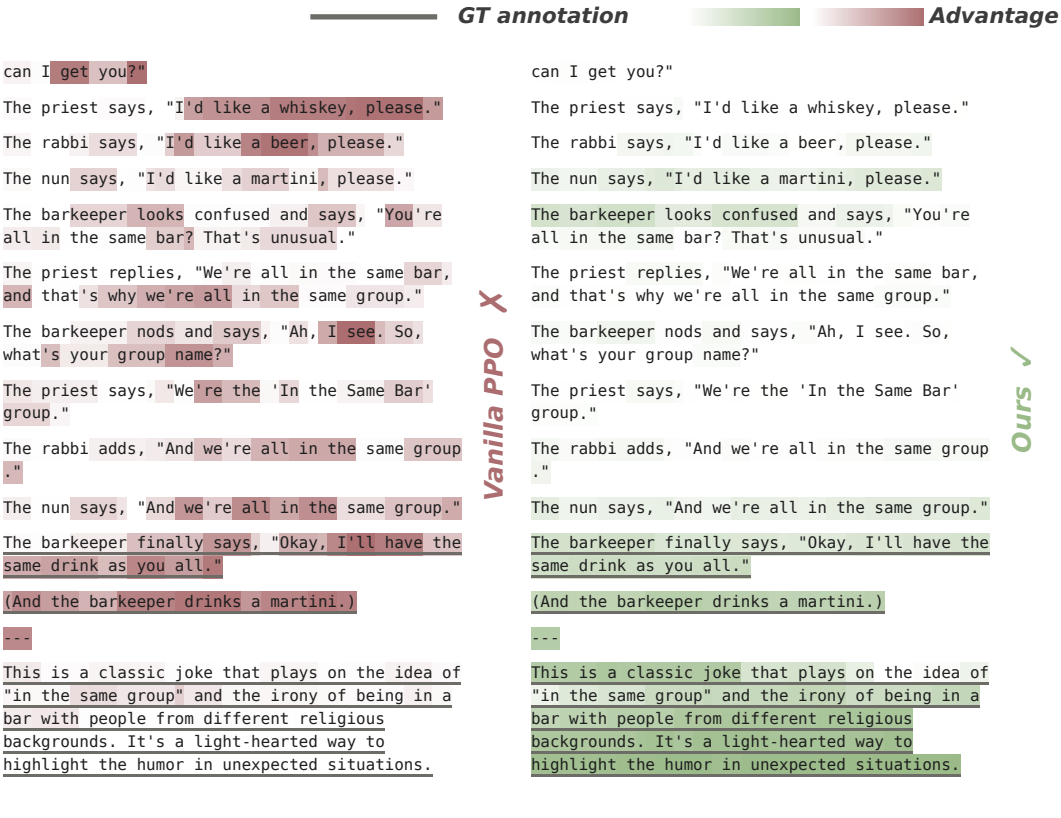


Table 5: **Case 2: Joke Writing** (Chat, 1.7B critic). A joke set-up with no further instruction. The rubrics ask for a coherent punchline, humor that avoids clergy stereotypes, and a follow-up explanation of how the joke works.

Query

I'd like to play DND, with you as the dungeon master. Will you play?

Annotated rubrics

1. Explicitly agrees to act as Dungeon Master.
2. Groups the follow-up questions into labeled sections.
3. Uses bold, bullets, or numbering for the setup questions.
4. Ends by inviting the user to supply the details.



Table 6: **Case 3: Dungeon Master** (Chat, 1.7B critic). The user asks the assistant to run a D&D session. The rubrics ask it to agree and then set the game up: rule set, play style, tone, character details and dice handling, laid out as clearly formatted questions.

Query
Tell my funny joke about space

Annotated rubrics
1. Opens with a brief courteous lead-in before the jokes.
2. Uses a space term such as “Milky Way” or “black hole”.

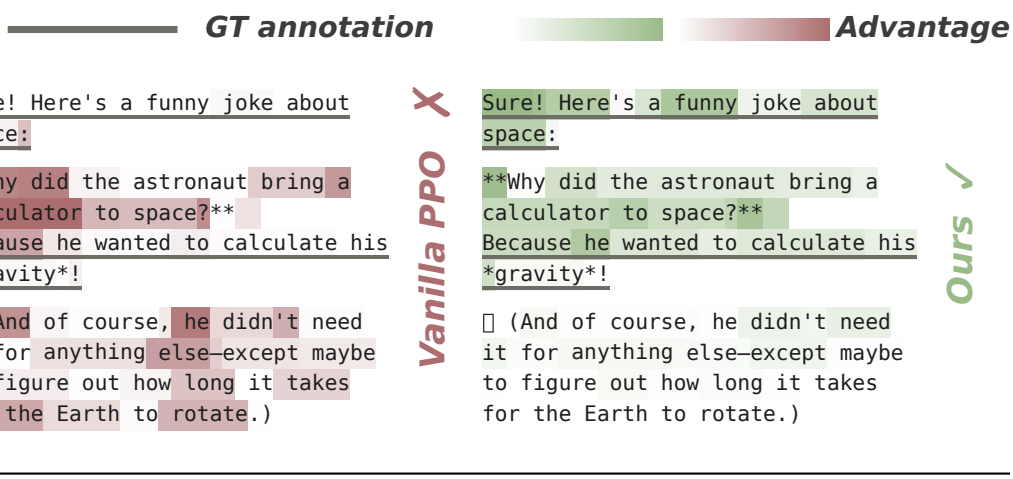


Table 7: **Case 4: Space Joke** (Chat, 1.7B critic). A request for a funny joke about space. The rubrics ask for original, clean, space-themed jokes in a recognizable set-up and punchline format.

Query
wlp3s0

Annotated rubrics
1. Notes the interface may instead be named `wlan0` or similar.
2. Asks the user for their goal or Linux distribution.

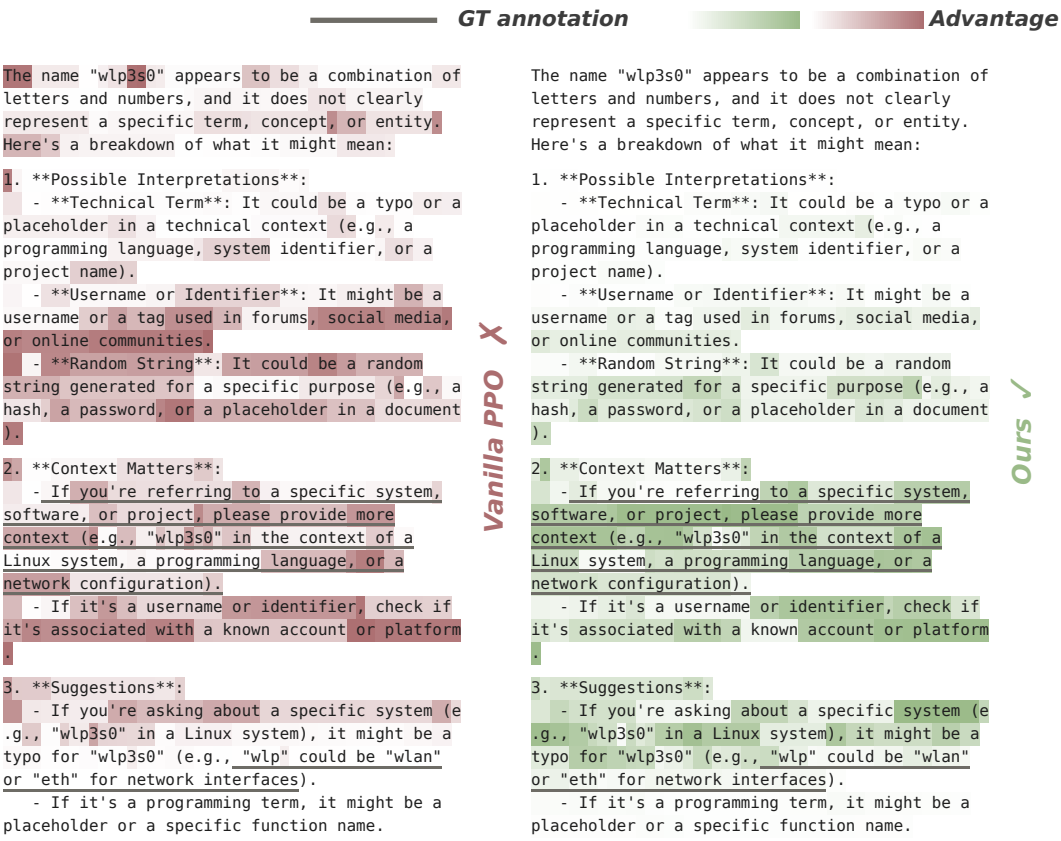


Table 8: **Case 5: Network Interface** (Chat, 4B critic). The user sends only `wlp3s0`, a Linux wireless interface name. The rubrics ask the model to identify it, decode the naming convention, and give diagnostic commands.

Query

What are the key material-related challenges and limitations of using Selective Laser Sintering (SLS) for fabricating biomedical scaffolds?

Annotated rubrics

1. Only a limited palette of medical-grade powders is compatible.
2. Particle size and flow govern layer quality and porosity.
3. Biodegradable polymers have a narrow sintering window.
4. Pore size and interconnectivity are hard to control.
5. Residual porosity and anisotropic strength.
6. Filler segregation in polymer-ceramic composites.
7. Material-parameter coupling; no predictive models.
8. Granular surface finish affects cell adhesion.

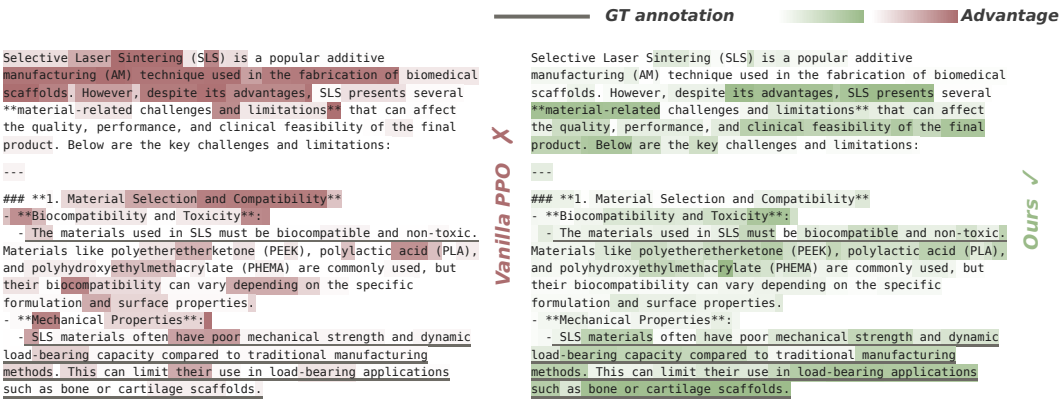


Table 9: **Case 6: SLS Scaffolds** (Science, 4B critic). A materials-science question on the limits of selective laser sintering for biomedical scaffolds. The rubrics enumerate the expected points: the narrow sintering window, thermal degradation, pore control, residual porosity and anisotropic strength, and composite segregation.

Query

Write the number “Six million, forty thousand, seven” in digits.

Annotated rubrics

1. The value equals 6,040,007.
2. Commas separate groups of three digits.
3. No leading zeros.
4. No number words remain.
5. The millions digit is 6.
6. The thousands group reads 040.
7. The ones group reads 007.
8. No quotation marks, colons, or letters.

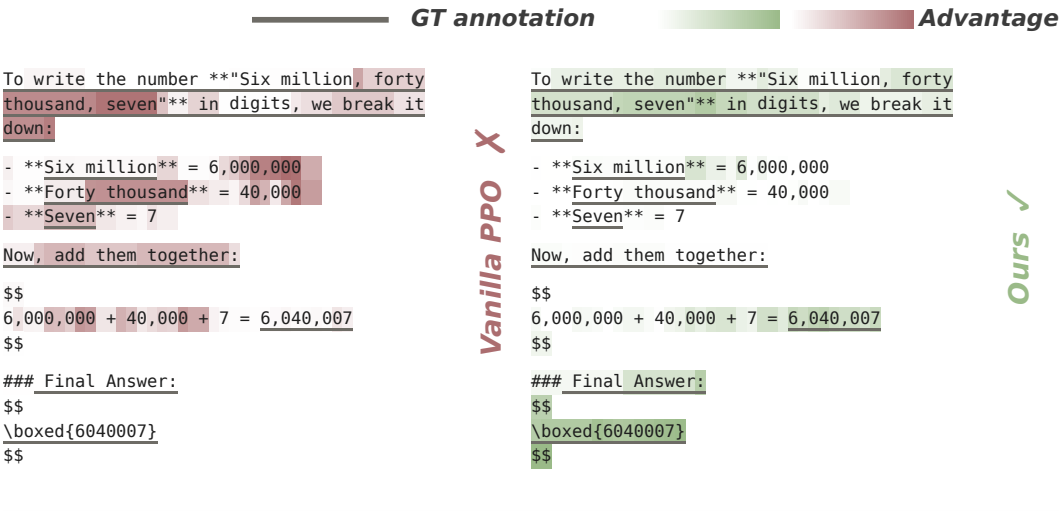


Table 10: **Case 7: Number to Digits** (Science, 0.6B critic). Convert a number written in words into digits. The rubrics ask for the value 6,040,007 with correct digit grouping and, strictly, for a response containing nothing but the digit string.

Query

How does the number of mutations in the Pfdhfr gene influence the risk of therapeutic failure according to the compiled studies?

Annotated rubrics

1. Single or double mutations give only reduced sensitivity.
2. States the scope limits or assumptions.
3. Uses precise failure terminology.

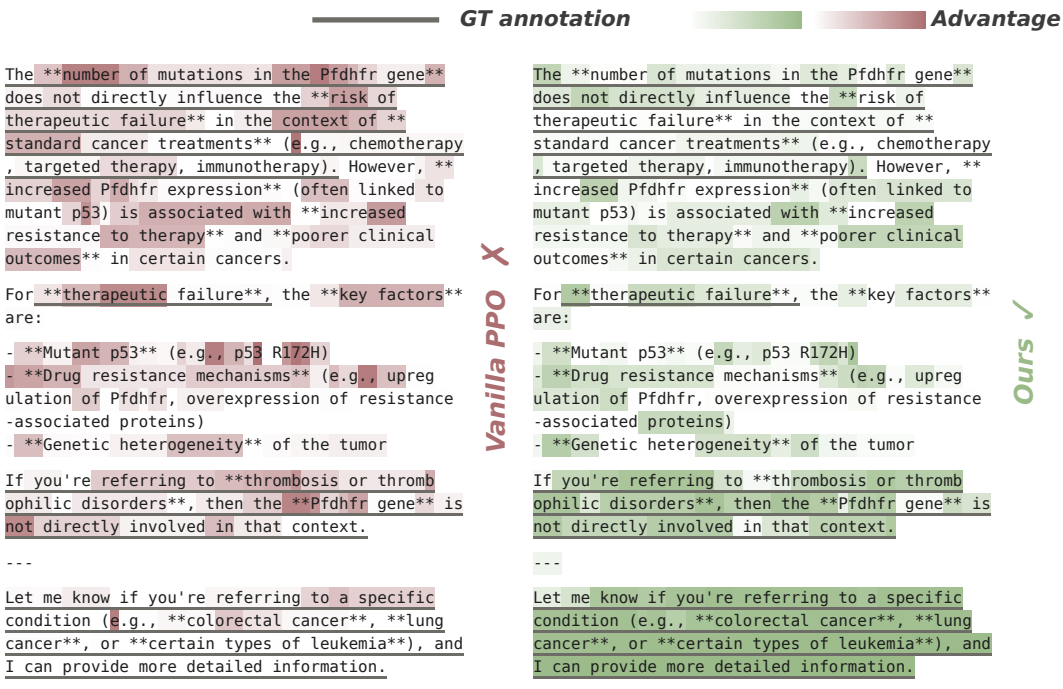


Table 11: **Case 8: Pfdhfr Mutations** (Science, 0.6B critic). A malaria pharmacogenetics question on how the number of Pfdhfr mutations affects sulfadoxine–pyrimethamine treatment failure. The rubrics ask for the stepwise dose–response relation across codons 51, 59, 108 and 164, and the role of the partner gene Pfdhps.

A.2 Training EV

Figure 5 reports explained variance during training for $\mathcal{D}$VE and vanilla PPO across all domains and model scales, extending the results shown in Figure 3 of the main body of our paper. Explained variance falls below zero early in training and during occasional instabilities, before the critic has fitted the reward scale; the vertical axis is restricted to the region where the two methods differ, so those excursions are clipped. Thin lines show raw per-step values and thick lines show smoothed trends.

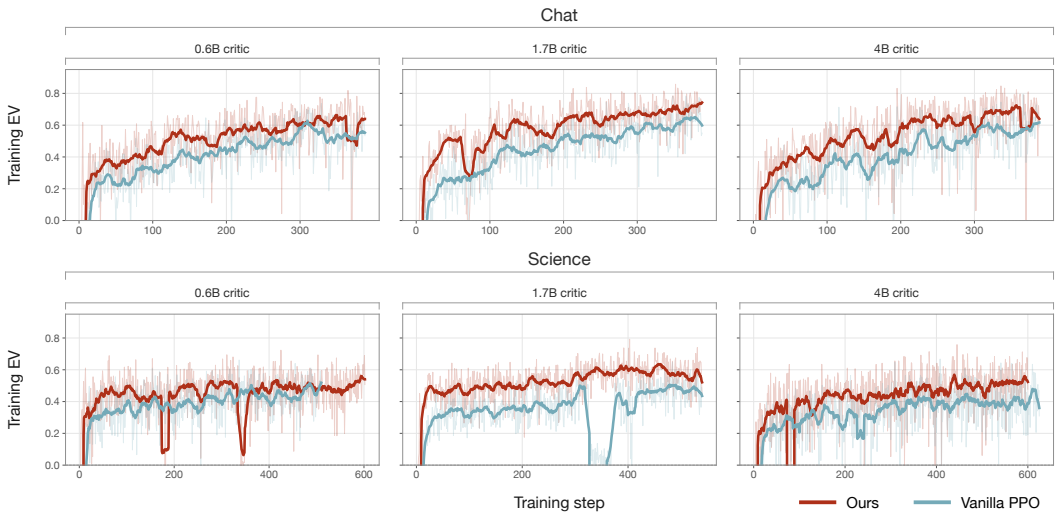


Figure 5: Training explained variance for $\mathcal{D}$VE and vanilla PPO. Higher is better. The vertical axis is clipped at zero. Thin lines show raw values; thick lines show smoothed trends.

A.3 Evaluation Variability

Figure 6 supplements Table 1 with evaluation variability on Chat (Arena-Hard-V2) and Science (ResearchQA). For each trained checkpoint, we conduct three independent evaluation runs, regenerating responses and repeating evaluation in each run. Points show the mean score improvement over the corresponding base model, and error bars show one standard deviation across runs. The base-model score serves as a fixed reference. These bars describe evaluation variability for a fixed checkpoint, rather than variability across independent training runs.

$\mathcal{D}$VE achieves larger mean improvements than vanilla PPO in all six domain and scale combinations. The gains are particularly clear at 0.6B in both domains and at 4B on the chat domain. At 1.7B on Chat, the mean improvements are close, and the error bars overlap. The figure therefore complements the mean scores in Table 1 by showing the scale of evaluation variability alongside the observed performance differences.

A.4 Datasets

A.4.1 Training Sets

Table 12 summarizes the three training sets. The instruction-following set contributes 16,712 queries and 121,127 rubrics, with a median of 7 rubrics per query and a maximum of 17. Chat and Science are drawn from RubricHub and are considerably denser: both have a median of 27 rubrics per query, yielding 246,796 rubrics over 9,312 Chat queries and 777,960 over 28,918 Science queries, with maxima of 50 and 54. Rubric weights are uniform in instruction following, while Chat and Science use 17 and 18 distinct integer weights in the range 0 to 20. Instruction-following rubrics are verified by a deterministic checker 62.2% of the time and by the LLM judge otherwise; all Chat and Science

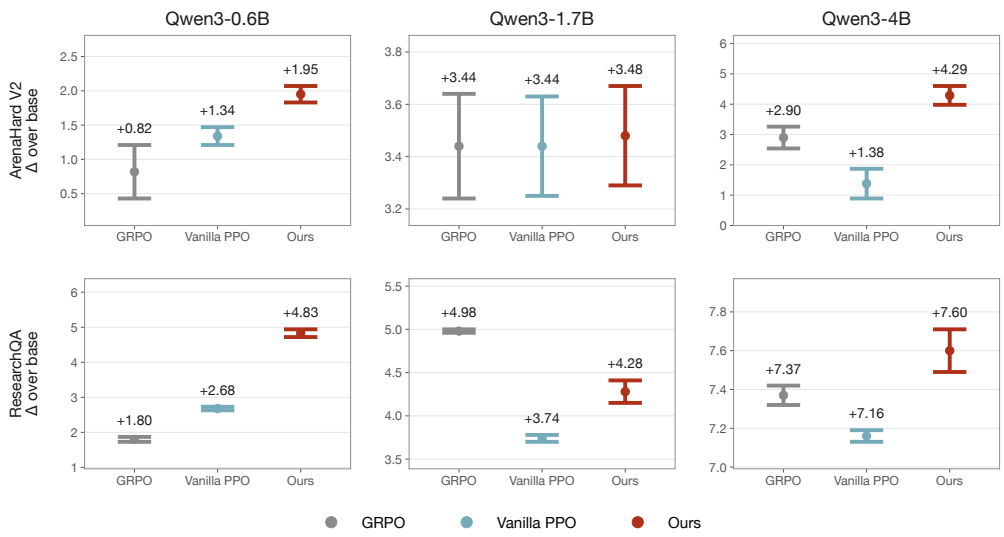


Figure 6: Evaluation variability on Chat (top) and Science (bottom). Columns correspond to Qwen3 models at 0.6B, 1.7B, and 4B. Points show mean improvements over the base model in percentage points; error bars indicate ± one standard deviation across three independent evaluation runs.

	Instruction Following	Chat	Science
Source	HiR-16K	RubricHub	RubricHub
Training queries	16,712	9,312	28,918
Total rubrics	121,127	246,796	777,960
Rubrics per query $\lvert\mathcal{C}(x)\rvert$			
min / max	5 / 17	4 / 50	9 / 54
Q1 / median / Q3	6 / 7 / 8	22 / 27 / 31	22 / 27 / 31
mean	7.25	26.50	26.90
Rubric weights w_k			
range	1 (uniform)	0–20	0–20
distinct values	1	17	18
Rubric verification			
rule-based	62.2%	–	–
LLM judge	37.8%	100%	100%
Query length (whitespace words)			
median / p90 / max	111 / 318 / 1,721	14 / 130 / 2,869	29 / 80 / 1,473
Optimizer steps	700	388	1,208

Table 12: Training data statistics for the three rubric-based domains. Optimizer steps assume a batch size of 96 over four epochs. Query length is reported in whitespace-delimited words rather than tokens.

rubrics are graded by the judge. Queries are longest in instruction following, with a median of 111 whitespace words, and shortest in Chat, with a median of 14.

A.4.2 Policy Benchmarks

We evaluate policy quality on five benchmarks spanning the three training domains. Evaluation protocol differs sharply across them: all three instruction-following benchmarks are checked entirely by deterministic programs and involve no LLM call at evaluation time, whereas Chat and Science are scored by `deepseek-v3.2` at temperature 0.

Instruction Following. We report strict prompt-level and instruction-level accuracy on IFEval and IFBench, and instruction-level accuracy on MulDimIF, giving 2,035 prompts and 5,349 individual instructions in total. All three benchmarks are verified entirely by deterministic checkers, without LLM-as-judge.
Because `deepseek-v3.2` serves as the rubric judge during *training*, the entire *evaluation* of instruction following is free of any dependence between the training reward and the evaluation metric.

Benchmark	Prompts	Instr.	Judge
IFEval	541	834	rule
IFBench	294	335	rule
MulDimIF	1,200	4,180	rule
Total	2,035	5,349	

Table 13: Instruction-following benchmarks.

Chat. We evaluate on the Arena-Hard-v2 prompt set, comprising 500 hard prompts and 250 creative writing prompts, with the released `gemini-2.0-flash-001` responses as a fixed baseline opponent rather than regenerating them. The released baseline covers 748 of the 750 prompts, so we score on that intersection. Each prompt is judged twice, with the candidate and baseline responses presented in both orders so that position bias enters symmetrically. None of the 750 evaluation prompts appears among the 9,312 Chat training queries.

Arena-Hard-v2	
Prompts (hard / creative)	750 (500 / 250)
Prompts scored	748
Comparisons per run	1,496
Judgments per checkpoint	4,488
Overlap with training set	0

Table 14: Chat benchmark.

Science. ResearchQA releases a test split of 3,750 questions, a test-mini split of 776, and a validation split of 703. Rather than using a released split directly, we filter ResearchQA to questions with zero overlap with the science training and validation sets, and evaluate on 300 such questions, comprising 2,395 rubric criteria. We present each criterion to the judge in a separate call.

ResearchQA	
Questions evaluated	300
Rubric criteria	2,395
Criteria per question	7.98
Judging calls per run	2,395
Judgments per checkpoint	7,185

Table 15: Science benchmark.

A.5 Evaluating Value Estimation Test Set

Training rollouts are produced by each method's own evolving policy, so explained variance measured during training reflects both critic quality and differences in the rollout distribution. To remove that dependence we score every critic on a fixed set of responses. This section describes how those response sets were built and reports their statistics.

Prompt selection. For each of Chat and Science we draw 2,000 prompts uniformly at random from the domain's training pool using a fixed seed, and discard prompts carrying no rubrics. This leaves 1,998 prompts for Chat and 2,000 for Science. The two response sets within a domain use exactly the same prompts, so the near-OOD and far-OOD conditions differ only in the policy that produced the responses. Prompts are formatted through the same pipeline used to compute rewards during training, namely the chat template followed by truncation to 2,048 tokens and decoding without special tokens, so that the critic sees inputs in the format it was trained on.

Response generation. The near-OOD responses come from Qwen3-1.7B served locally, and the far-OOD responses from DeepSeek-V3.2 through its API. Both use temperature 0.7 and a 2,048-token limit, with top-p set to 1.0 for the local model. Each response is then scored by the same rubric judge used during training: every rubric receives a binary verdict $R(x, y, c_k) \in \{0, 1\}$, and the verdicts are combined with the normalized rubric weights into a scalar reward $R \in [0, 1]$.

	Chat		Science	
	Near-OOD	Far-OOD	Near-OOD	Far-OOD
Responses	1,998	1,998	2,000	2,000
Rubrics per prompt	26.4 (8–40)		26.8 (10–40)	
Mean R	0.393	0.650	0.386	0.753
Std. R	0.229	0.206	0.249	0.169
Median R	0.391	0.692	0.365	0.779
$R = 0$	3.4%	1.3%	3.0%	0.4%
$R = 1$	0.5%	1.6%	0.7%	6.1%
Rubric satisfaction	0.403	0.656	0.398	0.759
Response length (chars)	2,329	2,954	3,849	4,584

Table 16: Statistics of the fixed response sets used to evaluate value estimation. Near-OOD responses are generated by Qwen3-1.7B and far-OOD responses by DeepSeek-V3.2, from an identical prompt set within each domain. Rubrics per prompt are reported as the mean with the observed range and are shared across the two conditions.

Properties of the two conditions. Three features of Table 16 matter for interpreting the value-estimation results. First, near-OOD responses have more reward variance than far-OOD responses (0.229 and 0.249 against 0.206 and 0.169). Because explained variance is divided by $\mathrm{Var}(R)$, a small denominator makes the metric unstable; the near-OOD condition is better behaved for this reason. Second, the near-OOD rubric satisfaction rate of roughly 0.40 sits close to the rate observed during training, whereas the far-OOD rate of 0.66 to 0.76 is substantially higher, reflecting both stronger and more homogeneous responses. Third, the reward distributions of the two conditions barely overlap, with mean rewards differing by more than 0.25, so MAE should be compared within each condition rather than across conditions.

Scope. The prompts are drawn from the training pool rather than from a held-out split. The *out-of-distribution* refers to the response distribution, since no critic was trained on rollouts from either of these two policies. The prompts themselves are in-distribution by construction, which is deliberate: it isolates the effect of policy shift from that of prompt shift. We also note that the far-OOD responses come from the same model family that serves as the rubric judge, which plausibly contributes to the higher satisfaction rate observed in that condition. Since both critics are scored on the identical set of responses and rewards, this affects the difficulty of the far-OOD condition rather than the comparison between methods within it.

A.6 Token-Level Relevance Annotation

We annotate 1,00 responses to RL training prompts in each of the Chat and Science domains. For each (response, rubric pair), an LLM annotator identifies character spans containing the evidence needed to judge whether the rubric is satisfied. Relevant spans may indicate either satisfaction or violation of the rubric. Rubric text, order, and weights follow the training data. The annotations distinguish local from global criteria and flag evidence that does not match the response. These LLM-generated annotations serve as reference relevance labels.

We map the character spans to Qwen3 tokens using tokenizer offsets, labeling a token as relevant if its character interval overlaps any annotated span. Labels are binary and do not encode degrees of relevance. We exclude global rubrics, rubrics whose evidence cannot be matched to the response, and rubrics whose tokens are all relevant or all irrelevant, since none support within-response ranking. We also exclude responses exceeding the 4,096-token critic context. After filtering, the evaluation contains 13,881 rubrics in Chat and 18,441 in Science.

A.7 Training Details

We implement all experiments in verl (Sheng et al., 2025). $\mathcal{D}$VE and all baselines share the same actor optimization, KL, rollout, and sequence-length settings, and differ only where the algorithm requires it.

Table 17 lists the PPO training configuration.

Hyperparameter	Value
PPO training batch size	96
PPO mini-batch size	24
PPO epochs per batch	1
Actor learning rate	1×10^{-6}
Critic learning rate	1×10^{-5}
Learning-rate warmup ratio	10gan jganj
Actor gradient clipping	1.0
PPO clipping ratio ϵ (lower / upper)	0.2 / 0.2
Entropy coefficient	0
KL loss	Enabled
KL loss coefficient	0.01
KL loss type	`low_var_kl`
Discount factor γ	1.0
GAE parameter λ	1.0
Maximum prompt length	2,048 tokens
Maximum response length	4,096 tokens
Rollout temperature	1.0
Rollout responses per prompt	1
Total training epochs	4
Data seed	1
Distributed training strategy	FSDP
Dynamic batching	Enabled
Maximum PPO tokens per GPU	8,192

Table 17: Hyperparameters for PPO-based training.

Table 18 lists the configurations used for the two baselines.

Hyperparameter	PPO baseline	GRPO baseline
Training batch size	96	96
Mini-batch size	24	24
Epochs per batch	1	1
Actor learning rate	1×10^{-6}	1×10^{-6}
Critic learning rate	1×10^{-5}	– (no critic)
Learning-rate warmup ratio	10	10
Actor gradient clipping	1.0	1.0
PPO clipping ratio ϵ (lower / upper)	0.2 / 0.2	0.2 / 0.2
Entropy coefficient	0	0
KL loss	Enabled	Enabled
KL loss coefficient	0.01	0.01
KL loss type	`low_var_kl`	`low_var_kl`
Discount factor γ	1.0	– (no GAE)
GAE parameter λ	1.0	– (no GAE)
Maximum prompt length	2,048 tokens	2,048 tokens
Maximum response length	4,096 tokens	4,096 tokens
Rollout temperature	1.0	1.0
Rollout responses per prompt	1	5
Total training epochs	4	4
Distributed training strategy	FSDP	FSDP
Dynamic batching	Enabled	Enabled
Maximum PPO tokens per GPU	8,192	8,192

Table 18: Hyperparameters for the PPO and GRPO baselines.

Both share the optimizer, KL, and sequence-length settings of $\mathcal{D}$VE reported in Appendix A.7, and differ only where the algorithm requires it. Two differences are worth stating explicitly. GRPO has no critic, so the critic learning rate and critic warmup do not apply to it. GRPO also samples five

responses per prompt, since its advantage is defined relative to a group, whereas PPO and $\mathcal{D}$VE sample one; GRPO therefore consumes five times the rollout budget per gradient step.

A.8 Prediction-Error Analysis of $\mathcal{D}$VE

Notation and evaluation setting. We restate the notation from § 2.3 for completeness. Let x denote a query, $y = (y_1, \ldots, y_{|y|})$ a response, and $y_{<t}$ its prefix at a valid response-token position $t \in \{1, \ldots, |y|\}$. Each query is associated with a checklist $\mathcal{C} = \{(c_k, w_k)\}_{k=1}^{|\mathcal{C}|}$, where c_k is a rubric and $w_k \geq 0$ its weight. Assume $\sum_k w_k > 0$, and write $\hat{w}_k = w_k / \sum_{j=1}^{|\mathcal{C}|} w_j$ for the normalized weights. The checklist and its weights are fixed for each query.

The reward assigned to response y under rubric c_k is $R(x, y, c_k)$, and the scalar outcome reward is

$$R(x, y, \mathcal{C}) = \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k R(x, y, c_k). \tag{11}$$

The shared critic in $\mathcal{D}$VE produces the rubric-level prediction $V_{\theta,t}^k = V_\theta(x, y_{<t}, c_k)$. For brevity, denote its aggregated scalar prediction by $V_{\mathcal{D}VE,t} = \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k V_{\theta,t}^k$. Let $V_{\text{vanilla},t}$ denote the vanilla critic's scalar prediction from $(x, y_{<t})$ alone.

All expectations below are taken under the same fixed distribution of queries, responses, and valid token positions. The critic parameters are held fixed, and all rewards and predictions are assumed to have finite second moments. Both critics are evaluated against the same scalar outcome reward:

$$\begin{aligned} \text{MSE}_{\text{vanilla}} &= \mathbb{E}\left[\left(R(x, y, \mathcal{C}) - V_{\text{vanilla},t}\right)^2\right], \\ \text{MSE}_{\mathcal{D}VE} &= \mathbb{E}\left[\left(R(x, y, \mathcal{C}) - V_{\mathcal{D}VE,t}\right)^2\right]. \end{aligned} \tag{12}$$

These quantities measure scalar prediction error, rather than the criterion-level training loss of $\mathcal{D}$VE.

We use θ generically to denote method-specific critic parameters, without implying parameter sharing between the vanilla and rubric-conditioned critics.

Proposition 1 (Prediction error under exact rubric-level recovery)**.** *Suppose the shared critic exactly recovers each rubric-level conditional mean: $V_{\theta,t}^k = \mathbb{E}[R(x, y, c_k) \mid x, y_{<t}]$ almost surely for all $k = 1, \ldots, |\mathcal{C}|$. Then its aggregated prediction satisfies $V_{\mathcal{D}VE,t} = \mathbb{E}[R(x, y, \mathcal{C}) \mid x, y_{<t}]$. Consequently, for any vanilla critic $V_\theta(x, y_{<t})$ satisfying the above conditions,*

$$\begin{aligned} \text{MSE}_{\text{vanilla}} - \text{MSE}_{\mathcal{D}VE} &= \mathbb{E}\left[\left(V_\theta(x, y_{<t}) - V_{\mathcal{D}VE,t}\right)^2\right] \\ &= \mathbb{E}\left[\left(V_\theta(x, y_{<t}) - \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k V_{\theta,t}^k\right)^2\right] \geq 0. \end{aligned} \tag{13}$$

The inequality is strict if and only if $\Pr(V_\theta(x, y_{<t}) \neq V_{\mathcal{D}VE,t}) > 0$.

Proof. We first establish that the aggregated prediction recovers the conditional mean of the scalar reward. Because the checklist and rubric weights are fixed for each query, they are fixed when conditioning on $(x, y_{<t})$. Using the exact-recovery assumption and linearity of conditional expectation,

$$\begin{aligned} V_{\mathcal{D}VE,t} &= \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k V_{\theta,t}^k \\ &= \sum_{k=1}^{|\mathcal{C}|} \hat{w}_k \mathbb{E}[R(x, y, c_k) \mid x, y_{<t}] \\ &= \mathbb{E}\left[\sum_{k=1}^{|\mathcal{C}|} \hat{w}_k R(x, y, c_k) \,\middle|\, x, y_{<t}\right] \\ &= \mathbb{E}[R(x, y, \mathcal{C}) \mid x, y_{<t}]. \end{aligned} \tag{14}$$

Therefore, the prediction residual has zero conditional mean: $\mathbb{E}[R(x, y, \mathcal{C}) - V_{\mathcal{DVE},t} \mid x, y_{<t}] = 0$.

We next compare the two scalar prediction errors directly. Define $\Delta_t = V_\theta(x, y_{<t}) - V_{\mathcal{DVE},t}$. Since $R(x, y, \mathcal{C}) - V_\theta(x, y_{<t}) = R(x, y, \mathcal{C}) - V_{\mathcal{DVE},t} - \Delta_t$, expanding the squared errors gives

$$\begin{aligned} \mathrm{MSE}_{\mathrm{vanilla}} - \mathrm{MSE}_{\mathcal{DVE}} = \mathbb{E}[\Delta_t^2] \\ - 2\mathbb{E}\left[\Delta_t\big(R(x, y, \mathcal{C}) - V_{\mathcal{DVE},t}\big)\right]. \end{aligned} \tag{15}$$

With the critic parameters fixed, both scalar predictions are functions of $(x, y_{<t})$, and so is Δ_t. Applying the law of iterated expectations, the cross term vanishes:

$$\begin{aligned} &\mathbb{E}\left[\Delta_t\big(R(x, y, \mathcal{C}) - V_{\mathcal{DVE},t}\big)\right] \\ &\quad = \mathbb{E}\left[\Delta_t \mathbb{E}\left[R(x, y, \mathcal{C}) - V_{\mathcal{DVE},t} \mid x, y_{<t}\right]\right] = 0. \end{aligned} \tag{16}$$

Substituting this identity into Eq. (15) yields $\mathrm{MSE}_{\mathrm{vanilla}} - \mathrm{MSE}_{\mathcal{DVE}} = \mathbb{E}[\Delta_t^2] \geq 0$. Replacing Δ_t with its definition gives Eq. (13), which is the MSE difference stated in Eq. (7) of the main text.

Finally, $\mathbb{E}[\Delta_t^2] = 0$ if and only if $\Delta_t = 0$ almost surely. Thus, the inequality is strict exactly when the two scalar predictions differ with positive probability. □

Relation to the uniformly weighted critic loss. The criterion-level loss in § 2.2 is

$$\mathcal{L}_V(\theta) = \frac{1}{|\mathcal{C}|} \sum\nolimits_{k=1}^{|\mathcal{C}|} \frac{1}{|y|} \sum\nolimits_{t=1}^{|y|} \left(V_{\theta,t}^k - \hat{V}_t^k\right)^2, \tag{17}$$

where $\hat{V}_t^k$ is the return target constructed from the rubric-specific reward sequence. This objective weights rubric-level prediction tasks uniformly, whereas $\hat{w}_k$ specifies how their predictions reconstruct the scalar value.

To connect this loss to the proposition, consider complete undiscounted return targets in the terminal-only reward setting with $\gamma = 1$. In this setting, $\hat{V}_t^k = R(x, y, c_k)$ at every valid token position. For this connection, take the common distribution in the proposition to be the distribution induced by the population loss's query, response, and token-position averaging. Each criterion-level squared-loss term is then minimized by the corresponding conditional mean. If a single shared critic simultaneously recovers all these means, it minimizes every term and therefore their uniformly weighted population loss.

Thus, uniform weighting of the criterion-level loss is compatible with exact recovery of the scalar conditional mean. The training weights need not match the reward weights $\hat{w}_k$: the former allocate supervision across criteria, while the latter determine how the predictions reconstruct the scalar value. This observation concerns the population objective under simultaneous exact recovery; it does not imply that finite-data optimization attains that recovery.

Interpretation. Under the exact-recovery assumption of Proposition 1, the weighted rubric-level predictions reconstruct the conditional mean of the scalar reward. The MSE gap in Eq. (13) is therefore exactly the expected squared difference between the vanilla critic's prediction and this reconstructed value. This comparison is made under a common evaluation distribution, while the prediction quality of the learned critics is evaluated empirically in § 4.4.

A.9 Prompts

A.9.1 Critic Input Templates

The critic receives the query and the response prefix formatted with the policy's chat template. $\mathcal{D}$VE prepends a single rubric to this input, so that the value it predicts is the one conditioned on that criterion; the ablated variant prepends the full rubric set instead. The three templates below differ only in this prefix.

Critic input, vanilla PPO

```
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
{response}<|im_end|>
```

Critic input, $\mathcal{D}$VE

```
[VALUE ESTIMATION FOCUS] When estimating the value of the response
below, judge only the following criterion:
{rubric_k}
---
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
{response}<|im_end|>
```

Critic input, ablated variant without per-rubric aggregation

```
[VALUE ESTIMATION FOCUS] When estimating the value of the response
below, consider ALL of the following criteria together:
- {rubric_1}
- {rubric_2}
- {rubric_3}
...
---
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
{response}<|im_end|>
```

A.9.2 Judge Prompts

Rubric judge (Chat and Science training). A single prompt grades every rubric of a response in one call and is shared by both domains.

Rubric judge prompt, used during Chat and Science training

You are an expert evaluator. 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"
}
```
````

Rubric judge (Instruction Following training). Instruction-following evaluation uses no judge prompt. All three benchmarks are scored by deterministic programs, and the model under evaluation receives the raw benchmark prompt through its chat template with no additional wrapping. An LLM judge is used only during training, as a fallback for the subset of constraints that admit no deterministic checker, with the prompt below.

LLM fallback judge prompt, used during instruction-following training

Based on the provided Input (if any) and Generated Text, judge whether the generated text fulfills the Criteria Item with either a YES or NO choice. Your selection should be based on your judgment as well as the following rules:

- **YES**: Select "YES" if the generated text entirely fulfills the condition specified in the Criteria Item. However, note that even minor inaccuracies exclude the text from receiving a "YES" rating. As an illustration, consider a Criteria Item "Each sentence in the generated text uses a second person". If even one sentence does not use the second person, the answer should NOT be "YES". To qualify for a "YES" rating, the generated text must be entirely accurate and satisfy the criteria.

- **NO**: Opt for "NO" if the generated text fails to meet the criteria or provides no information that could be utilized to judge. For instance, the Criteria Item asks "Is the second sentence in the generated text a compound sentence?" and the generated text only has one sentence. It offers no relevant information to judge whether this criteria is met. Consequently, the answer should be "NO".

Input: {prompt}
Generated Text: {response}
Criteria Item: {rubric}

You only need to judge whether the generated text satisfies the given Criteria Item and do NOT affect by other requirements in Input (if any). Return either a "YES" or "NO" choice without any additional text in your response.

Rubric judge (Chat evaluation). Arena-Hard-v2 is scored by pairwise comparison against the fixed baseline responses, with the candidate and baseline presented in both orders.

Pairwise judge prompt, used for Arena-Hard-v2 evaluation

You are judging which of two AI assistant answers is better for the user question. Consider correctness, helpfulness, completeness.

```
[Question]
{question}
[Answer A]
```

{answer_a}
[Answer B]
{answer_b}

Which answer is better? Reply EXACTLY one token: A (if A better), B (if B better), or T (if roughly equal).

Rubric judge (Science evaluation). ResearchQA is graded one rubric criterion at a time on a five-point coverage scale, which we normalize to $[0, 1]$ and average over the criteria of a question.

Rubric coverage judge prompt, used for ResearchQA evaluation

You are an expert evaluator judging how well a model's response covers a specific rubric criterion for a research question.

[Research Question]
{query}
[Model Response]
{response}
[Rubric Criterion]
{criterion}

On a 5-point scale, rate how well the response covers the criterion above:

0 = Not at all (the criterion is entirely absent or wrong)
1 = Barely (only a fleeting or largely incorrect mention)
2 = Moderately (partially covered, notable gaps)
3 = Mostly (largely covered, minor gaps)
4 = Completely (fully and correctly covered)

Answer with exactly one digit from 0 to 4.

A.9.3 Token-Level Relevance Annotation Prompt

Following Xu et al. (2026b), we obtain token-level relevance labels by prompting an LLM annotator rather than by training a discriminator: for each response–rubric pair, the annotator first classifies the rubric as global or local, then, for local rubrics, marks the exact spans of the response that satisfy or violate it. The prompt below is used for this annotation.

Prompt template for token-level relevance annotation

```
You are a training data annotator for a token-level discriminator
model.  Your task is to classify the Criteria and label which
tokens in the Response are relevant to it.

Criteria:
<<criteria>>

Response:
<<response>>

Please follow these steps:

Step 1:  Scope Classification
Determine whether this criteria is GLOBAL or LOCAL:
- GLOBAL: The criteria applies to ALL tokens of the response as a
whole (e.g., language/writing style requirement, tone requirement,
word count limit, paragraph count limit, overall format/structure
requirement, topic requirement, count, frequency, or numeric
threshold)
- LOCAL: The criteria only concerns SPECIFIC parts of the response
(e.g., a specific phrase must appear, a specific section must
exist, certain keywords must/must not be used, a particular element
must be included/excluded)
```

```
Step 2: Polarity Classification
If criteria is LOCAL, then determine whether this criteria is
POSITIVE or NEGATIVE:
- POSITIVE: Requires something to be present or done (e.g., "must
include X", "should start with Y", "the first paragraph should be
X")
- NEGATIVE: Requires something to be absent or avoided (e.g., "do
not use X", "must not include Y", "exclude the word Z", "avoid
mentioning W")

Step 3: Output
- If GLOBAL, then output: {"type": "all relevant", "scope":
"global", "polarity": null, "relevant_texts": []}
- If LOCAL + POSITIVE:
  - If relevant text segments EXIST in the response (i.e., the
criteria IS satisfied by some portion of the text), then output:
{"type": "partial relevant", "scope": "local", "polarity":
"positive", "relevant_texts": ["exact segment 1", "exact segment
2"]}
  - If NO relevant text segments exist (i.e., the criteria is NOT
satisfied at all), then output: {"type": "all irrelevant",
"scope": "local", "polarity": "positive", "relevant_texts": []}
- If LOCAL + NEGATIVE:
  - If text segments that VIOLATE the criteria EXIST in the
response, then output: {"type": "partial relevant", "scope":
"local", "polarity": "negative", "relevant_texts": ["exact
violating segment 1"]}
  - If NO violating text exists (i.e., the criteria IS satisfied /
not violated), then output: {"type": "all irrelevant", "scope":
"local", "polarity": "negative", "relevant_texts": []}

Important Notes:
1. The text segments in relevant_texts must be exact original text
from the Response (exact match, including punctuation and spaces).
2. Each text segment should be a continuous, complete sentence or
phrase.
3. Output only JSON format, without any markdown code block
markers or other text.
4. Ensure the JSON format is correct and can be directly parsed.
5. scope and polarity are required so the taxonomy decision can be
checked; they must be consistent with type.
```