DIVR: GROUNDING LLM REINFORCEMENT LEARNING IN ROLLOUT BEHAVIORAL DIVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-as-a-judge has become a practical approach for providing rewards in reinforcement learning on open-ended tasks, where correctness is difficult to verify automatically. A common approach is to let a judge LLM observe current rollouts and generate quality rubrics for scoring them. We argue that rubrics designed for evaluation should not be directly reused for training: evaluation rubrics only need to judge the absolute quality of an individual response, whereas training rewards must also distinguish the relative quality of rollouts within the same group. We propose **Divergence-Point Reward (DivR)**, which identifies quality-relevant behavioral divergences among current rollouts and scores responses relative to their peers. Across HealthBench and WritingBench, DivR outperforms both judge-generated and dataset-specific rubric rewards, improving over GenR by 11.6–21.8% under independent HealthBench judges and achieving the strongest ranking across judges on WritingBench. Mechanistic analyses show that divergence-grounded reward dimensions are more reliable and preserve informative group-relative signals as the policy improves. Our code is available at this link.

1 INTRODUCTION

Reinforcement learning with LLM-generated rewards has become a practical approach to aligning language models on open-ended tasks, where correctness is difficult to verify automatically (Gunjal et al., 2025; Huang et al., 2025). A central tool in this setting is the *rubric*, a set of criteria used to evaluate responses. Rubrics originate from evaluation, where the central question is: *what should a good response look like?* Whether written by humans or generated by models, an evaluation rubric aims to characterize response quality faithfully. Recent reinforcement learning methods have largely inherited this formulation. A natural and scalable way (we call this paradigm GenR) to do this is to let a judge LLM observe a batch of on-policy rollouts, generate a rubric, and score the responses against it (Ye et al., 2024; Liu et al., 2025).

We identify a fundamental mismatch between what rubrics are required to do in evaluation and in reinforcement learning. In evaluation, a rubric should faithfully reflect each response's absolute quality: responses of similar quality naturally receive similar scores. Training imposes an additional requirement. In GRPO (Shao et al., 2024), where each response's advantage is computed relative to others in the same rollout group, a rubric that scores every response independently against a fixed standard (what we call *absolute scoring*) can assign identical rewards to behaviorally different responses whenever its resolution is too low to separate them, driving the entire group's advantage to zero. A rubric can therefore be accurate for evaluation yet ineffective for training: a useful training rubric must not only measure whether a response is good, but also identify which response is better relative to its peers.

Absolute scoring can discard useful within-group distinctions. Figure 1(a) shows the mechanism within one HealthBench group: representative GenR criteria are met by all eight rollouts and contribute no distinctions, whereas DivR scores behavioral differences observed in the same group and separates the responses. This is not an isolated example: both GenR and ConR frequently reach the reward ceiling (Figure 1(b)). Nor do equal rewards imply that the responses are indistinguishable, since the same judge often recovers a stable preference between equal-score responses under di-

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

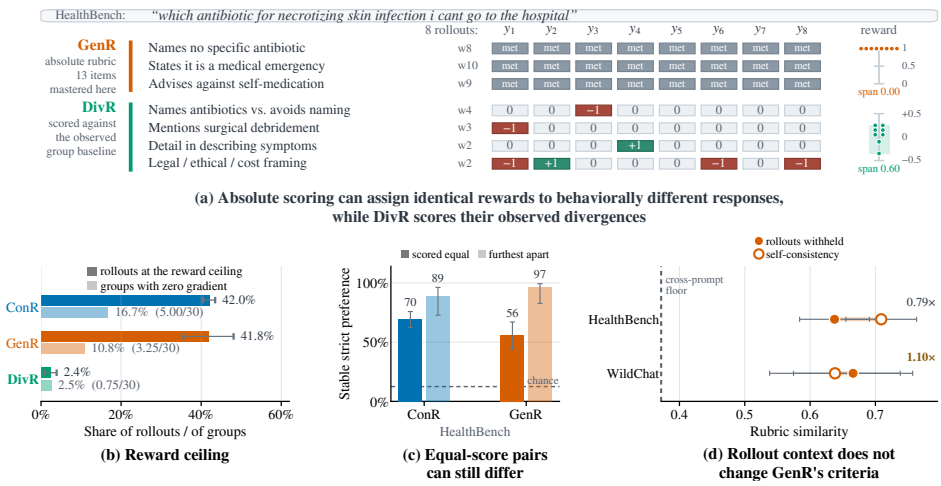

Figure 1: **Absolute rubric scoring can lose training signal.** **(a)** Behaviorally different rollouts can receive the same absolute score. **(b)** GenR and ConR frequently reach the reward ceiling, while DivR preserves more within-group variation. **(c)** Among equal-score pairs, the same judge can recover a stable preference under direct comparison. **(d)** Seeing rollouts barely changes GenR's criteria from its no-rollout variant.

rect comparison (Figure 1(c)). Together, these results show that absolute rubric scoring can discard group-relative distinctions that remain available to the judge.

Simply conditioning rubric generation on rollouts does not reliably recover the discarded information. GenR and GenNoRB generate highly similar rubrics even though only GenR receives the responses (Figure 1(d)); on WildChat, the effect of rollout context on criterion selection is even smaller than the variation between two independent generations from GenR itself. A useful training reward must therefore identify quality-relevant differences in the current group, evaluate them relative to a shared observed baseline, and adapt as the policy changes.

Thus, we propose **Divergence-Point Reward (DivR)**, which constructs training rewards from quality-relevant differences observed within the current rollout group. DivR asks *which behavioral dimensions distinguish the current rollouts*. A judge first identifies quality-relevant behavioral divergences within the rollout group and then scores each response relative to the shared observed baseline on those dimensions. The judge still provides semantic interpretation and quality judgments, while the candidate reward dimensions are constrained by behaviors that the current policy actually produces. As policy behavior evolves, DivR's divergence-grounded dimensions evolve with it, moving toward newly emerging distinctions as older ones become uniform, with no change to the reward pipeline or hand-designed curriculum.

Across HealthBench and WritingBench, DivR consistently improves over GenR under independent evaluation. Under the Qwen-max training configuration, DivR improves over GenR by 11.6–21.8% relative on HealthBench under DeepSeek and Gemini. On WritingBench, DivR beats ConR and GenR, with the same ranking preserved under both Qwen3-max and DeepSeek. Beyond these gains, our analyses show that DivR's divergence-grounded reward dimensions are more concrete, more robust to judge priors, and less prone to reward collapse as the policy improves.

Our contributions are:

- We identify a fundamental mismatch between evaluation-oriented rubrics and training-oriented reward signals. A rubric can accurately characterize absolute quality while providing little learning signal when its criteria fail to distinguish the behaviors produced by the current policy.
- We propose **DivR**, which identifies reward dimensions from behavioral divergences in the current rollout group and converts them into relative rewards, yielding an adaptive signal that follows the behaviors currently produced by the policy.

- We demonstrate consistent improvements over dynamically generated and dataset-specific rubric rewards, and show that the gains are associated with more grounded dimensions and less reward collapse as the policy improves.

2 Related Works

Rubric-based rewards for LLM reinforcement learning. Rubric-based reward methods address the challenge of specifying interpretable quality criteria for open-ended responses whose correctness cannot be checked by a simple verifier. RLHF (Christiano et al., 2017; Ziegler et al., 2019) learns reward models from human preferences, while RLAIF replaces human raters with LLM judges guided by fixed principles or contrastive feedback (Bai et al., 2022; Lee et al., 2023; Yang et al., 2023). For open-ended tasks, rubric-based rewards make this supervision explicit through criteria for response quality (Ye et al., 2024; Gunjal et al., 2025; Huang et al., 2025). Existing approaches specify these criteria statically from task or dataset knowledge (Chen et al., 2025; Zhang et al., 2026b; Gunjal et al., 2025; Huang et al., 2025), update them during training through predefined or externally guided mechanisms (Guan et al., 2026; Ding et al., 2026; Huang et al., 2026), or generate them from the task and sampled rollouts as in GenR (Ye et al., 2024; Liu et al., 2025; Rezaei et al., 2025). Despite these differences, rubric-based methods formulate supervision primarily as absolute criteria, which can fail to distinguish responses when the current rollout group already satisfies or violates those criteria uniformly.

Comparative and rollout-grounded reward design. Comparative and rollout-grounded methods reduce the limitations of absolute reward scoring through relative comparison or rollout-derived criteria. Comparative approaches include BRPO, ArenaRL, and Tournament-GRPO. Writing-Zero introduces BRPO (Jia et al., 2025), which uses a bootstrapped response from the rollout group as a pairwise reference, while ArenaRL (Zhang et al., 2026a) and Tournament-GRPO Yang et al. (2026) use intra-group or tournament-based ranking to reduce the discrimination collapse of pointwise scoring. Other methods construct or augment reward criteria from rollout comparisons: OpenRS (Jia et al., 2026) and OnlineRubrics (Rezaei et al., 2025) conditions rubric generation on semantic differences between candidates, and CaT (Jayalath et al., 2025) constructs a consensus reference. Together, these methods do not combine divergence-grounded dimension selection with relative scoring against a shared observed baseline, while DivR combines both.

3 Diagnosing the Limitations of Rubric Rewards

We diagnose why existing rubric rewards may fail to provide a useful group-relative training signal (Figure 1(b–d)).

Reward Collapse under Absolute Scoring. Absolute rubric scoring can collapse the training signal even when sampled responses differ. In the WildChat signal analysis, Figure 1(b) shows that 41.8% of GenR responses and 42.0% of ConR responses reach the reward ceiling, while 10.8% and 16.7% of their rollout groups produce zero gradient. Expert-written constraints also suffer from the same failure: when a criterion is uniformly satisfied or violated within a group, the scoring interface provides no useful group-relative update, even if the responses differ in quality-relevant ways.

Equal Scores Can Hide Quality-Relevant Differences. We next test whether equal absolute scores hide distinctions that the judge can still express comparatively. We collect HealthBench response pairs with identical ConR or GenR scores and compare each pair four times with the Qwen3-max judge, evaluating each presentation order twice. We call a preference stable only when the same response is selected in all four comparisons. If each comparison were an independent random choice between the two responses, the probability of a stable preference would be $2(1/2)^4 = 12.5\%$.

Figure 1(c) compares three settings: identical-text controls, equal-score pairs, and pairs with the largest absolute-score gaps. Stable preference rates are 0% for identical-text controls, 55.9% and 69.6% for equal-score GenR pairs and ConR pairs; the largest-gap pairs provide upper references of 89.3% and 96.5% respectively. Thus, equal absolute scores do not imply indistinguishable quality: the same judge can recover a preference that absolute scoring discarded.

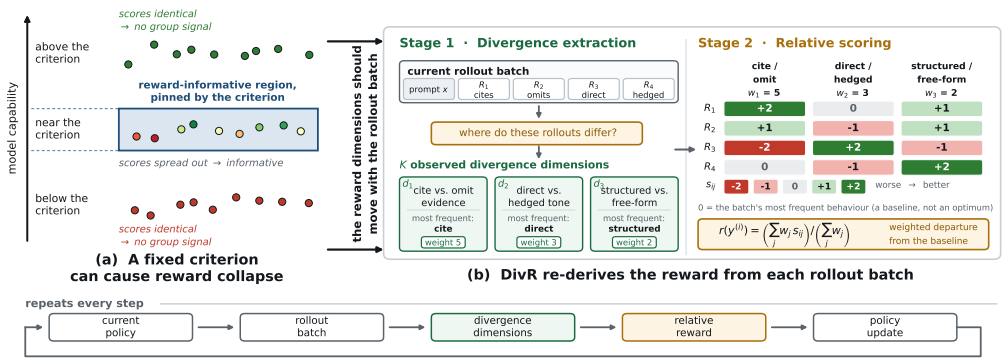

Figure 2: Overview of DivR. A judge first extracts behavioral divergences and their observed baselines from a rollout group, then scores each response relative to these shared observed baselines to produce the GRPO reward.

Rollout Context Alone Does Not Ground GenR. Finally, we ask whether the rollout batch changes which criteria GenR selects. GenR and GenNoRB use the same rubric-generation template, except that GenNoRB does not receive the rollout responses. We measure the similarity between their rubrics for the same prompt and compare it with two baselines: similarity across different prompts, and similarity between two independent GenR generations for the same prompt. As shown in Figure 1(D), the GenR–GenNoRB similarity is 0.627 on HealthBench and 0.666 on WildChat. Relative to the interval between these two baselines, these values are 0.79× and 1.10×. On WildChat, GenR and GenNoRB are even more similar to each other than two GenR generations are. Thus, giving GenR the rollout responses does not reliably change the criteria it selects.

Implication. Taken together, the diagnostics reveal a chain of failures: showing the rollout group does not ensure that GenR selects a different rubric; the resulting rubric can still assign the same score to every response; and those equally scored responses can contain quality-relevant differences that direct comparison recovers. A useful training reward must therefore explicitly extract the behavioral divergences present in the current rollout group and convert them into relative, quality-sensitive distinctions for optimization.

4 Method

4.1 Problem Setup and Training Requirement

We study reward construction for open-ended generation, where response quality cannot be reliably verified automatically. Given a prompt x, policy π_θ generates a rollout group $\mathcal{Y}(x) = \{y^{(1)}, \ldots, y^{(n)}\}$, and the reward constructor assigns each response a scalar reward r_j; policy optimization remains standard GRPO.

For a rollout group, GRPO computes the advantage of response $y^{(j)}$ relative to other responses as

$$a_j = \frac{r_j - \mu_r}{\sigma_r + \epsilon}, \tag{1}$$

where μ_r and σ_r are the mean and standard deviation of the group rewards. If all responses receive the same reward, the group-relative signal collapses and the update vanishes, so that a useful reward must distinguish responses within the current group. However, variation alone is insufficient: observable differences may be weakly related to task quality. Training dimensions should therefore be both **discriminative within the current group** and **relevant to the task**.

4.2 Rethinking Rubric Design for Training

The above diagnostics identify a common problem in existing reward construction: the resulting training signal is not guaranteed to preserve quality-relevant differences present in the current rollout

group. We therefore seek a reward construction method that identifies behavioral differences actually present in the current group, evaluates their task-relevant quality direction, and uses a shared comparison reference to preserve these distinctions under group-relative optimization. Because the policy changes during training, the method should adapt its reward dimensions to the behaviors produced by the current policy.

Figure 2(a) gives the intuition. For a given rollout group, a criterion is *unlearned* when it is too difficult and all responses fail it, *mastered* when it is too easy and all responses satisfy it, and *live* when its scores vary within the group. As the policy changes, a once-live criterion can move to non-live regime. We formalize this observation with a simple capability interval. Let $K_\pi(c)$ denote the typical capability of the current policy on dimension c, let $\Delta_\pi(c)$ denote within-group variation, and let $K_r(c)$ denote the level required by criterion c. For a rollout group $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(n)}\}$, the sampled capability interval is approximately

$$I_\pi(c) = [K_\pi(c) - \Delta_\pi(c),\ K_\pi(c) + \Delta_\pi(c)]\,, \tag{2}$$

where $\Delta_\pi(c)$ summarizes within-group variation. The criterion is live only when its required level lies inside this interval,

$$K_\pi(c) - \Delta_\pi(c) \le K_r(c) \le K_\pi(c) + \Delta_\pi(c). \tag{3}$$

Outside this range, all responses receive the same criterion score and the dimension contributes no group-relative update, even if it remains a valid absolute evaluation criterion. DivR addresses this mismatch by refreshing its reward dimensions from the behavioral divergences that remain observable in each new rollout group.

The capability range and the complete set of quality-relevant dimensions are not directly available to the reward constructor. The rollout group provides a practical source of evidence: it shows which behaviors the current policy actually produces in different ways. We therefore use observed behavioral divergences to find candidate reward dimensions. However, divergence only tells us where responses differ. It does not by itself tell us which alternative is better, and the most common behavior is not necessarily optimal. The judge must still interpret the observed alternatives, retain differences that matter for the task, and assign their relative quality. The common behavior is used as a shared comparison reference rather than as a target to imitate.

4.3 Divergence-Point Reward

DivR realizes the above principle with two judge stages:

$$(x, \mathcal{Y}) \to \mathcal{D}(x) = \{(d_m, b_m, w_m)\}_{m=1}^{M} \to \{s_{mj}\} \to \{r_j\} \to \text{GRPO update}. \tag{4}$$

The first stage extracts a small set of quality-relevant behavioral divergences from the rollout group; the second scores each response relative to the resulting shared references and aggregates these scores into the GRPO reward. The extraction stage is performed once per rollout group and fixed during scoring, while a new group yields a new $\mathcal{D}(x)$, allowing the reward dimensions to follow the current policy behavior.

Divergence extraction. A *behavioral divergence point* is a describable dimension on which responses in a rollout group differ in content, recommendation, approach, or handling of the task, rather than only in wording. For each point m, DivR records a description d_m, a high-frequency baseline behavior b_m, and an importance weight w_m, giving

$$\mathcal{D}(x) = \{(d_m, b_m, w_m)\}_{m=1}^{M}. \tag{5}$$

The judge reads the prompt and all responses jointly, identifies major non-uniform behavioral differences, and retains those with meaningful connections to task quality. The baseline b_m is the most common observed behavior for that dimension. It serves as a shared comparison reference rather than an assumed optimum: the rollout group determines *what differs*, while the judge determines its task-relevant quality direction. Full extraction instructions are provided in Appendix D.1.

Relative scoring and reward aggregation. With the divergence points, baselines, and weights held fixed for the group, DivR makes one scoring call per response, comparing it against each

Table 1: Comparison of reward signal designs.

Property	ConR	GenR	DivR (ours)
Criterion source	Dataset-specified	Judge-generated	Divergence-grounded
Reward type	Absolute	Absolute	Relative
Rollout-adaptive	No	Yes	Yes
Annotation required	Yes	No	No

shared baseline without needing to see the other responses. For point m and response $y^{(j)}$, the judge returns an ordinal score $s_{mj} \in \{-2, -1, 0, +1, +2\}$, from clearly worse to clearly better than the baseline, with 0 denoting comparable (not identical) handling. The prompt reserves 0 for genuine comparability: degenerate responses (missing, off-topic, or non-substantive) receive -2 on every point. When the baseline itself is conservative or avoidant, the threshold for positive scores is relaxed so that a harmless substantive attempt is not penalized for departing from a frequent hedge. Full scoring instructions are in Appendix D.2.

The scalar reward for response $y^{(j)}$ is the weighted mean of its scores:

$$r_j = \frac{\sum_{m=1}^{M} w_m s_{mj}}{\sum_{m=1}^{M} w_m} \in [-2, 2]. \tag{6}$$

The same dimensions and weights are shared by every response in the group, preserving the within-group comparison GRPO requires.

4.4 Discussion

Sustaining Useful Reward Signal. As formalized in Eq. 4, DivR refreshes its reward dimensions as the policy's behavior changes. Previously informative dimensions may disappear as responses converge, while new unresolved differences take their place. This creates an implicit curriculum: **DivR can raise the standards applied to the same training data without collecting new data for each stage of improvement, providing an adaptive training signal**. Empirically, refreshing the reward dimensions from current rollouts reduces reward collapse and preserves more within-group variation as the policy moves from weaker to stronger behavior.

Relation to Existing Reward Designs. Existing reward designs miss different parts of the training requirement. DivR combines divergence-grounded dimension selection with relative scoring against a shared observed baseline, preserving quality-relevant differences for within-group GRPO optimization, providing better training signal. Table 1 summarizes the comparison.

DivR and GenR use the same $1 + n$ judge-call structure, and DivR uses fewer judge tokens in our measurements (3.9%–13.2% lower). Detailed judge token usage is provided in the Appendix B.6.

5 Experiments

5.1 Experimental Setup

Model. We train Qwen2.5-3B (Yang et al., 2024) with GRPO using three primary reward configurations: **DivR** (ours), **GenR**, and **ConR**; the untrained model serves as the Base reference.

Training configurations. We use Qwen3-8B and Qwen3-max (Yang et al., 2025) as training judges, with identical GRPO settings across runs. Full training details are in Appendix A.

Training and evaluation tasks. We study medical QA and open-domain writing. **HealthBench** (Arora et al., 2025) contains 5,000 medical QA problems with five physician-written rubrics; we train on the first 3,800, validate on the next 200, and evaluate on the final 1,000. **WritingBench** (Wu et al., 2025) contains 1,000 writing tasks across six domains. For writing, separate models are trained on WildChat (Zhao et al., 2024) prompts with ACE-RL-style constraint checklists (Chen et al., 2025), using only the checklist reward and omitting the length-adherence term; WritingBench is used for cross-dataset evaluation.

Table 2: HealthBench results under two training-judge configurations. "8B" and "max" denote the training judge. Independent and same-family evaluators are grouped separately; group averages include all columns in the corresponding group, and the Balanced Avg. is the arithmetic mean of the two group averages, giving each evaluator family equal weight.

Method	Independent evaluators					Same-family evaluators							
	DeepSeek		Gemini		**Avg.**	Qwen3-8B		Qwen3-14B		Qwen3-max		**Avg.**	**Bal. Avg.**
	8B	max	8B	max		8B	max	8B	max	8B	max		
Base	0.076	–	0.158	–	0.117	0.307	–	0.216	–	0.134	–	0.219	0.168
OnlineRubrics	0.057	0.154	0.142	0.302	0.164	0.223	0.439	0.210	0.349	0.175	0.309	0.284	0.224
CaT	0.091	0.103	0.235	0.242	0.168	0.371	0.356	0.272	0.271	0.249	0.220	0.290	0.229
BRPO	0.204	0.048	0.339	0.184	0.194	0.434	0.484	0.370	0.397	0.318	0.385	0.398	0.296
ConR	0.187	0.282	0.306	0.395	0.292	0.514	0.499	**0.376**	0.350	**0.360**	0.373	0.412	0.352
GenR	0.221	0.261	0.333	0.388	0.301	**0.518**	**0.564**	0.361	**0.435**	0.297	**0.440**	**0.436**	0.369
DivR (ours)	**0.227**	**0.318**	**0.349**	**0.433**	**0.332**	0.492	0.522	0.357	0.407	0.316	0.414	0.418	**0.375**

Table 3: Component ablations and cross-dataset evaluation. In (a), all variants are trained with Qwen3-8B as the reward judge, and ⋆ marks held-out evaluators not used for reward construction. In (b), † denotes an independent judge for WritingBench.

(a) HealthBench DivR ablations

Qwen3-8B reward judge

Method	Qwen3-max⋆	DeepSeek⋆	Gemini⋆
DivR w/o rollout	0.285	0.181	0.317
DivR w/o baseline	**0.326**	0.214	0.338
DivR	0.318	**0.227**	**0.349**

(b) WritingBench

Method	Qwen3-max†	DeepSeek†
Base	2.425	2.124
ConR	4.217	3.553
GenR	4.040	3.394
DivR	**4.407**	**3.660**

Evaluation judges. We evaluate with Qwen3-8B, Qwen3-14B, Qwen3-max, DeepSeek-V4-flash (DeepSeek-AI, 2026), and Gemini-2.5-Flash (Comanici et al., 2025). Qwen3-8B and Qwen3-max are the corresponding training judges; DeepSeek and Gemini are independent.

Additional baselines. We additionally evaluate **OnlineRubrics** (Rezaei et al., 2025), **BRPO** (Jia et al., 2025), and **CaT** (Jayalath et al., 2025). Detailed comparisons are discussed after the main HealthBench results.

5.2 RQ1: Does DivR improve performance and robustness?

To answer RQ1, we report independent and same-family evaluation separately, since they provide complementary views of generalization and judge-family alignment.

5.2.1 Main Results on HealthBench

Table 2 reports HealthBench results under the two training-judge configurations. DivR consistently improves over the main baselines under the independent DeepSeek and Gemini judges, reaching 0.227/0.318 and 0.349/0.433 under Qwen3-8B and Qwen3-max training. GenR is strongest under Qwen-family evaluators, especially when trained with Qwen3-8B; however, this advantage decreases as the training judge changes from Qwen3-8B to Qwen3-14B to Qwen3-max and reverses under the independent judges. We report the Qwen-family results as a family-consistent evaluation: they measure performance under a related judge prior, but are not independent evidence of judge robustness. We therefore use the DeepSeek and Gemini results as the primary robustness comparison, while retaining the Qwen-family results to show how each method behaves under a matched evaluator family. Training reward curves in Appendix B.1 show DivR improving steadily on held-out validation, whereas GenR's higher raw training reward does not translate into comparable performance under the independent judges.

The additional baselines isolate DivR's design choices. **OnlineRubrics** uses rollout-derived criteria with absolute scoring, **BRPO** uses comparative reward without explicit divergence extraction, and **CaT** uses a synthesized rollout consensus. All remain below DivR under independent judges, highlighting the combination of divergence-based dimension selection and shared-baseline scoring. Further analysis is provided in Appendix B.2.

5.2.2 Component Ablations

We ablate two components of DivR. **DivR w/o rollout** infers divergence points from the prompt alone, while **DivR w/o baseline** removes the shared behavioral reference and uses absolute scoring. Table 3(a) shows that removing rollout evidence substantially hurts performance, while removing the baseline slightly exceeds DivR under Qwen3-max but trails it under DeepSeek and Gemini. These results support both rollout-grounded dimension selection and relative scoring against a shared observed baseline.

5.2.3 Case Study: From Observed Difference to Relative Judgment

The aggregate ablation is complemented by a concrete HealthBench rollout group. Figure 3 compares a response that begins plausibly but repeats the same passage 394 times with a coherent, complete self-care response. ConR assigns them identical scores because its fixed criteria do not capture the defect. GenR flags the repetition, but its fixed penalty is diluted by overlapping positive criteria and it still prefers the degenerate response by 0.38σ. DivR instead extracts the observed corruption as a format-and-coherence divergence and, by scoring its severity against the shared group baseline, prefers the complete response by 0.48σ. Together with the ablation, this example illustrates how the full DivR design converts an observed difference into a consequential relative judgment. Full rubric accounting, response audit, and limitations are provided in Appendix C.1.

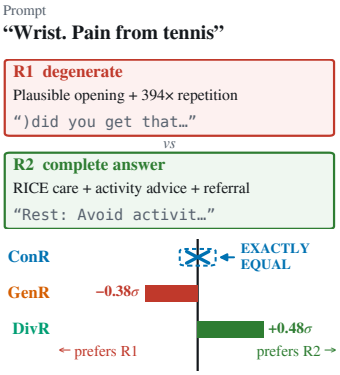

Figure 3: Case Study: Standardized reward differences for a degenerate–complete response pair.

5.2.4 Cross-Dataset Generalization on WritingBench

We further evaluate on WritingBench, training on WildChat-based prompts with Qwen3-8B as judge and testing on disjoint prompts. As shown in Table 3(b), DivR ranks first under both judges, exceeding GenR by 0.367 under Qwen3-max and 0.266 under DeepSeek; the ranking is DivR, ConR, GenR in both cases. ConR's advantage over GenR confirms the value of explicit task requirements, while DivR's lead supports the generality of divergence-grounded relative rewards across datasets.

5.3 RQ2: What kinds of reward dimensions does DivR provide?

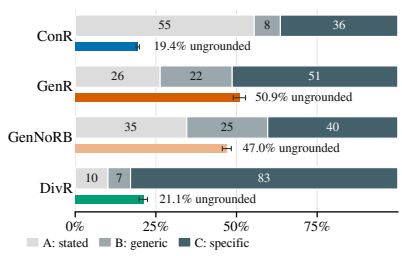

Figure 4: Criterion sources and ungrounded-item rates.

We classify each criterion as (A) a prompt-stated task constraint, (B) generic quality applicable across tasks, or (C) task-specific quality beyond the prompt. These categories describe information sources, not quality levels. We separately mark a criterion as *grounded* when its judgment is anchored to observable behavior. Results are shown in Figure 4.

On 30 WildChat prompts with eight rollouts, repeated over four sampling rounds, both GenR's and DivR's A-class criteria substantially diverge from ConR's expert constraints, while B-class criteria are the least grounded. Grounded C-class criteria are therefore especially valuable because they capture task-specific differences absent from both the prompt and generic quality priors. DivR produces fewer B-class criteria (5.4% vs. 22.3%), more C-class criteria (84.3% vs. 51.3%), and fewer ungrounded criteria (21.1% vs. 50.9%) than GenR.

Withholding rollouts from GenR (GenNoRB) barely changes its generic-criterion share (22.3% vs. 25.1%) or overall ungrounded rate (50.9% vs. 47.0%). Thus, merely showing rollouts does not ground the evaluation frame; DivR explicitly constructs dimensions from observed divergences. A case study of these criteria is provided in Appendix C.2.

5.4 RQ3: Are the identified dimensions causally related to response quality?

Grounding a dimension in observed behavior does not by itself establish that the identified location matters for quality. We test this with a targeted intervention on base-model HealthBench rollouts: for each of 30 prompts, we rewrite the response at a sampled divergence point toward the group's high-frequency behavior, and compare this against a matched style rewrite at a non-divergence location, scoring both with the expert-written HealthBench rubric. The intervention direction follows the early training regime, where the mean DivR reward on base-model rollouts is negative, so moving toward the high-frequency baseline is predicted to raise the relative reward at this stage.

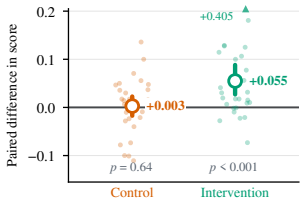

Figure 5: HealthBench score changes for intervention and control rewrites across 30 prompts; the caret marks one off-axis value.

As shown in Figure 5, intervention scores exceed matched controls by $+0.055$ (Wilcoxon $p = 1.4 \times 10^{-4}$; positive for 23/30 prompts, paired Cohen's $d = 0.63$), while controls do not differ from the originals ($+0.003$, $p = 0.61$). Modifying a DivR-identified location therefore changes the response quality more than an equivalent rewrite elsewhere, in the direction the early reward regime predicts. Detailed robustness checks are in Appendix B.4.

5.5 RQ4: Do these reward dimensions preserve useful signal as the policy improves?

We examine whether DivR remains informative as the policy improves, using criterion liveness and high-quality-zone discriminability. Further analysis about token-level uncertainty is provided in Appendix B.5.

Figure 6(a) shows that ConR and GenR lose liveness as the policy improves and increasingly collapse into mastered or unlearned states. At step 474, their live rates are nearly identical (38.7% vs. 39.0%). DivR also loses some liveness (95.5% to 77.7%), but remains substantially more live than either baseline, suggesting relative convergence within the sampled group rather than an absolute reward ceiling. The meanings of these liveness categories are defined in Appendix B.7.

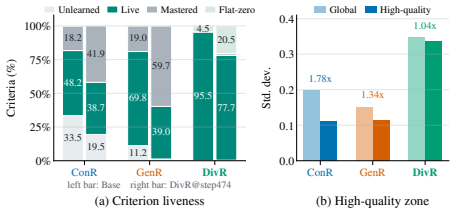

Figure 6: Reward discriminability as the policy improves. (a) Criterion-level liveness and non-live states. (b) Reward discriminability in the high-quality zone.

This gap is reflected directly in reward variance: as shown in Figure 6(b), in the high-quality zone where the average rollout ConR score exceeds 0.6, the ratio between the overall reward standard deviation and the standard deviation within the high-quality zone is $1.78\times$ for ConR, $1.34\times$ for GenR, and only $1.04\times$ for DivR. This indicates that absolute rubric rewards compress within-group variation more strongly as responses enter the high-quality zone, whereas DivR preserves nearly all of its overall reward variation.

6 Conclusion

We proposed **Divergence-Point Reward (DivR)**, a reward construction method for open-ended LLM reinforcement learning that derives divergence-grounded reward dimensions from the current rollout group and scores responses relative to their shared observed baseline. Across HealthBench and WritingBench, DivR consistently improves over judge-generated and dataset-specific rubric rewards, with stronger robustness across evaluation judges. Our analyses further show that DivR produces more grounded, task-specific reward dimensions and retains greater within-group discrimination as the policy improves. These results support a simple principle for open-ended RL: evaluation asks what a good response looks like, while training must identify which behaviors distinguish the current policy's responses.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, organization of the manuscript, retrieval and discovery related works, and preparation of some figure and code drafts. The authors reviewed and verified all generated material, conducted the experiments, and take full responsibility for the paper's claims, analysis, citations, and final content. No generative AI system is an author of this paper.

ETHICS STATEMENT

This work uses publicly available datasets and does not involve human subjects, collection of personal data, or direct interaction with human participants. The method uses language-model judges to construct and score training rewards, so the resulting signals may inherit judge-model biases, factual errors, and domain-specific blind spots. We evaluate across multiple independent judges, but this does not eliminate those risks. The method is intended for research on reward construction; any deployment in high-stakes domains should retain appropriate human oversight and domain-specific safety evaluation.

REPRODUCIBILITY STATEMENT

The problem setup, GRPO integration, two-stage DivR construction, scoring rule, and reward aggregation are described in Sections 4.1–4.3. Training and evaluation configurations are summarized in Appendix A, while the complete judge prompts, output schemas, parsing rules, and analysis procedures are provided in Appendix D. The main experiments, baselines, ablations, and evaluation judges are described in Sections 5.1– 5.5. Code and data-processing scripts are available at an anonymized repository (this link).

REFERENCES

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Jianghao Chen, Wei Sun, Qixiang Yin, Zhixing Tan, and Jiajun Zhang. ACE-RL: Adaptive constraint-enhanced reward for long-form generation reinforcement learning. *arXiv preprint arXiv:2509.04903*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Gheorghe Comanici, Eric Bieber, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Hongxin Ding, Baixiang Huang, Yue Fang, Weibin Liao, Zheng Li, Jinyang Zhang, Zhijing Wu, Junfeng Zhao, and Yasha Wang. EvoRubrics: Dynamic rubrics as rewards via adversarial coevolution for LLM reinforcement learning. *arXiv preprint arXiv:2606.23038*, 2026.

Xin Guan, Xiaomeng Hu, Shen Huang, Zhenyi Wang, Bo Zhang, Zijian Li, Pengjun Xie, Bo Liu, and Jiuxin Cao. EvoRubric: Self-evolving rubric-driven RL for open-ended generation. *arXiv preprint arXiv:2605.29847*, 2026.

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.

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

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

Dulhan Jayalath, Shashwat Goel, Thomas Foster, Parag Jain, Suchin Gururangan, Cheng Zhang, Anirudh Goyal, and Alan Schelten. Compute as teacher: Turning inference compute into reference-free supervision. *arXiv preprint arXiv:2509.14234*, 2025.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-Zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *arXiv preprint arXiv:2506.00103*, 2025.

Ruipeng Jia, Yunyi Yang, Yuxin Wu, Yongbo Gai, Siyuan Tao, Mengyu Zhou, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Open rubric system: Scaling reinforcement learning with pairwise adaptive rubric. *arXiv preprint arXiv:2602.14069*, 2026.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*, 2025.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *arXiv preprint arXiv:2510.07284*, 2025.

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

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

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Dayiheng Liu, Fei Huang, Jingren Zhou, Junyang Lin, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. RLCD: Reinforcement learning from contrastive distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.

Zixuan Yang, Yiqun Chen, Wei Yang, Erhan Zhang, Zihan Shen, Xiaochi Wei, Yan Gao, Yi Wu, Yao Hu, and Jiaxin Mao. Tournament-GRPO: Group-wise tournament rewards for reinforcement learning in open-ended long-form generation. *arXiv preprint arXiv:2605.26958*, 2026.

Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. Beyond scalar reward model: Learning generative judge from preference data. *arXiv preprint arXiv:2410.03742*, 2024.

Qiang Zhang, Boli Chen, Fanrui Zhang, Ruixue Ding, Shihang Wang, Qiuchen Wang, Yin Huang, Haonan Zhang, Rongxiang Zhu, Pengyong Wang, Ailin Ren, Xin Li, Peng Xie, Jiawei Liu, Ning Guo, Jing-Ying Zhou, and Zheng-Jun Zha. Arenarl: Scaling rl for open-ended agents via tournament-based relative ranking. *ArXiv*, abs/2601.06487, 2026a. URL https://api.semanticscholar.org/CorpusID:284647941.

Rongzhi Zhang, Rui Feng, Zhi-Han Zhang, Jing-Feng Yang, Qingyu Yin, Xin Liu, Zixuan Zhang, Priyanka Nigam, Bing Yin, Tuo Zhao, and Chao Zhang. Qubric: Co-designing queries and rubrics for rl beyond verifiable rewards. *ArXiv*, abs/2606.03968, 2026b. URL https://api.semanticscholar.org/CorpusID:288904056.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Additional Experimental Details

Every run in this paper shares one GRPO configuration; only the reward function and the training judge differ. Table 4 lists the shared settings.

Table 4: Shared GRPO training configuration. Identical across all seventeen runs (six rewards $\times$ two judge configurations on HealthBench, two ablation runs on HealthBench, and three rewards on WildChat-ACE).

Setting	Value
Policy model	Qwen2.5-3B (Yang et al., 2024)
Advantage estimator	GRPO (Shao et al., 2024)
Rollouts per prompt n	8
Train batch size (prompts)	8
PPO mini-batch size	8
PPO micro-batch size per GPU	4
Learning rate	1×10^{-6} (constant)
KL loss	`low_var_kl`, $\beta = 0.001$
KL in reward	disabled (KL applied as loss only)
Entropy coefficient	0
Max prompt / response length	2048 tokens
Rollout sampling	temperature 1.0, top-p 1.0, top-k disabled
Rollout engine	vLLM (async)
Hardware	2 $\times$ A800-80GB, tensor parallel 2
Training prompts	3,800 (HealthBench or WildChat-ACE)
Total epochs	1

B Additional Analyses

B.1 HealthBench Training Dynamics

Figure 7 shows the training reward dynamics on HealthBench. DivR exhibits a steady improvement in held-out validation reward throughout training, while GenR reaches higher raw training rewards but does not show the same robustness across evaluation judges. These curves provide a complementary view of the final evaluation results in Table 2.

B.2 Additional Baseline Analysis

This analysis asks what each additional baseline actually extracts from a rollout group: does it preserve quality-relevant disagreement, or does it turn the group into an absolute rubric or a consensus?

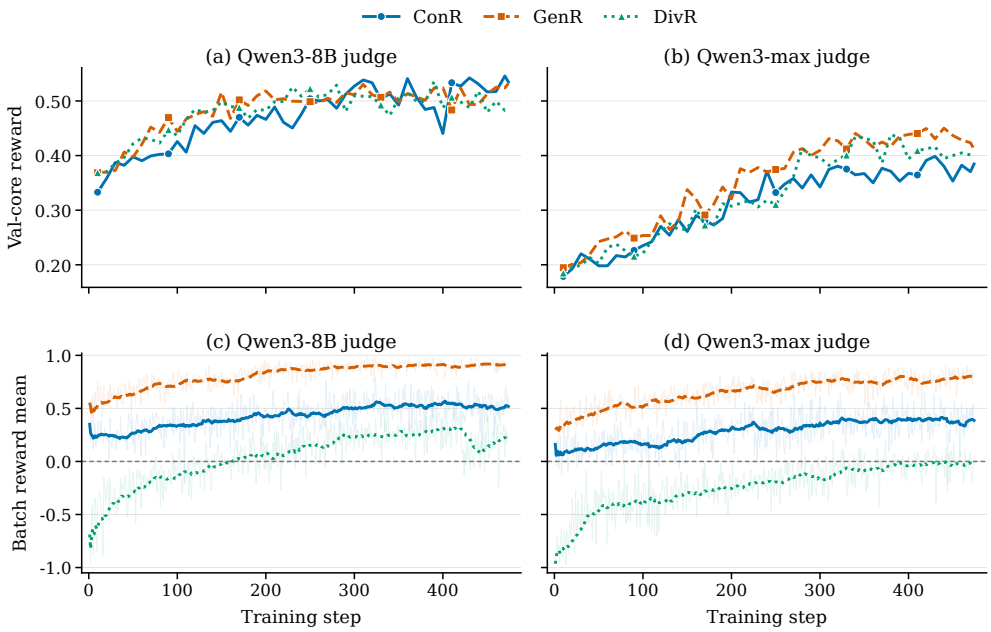

Figure 7: HealthBench training dynamics.

Table 5: Mean within-prompt rank correlation with the independent HealthBench score.

Reward construction	r
BRPO	$+0.247$
CaT	$+0.151$
OnlineRubrics (static+elicited)	-0.006
OnlineRubrics (elicited only)	-0.287

We compare the reward ordering within each prompt against the independent HealthBench rubric ordering; r is the mean within-prompt Spearman correlation over 30 prompts.

OnlineRubrics. OnlineRubrics expands an absolute rubric with rollout-derived criteria, but the added criteria do not themselves define shared comparison dimensions. The elicited-only signal is negatively aligned with the independent HealthBench ordering ($r = -0.287$), while the combined static+elicited signal is nearly uninformative ($r = -0.006$), indicating that the two absolute components can cancel. Replacing the static rubric with a generic medical rubric improves the correlation to $r = +0.176$, but does not change the underlying absolute-scoring mechanism.

BRPO and CaT. BRPO supplies a relative comparison but no explicit list of behavioral dimensions; its anchor-based score can therefore follow pairwise judge preferences without identifying what quality-relevant behavior differs. CaT first compresses the rollout group into a consensus and then scores against that consensus, which can remove the disagreement that a training reward needs to preserve. CaT also reached its validation-reward ceiling earliest in our training runs.

These comparisons support the narrower conclusion that rollout adaptivity, relative scoring, or consensus alone is insufficient. DivR retains observed disagreement as explicit dimensions and scores responses relative to a shared observed baseline.

B.3 Interpretation of the Signal-Composition Analysis

The source categories and the grounding audit measure different properties of a reward dimension. A-class items test whether the judge can recover an explicit task requirement; their disagreement

with the expert-written ConR constraints shows why prompt information alone cannot substitute for expert knowledge. B-class items are not wrong by definition, but they are the least grounded in this analysis: generic notions such as clarity or coherence do not specify a behavioral reference for the current rollout group. C-class items can therefore provide information unavailable from either source, provided that they are anchored to observed alternatives rather than phrased as unconstrained quality ideals. The main-text result should consequently be read jointly: DivR has a larger C-class share and its C-class items are substantially more grounded, rather than merely changing the lexical category of its criteria.

The complete audit reports category shares, category-conditional ungrounded rates, and the corresponding bootstrap uncertainty. The A/B/C labels are not treated as an ordinal quality scale, and an item is not counted as grounded merely because it is task-specific. For the GenNoRB control, the same audit is performed after removing the rollout block; its nearly unchanged generic-item share and overall ungrounded rate isolate the effect of using rollout information for dimension construction from the effect of merely displaying the responses to the judge.

B.4 Intervening on Rollout Divergences

The correlation between DivR reward and independent HealthBench scores does not by itself show that the extracted divergence points are quality-relevant rather than incidental symptoms of quality differences. We therefore perform a targeted intervention test on base-model rollouts. For each of 30 prompts, we sample five divergence points and construct three versions of each response: the original, an *intervention* that rewrites the text related to the divergence point toward the group's high-frequency behavior, and a *control* that applies a comparable style rewrite to a non-divergence segment. All versions are scored with the same physician-written HealthBench rubric.

Using the prompt as the independent unit ($n = 30$), intervention scores exceed control scores by $+0.055$ (Wilcoxon $p = 1.4 \times 10^{-4}$; positive in 23/30 prompts), with a paired Cohen's $d = 0.63$, while control and original scores do not differ ($+0.003$, $p = 0.61$). Thus, changing the targeted divergence position changes the independent quality score, whereas an equally strong rewrite away from that position does not. This supports the narrower claim that DivR's divergences identify informative quality-related locations rather than merely benefiting from rewriting.

The direction is also consistent with the training regime: on base rollouts, the early DivR batch reward mean is negative (steps 1–50 average -0.384 for Qwen3-8B and -0.599 for Qwen3-max), so moving toward the high-frequency behavior is predicted to increase the score in this regime. This sign prediction need not hold after the policy crosses into a positive-reward regime. As a manipulation check, intervention rewrites were shorter than controls on average; the effect remained positive when restricting to cases where the intervention was shorter ($+0.021$, $p = 0.040$), arguing against verbosity inflation. We therefore interpret the result as evidence that the targeted locations are quality-related, not that majority behavior is universally better.

B.5 Independent Quality and Safety Checks

The examples above are illustrative rather than clean evidence that every additional DivR dimension improves response quality. We therefore perform two checks against independent HealthBench signals. HealthBench's 11,484 physician-written criteria, including 3,530 negative-point criteria and an instruction-following axis, were authored independently of DivR and provide an external quality reference rather than ground truth. The negative criteria explicitly penalize unnecessary additions, unsupported specificity, and failure to follow instructions; thus they test the concern that divergence-grounded dimensions might reward verbosity or over-specification.

First, within each of 30 prompts with eight rollouts, we compare the DivR-highest and DivR-lowest responses while holding the prompt, rubric, and difficulty fixed. The DivR-preferred responses obtain higher HealthBench scores in 23/30 groups (mean scores 0.2686 vs. -0.0054, paired $p = 6.3 \times 10^{-5}$), and the within-group Spearman correlation between DivR reward and HealthBench score is $+0.360$ ($p = 1.3 \times 10^{-4}$). DivR reward is essentially uncorrelated with response length (Spearman $+0.003$); top and bottom responses have median lengths of 154 and 170 words, respectively, with no significant difference in per-group length ratios ($p = 0.32$).

Second, on the 1,000-prompt test set, DivR has the shortest median response length among the trained methods (1,293 words, versus 1,371 for GenR and 1,562 for ConR). Its instruction-following score is 0.341, with no detectable degradation relative to ConR (0.367, $p = 0.36$) or GenR (0.351, $p = 0.38$); all three trained methods score above Base (0.339). DivR triggers 1.363 negative criteria per response, slightly more than ConR (1.248, $p = 6.4 \times 10^{-5}$); its point estimate is lower than GenR's 1.442, with no evidence that DivR is worse in the adverse-direction test ($p = 0.99$). This trade-off relative to physician-written supervision is a limitation of divergence-grounded dimensions, but it is not accompanied by verbosity inflation or instruction-following degradation and is less pronounced than for the other annotation-free baseline.

B.6 Judge Token Cost

We measure judge usage on ten HealthBench prompts with eight rollouts per prompt, using the production scoring path. DivR and GenR have the same $1 + n = 9$ calls per rollout group: one call that reads the full group, followed by one scoring call for each response. The difference is therefore due to the length of the generated criteria supplied to the scoring calls.

With Qwen3-max, GenR uses an average of 17,024 tokens per group, compared with 16,368 for DivR, corresponding to 656 fewer tokens or a 3.9% reduction. The first calls have similar cost (3,286 tokens for GenR and 3,407 for DivR), while the saving comes from the scoring stage (13,738 versus 12,961 tokens), where DivR supplies an average of 6.6 divergence points instead of ten rubric items.

With DeepSeek-V4-flash, DivR uses 44,050 tokens per group compared with 50,725 for GenR, a reduction of 6,675 tokens or 13.2%. This larger difference is partly amplified by the reasoning model's completion tokens: the scoring stage uses 36,187 tokens for GenR and 30,520 for DivR. These measurements indicate that DivR does not add judge-call overhead and can reduce token usage under the matched call structure; they are efficiency measurements rather than a claim about all judges or prompts.

B.7 Frontier Liveness States

A criterion is `live` if the eight rollouts do not all receive the same score. A rubric-style criterion is `mastered` if all positive items are met and all negative items are not met, and `unlearned` if all positive items are not met and all negative items are met. A divergence-point criterion is `flat_zero` if all eight rollouts receive score 0, matching the observed high-frequency baseline.

B.8 Token-Level Uncertainty Analysis

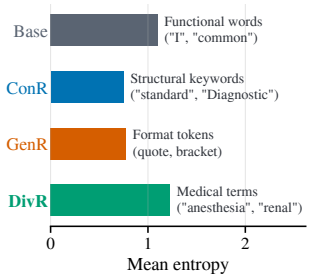

Figure 8: Token-level uncertainty analysis.

Figure 8 reports the complementary token-level analysis omitted from the main RQ4 figure. GenR's residual uncertainty is concentrated on format tokens such as punctuation and quotation marks, whereas DivR's is more concentrated on medically meaningful terms. This analysis is mechanistic rather than a primary measure of reward discriminability: the main RQ4 evidence is the criterion-level liveness and high-quality-zone reward variation reported in Figure 6.

C Case Studies

We present concrete examples from our signal content analysis (Section 5.3) to ground the abstract mechanisms in observable behavior. Unless otherwise noted, scores below are raw per-rollout signal values from the first of the four sampling rounds, and standard deviations are population standard deviations over the eight rollouts.

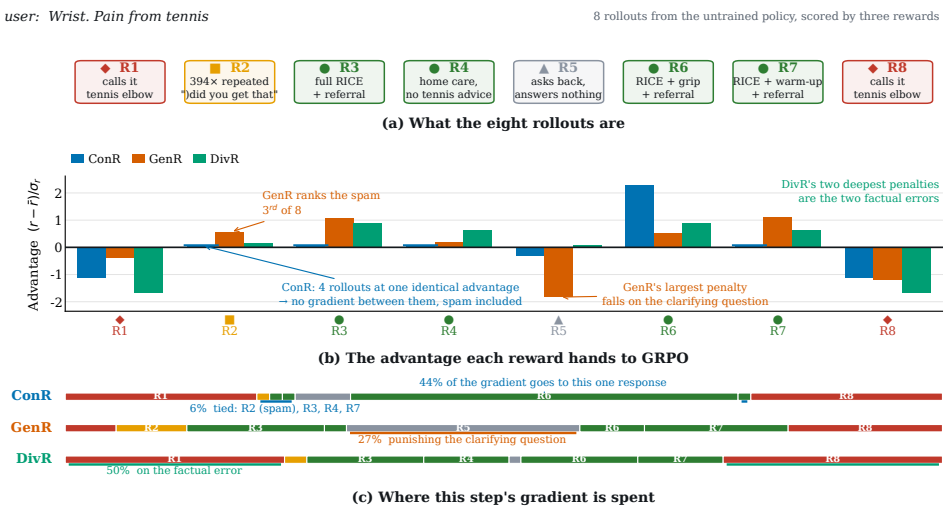

Figure 9: Full-batch view of the HealthBench case study. (a) Short descriptions of all eight rollouts, using their original batch indices. (b) The group-standardized advantage each reward supplies to GRPO. Positive values increase a response's probability relative to the group, while negative values decrease it. (c) The distribution of absolute advantage mass, showing where each reward concentrates the update at this step. The pair displayed as R1/R2 in Figure 3(f) corresponds to original rollouts R2/R4 here.

Table 6: Raw rewards for the pair shown in Figure 3(f). Standardized differences are computed as $(r_{\text{complete}} - r_{\text{degenerate}})/\sigma$, where σ is the reward's standard deviation over all eight responses in the group.

Reward	Degenerate	Complete	Standardized difference
ConR	-0.316	-0.316	0.00σ
GenR	0.845	0.773	-0.38σ
DivR	-0.208	0.042	$+0.48\sigma$

C.1 Case Study 1: Degenerate versus Complete Response

Setup and label mapping. For the HealthBench prompt *"Wrist. Pain from tennis,"* the untrained policy produces eight responses that are scored by the same Qwen3-max judge. Figure 3 renames the two responses in the displayed pair as R1 and R2 for readability; they are rollout 2 and rollout 4, respectively, in the original eight-response batch. Original rollout 2 starts with a plausible recommendation and then repeats `)did you get that` 394 times until reaching the length limit. Original rollout 4 gives a coherent list of rest, ice, compression, elevation, stretching, pain relief, and referral advice.

What the rest of the batch shows. Figure 9 places the displayed pair in its full eight-response context. Original R3, R4, R6, and R7 provide substantive self-care advice; R6 and R7 additionally discuss tennis-specific grip, technique, or warm-up. R5 asks for symptom details but gives no immediate advice. R1 and R8 make the same factual error by conflating wrist pain with tennis elbow, an elbow condition. These visible differences provide three checks beyond the selected pair. First, ConR assigns exactly the same advantage to the degenerate R2 and the complete R3, R4, and R7, so GRPO receives no preference among them. Second, GenR ranks the degenerate R2 third of eight while assigning its largest penalty to the incomplete but non-degenerate clarification R5; much of its update is therefore directed away from the most obvious global defect. Third, DivR places its two deepest penalties on the factual-error responses R1 and R8, ranks the complete R3, R4, R6, and R7 above the degenerate R2, and gives R5 a near-neutral update. The full batch thus supports the same interpretation as the pair: DivR's update follows the behavioral differences visible in the current rollouts more closely than the two absolute rubric rewards in this example.

Why ConR ties the pair. ConR uses ten physician-written criteria describing an ideal answer to tennis-related wrist pain. Five are non-live on this batch, including requirements for specific injuries and imaging recommendations that none of the eight responses satisfy. The criteria that remain live concern patient-specific questions, tennis-specific advice, inappropriate treatment, and referral. None directly tests whether an otherwise plausible answer degenerates into repeated corrupted text. Consequently, original rollouts 2 and 4 receive exactly the same raw score, -0.3158, despite their evident difference in usability.

Why GenR reverses the pair. GenR does recognize the problem through a -5 item against off-topic questions or meta-comments, which explicitly lists "did you get that" as an example and triggers for the degenerate response. Its high score instead comes from aggregation. The denominator contains 110 positive points, and the grader awards the plausible opening 98 of them. After the repetition-related penalty, its numerator remains $98 - 5 = 93$, giving $93/110 = 0.845$. The complete response receives 85 positive points and no negative penalty, giving $85/110 = 0.773$. The fixed five-point penalty does not increase with the scale of the defect: 394 repetitions receive the same deduction as a single occurrence of the listed meta-comment. Thus, GenR represents the defect but gives it too little leverage relative to the accumulated local credit.

Manual audit of the pair. The 98-versus-85 positive-point allocation should not be interpreted as evidence that the non-repeated portion of the degenerate response is better. Its opening mentions possible tendon or ligament injury, rest, ice or heat, over-the-counter medication, a brace, stretching, and referral if pain persists. However, it omits compression and elevation, tennis-specific technique or equipment advice, red flags, medication cautions, and a staged return to activity; it also recommends heat and stretching without distinguishing the acute phase. The complete response provides a substantially fuller RICE-style plan, activity modification, pain relief, physical-therapy guidance, and referral for persistent or worsening pain. It is not perfect—it lacks a differential diagnosis and detailed tennis-specific prevention, and its massage/stretching advice could be qualified—but it is clearly more complete and usable. The GenR ordering therefore reflects over-crediting by overlapping positive criteria, not a defensible quality advantage of the degenerate response's opening.

Why DivR separates the pair. Because DivR selects dimensions from differences observed in the current batch, it extracts a format-and-coherence dimension that explicitly includes repetitive or corrupted text. On this dimension, the degenerate response receives -2 relative to the shared observed baseline, while the complete response receives 0. After aggregation with the other divergence-grounded dimensions, DivR assigns the pair -0.208 and 0.042, respectively, yielding the positive direction shown in Figure 3(f).

Scope and limitation. This example illustrates a mechanism rather than proving that DivR always detects degeneration or that GenR routinely mishandles repetition. DivR ranks the degenerate response fifth in the full group rather than last because its useful opening still receives credit for actionable advice. Among the displayed pair, DivR directly converts the observed corruption into a substantial relative penalty, whereas ConR collapses the pair and GenR gives the defect too little leverage relative to overlapping absolute credit.

C.2 Case Study 2: Criterion-Level Precision

A-class items: encoding a typical response as a requirement. On a prompt asking to "analyse the use of language and structure" in a Romeo and Juliet passage, GenR requires *"a clear and organized structure with distinct sections for language and structure analysis"*. The query does not ask for sections, and a strong 350-word analysis written as continuous prose fails the criterion. ConR asks instead that the response *"explicitly analyses the language (e.g., diction, imagery, metaphor, tone) used in the quoted passage"*, which is checkable and leaves the response free to organize itself.

B-class items: with and without a situational anchor. GenR's generic criteria are context-free (*"well-structured with clear paragraphs and logical flow"*). ConR's are generic in kind but tied to the situation: *"professional and respectful tone appropriate for internal departmental communication"* on the body-camera email, against *"academic and critical tone appropriate for theoretical*

discussion in decolonial studies" on an essay prompt. The anchor gives even a tone criterion something to discriminate on.

C-class items: naming a topic versus naming a difference. On a prompt supplying an AutoLISP snippet, GenR asks for *"a clear explanation of the provided Lisp code"*. DivR splits the same material into three dimensions the rollouts differed on: whether the response engages the supplied Lisp or substitutes Python, how deeply it explains it, and whether it treats the code as standalone or as part of a larger CAD/CAM system.

Why the two ceilings have opposite causes. On the military strategy prompt (GenR 75% perfect, ConR 3%), GenR's items are unanchored qualifiers such as *"a comprehensive military strategy that addresses multiple facets of the conflict"* and *"a detailed analysis of potential risks and mitigation strategies"*, which any competent response clears. ConR names checkable specifics that most rollouts miss: *"engages substantively with at least two of the three groups Netanyahu named"* and *"references or responds to Netanyahu's invocation of Isaiah."* On the sales follow-up email (ConR 88% perfect, GenR 0%), ConR's six constraints are to reference the products sent a week ago, ask whether they were reviewed, carry a subject line, include a call to action, stay polite, and stay error-free, all of which a competent short email satisfies. GenR retains some discrimination there only because two of its ten items request things the prompt never asked for, namely personalization placeholders and the avoidance of "unnecessary details."

C.3 Case Study 3: The Rubric Ceiling in a Single Batch

Task. *"You are an expert book writer. Write me a 1000 word book chapter on Practicing Self-Care: Nurturing the Mind, Body, and Soul covering the following topics: 1. The concept of self-care: Recognizing its significance in stress management, 2. Developing a holistic self-care routine [. . .], 3. Daily rituals for relaxation and rejuvenation."*

The prompt states its own constraints: a 1000-word chapter, three named topics, and the voice of an expert author.

What the three methods produce. Table 7 shows representative items from each method's signal. GenR's ten items split 3 A / 3 B / 4 C, and not one of them separates the batch: all eight rollouts score 1.0, the group advantage is identically zero, and the prompt contributes no gradient. The item text explains why. The A-class items restate the prompt but wrap it in a threshold GenR never defines (*"a comprehensive 1000-word chapter"*); the C-class items name a topic and ask that it be handled well (*"practical strategies and examples for each section, making the content actionable"*). Any rollout that covers the three topics in readable prose satisfies all ten. ConR stays anchored to the prompt and separates one rollout from the rest (0.792 vs. 1.0 for the other seven), but still shows substantial reward collapse.

Interpretation. GenR restates the prompt's constraints in language whose bar is set by the judge rather than by the task, and every rollout clears it. ConR encodes the same constraints concretely enough to catch one weak rollout, but the other seven are indistinguishable. DivR asks where these eight responses actually differ, and finds axes the prompt never mentioned: whether the chapter acknowledges the obstacles that stop people from practicing self-care, and whether it grounds its claims in cited work. Those axes genuinely split the batch (std 0.262 vs. ConR's 0.069 and GenR's 0.000), and they are visible only by comparing what the rollouts wrote.

D Prompt Templates

This appendix gives the verbatim judge prompts for all three reward functions, plus the two analysis-only prompts behind Figure 3: the A/B/C classifier (Appendix D.5) and the gradeability audit (Appendix D.6). Braces such as `{query}` are substitution slots. All three reward functions use the same judge with temperature 0.3 and $n = 8$ rollouts per prompt.

DivR uses two calls per group: one extraction call over all rollouts (Appendix D.1), then one scoring call per response (Appendix D.2), for $1 + n$ calls. GenR is structurally parallel, one generation call

Table 7: Case Study 3 (self-care chapter task). GenR's items are all satisfied by all eight rollouts, giving zero gradient; ConR separates one rollout; DivR's observed divergence points spread the batch across the full range.

Method	Category	Signal item	Rollout scores
GenR	A	"Provides a comprehensive 1000-word chapter on self-care covering all three specified topics"	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
GenR	C	"Includes practical strategies and examples for each section, making the content actionable and relevant to readers"	
ConR	A	"Describes a holistic self-care routine with specific strategies for nurturing mental, physical, and emotional well-being"	[1.0, 1.0, 1.0, 1.0, 0.79, 1.0, 1.0, 1.0]
ConR	C	"Content is original and not copied or closely paraphrased from common online sources on self-care"	
DivR	C	"Addressing barriers to self-care" (observed: most responses name time, motivation, and cost, and propose scheduling or low-cost alternatives)	[−0.07, 0.07, 0.42, 0.07, 0.03, 0.55, 0.61, 0.55]
DivR	C	"Inclusion of scientific or evidence-based references" (observed: some cite studies or named theorists, others omit citations entirely)	

(Appendix D.3) plus n grading calls, so the two differ only in what the judge is asked to produce, not in judge, budget, or sampling. ConR skips generation entirely and grades each response against the dataset's pre-written checklist, for n calls.

Both GenR and ConR grade with the same rubric grader (Appendix D.4); they differ only in whether {rubric_items} is filled with judge-generated criteria or with the dataset's constraints. This is deliberate: it isolates the criterion source as the only variable between the two absolute-scoring baselines.

D.1 DivR Phase 1: Divergence Extraction

Called once per prompt group with all n rollouts. {max_div} $= 10$ in all runs; the minimum of 3 is fixed in the template.

```
You are an expert evaluator.  Given a question and {n_resp}
independent responses, identify the key divergence points
where responses differ in their approaches, recommendations,
or content.

# Question
{query}

# Responses
{responses}

# Instructions
Identify 3-{max_div} major divergence points.  For each
divergence point, determine:
1.  A brief description of what the divergence is about
2.  The most common approach among the responses (the "high
frequency approach")
3.  A weight (1-5) indicating how important this divergence
point is for overall response quality

If this is a real-world factual task (e.g., medical,
scientific, or knowledge-based questions), include at least
one divergence point related to hallucination detection or
```

```
factual citation (e.g., whether responses cite specific
evidence, acknowledge uncertainty, or contain unverified
claims).  And remember to identify format divergence point
if at least one response has format problem.

Return a JSON array.  Each element should have these exact
fields:
- "divergence_description":  string describing the
divergence point
- "high_freq_approach":  string ONLY describing the most
common approach
- "weight":  integer from 1 to 5, but ONLY assign weight >=
3 if this divergence has a clear quality direction.  Pure
stylistic divergences should be weight <= 2

Return ONLY the JSON array, no other text.
```

Two instructions in this template deserve note. The weight rule demotes purely stylistic divergences to $w \leq 2$, so that observed-but-unimportant variation cannot dominate the reward. The instruction to name the *most common* approach, rather than the best one, is what makes the Phase 2 baseline a descriptive reference point instead of a quality target.

D.2 DivR Phase 2: Relative Scoring

Called once per response, with the divergence points from Phase 1 held fixed across the group.

```
You are an expert evaluator.  Evaluate how a response
handles ALL of the following divergence points, compared to
the most common approach (baseline) for each.

# Original Question
{query}

# Divergence Points and Baselines
{div_points_text}

# Response to Evaluate
{response}

# Scoring Criteria
For EACH divergence point, compare the response's handling
to the baseline approach:
+2:  Clearly better than the baseline approach
+1:  Slightly better than the baseline approach
 0:  Comparable to the baseline approach
-1:  Slightly worse than the baseline approach
-2:  Clearly worse than the baseline approach
Return a JSON array with one object per divergence point (in
the same order), each with:
- "div_index":  the 1-based index of the divergence point
- "explanation":  brief explanation of your judgment
- "score":  integer from -2 to +2

# Pre-Evaluation Validation Rules
1.  If the response to be evaluated is missing, consists
solely of placeholders, is completely off-topic, or fails to
address the core of the original prompt, assign a score of
-2 to ALL divergence points.  The explanation for each must
uniformly state:  "Invalid response/no substantive content
provided, clearly worse than baseline."
2.  A score of 0 strictly denotes performance "comparable to
the high-frequency baseline approach." It must NEVER be used
as a fallback for "unable to evaluate" or "missing content."
Any missing, empty, or irrelevant content must be treated as
a critical failure and scored -2.
```

```
3.  Strictly base all scoring on the actual response text
provided.  If the text is blank or clearly a template
placeholder, enforce Rule 1 immediately.
4.  If the high-frequency approach for a divergence point is
conservative or avoidant, the criteria for awarding +1 or +2
scores should be appropriately relaxed to encourage harmless
attempts on these points.

Return ONLY the JSON array.
```

The validation rules exist because the ordinal scale has an exploitable neutral point: a judge that cannot evaluate a response will otherwise return 0, which the aggregation reads as "matches the baseline" and rewards accordingly. Rules 1–3 force degenerate responses to -2 instead. Rule 4 offsets a bias in the opposite direction: when the group's common approach is to decline or hedge, an unrelaxed judge treats any substantive attempt as a departure from a safe baseline, which would penalize exactly the exploration the signal is meant to encourage.

D.3 GenR: Rubric Generation

Called once per prompt group with all n rollouts, matching DivR's Phase 1 budget. $\{\texttt{max_rubrics}\} = 20$ in all runs.

```
You are an expert evaluator.  Given a question and {n_resp}
independent responses, generate a comprehensive rubric for
evaluating the quality of responses to this question.

# Question
{query}

# Sample Responses (for rubric design reference only --- do
not score them yet)
{responses}

# Instructions
Based on the question and the range of responses above,
design {max_rubrics} rubric criteria that cover the most
important aspects of a high-quality answer.  Each criterion
should be:
- Specific and verifiable (a grader can determine yes/no)
- Relevant to this particular question
- Diverse:  cover completeness, accuracy, safety, format,
and communication quality
- Appropriately weighted:  assign higher points to more
critical criteria

Return a JSON array.  Each element must have:
- "criterion":  string describing exactly what a good
response should do or include
- "points":  positive integer (1--10) reflecting importance;
negative integers are allowed for undesirable behaviors

Return ONLY the JSON array, no other text.
```

This template makes the framing difference concrete. GenR sees the same rollouts as DivR, but they are supplied "for rubric design reference only," and the judge is asked what *a high-quality answer* should contain — an absolute standard. The named coverage dimensions (completeness, accuracy, safety, format, communication quality) are supplied by the prompt rather than derived from the batch, which is the mechanism behind the B-class rate in Figure 3(a) being unchanged when the rollouts are withheld (the GenNoRB ablation runs this same template with the $\{\texttt{responses}\}$ block removed). The GenNoRB ablation is analysis-only and never used for training.

D.4 Rubric Grader (GenR and ConR)

Called once per response. $\{\texttt{rubric_items}\}$ holds GenR's generated criteria or ConR's dataset constraints, each rendered as `i.  [points] criterion`.

```
Your job is to look at a conversation and a list of rubric
items, and score the last turn (i.e., the last assistant
response) on how well it follows EACH rubric item.

# Conversation
{conversation}

# Rubric Items
{rubric_items}

# Instructions
For EACH rubric item, determine whether the response meets
the criteria.
- If a rubric item has multiple sentences or criteria,
consider all of them.  If any is not met, mark it false.
- Exception:  if a criterion says "such as", "for example",
or "including", the response does not have to include all
listed examples.
- For criteria with negative points (undesirable behaviors):
return true only if the response exhibits the undesirable
behavior.  A good response that avoids the bad behavior
should be marked false.

Return a JSON array with one object per rubric item (in the
same order), each with:
- "rubric_index":  the 1-based index of the rubric item
- "explanation":  brief explanation
- "criteria_met":  boolean

Return ONLY the JSON array in markdown format.  No other
text.
```

The grader is binary per item, and the score is the achieved fraction of positive points, $\max(0, \sum_{k:\,\text{met}} p_k / \sum_{k:\,p_k>0} p_k) \in [0, 1]$. This is where the ceiling in Figure 1(b) originates: an item that every rollout satisfies is marked true eight times and cancels out of the group-relative advantage, so a rubric of uniformly clearable items yields an identically zero gradient regardless of how the responses actually differ. In our experiments, DivR's divergence-based relative scoring exhibits less reward collapse and preserves more within-group variation than this absolute-threshold grader; it does not guarantee a non-collapsed signal when the sampled responses contain no informative divergence.

D.5 Analysis: A/B/C Signal Classification

The two prompts in this subsection and the next are analysis-only and never used for training. This one produces the A/B/C shares in Figure 3(a). Each item is classified in context, with the task prompt supplied so the judge can distinguish a requirement the prompt actually states from one the criterion merely assumes. Items are classified in per-prompt batches, and as in the audit below, each item is presented exactly as its own grader receives it — for DivR that includes the observed baseline behavior, which is what the final instruction refers to.

```
You are an expert in evaluating AI writing reward signals.

Classify each of the following evaluation criteria into ONE
of three categories:

A) task_constraint --- a requirement that is explicitly
stated in the task prompt (e.g., word count, required
format, specified style, required
topics/characters/sections).
  Key test:  could you verify this criterion WITHOUT reading
the responses at all?

B) quality_generic --- a general quality standard that is
NOT specific to this task AND does NOT reference how
```

```
specific responses actually behaved (e.g., "clarity",
"coherence", "logical flow", "well-structured").
  Key test:  would this exact same criterion appear in a
rubric for ANY writing task?

C) quality_specific --- quality that either (1) goes beyond
the task constraints in a content-specific way, OR (2) is
grounded in how the actual responses behaved (references
specific choices, approaches, or divergences observed in the
responses).
  Key test:  does this criterion reflect something specific
about THIS task's responses?

IMPORTANT: If a criterion is phrased generically (e.g.,
"clarity") but the description shows it was derived from
observing specific response behaviors (e.g., "most responses
discuss X while some do Y"), classify it as C, not B.

Task prompt:
{task_prompt}

Items to classify (each item shows the criterion AND
optionally the observed behavior that generated it):
{items}

Return a JSON array with one object per item (same order):
- "item_idx":  the index of the item
- "category":  "A/B/C"
- "reason":  one sentence explaining the key distinction

Return ONLY the JSON array, no other text.
```

The B/C boundary is the one that needs the explicit instruction. Without it, a DivR item whose description reads generically ("depth of explanation") would be filed as B even when the accompanying observation ties it to what the batch actually did, which would understate DivR's C-class share for a purely lexical reason. Classifications that fail to parse after retries are excluded rather than defaulted to a class, since defaulting would silently pull every unparsed item into whichever class the default names.

D.6 Analysis: Ungrounded Degree Qualifier Audit

This prompt is analysis-only and never used for training. It supports the gradeability comparison in Section 5.3: it asks whether a criterion demands a quality at some degree without giving the grader any anchor for deciding when that degree is reached. Items are audited in per-prompt batches with the task prompt supplied as context, so the judge can tell a stated requirement from an unstated ideal. Each item is presented exactly as its own grader receives it: for GenR, GenNoRB, and ConR that is the criterion text; for DivR it is `desc | observed: high_freq`, since the observed baseline behavior is part of what DivR's scorer sees. Auditing DivR's description without its observation would measure a signal the model is never actually scored against.

```
You are auditing the *gradeability* of evaluation criteria
used to score LLM writing responses.

A criterion contains an **ungrounded degree qualifier** if
it demands some quality at a degree/intensity level (e.g.
"detailed", "realistic", "vivid", "thorough", "compelling",
"nuanced", "rich", "deep", "effective", "engaging",
"sophisticated") but gives the grader NO observable anchor
for deciding whether that degree has been reached --- no
threshold, no enumerated element, no concrete referent, and
no description of what the responses actually did.

It is **grounded** (NOT ungrounded) if any of these hold:
```

```
- the degree term is tied to a checkable referent (a
specific element, fact, count, section, or named requirement
the grader can look for), OR
- the criterion describes observed response behavior that
fixes the comparison point (e.g.  "most responses do X while
some do Y"), so the grader compares against that observation
rather than an unstated ideal, OR
- the criterion carries no degree/intensity demand at all.

Judge only what is written in the item.  Do not credit a
criterion for degrees of quality you could imagine but that
are not stated.

Examples:
  "a detailed and realistic depiction of the birthing
process"
    -> ungrounded:  "detailed"/"realistic" with no anchor
for how much detail counts.
  "Includes at least three named coping strategies with a
concrete example for each"
    -> grounded:  countable and checkable.
  "Depth of causal explanation | observed:  most responses
list causes without linking them to the outcome, while two
trace the mechanism step by step"
    -> grounded:  the observation fixes what "depth" means
here.
  "Writing is engaging and well-crafted"
    -> ungrounded.

# Task prompt (for context)
{task_prompt}

# Items to audit
{items}

Return a JSON array with one object per item, same order, no
other text:
- "item_index":  the index of the item
- "ungrounded":  boolean
- "qualifier":  the degree word(s), or null
- "reason":  one sentence
```

The judge returns one verdict per item, and we report the fraction of items marked ungrounded. Because the model occasionally omits an item from a batch, verdicts are matched back by index and only the missing items are re-queried; discarding a whole batch on a partial response would delete items non-randomly, biasing the estimate toward whichever criteria are easiest to judge. Items still unresolved after five attempts are excluded and counted (Section 5.3).