ENGRICH: ENHANCING GENERATIVE REWARD MODELING WITH CRITIQUES FROM HUMANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative reward models (GRMs) are important for LLM optimization. Unlike scalar reward models, GRMs generate natural-language critiques alongside preference judgments, providing finer-grained evaluation signals. Their effectiveness depends heavily on critique reliability. However, existing GRM training typically uses final preference correctness as outcome supervision. Because the preference outcome space is highly constrained, unreliable critiques can still yield correct outcomes and thus be reinforced. Recent work leverages human critiques for process supervision, but such critiques are scarce and are often reduced to scalar rewards, leaving their fine-grained evaluative information underutilized. We argue that evaluative criteria learned from human critiques can be generalized to broader outcome-only preference data. To this end, we propose **EnGRICH**, a GRM training framework that pairs the GRM with a training-time MetaCritic learned from a small set of human critiques. MetaCritic constructs response-specific rubrics and uses them to evaluate the evidence coverage and correctness of generated critiques. The resulting signals provide both process rewards for fine-grained credit assignment and structured guidance for exploring better critiques. During GRM training, MetaCritic is further optimized to generalize human-grounded evaluative criteria to outcome-only data. At inference, the trained GRM operates independently. Experiments across seven reward-model benchmarks show that EnGRICH consistently improves over competitive baselines, while further analyses validate the effectiveness of its core mechanisms.[1]

1 INTRODUCTION

Reward models (RMs) play an important role across LLM training and inference (Lambert et al., 2025b; Zhang et al., 2025). RMs typically take one of two forms: scalar reward models, which directly assign numerical scores to responses, and generative reward models (GRMs), which compare candidate responses by generating language-based critiques and predicting the preferred response (Ye et al., 2024; Lai et al., 2026). By verbalizing the evaluation process, GRMs can leverage the reasoning capabilities of pretrained LLMs and provide richer evaluation signals than scalar RMs (Chen et al., 2025). This advantage is particularly valuable in non-verifiable domains such as creative writing and open-ended instruction following, where evaluation often requires nuanced semantic judgment (Xu et al., 2026).

A common approach to training GRMs is reinforcement learning with verifiable rewards (RLVR), which rewards agreement between the model's final judgment and the preference label (Liu et al., 2025; Chen et al., 2026). In conventional RLVR tasks such as mathematics and coding, outcome rewards provide strong supervision because their large solution spaces make correct solutions difficult to obtain by chance (Wang et al., 2026c). However, pairwise GRMs operate over a much more restricted outcome space, where the final judgment reduces to choosing between two responses. Consequently, a GRM may reach the correct preference judgment even when its critique is unreliable (Lai et al., 2026). Such spurious successes can still be reinforced during training and subsequently affect downstream tasks such as policy optimization and candidate selection. Therefore, reliable GRM training requires supervision of both the final preference judgment and the quality of its supporting critique.

[1] We release our code at `https://anonymous.4open.science/r/ENGRICH`.

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.

Motivated by this limitation, recent work has explored finer-grained supervision for GRM critiques along two directions. One line derives process supervision from human critiques: RM-NLHF (Wang et al., 2026c) uses critique similarity as a process reward and extends it to outcome-only data, while RC-GRPO (Wang et al., 2026a) rewards rationale consistency with human critiques. However, these methods reduce human critiques to scalar process rewards, leaving their fine-grained diagnostic information underutilized beyond scoring. Another line uses explicit rubrics to structure evaluation: Rubric-ARM (Xu et al., 2026) jointly optimizes a rubric generator and judge through alternating RL. However, the rubric generator is supervised only by preference correctness rather than human critiques, and both modules are required at inference.

Prior work highlights the value of human critiques for fine-grained supervision and rubrics as explicit evaluative criteria. However, since high-quality human critiques are costly to collect, only a small fraction of preference data is accompanied by them (Akyürek et al., 2023). We therefore seek to learn from these limited critiques what evidence a well-grounded GRM critique should cover and extend such evaluative criteria to outcome-only preference data. Rather than reducing this supervision to scalar rewards alone, we retain its diagnostic structure, allowing the same human-grounded process supervision to support both process credit assignment and diagnosis-guided exploration (Li et al., 2026).

To this end, we propose **EnGRICH** (**En**hancing **G**enerative **R**eward Modeling w**I**th **C**ritiques from **H**umans), a training framework that pairs a GRM with a MetaCritic. MetaCritic is initialized from human critiques to construct response-specific rubrics and diagnose the evidence coverage and correctness of GRM critiques. These diagnoses support GRM optimization in two complementary ways: they provide process rewards for finer-grained credit assignment and structured guidance for exploring better critiques. During GRM training, MetaCritic is further optimized to generalize human-grounded evaluative criteria to outcome-only data. MetaCritic is used only during training; inference requires only the standalone GRM.

Using EnGRICH, we train GRMs on limited human critiques together with outcome-only preference data and evaluate them across diverse reward-model benchmarks. EnGRICH consistently outperforms both outcome-supervised and prior process-supervised baselines, while further analyses validate the effectiveness of its process supervision and exploration mechanisms.

2 Related Work

Generative Reward Models and Fine-Grained Supervision. Scalar reward models learn scalar rewards from pairwise preferences (Christiano et al., 2017; Ouyang et al., 2022). LLM evaluators such as JudgeLM (Zhu et al., 2025), Prometheus (Kim et al., 2024), and Critic-RM (Yu et al., 2025) extend reward modeling with natural-language critiques, while recent GRMs such as GenRM (Mahan et al., 2024), DeepSeek-GRM (Liu et al., 2025), and RM-R1 (Chen et al., 2026) generate reasoning before predicting preferences. Yet their training still largely relies on final preference correctness, which may reinforce unreliable critiques that yield correct judgments. RC-GRPO (Wang et al., 2026a) directly derives scalar rationale-consistency rewards from human critiques. RM-NLHF (Wang et al., 2026c) further learns a scalar MetaRM to extend critique-based process rewards to outcome-only data, while R-Align (Lai et al., 2026) similarly uses a Meta-Judge to assess rationale alignment with gold rationales. These methods improve process supervision but largely compress fine-grained diagnostic information into scalar or holistic signals. Another line uses explicit rubrics to structure evaluation. Rubric-ARM (Xu et al., 2026) jointly learns a rubric generator and a judge, but optimizes rubric generation through preference correctness without explicit supervision from human critiques, and requires both modules at inference. EnGRICH instead learns human-grounded, instance-specific rubrics from limited human critiques, generalizes them to outcome-only preference data, and uses the resulting diagnoses both as process rewards and as feedback for targeted revision.

Learning from Natural-Language Feedback. Our use of fine-grained diagnostic signals to guide evaluation trajectories is inspired by broader work on learning from natural-language feedback. ILF (Scheurer et al., 2024) trains on feedback-conditioned refinements, while Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), and RL4F (Akyürek et al., 2023) use critiques or reflections to revise model outputs. More recent methods incorporate such refinement into online reinforcement learning. Critique-GRPO (Zhang et al., 2026) learns from both initial solutions and

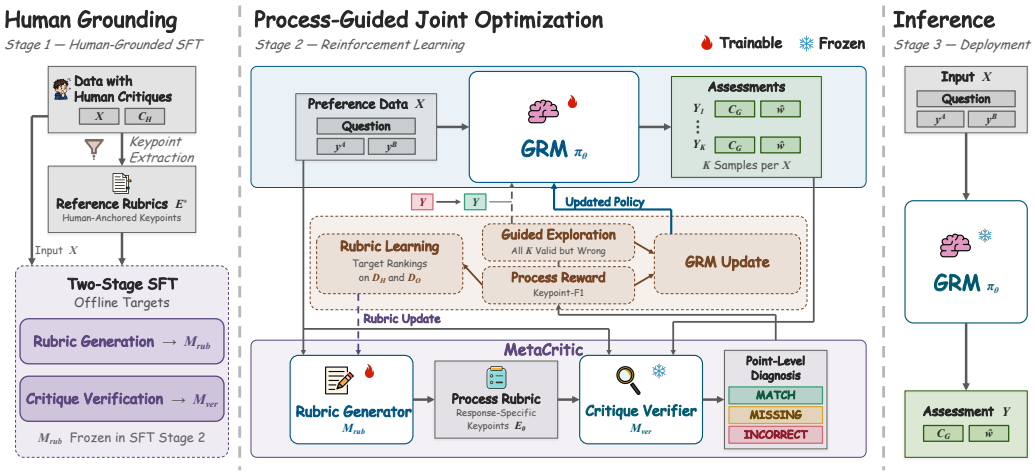


Figure 1: Illustration of EnGRICH.

critique-guided refinements, while MulFeRL (Li et al., 2026) invokes verbal feedback on failed rollouts and assigns credit to verifier-confirmed progress. These methods primarily improve problem-solving outputs or policies. We argue that fine-grained supervision can serve two complementary roles in GRMs: providing scalar process rewards for process credit assignment and keypoint-level diagnosis for guided exploration.

3 Method

3.1 Problem Formulation

A GRM evaluates two candidate responses by generating a comparative critique followed by a preference judgment. Given an input $X = (x, y^A, y^B)$, where x is the question and y^A, y^B are the candidate responses, the GRM π_θ produces an assessment $Y = (C_G, \hat{w})$, where C_G is the generated critique and $\hat{w} \in \{A, B\}$ is the predicted preference. Given the cost of collecting human critiques (Akyürek et al., 2023), we consider a training setting where only a small fraction of the data is accompanied by human critiques, while the majority provides only preference labels. We therefore define a human-critique subset $\mathcal{D}_H = \{(X, w, C_H)\}$ and a larger outcome-only subset $\mathcal{D}_O = \{(X, w)\}$, where w denotes the gold preference and C_H is the corresponding human critique.

Our goal is to learn from the limited human critiques the key evaluation points that characterize a well-grounded critique, and to generalize these evaluative criteria to supervise GRM training on broader preference data. This enables the GRM to produce reliable critiques and accurate preference judgments across diverse tasks.

3.2 Overview

As illustrated in Figure 1, **EnGRICH** turns a small set of human critiques into reusable process supervision for GRM optimization. MetaCritic uses the rubric generator M_{rub} to construct response-specific rubrics and the verifier M_{ver} to diagnose the evidence coverage and correctness of GRM critiques. The resulting signals support GRM optimization in two complementary ways: they provide process rewards for fine-grained credit assignment and structured guidance for exploring better critiques when outcome supervision provides limited learning signal. During GRM training, MetaCritic is further optimized with target rankings on $\mathcal{D}_H$ and $\mathcal{D}_O$ to generalize human-grounded evaluative criteria to outcome-only data. MetaCritic is used only during training, while the GRM operates independently at inference. We next describe the core mechanisms of EnGRICH in detail.

3.3 From Human Critiques to Process Supervision

Human critiques reveal the keypoints that support a preference judgment. A reliable GRM assessment should predict the correct preference and accurately address the keypoints. The challenge

is therefore twofold: transforming limited human critiques into keypoint-level process supervision, and generalizing the resulting human-grounded evaluative criteria to broader outcome-only data. We introduce MetaCritic to construct response-specific keypoints, assess GRM critiques against them, and extend this supervision beyond the human-critiqued data.

MetaCritic consists of a rubric generator M_{rub} and a critique verifier M_{ver}. Given an input X, M_{rub} generates a rubric $\mathcal{E}$ consisting of the keypoints that a well-grounded critique of X should cover. Given X, a rubric $\mathcal{E}$, and a generated critique C_G, M_{ver} labels each keypoint as `MATCH` if C_G covers it correctly, `MISSING` if C_G omits it or covers it incompletely, and `INCORRECT` if C_G misstates it. Let m, u, and e denote the numbers of keypoints with each label, respectively.

This design turns the same keypoint verification into two complementary training signals. We aggregate the keypoint labels into a scalar process reward that quantifies critique quality beyond preference correctness, while retaining the rubric and its labels as structured feedback to guide exploration toward better critiques. To make the process reward account for both keypoint coverage and correctness, we draw on precision and recall in information retrieval (Manning et al., 2008) and define it as *Keypoint-F1*:

$$P = \frac{m}{m+e}, \qquad R = \frac{m}{m+u+e}, \qquad q(\mathcal{E}, C_G) = \frac{2PR}{P+R} = \frac{2m}{2m+u+2e}. \tag{1}$$

where `INCORRECT` keypoints reduce both precision and recall, whereas `MISSING` keypoints reduce only recall[2].

Human-grounded initialization of MetaCritic. We initialize MetaCritic on $\mathcal{D}_H$ in two SFT stages. First, M_{rub} learns to generate rubrics from inputs X in $\mathcal{D}_H$ using offline targets that a frozen external LLM $\mathcal{A}$ derives from the human critique C_H. We then freeze M_{rub} and train M_{ver} to label critiques sampled from the initial GRM against rubrics generated by M_{rub}, matching its inputs during GRM training. The corresponding keypoint-level labels are also constructed by $\mathcal{A}$. Both components are implemented as independent LoRA adapters over a shared frozen backbone with the same architecture as the GRM.

3.4 Learning with Human-Grounded Process Supervision

Keypoint verification supervises the GRM in two ways: scalar process rewards distinguish critique quality among correct assessments, while keypoint-guided exploration recovers learning signal in groups where all assessments fail. Alongside GRM optimization, we train M_{rub} to extend the evaluative criteria learned from human critiques to outcome-only data.

Optimizing the GRM with process supervision. For each X, we sample K assessments $Y_k = (C_{G,k}, \hat{w}_k)$ from the GRM π_θ and a rubric $\mathcal{E}_0 \sim M_{\text{rub}}(\cdot \mid X)$, and combine preference correctness with Keypoint-F1:

$$R_k = \begin{cases} -1, & \text{invalid format,} \\ 0, & \text{valid format, } \hat{w}_k \neq w, \\ 1 + \lambda\, q(\mathcal{E}_0, C_{G,k}), & \text{valid format, } \hat{w}_k = w, \end{cases} \tag{2}$$

where $\lambda \geq 0$ weights the process reward. We grant process credit only to correct predictions, since rewarding process quality for outcome-incorrect assessments can decouple process from outcome and invite reward hacking (Sun et al., 2025). Thus, Keypoint-F1 provides bounded process-level differentiation among correct assessments.

For failed assessments, keypoint verification instead guides exploration to recover the learning signal. When all K assessments have valid formats but incorrect predictions, their rewards tie at zero and group-relative advantages vanish. Yet these assessments need not fail in the same way: M_{ver} reveals which keypoints in $\mathcal{E}_0$ are `MISSING` or `INCORRECT` in each critique.

Inspired by prior refinement methods with external guidance (Zhang et al., 2026; Li et al., 2026), we use the sampled rubric $\mathcal{E}_0$ and its keypoint labels as privileged information to guide exploration toward a critique that better addresses the identified gaps. Specifically, we randomly select one failed assessment in the group for revision. The exploration prompt includes the rubric and keypoint

[2]We omit the fixed input X from the notation of $q(\mathcal{E}, C_G)$ for brevity.

labels but withholds the gold preference w and prediction correctness, so that the new assessment is guided by keypoint-level information while preventing the model from simply flipping its prediction based on outcome information. If the resulting assessment has a valid format and predicts the correct preference, it replaces its parent and is rescored under $\mathcal{E}_0$; otherwise, the group remains unchanged. Successful exploration thus converts keypoint-level guidance into reward contrast, restoring learning signal in an otherwise uninformative group. We optimize the GRM with GRPO (Shao et al., 2024), centering rewards within the resulting group:

$$A_k = R_k - \mathrm{mean}_j(R_j), \qquad \mathcal{L}_{\mathrm{GRM}}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \ell(\theta; X, Y_k, A_k), \tag{3}$$

where ℓ is the token-level GRPO loss. For assessments obtained through guided exploration, the standard importance ratio does not match the policy that generated them. Following prior work (Zhang et al., 2026; Yan et al., 2025), we evaluate them under the policy conditioned only on X and apply $f(r_t) = r_t/(r_t + \gamma)$, where $r_t = \pi_\theta(y_t \mid X, y_{<t})/\pi_{\theta_{\mathrm{old}}}(y_t \mid X, y_{<t})$ and $0 < \gamma < 1$. This encourages the GRM to internalize successful explorations from X alone.

Generalizing human-grounded rubric generation. Human-grounded SFT in §3.3 initializes M_{rub} with evaluative criteria from human critiques. During GRM training, we further optimize M_{rub} on $\mathcal{D}_H$ and $\mathcal{D}_O$ to retain these criteria and generalize them to outcome-only data, while keeping M_{ver} fixed for consistent keypoint verification.

We train M_{rub} with complementary supervision for human-grounded retention and outcome-guided adaptation. On $\mathcal{D}_H$, the human-derived reference rubric $\mathcal{E}^*$ scores the same GRM critiques with Keypoint-F1. Together with preference correctness, these scores define a reference ranking: correctness determines the primary ordering, while critiques with the same correctness are ordered by their scores under $\mathcal{E}^*$. Each generated rubric then scores the same critiques, and is rewarded according to how well its induced ranking matches this reference ranking.

On $\mathcal{D}_O$, where human critiques are unavailable, preference correctness provides coarser target rankings under the same reward form. These rankings provide RL supervision for M_{rub}, initialized by human-grounded SFT on $\mathcal{D}_H$, to generalize the evaluative criteria learned from human critiques to broader outcome-only data without reference rubrics. Meanwhile, continued supervision on $\mathcal{D}_H$ encourages M_{rub} to retain fine-grained distinctions in keypoint coverage and correctness beyond the final judgment.

Specifically, for each X, we sample $\{\mathcal{E}_g\}_{g=0}^{G_X-1}$ from $M_{\mathrm{rub}}(\cdot \mid X)$ and apply them to the same K original GRM critiques; only $\mathcal{E}_0$ supplies GRM process rewards and exploration guidance. Let $c_i = \mathbb{I}[\hat{w}_i = w]$ denote preference correctness. On $\mathcal{D}_H$, let $q_i^* = q(\mathcal{E}^*, C_{G,i})$ denote the reference-rubric score. For format-valid assessments, we define:

$$\begin{aligned} \mathcal{P}_X &= \begin{cases} \{(i,j) : c_i > c_j\} \cup \{(i,j) : c_i = c_j,\ q_i^* > q_j^*\}, & \text{on } \mathcal{D}_H, \\ \{(i,j) : c_i > c_j\}, & \text{on } \mathcal{D}_O, \end{cases} \\ r_g^{\mathrm{rub}} &= \frac{1}{|\mathcal{P}_X|} \sum_{(i,j)\in\mathcal{P}_X} \log\sigma(q(\mathcal{E}_g, C_{G,i}) - q(\mathcal{E}_g, C_{G,j}))\,. \end{aligned} \tag{4}$$

where σ is the logistic sigmoid. Inputs with $\mathcal{P}_X = \varnothing$ are skipped for rubric optimization. The resulting rewards are centered across the G_X sampled rubrics without standard-deviation normalization and used to optimize M_{rub} with GRPO.

4 Experiments

4.1 Experimental Settings

Datasets. Following the data-mixture design of RM-NLHF (Wang et al., 2026c), we construct a 60K pairwise preference mixture with 12K human-critique examples $\mathcal{D}_H$ and 48K outcome-only examples $\mathcal{D}_O$. $\mathcal{D}_H$ is drawn from HelpSteer3 (Wang et al., 2025) and contains natural-language human critiques. $\mathcal{D}_O$ combines the Skywork-Reward-Preference-80K mixture (Liu et al., 2024a), Code-Preference-Pairs, Math-Step-DPO-10K (Lai et al., 2024), HH-RLHF (Bai et al., 2022), Tulu-3 (Lambert et al., 2025a), and LitBench-Train (Fein et al., 2026), covering general assistance, code,

Table 1: Results across seven reward-model benchmarks. **Bold** and underlined indicate the best and second-best controlled results within each backbone; Overall is the unweighted mean.

Model / Method	HS3	RB2	RM-B	SCAN	HREF	LitBench	WQ	Overall
Open-Weight Reward Models								
GRAM-Qwen3-4B	74.17	66.01	66.87	74.44	60.78	62.24	55.03	65.65
GRAM-R^2-3B	76.37	50.94	64.18	69.97	63.19	62.64	59.16	63.78
CE-RM-4B	73.44	70.76	79.50	68.85	59.66	57.42	52.78	66.06
UltraRM-13B	74.60	53.07	57.20	74.31	58.26	69.80	59.06	63.76
RM-R1-14B	76.42	65.25	81.11	77.32	64.79	65.10	58.86	69.84
UniRRM-14B	79.14	72.65	82.65	80.83	62.91	63.08	56.76	71.15
Controlled Training								
Qwen3-4B								
Base	72.07	52.34	69.88	75.72	52.12	62.76	55.49	62.91
SFT	72.23	46.23	65.14	72.20	57.02	60.50	56.56	61.41
DPO	74.29	59.98	72.05	76.20	56.97	64.27	57.02	65.83
GRPO	75.16	62.28	73.81	74.60	56.22	63.33	56.10	65.93
RC-GRPO	74.53	60.37	73.62	74.44	58.16	63.45	56.46	65.86
RM-NLHF	75.00	61.50	73.98	75.24	57.20	63.79	57.02	66.25
EnGRICH	**77.06**	**65.42**	**75.12**	**77.48**	**62.60**	**65.69**	**61.15**	**69.22**
Gemma3-12B								
Base	71.29	44.14	65.84	76.04	57.75	67.04	59.62	63.10
SFT	74.34	52.90	66.47	75.72	61.73	63.75	61.46	65.20
DPO	76.42	52.81	69.82	77.48	60.38	69.33	61.77	66.86
GRPO	75.78	55.61	72.56	75.24	64.27	66.67	61.92	67.44
RC-GRPO	74.90	53.62	71.21	75.56	63.50	65.81	61.61	66.60
RM-NLHF	75.62	54.80	71.90	76.04	63.40	67.20	61.51	67.21
EnGRICH	**77.71**	**59.85**	**74.66**	**78.27**	**66.91**	**69.58**	**65.24**	**70.32**

mathematics, safety, instruction following, and writing. We evaluate on seven benchmarks: HelpSteer3 (HS3) (Wang et al., 2025), RewardBench 2 (RB2) (Malik et al., 2026), RM-Bench (RM-B) (Liu et al., 2024b), SCAN-HPD (SCAN) (Wang et al., 2026b), HREF-HA (HREF) (Lyu et al., 2025), LitBench (Fein et al., 2026), and WQ-Arena (WQ) (Chakrabarty et al., 2025). [3]

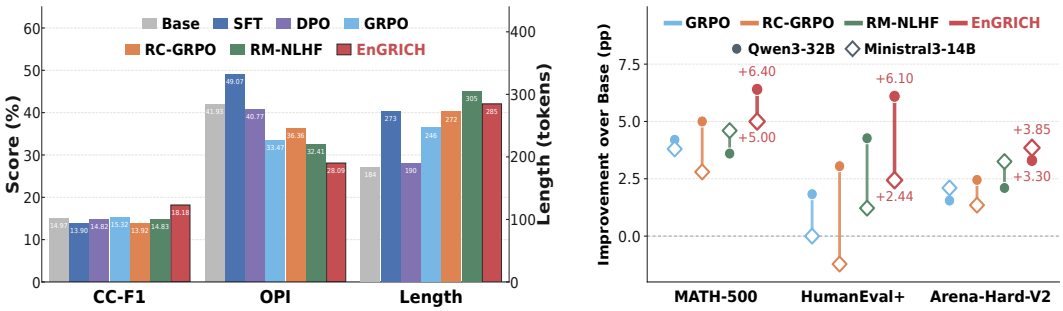
Baselines. We compare against six baselines: (1) *Base*, the backbone used directly as a GRM; (2) *SFT*, finetuned on human critiques in $\mathcal{D}_H$; (3) *DPO* (Rafailov et al., 2024), optimizing pairwise preferences; (4) *GRPO* (Shao et al., 2024), using format and preference correctness as rewards; (5) *RC-GRPO* (Wang et al., 2026a), which requires human rationales and is thus limited to $\mathcal{D}_H$, using rationale consistency as process supervision; and (6) *RM-NLHF* (Wang et al., 2026c), which learns process rewards from human-critique similarity on $\mathcal{D}_H$ and extends them to $\mathcal{D}_O$ with a scalar estimator. Both process-supervised baselines combine their process rewards with correctness and are optimized with GRPO. Representative public reward models are included for reference only due to differences in training data and backbones.

Implementation Details. We use Qwen3-4B (Yang et al., 2025) in non-thinking mode and Gemma3-12B-Instruct (Gemma3-12B) (Team et al., 2025) as backbones. The GRM, rubric generator M_{rub}, and critique verifier M_{ver} use separate LoRA (Hu et al., 2021) adapters over a frozen backbone. For training examples with human-derived supervision available from prior work (Wang et al., 2026c), a frozen Qwen3.7-Max[4] teacher constructs reference rubrics and annotates keypoint-level relations. The two MetaCritic SFT stages each run for one epoch, while the SFT baseline is trained for three epochs. Each RL method is trained for one pass over its available training data. Training uses 32 H20 GPUs and inference uses 8 H20 GPUs. At inference, the GRM uses greedy decoding. [5]

[3] Full dataset details are provided in § C.

[4] `https://www.alibabacloud.com/help/en/model-studio/qwen3-7-max`

[5] Implementation details, settings, and reward-model sources are provided in § D.



(a) Direct critique evaluation.

(b) Downstream revision evaluation.

Figure 2: Critique quality evaluation. All compared methods use Qwen3-4B as the GRM backbone. Left: CC-F1, OPI, and average assessment length on HelpSteer3 validation set; higher CC-F1 and lower OPI indicate better critique quality. Right: response-revision gains over direct-answer baselines.

4.2 Main Results

As shown in Table 1, EnGRICH consistently improves performance across all seven reward-model benchmarks and both backbones, demonstrating its effectiveness across diverse evaluation settings. The comparison with process-supervised baselines further highlights the mechanism of EnGRICH. RC-GRPO uses human-derived rationale consistency as process supervision, while RM-NLHF extends human-critique supervision through a learned scalar process reward. EnGRICH instead couples rubric learning with GRM optimization: M_{rub} learns human-grounded evaluative criteria on $\mathcal{D}_H$ and generalizes them to outcome-only data $\mathcal{D}_O$, while keypoint verification turns these criteria into both scalar process rewards for fine-grained credit assignment and keypoint-level guidance for exploration in all-failed groups. This combination allows limited human critiques to supervise both critique quality and exploration, leading to stronger and more consistent gains.

4.3 Critique Quality Evaluation

We further evaluate the quality of the generated critiques from direct and downstream perspectives. For direct evaluation, we adopt two critique-quality metrics: **Common Correct F1 (CC-F1)** (Wang et al., 2026a) and **Outcome Process Inconsistency (OPI)** (Wang et al., 2026c). CC-F1 is computed on examples where all compared GRMs predict the correct preference and measures the F1 agreement between GRM critiques and human critiques. OPI measures the fraction of correctly predicted examples with unreliable critiques, defined by core-argument agreement with the human critique of at most 0.5. Both metrics are computed on the HelpSteer3 validation set with GPT-5.5 as the judge. As shown in Figure 2a, EnGRICH achieves higher CC-F1 and lower OPI than the baselines, while comparable critique lengths indicate that the gains do not simply come from longer critiques. For downstream evaluation, we use generated critiques to revise responses on MATH-500 (Lightman et al., 2023), HumanEval+ (Liu et al., 2023), and Arena-Hard-v2.0 (Li et al., 2025), with frozen Qwen3-32B (Yang et al., 2025) and Ministral3-14B (Liu et al., 2026) as editors. Figure 2b shows that EnGRICH consistently yields larger revision gains, suggesting that its critiques provide more informative and actionable evaluative evidence. Together, the direct and downstream evaluations show that EnGRICH improves critique quality rather than merely increasing preference accuracy.

5 Analysis

5.1 Ablation Study

In Table 2, we ablate four components of EnGRICH to examine process credit assignment, keypoint-guided exploration, rubric generalization, and the transfer of process supervision to outcome-only data. *w/o Process Reward* sets $\lambda = 0$, removing Keypoint-F1 while retaining keypoint-guided exploration. *w/o Guided Exploration* removes exploration in all-failed groups while retaining the process reward. *w/o Rubric Generalization* freezes M_{rub} after human-grounded SFT, removing its

Table 2: Ablation study of EnGRICH on Gemma3-12B.

	Benchmark Performance							*Critique Quality*	
Method	**HS3**	**RB2**	**RM-B**	**SCAN**	**HREF**	**LitBench**	**WQ**	**CC-F1**	**OPI**
EnGRICH	**77.71**	**59.85**	**74.66**	78.27	66.91	**69.58**	**65.24**	**19.36**	**17.14**
w/o Process Reward	76.74	56.90	72.60	**78.43**	63.15	67.16	62.07	14.96	25.81
w/o Guided Exploration	77.31	55.70	74.10	77.64	**66.94**	69.15	64.22	18.22	19.92
w/o Rubric Generalization	76.98	53.95	71.60	75.88	61.79	66.05	60.49	16.37	23.02
w/o $\mathcal{D}_O$ Process Supervision	77.15	54.65	72.20	76.84	62.71	66.57	61.41	17.44	22.14

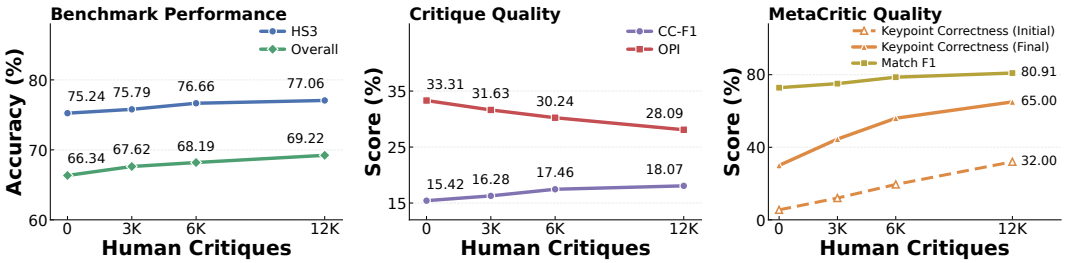


Figure 3: Scaling human critique supervision with Qwen3-4B as backbone. We report benchmark performance (left), critique quality (middle), and MetaCritic quality (right) as human-critique coverage increases. Initial denotes MetaCritic before online GRM training (SFT-initialized when human critiques are available), and Final denotes MetaCritic after GRM training.

continued optimization during GRM training. Finally, *w/o $\mathcal{D}_O$ Process Supervision* removes both Keypoint-F1 and keypoint-guided exploration on $\mathcal{D}_O$, leaving preference correctness as the only GRM supervision on this subset.

The ablation results show complementary benefits from the core components of EnGRICH. Removing the process reward causes the largest degradation in critique quality while also lowering performance on most benchmarks, supporting the importance of Keypoint-F1 for fine-grained credit assignment among preference-correct assessments. Removing guided exploration also generally degrades benchmark performance, showing complementary gains from recovering learning signal in all-failed groups. Freezing M_{rub} causes the largest average drop in benchmark performance and clear degradation in critique quality, supporting the continued generalization of human-grounded evaluative criteria beyond their SFT initialization. Finally, restricting process supervision to $\mathcal{D}_H$ underperforms full EnGRICH across all benchmarks and critique-quality metrics, showing the benefit of extending human-grounded process supervision to $\mathcal{D}_O$.

5.2 Scaling Human Critique Supervision

We vary the size of the human-critiqued subset $\mathcal{D}_H$ over $\{0, 3\text{K}, 6\text{K}, 12\text{K}\}$, while keeping all 60K preference pairs and labels fixed; the remaining examples form the outcome-only subset $\mathcal{D}_O$. Figure 3 reports benchmark performance, critique quality, and MetaCritic quality. For MetaCritic, GPT-5.5 evaluates sampled HelpSteer3 and RM-Bench examples: *Keypoint Correctness* measures the fraction of generated rubrics whose keypoints are all correct, while *Match F1* is the macro-F1 of M_{ver} over `MATCH`, `MISSING`, and `INCORRECT` keypoint labels.

Increasing human-critique supervision consistently improves all three aspects, while rubric generalization further improves rubric quality across different sizes of $\mathcal{D}_H$. Notably, the gains emerge even with limited human critiques. These results suggest that EnGRICH can generalize human-grounded evaluative criteria to outcome-only data and reuse them as both scalar process rewards and keypoint-level guidance for exploration during GRM optimization. Thus, limited human critiques can provide reusable supervision beyond $\mathcal{D}_H$. Moreover, M_{ver} remains consistently accurate, with only marginal gains from additional human critiques. This supports keeping M_{ver} fixed and suggests that the base model may itself serve as the verifier, further simplifying EnGRICH.

Table 3: Downstream performance of Qwen3-4B-Instruct-2507 policies trained with Gemma3-12B as the GRM backbone.

Policy Setting	IFEval P-Strict	AlpacaEval 2 LC	Arena-Hard v2 CW
Initial policy	84.84	39.63	15.24
Outcome-GRPO	83.73	40.54	19.19
RC-GRPO	83.55	40.21	18.87
RM-NLHF	84.10	40.43	19.74
EnGRICH	**87.06**	**42.80**	**21.43**

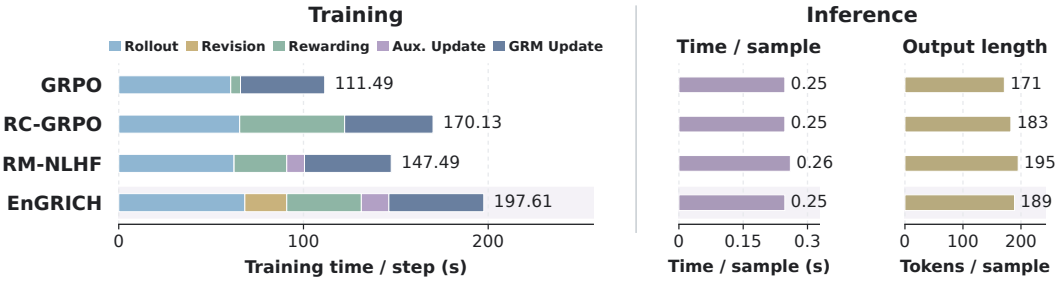


Figure 4: Training and inference efficiency of RL-based GRMs on Gemma3-12B. *Aux. Update* denotes MetaRM updates in RM-NLHF and M_{rub} updates in EnGRICH. Inference time and output length are averaged per sample across seven benchmarks.

5.3 Transfer to Policy Optimization

Beyond reward-model performance and critique quality, we examine whether the gains of EnGRICH transfer to downstream policy optimization. For each GRM variant, we initialize a policy from Qwen3-4B-Instruct-2507 (Yang et al., 2025) and optimize it with GRPO for 100 steps on HelpSteer3 training prompts, using the corresponding GRM as the reward model. As shown in Table 3, the policy trained with the EnGRICH GRM achieves the strongest performance across all three downstream benchmarks. These results indicate that the gains of EnGRICH extend beyond reward-model evaluation to downstream policy optimization.

5.4 Efficiency

As shown in Figure 4, EnGRICH introduces additional training overhead over GRPO from Keypoint-F1 verification, keypoint-guided exploration and rubric generalization. This overhead is comparable to RC-GRPO and RM-NLHF, which respectively add rationale-consistency scoring and a learned process-reward model, whereas EnGRICH uses MetaCritic to generalize human-grounded evaluative criteria and provide keypoint-level verification. At inference, the GRM operates independently of MetaCritic, with amortized inference time per sample and assessment length comparable to the baselines.

6 Conclusion

We study reliable training of generative reward models beyond final preference correctness. We propose **EnGRICH**, which treats human critiques not merely as additional supervision, but as a source of reusable evaluative criteria that can be generalized to broader outcome-only data. Through MetaCritic, these criteria provide both fine-grained process rewards for credit assignment and structured guidance for exploring better critiques, while the trained GRM operates independently at inference. Experiments across seven reward-model benchmarks and downstream policy optimization demonstrate consistent improvements, and further analyses validate the complementary roles of process supervision, guided exploration, and rubric generalization. Future work may explore how to better extract, generalize, and reuse the evaluative criteria in human critiques, enabling more reliable process supervision across broader data and evaluation settings.

AI USE STATEMENT

In this work, GPT-5.6 was used for grammar checking, language polishing, and assistance with literature search and the identification of relevant prior work. Qwen3.7-Max was used to construct reference rubrics from human-derived keypoints and annotate keypoint-level relations for MetaCritic initialization. GPT-5.5 was used as an external evaluator for critique-quality, MetaCritic, and Arena-Hard evaluations. All AI-assisted writing was manually reviewed by the authors, and suggested literature was verified against the original sources before being cited or discussed. We take full responsibility for the final content and experimental results of this work.

ETHICS STATEMENT

Our work uses existing preference datasets and benchmarks and does not involve the collection of new human-subject data. As with other reward-modeling methods, the learned evaluative criteria may reflect biases or limitations present in the underlying data and human feedback. We encourage careful evaluation across diverse settings when applying the resulting models. We follow the licenses and intended research uses of the datasets and models employed in our experiments.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an anonymous link to our source code and implementation. The main paper describes the proposed method and experimental setup, while the appendix provides additional implementation details, training configurations, dataset and evaluation settings, and baseline specifications needed to reproduce our experiments.

REFERENCES

Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs, 2023. URL https://arxiv.org/abs/2305.08844.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

Tuhin Chakrabarty, Philippe Laban, and Chien-Sheng Wu. AI-Slop to AI-Polish? Aligning Language Models through Edit-Based Writing Rewards and Test-time Computation, 2025. URL https://arxiv.org/abs/2504.07532.

Bin Chen, Xinzge Gao, Chuanrui Hu, Penghang Yu, Hua Zhang, and Bing-Kun Bao. Reasongrm: Enhancing generative reward models through large reasoning models, 2025. URL https://arxiv.org/abs/2506.16712.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. Rm-r1: Reward modeling as reasoning, 2026. URL https://arxiv.org/abs/2505.02387.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4299–4307, 2017.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Daniel Fein, Sebastian Russo, Violet Xiang, Kabir Jolly, Rafael Rafailov, and Nick Haber. LitBench: A benchmark and dataset for reliable evaluation of creative writing. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European*

Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 7740–7755, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.362. URL https://aclanthology.org/2026.eacl-long.362/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models, 2024. URL https://arxiv.org/abs/2310.08491.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms, 2024. URL https://arxiv.org/abs/2406.18629.

Yanlin Lai, Mitt Huang, Hangyu Guo, Xiangfeng Wang, Haodong Li, Shaoxiong Zhan, Liang Zhao, Chengyuan Yao, Yinmin Zhang, Qi Han, Chun Yuan, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. R-align: Enhancing generative reward models through rationale-centric meta-judging, 2026. URL https://arxiv.org/abs/2602.06763.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025a. URL https://arxiv.org/abs/2411.15124.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797, 2025b.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34209–34231, 2025.

Xuancheng Li, Haitao Li, Yujia Zhou, YiqunLiu, and Qingyao Ai. Mulferl: Enhancing reinforcement learning with verbal feedback in a multi-turn loop, 2026. URL https://arxiv.org/abs/2601.22900.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz,

Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms, 2024a. URL https://arxiv.org/abs/2410.18451.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style, 2024b. URL https://arxiv.org/abs/2410.16184.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling, 2025. URL https://arxiv.org/abs/2504.02495.

Xinxi Lyu, Yizhong Wang, Hannaneh Hajishirzi, and Pradeep Dasigi. HREF: Human Response-Guided Evaluation of Instruction Following in Language Models, 2025. URL https://arxiv.org/abs/2412.15524.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models, 2024.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation, 2026. URL https://arxiv.org/abs/2506.01937.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2024. URL https://arxiv.org/abs/2305.18290.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale, 2024. URL https://arxiv.org/abs/2303.16755.

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

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Zetian Sun, Dongfang Li, Zhuoen Chen, Yuhuai Qin, and Baotian Hu. Stabilizing long-term multi-turn reinforcement learning with gated rewards, 2025. URL https://arxiv.org/abs/2508.10548.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Binghai Wang, Yantao Liu, Yuxuan Liu, Tianyi Tang, Shenzhi Wang, Chang Gao, Chujie Zheng, Yichang Zhang, Le Yu, Shixuan Liu, Tao Gui, Qi Zhang, Xuanjing Huang, Bowen Yu, Fei Huang, and Junyang Lin. Outcome accuracy is not enough: Aligning the reasoning process of reward models, 2026a. URL https://arxiv.org/abs/2602.04649.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. HelpSteer3-Preference: Open Human-Annotated Preference Data across Diverse Tasks and Languages, 2025. URL https://arxiv.org/abs/2505.11475.

Zongqi Wang, Tianle Gu, Chen Gong, Xin Tian, Siqi Bao, and Yujiu Yang. SCAN: Structured Capability Assessment and Navigation for LLMs, 2026b. URL https://arxiv.org/abs/2505.06698.

Zongqi Wang, Rui Wang, Yuchuan Wu, Yiyao Yu, Pinyi Zhang, Shaoning Sun, Yujiu Yang, and Yongbin Li. Reward modeling from natural language human feedback, 2026c. URL https://arxiv.org/abs/2601.07349.

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, 2026. URL https://arxiv.org/abs/2602.01511.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL https://arxiv.org/abs/2504.14945.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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, 2024. URL https://arxiv.org/abs/2410.03742.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. Self-generated critiques boost reward modeling for language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11499–11514, Albuquerque, New Mexico, apr 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.573.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025. URL https://arxiv.org/abs/2408.15240.

Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback, 2026. URL https://arxiv.org/abs/2506.03106.

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

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges, 2025. URL https://arxiv.org/abs/2310.17631.

A Algorithm Details and Analysis

This section supplements the main method with the initialization losses, training procedure, and token-level optimization objectives. We also establish properties of the resulting supervision signals. For clarity, let ϕ and ψ denote the trainable parameters of M_{rub} and M_{ver}, respectively. The backbone remains frozen, and the two components use independent LoRA adapters. After initialization, ψ remains fixed, while θ and ϕ are optimized using their respective objectives.

A.1 HUMAN-GROUNDED INITIALIZATION

Rubric generator. For each $(X, w, C_H) \in \mathcal{D}_H$, the frozen external annotator $\mathcal{A}$ extracts a reference rubric $\mathcal{E}^*$ from the human critique, as described in §3.3. This gives the initialization set $\mathcal{D}_{\text{rub}}^{\text{init}} = \{(X, \mathcal{E}^*)\}$. Writing $\mathcal{E}^* = (e_1^*, \ldots, e_{T_E}^*)$ for its tokenization, we minimize:

$$\mathcal{L}_{\text{rub}}^{\text{init}}(\phi) = -\mathbb{E}_{(X,\mathcal{E}^*)\sim\mathcal{D}_{\text{rub}}^{\text{init}}}\left[\frac{1}{T_E}\sum_{t=1}^{T_E}\log M_{\text{rub},\phi}(e_t^* \mid X, e_{<t}^*)\right]. \tag{5}$$

Only X is supplied to the generator. The human critique and gold preference are not part of its conditioning input.

Critique verifier. We next freeze the initialized rubric generator and use it to construct rubrics for inputs in $\mathcal{D}_H$. Critiques are sampled from the initial GRM, and $\mathcal{A}$ supplies a target status for each rubric keypoint. Let Z^* be the serialized status sequence and define:

$$\mathcal{D}_{\text{ver}}^{\text{init}} = \{(X, \mathcal{E}, C_G, Z^*)\}.$$

For $Z^* = (z_1^*, \ldots, z_{T_Z}^*)$, the verifier loss is:

$$\mathcal{L}_{\text{ver}}^{\text{init}}(\psi) = -\mathbb{E}_{(X,\mathcal{E},C_G,Z^*)\sim\mathcal{D}_{\text{ver}}^{\text{init}}}\left[\frac{1}{T_Z}\sum_{t=1}^{T_Z}\log M_{\text{ver},\psi}\left(z_t^* \mid X, \mathcal{E}, C_G, z_{<t}^*\right)\right]. \tag{6}$$

Thus, verifier initialization uses generated rubrics rather than only reference rubrics. Both losses apply to target tokens, not conditioning tokens. The external annotator is used offline; subsequent training uses the initialized MetaCritic and the stored reference rubrics.

A.2 TRAINING PROCEDURE

Original and retained assessments. For each input X, we distinguish the original group $\mathcal{Y}^0 = \{Y_k^0\}_{k=1}^K$ from the retained group $\widetilde{\mathcal{Y}} = \{\widetilde{Y}_k\}_{k=1}^K$ used for the GRM update. Initially, $\widetilde{Y}_k = Y_k^0$. Rubric learning always uses $\mathcal{Y}^0$, whereas GRM learning uses $\widetilde{\mathcal{Y}}$ after any successful replacement. This distinction prevents a rubric from being evaluated on a revision that it helped generate.

At the start of a rollout batch, let θ_{old} and ϕ_{old} denote the parameter snapshots used for sampling. For each input, we sample $K \geq 2$ original assessments and G_X rubrics. All candidate rubrics are evaluated against the same original critiques using the frozen verifier. On $\mathcal{D}_H$, the stored reference rubric is evaluated against these same critiques. The resulting scores determine the target pairs and rubric rewards in Eq. 4. Only $\mathcal{E}_0$ supplies GRM rewards and exploration guidance.

One-step guided exploration. When all original assessments are format-valid and outcome-incorrect, we sample a parent index J uniformly from $\{1, \ldots, K\}$. Let $Z_{0,J}$ denote its keypoint labels under $\mathcal{E}_0$. The GRM generates one revision conditioned on $(X, Y_J^0, \mathcal{E}_0, Z_{0,J})$ and the revision instruction. Neither w nor an explicit correctness indicator is included. A format-valid, outcome-correct revision replaces its parent and is rescored under $\mathcal{E}_0$; otherwise, the original group is retained. The group size therefore remains K.

Let $d_k = 1$ only when $\widetilde{Y}_k$ is an accepted revision, and $d_k = 0$ otherwise. Acceptance depends on format and preference correctness, not on an additional improvement threshold for Keypoint-F1. The revised assessment is not substituted into the rubric-learning buffer, and revision success is not added to the rubric reward. Algorithm 1 summarizes the procedure.

A.3 TOKEN-LEVEL OPTIMIZATION OBJECTIVES

GRM objective. We explicitly write the policy-surrogate terms using per-sequence token averaging. Let $\widetilde{Y}_k = (y_{k,1}, \ldots, y_{k,T_k})$ denote the retained assessment, including its critique and final preference. Define:

$$r_{k,t}(\theta) = \frac{\pi_\theta(y_{k,t} \mid X, y_{k,<t})}{\pi_{\theta_{\text{old}}}(y_{k,t} \mid X, y_{k,<t})}, \tag{7}$$

$$\mathcal{C}_\epsilon(r, a) = \min\{ra, \text{clip}(r, 1-\epsilon, 1+\epsilon)a\}. \tag{8}$$

Algorithm 1 EnGRICH training

Require: $\mathcal{D}_H$, $\mathcal{D}_O$, initial GRM and MetaCritic; $K \geq 2$, rubric group sizes G_X, and optimization hyperparameters.

Construct offline reference rubrics with $\mathcal{A}$.
Initialize M_{rub} using Eq. 5.
Freeze M_{rub}; generate rubrics and initial GRM critiques, and obtain verifier targets from $\mathcal{A}$.
Initialize M_{ver} using Eq. 6; then fix ψ.
while training is not complete **do**
  Snapshot $\theta_{\mathrm{old}} \leftarrow \theta$ and $\phi_{\mathrm{old}} \leftarrow \phi$; sample a batch $\mathcal{B}$.
  Initialize GRM and rubric update buffers.
  for each input X in $\mathcal{B}$ **do**
    Sample $\mathcal{Y}^0 \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid X)$ with K assessments.
    Sample $\{\mathcal{E}_g\}_{g=0}^{G_X-1} \sim M_{\mathrm{rub},\phi_{\mathrm{old}}}(\cdot \mid X)$.
    Verify format-valid original critiques under each $\mathcal{E}_g$; cache their labels and scores.
    if $X \in \mathcal{D}_H$ **then**
      Score original critiques under the stored reference rubric $\mathcal{E}^*$.
    end if
    Construct $\mathcal{P}_X$ from format-valid original assessments using Eq. 4.
    if $\mathcal{P}_X \neq \varnothing$ and $G_X \geq 2$ **then**
      Compute r_g^{rub} and centered advantages B_g; store the rubric update tuple.
    end if
    Set $\widetilde{\mathcal{Y}} \leftarrow \mathcal{Y}^0$ and $d_k \leftarrow 0$ for all k.
    if all original assessments are valid and incorrect **then**
      Sample J uniformly and generate one revision using $(X, Y_J^0, \mathcal{E}_0, Z_{0,J})$.
      if the revision is valid and correct **then**
        Replace $\widetilde{Y}_J$ with the revision; set $d_J \leftarrow 1$ and rescore it under $\mathcal{E}_0$.
      end if
    end if
    Compute R_k and A_k for $\widetilde{\mathcal{Y}}$ using Eqs. 2 and 3.
    Store $(X, \widetilde{\mathcal{Y}}, \{A_k, d_k\}_{k=1}^K)$ in the GRM update buffer.
  end for
  Update θ using Eq. 10.
  Update ϕ using Eq. 13 if the rubric update buffer is nonempty.
end while
return π_θ for standalone inference.

For the standard branch, this is the clipped GRPO surrogate (Shao et al., 2024). All advantages and denominator probabilities are treated as constants during differentiation. With $f(r) = r/(r+\gamma)$ and $0 < \gamma < 1$, the per-assessment loss is:

$$\ell_k(\theta) = -\frac{1}{T_k} \sum_{t=1}^{T_k} \left[(1 - d_k) \mathcal{C}_{\epsilon_{\mathrm{GRM}}}(r_{k,t}, A_k) + d_k A_k f(r_{k,t}) \right]. \tag{9}$$

The batch objective is:

$$\widehat{\mathcal{L}}_{\mathrm{GRM}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{X \in \mathcal{B}} \frac{1}{K} \sum_{k=1}^{K} \ell_k(\theta). \tag{10}$$

Here A_k is computed after replacement, without standard-deviation normalization. Every accepted revision has positive advantage because it is the only correct assessment in its retained group. Since $0 < f(r) < 1$, applying the standard clipped surrogate to the shaped ratio gives $\mathcal{C}_\epsilon(f(r), A_k) = A_k f(r)$ for these revisions. Thus, the guided branch in Eq. 9 also covers clipping applied after shaping.

Conditioning of accepted revisions. Although a revision is generated with additional diagnostic context, its training probabilities are evaluated with X as the only conditioning input. Both the numerator and denominator in Eq. 7 are recomputed under this context; feedback-conditioned rollout probabilities are not used as the denominator. The assessment must be re-encoded under X, rather than only masking the loss on feedback tokens while retaining the feedback in its context.

Policy shaping follows the general motivation of learning from guided trajectories (Zhang et al., 2026; Yan et al., 2025), but the argument of f here is the current-to-old ratio explicitly defined in

the main text. This is a shaped learning surrogate, not an unbiased importance-sampling correction for the diagnostic-conditioned behavior policy.

Rubric-generator objective. For each eligible input, compute the rubric rewards in Eq. 4 using only original critiques, and define:

$$B_g = r_g^{\mathrm{rub}} - \frac{1}{G_X} \sum_{h=0}^{G_X-1} r_h^{\mathrm{rub}}. \tag{11}$$

Let $\mathcal{E}_g = (e_{g,1}, \ldots, e_{g,U_g})$ denote the tokenized sampled rubric. Its token-level ratio is:

$$\rho_{g,t}(\phi) = \frac{M_{\mathrm{rub},\phi}(e_{g,t} \mid X, e_{g,<t})}{M_{\mathrm{rub},\phi_{\mathrm{old}}}(e_{g,t} \mid X, e_{g,<t})}. \tag{12}$$

Writing $\mathcal{B}_{\mathrm{rub}}$ for inputs with nonempty target pairs and $G_X \geq 2$, we minimize:

$$\widehat{\mathcal{L}}_{\mathrm{rub}}(\phi) = -\frac{1}{|\mathcal{B}_{\mathrm{rub}}|} \sum_{X \in \mathcal{B}_{\mathrm{rub}}} \frac{1}{G_X} \sum_{g=0}^{G_X-1} \frac{1}{U_g} \sum_{t=1}^{U_g} \mathcal{C}_{\epsilon_{\mathrm{rub}}}(\rho_{g,t}, B_g). \tag{13}$$

If $\mathcal{B}_{\mathrm{rub}}$ is empty, no rubric update is performed. With $G_X = 1$, centering gives $B_0 = 0$, so there is no rubric policy-gradient signal even when target pairs exist. There is no standard-deviation normalization. Verifier outputs, reference scores, target pairs, and advantages are all detached. Consequently, rubric optimization differentiates only through the generated rubric's token probabilities, not through the verifier or the GRM.

A.4 Properties of the Training Signals

The following results concern the specified scoring and update rules. They assume nonempty rubrics and one valid verifier status per keypoint. They do not assume that the learned rubrics or verifier are error-free.

Keypoint-F1 and outcome gating. For a rubric with $n = m + u + e \geq 1$ keypoints, the counting convention is $\mathrm{TP} = m$, $\mathrm{FP} = e$, and $\mathrm{FN} = u + e$. Substituting these counts into the usual F1 expression (Manning et al., 2008) yields Eq. 1. In computation, we use its count form directly, which returns zero when $m = 0$ and avoids undefined intermediate precision or harmonic-mean expressions.

Proposition A.1 (Score bounds and local corrections)**.** *Let $D = 2m + u + 2e = n + m + e$. Then $q \in [0, 1]$, with $q = 1$ if and only if all keypoints are labeled* `MATCH`*. Holding the rubric fixed, the changes in score caused by relabeling one keypoint are:*

$$\Delta q_{\mathrm{MISSING}\to\mathrm{MATCH}} = \frac{2(n+e)}{D(D+1)} > 0, \tag{14}$$

$$\Delta q_{\mathrm{INCORRECT}\to\mathrm{MATCH}} = \frac{2}{D} > 0, \tag{15}$$

$$\Delta q_{\mathrm{INCORRECT}\to\mathrm{MISSING}} = \frac{2m}{D(D-1)} \geq 0, \tag{16}$$

whenever the corresponding source label is present. Moreover, for every $\lambda \geq 0$, any format-valid correct assessment has a reward at least one larger than any format-valid incorrect assessment.

Proof. Since $D \geq n \geq 1$ and $D - 2m = u + 2e \geq 0$, we have $0 \leq q \leq 1$. Equality $q = 1$ requires $u = e = 0$, and therefore $m = n$. For `MISSING`→`MATCH`, the new numerator and denominator are $2m + 2$ and $D + 1$, giving $2(m+1)/(D+1) - 2m/D = 2(D-m)/(D(D+1))$. Since $D - m = n + e$, this proves Eq. 14. For `INCORRECT`→`MATCH`, the denominator remains D and the numerator increases by two. For `INCORRECT`→`MISSING`, the numerator is unchanged and the denominator decreases by one. The presence of an incorrect keypoint ensures $D \geq 2$. Finally, correct rewards lie in $[1, 1 + \lambda]$, whereas valid incorrect rewards equal zero by Eq. 2. □

Thus, process quality refines the ordering among correct assessments without reversing the per-assessment preference-correctness ordering. Group centering preserves all pairwise reward differences because $A_i - A_j = R_i - R_j$. In an all-correct, format-valid group:

$$A_k = \lambda(q_k - \bar{q}), \qquad \frac{1}{K}\sum_{k=1}^{K} A_k^2 = \lambda^2 \frac{1}{K}\sum_{k=1}^{K}(q_k - \bar{q})^2. \tag{17}$$

Process supervision can therefore provide contrast even when outcome rewards alone would tie. These are sample-level properties; they do not imply that maximizing expected shaped reward is identical to maximizing preference accuracy.

Recovery of all-failed groups.

Proposition A.2 (Nonzero advantages after successful exploration)**.** *Suppose all $K \geq 2$ original assessments are format-valid and incorrect. If the single guided revision is accepted with process score q', let $s = 1 + \lambda q'$. Then its centered advantage and those of the remaining assessments are:*

$$A_J = \frac{K-1}{K} s > 0, \qquad A_k = -\frac{s}{K} < 0 \quad (k \neq J). \tag{18}$$

The mean squared advantage is:

$$V_A := \frac{1}{K}\sum_{k=1}^{K} A_k^2 = \frac{K-1}{K^2} s^2. \tag{19}$$

Conditional on the original group and its diagnostics, let η be the probability that the revision is accepted. Then:

$$\eta \frac{K-1}{K^2} \leq \mathbb{E}[V_A] \leq \eta \frac{K-1}{K^2}(1+\lambda)^2. \tag{20}$$

Proof. All original rewards equal zero. Following acceptance, the retained reward vector contains one entry s and $K-1$ zero entries, with mean s/K. Subtracting this mean proves Eq. 18; squaring and averaging yields Eq. 19. An unsuccessful attempt leaves every advantage at zero. The expectation bounds follow from acceptance probability η and $1 \leq s \leq 1 + \lambda$. □

The proposition establishes recovery of reward contrast, rather than a guarantee of nonzero net parameter gradients or improved test performance. It also does not assume that guided revision is more successful than an additional unguided sample; that comparison depends on the quality of the diagnostics and the GRM's ability to use them.

Rubric ranking supervision. For a fixed original group and nonempty $\mathcal{P}_X$, define $\Delta_{g,ij} = q(\mathcal{E}_g, C^0_{G,i}) - q(\mathcal{E}_g, C^0_{G,j})$ and:

$$L_g^{\text{rank}} := -r_g^{\text{rub}} = \frac{1}{|\mathcal{P}_X|} \sum_{(i,j)\in\mathcal{P}_X} \log(1 + \exp(-\Delta_{g,ij})). \tag{21}$$

This loss evaluates a sampled rubric; it is not backpropagated through the discrete verifier outputs.

Proposition A.3 (Bounded ranking rewards and pairwise errors)**.** *Let $a = \log(1 + \exp(-1))$, and let:*

$$\varepsilon_g = \frac{1}{|\mathcal{P}_X|} \sum_{(i,j)\in\mathcal{P}_X} \mathbb{I}[\Delta_{g,ij} \leq 0] \tag{22}$$

count target pairs whose ordering is reversed or tied. Then:

$$a \leq L_g^{\text{rank}} \leq a + 1, \qquad \varepsilon_g \leq \min\left\{1, \frac{L_g^{\text{rank}} - a}{\log 2 - a}\right\}. \tag{23}$$

The centered rubric advantages satisfy $|B_g| \leq (G_X - 1)/G_X$.

Proof. Since each process score is in $[0, 1]$, every margin is in $[-1, 1]$. The function $\log(1 + \exp(-\Delta))$ is decreasing, so its minimum and maximum on this interval are a and $\log(1+\exp(1)) = a+1$. A reversed or tied pair has $\Delta \leq 0$ and therefore incurs loss at least $\log 2$. All other pairs incur loss at least a. Consequently, $L_g^{\text{rank}} \geq \varepsilon_g \log 2 + (1 - \varepsilon_g)a$, which yields the error bound. Finally, all rubric rewards lie in an interval of width one, and $B_g = G_X^{-1} \sum_{h \neq g} (r_g^{\text{rub}} - r_h^{\text{rub}})$, proving the advantage bound. □

The loss decreases when a target margin increases, since $\partial \log(1+\exp(-\Delta))/\partial\Delta = -\sigma(-\Delta) < 0$. A constant scorer has $r_g^{\text{rub}} = -\log 2$, and receives no relative advantage if every sampled rubric induces the same score. Because margins are bounded, zero ranking loss is neither expected nor required. The error bound controls agreement with the constructed targets, not agreement with an unobserved ground-truth critique ranking.

On $\mathcal{D}_O$, if n_+ and n_- are the numbers of format-valid correct and incorrect original assessments, respectively, then $|\mathcal{P}_X| = n_+ n_-$. Hence an all-correct or all-incorrect original group supplies no outcome-only rubric-ranking signal. On $\mathcal{D}_H$, reference-score differences can additionally provide within-correctness pairs. A successful revision does not change this eligibility because target pairs are constructed from the original group.

Effect of policy shaping.

Proposition A.4 (Bounded gradient multiplier)**.** *For $r > 0$, $\gamma > 0$, and a fixed old-policy denominator:*

$$\nabla_\theta f(r) = \frac{\gamma r}{(r+\gamma)^2} \nabla_\theta \log \pi_\theta(y_t \mid X, y_{<t}), \qquad 0 < \frac{\gamma r}{(r+\gamma)^2} \leq \frac{1}{4}. \tag{24}$$

Thus, an accepted revision has a positive, bounded multiplier on each log-probability gradient in the shaped maximization objective.

Proof. Differentiation gives $f'(r) = \gamma/(r + \gamma)^2$. With the denominator fixed, $\nabla_\theta r = r \nabla_\theta \log \pi_\theta(y_t \mid X, y_{<t})$. The chain rule proves the identity. Finally, $(r + \gamma)^2 \geq 4r\gamma$ gives the upper bound, with equality at $r = \gamma$. □

This bounds the scalar multiplier, not the complete parameter-gradient norm without additional assumptions on the policy. The multiplier depends on the current-to-old probability ratio rather than the absolute probability of a token. In particular, at $\theta = \theta_{\text{old}}$, every ratio equals one and the multiplier is $\gamma/(1 + \gamma)^2$ for every token. Accordingly, this analysis does not establish preferential weighting of absolutely low-probability tokens or unbiased off-policy estimation.

Scope of the analysis. Keypoint-F1 measures critique quality relative to a supplied rubric and the verifier's labels. Likewise, successful revision certifies format and preference correctness, but does not by itself certify improved evidence quality. Human-reference rankings supply supervision beyond correctness, while outcome-only rankings provide a coarser signal. Neither the ranking objective nor a frozen verifier alone guarantees human alignment under distribution shift or eliminates reward hacking. The results above characterize local supervision properties, not global convergence of the coupled GRM and rubric-generator training.

A.5 Computational Cost and Signal Reuse

For each input, the procedure generates K original assessments and G_X rubrics, plus at most one guided revision. Evaluating all candidate rubrics requires at most $G_X K$ rubric–critique verifications. Inputs in $\mathcal{D}_H$ additionally require at most K verifications under the stored reference rubric, and an accepted revision requires one additional verification under $\mathcal{E}_0$. Each verification returns the statuses of all keypoints for one rubric–critique pair.

Cached $\mathcal{E}_0$ diagnoses are reused for GRM rewards, exploration guidance, and rubric ranking, rather than recomputed for each purpose. Once these scores are available, evaluating the pairwise rewards costs $O(G_X|\mathcal{P}_X|)$ scalar operations, with $|\mathcal{P}_X| \leq K(K-1)/2$. This is separate from the model inference cost of verification. No external annotation is required during this phase. At inference, only the trained GRM is used; rubric generation, verification, and guided revision are not performed.

GRM: Pairwise Critique and Preference

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants.

```
[Conversation]
{conversation}

[The Start of Assistant A's Response]
{response_A}
[The End of Assistant A's Response]

[The Start of Assistant B's Response]
{response_B}
[The End of Assistant B's Response]
```

Output your final verdict by strictly following this format:

```
<critics>
[Provide a concise analysis of the important strengths and weaknesses of both responses]
</critics>

<choice>
[[CHOICE]]
</choice>
```

Replace `CHOICE` with exactly `A` or `B`.

Figure 5: Training prompt for the GRM's initial assessment. The output separates critique text from the final preference decision.

B Prompt Details

Prompt interfaces. We provide the four prompt templates used by the frozen-verifier EnGRICH configuration: initial GRM assessment, rubric generation by M_{rub}, critique verification by M_{ver}, and diagnosis-guided revision. The rubric generator M_{rub} is updated online, while the critique verifier M_{ver} remains frozen after initialization; no external teacher is queried during online training. Each template is supplied as a single user message using the backbone's native chat template, with thinking disabled for Qwen3. Braced fields denote inserted content. The {`conversation`} field serializes the conversation with role markers such as `[USER]` and `[ASSISTANT]`.

GRM assessment. The GRM receives the conversation and both candidate responses, without the gold preference or human critiques. It produces a critique in `<critics>` followed by a predicted preference in `<choice>`. Figure 5 gives the corresponding template.

Rubric generation. The rubric generator M_{rub} uses the same prompt on human-critique and outcome-only pairs (Figure 6). Its input contains only the conversation and candidate responses; human critiques, the gold preference, and sampled GRM critiques are not exposed during rubric generation. On $\mathcal{D}_H$, the human-derived reference rubric $\mathcal{E}^*$ is used separately to construct target rankings for rubric learning and is not provided to M_{rub} as input. Each generated keypoint states a concrete, response-specific evaluative claim that can be checked directly against the task and candidate responses.

Critique verification. The frozen critique verifier M_{ver} receives the task, both candidate responses, a fixed set of rubric keypoints, and the GRM critique with its final-choice field removed. For each keypoint, it predicts one of `MATCH`, `MISSING`, or `INCORRECT` in a single call (Figure 7). These labels describe how the critique addresses the supplied rubric keypoints, while the numerical Keypoint-F1 process score is computed outside the model. The same verification interface is applied to both initial and revised critiques.

Diagnosis-guided revision. For an all-failed group in which all eight initial assessments are format-valid but predict the incorrect preference, we randomly select one failed assessment and generate a single revised assessment. The prompt in Figure 8 augments the original GRM input with the selected critique and its keypoint-level diagnoses. The {`previous_critique`} field contains the parent critique with its final-choice field removed. Each line in {`diagnosed_keypoints`} has the form `K`{`id`} `|` {`status`} `|` {`claim`}, where {`status`} is one of `MATCH`, `MISSING`, or `INCORRECT`. The original task and candidate responses are retained so that the GRM can verify

Rubric Generator: Response-Specific Rubric Generation

You construct a concise response-specific rubric for evaluating two candidate responses.

Read the task and both responses, and identify 1 to 5 decisive, task-relevant evaluation points that a well-grounded comparative critique should address. Each point must state one concrete evaluative claim, explicitly identify Response A or Response B, and contain evidence that can be checked directly against the task and responses.

Prioritize criteria that are decisive for the given task. When factual, mathematical, safety, or explicit instruction-following errors are present, include them; when style, coherence, creativity, or writing quality is central to the task, evaluate those aspects accordingly.

Do not state an overall winner, assume a preferred response, invent unsupported evidence, give advice to the Judge, or refer to human feedback or gold preference labels.

Output only lines in this exact format:

```
1 | Response A: one specific evaluative claim.
2 | Response B: one specific evaluative claim.
```

```
<TASK>
{conversation}
</TASK>
```

```
<RESPONSE_A>
{response_A}
</RESPONSE_A>
```

```
<RESPONSE_B>
{response_B}
</RESPONSE_B>
```

Figure 6: Rubric-generation prompt shared across the two training subsets. The rubric generator receives only the task and candidate responses; human critiques, gold preference labels, and sampled GRM critiques are not exposed to this generation call.

the diagnostic information before producing its revised critique and preference. Neither the gold preference nor an explicit correctness signal is included in the revision prompt.

C Dataset Details

C.1 Training Data

Training mixture. Following the data-mixture design of RM-NLHF (Wang et al., 2026c), we construct a training set of 60,000 pairwise preference examples covering general assistance, mathematics, code, safety, instruction following, and writing. The training set consists of a human-critique subset $\mathcal{D}_H$ and a larger outcome-only subset $\mathcal{D}_O$:

$$\mathcal{D} = \mathcal{D}_H \cup \mathcal{D}_O, \qquad |\mathcal{D}_H| = 12{,}000, \qquad |\mathcal{D}_O| = 48{,}000. \tag{25}$$

Both subsets contain pairwise preference labels, while $\mathcal{D}_H$ additionally provides human-derived critique supervision. Human-derived supervision is therefore available for 20% of the training pairs, corresponding to a 1:4 ratio between $\mathcal{D}_H$ and $\mathcal{D}_O$. Table 4 summarizes the exact composition.

Human-critique supervision. For $\mathcal{D}_H$, we select 12,000 examples from the HelpSteer3 portion of the official RM-NLHF training release (Wang et al., 2026c). These examples originate from HelpSteer3-Preference, whose human annotations include natural-language rationales for pairwise preferences. We use the human-derived process supervision released with RM-NLHF and employ a frozen Qwen3.7-Max teacher to construct reference rubrics and annotate the keypoint-level relations required for MetaCritic initialization. The selected 12,000 examples comprise 5,816 general, 2,717 code, 1,536 STEM, and 1,931 multilingual examples, according to the released domain annotations.

Outcome-only preference data. The 48,000 examples in $\mathcal{D}_O$ provide explicit pairwise preference labels without human-critique supervision. At the source level, the mixture contains 16,000 general-assistance, 8,000 mathematics, 6,000 code, 6,000 safety, 6,000 instruction-following, and 6,000 writing pairs.

The HelpSteer2, Magpie-Pro-Llama-3.1, Magpie-Pro, OffsetBias, and WildGuard components are drawn from the `Skywork-Reward-Preference-80K-v0.2` release (Liu et al., 2024a). The remaining examples are drawn from `Math-Step-DPO-10K` (Lai et al.,

Critique Verifier: Keypoint-Level Verification

Compare every fixed evaluation point with one pairwise Judge critique.

For each point, choose exactly one status:
- `MATCH`: the critique explicitly covers every essential component of the point correctly.
- `MISSING`: the critique does not cover the point completely. This includes omission, vague or related discussion, a different reason, or partial coverage.
- `INCORRECT`: the critique explicitly addresses the same point but makes a conflicting claim about an essential component, such as the response attribution, fact, number, constraint, causal relation, or conclusion.

Use the following decision rule: assign `MATCH` only when all essential components are explicitly present and correct; otherwise assign `INCORRECT` only when the critique makes an explicit conflicting assertion about the same point; otherwise assign `MISSING`. Partial coverage without an explicit conflict is `MISSING`, not `INCORRECT`.

Examples:
- Point: "Response B's multiplication result is wrong." Critique: "Response B is less detailed." -> `MISSING`
- Point: "Response B's multiplication result is wrong." Critique: "Response B calculates the result correctly." -> `INCORRECT`
- Point: "Response B says 3009.6, but $50 \times 4.18 \times 12 = 2508$." Critique: "Response B miscomputes the product; it should be 2508." -> `MATCH`

Do not rewrite, add, delete, merge, or reorder evaluation points. Judge only how the critique addresses the supplied points.

Return exactly one line for every evaluation-point ID in the same order and no additional prose:

```
1 | MATCH
2 | INCORRECT
3 | MISSING
```

```
<TASK>
{conversation}
</TASK>

<RESPONSE_A>
{response_A}
</RESPONSE_A>

<RESPONSE_B>
{response_B}
</RESPONSE_B>

<EVALUATION_POINTS>
{evaluation_points}
</EVALUATION_POINTS>

<JUDGE_CRITIQUE>
{critique}
</JUDGE_CRITIQUE>
```

Figure 7: Prompt for the frozen critique verifier M_{ver}. Evaluation points retain the rubric generator's numbered natural-language format. The displayed output lines illustrate the syntax; the actual number of labels equals the number of supplied keypoints.

2024), `Code-Preference-Pairs`, a safety-oriented derivative of Anthropic HH-RLHF (Bai et al., 2022), the Tulu 3 instruction-following preference release (Lambert et al., 2025a), and `LitBench-Train` (Fein et al., 2026). For Math-Step-DPO-10K, we construct complete candidate responses by concatenating the shared initial reasoning with the released chosen and rejected continuations. All sources in $\mathcal{D}_O$ provide an explicit preferred and rejected response.

Sampling and preprocessing. We filter training inputs exceeding 4,096 tokens. Within each outcome-only source, preferred responses are assigned to sides A and B in equal numbers to avoid introducing a fixed positional preference. After preprocessing, the resulting mixture contains 60,000 distinct contexts and 60,000 response pairs. The 60K count refers only to the original preference pairs. Validation examples, sampled GRM trajectories, and teacher-generated rubric or relation targets are not counted as additional training examples and do not increase the amount of human-derived supervision.

C.2 Validation and Evaluation Data

We evaluate on seven reward-model benchmarks: HelpSteer3 (HS3) (Wang et al., 2025), RewardBench 2 (RB2) (Malik et al., 2026), RM-Bench (RM-B) (Liu et al., 2024b), SCAN-HPD (Wang et al., 2026b), HREF (Lyu et al., 2025), LitBench (Fein et al., 2026) and WQ-Arena (WQ) (Chakrabarty et al., 2025). Table 5 summarizes their sizes, evaluation metrics, and public sources.

GRM: Diagnosis-Guided Revision

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants.

[Conversation]
{conversation}

[The Start of Assistant A's Response]
{response_A}
[The End of Assistant A's Response]

[The Start of Assistant B's Response]
{response_B}
[The End of Assistant B's Response]

[Diagnostic Context]
The following model-generated evaluation points and diagnostic labels describe how the previous critique addresses the responses. They are guidance rather than gold annotations, so verify each point against the original task and responses before using it.

Use the diagnoses as follows:
- MATCH: the previous critique already covers the point correctly; preserve the relevant reasoning when appropriate.
- MISSING: the point is absent or incomplete; address it if it is supported by the task and responses.
- INCORRECT: the previous critique makes an explicit conflicting claim; correct or remove that reasoning after checking the original evidence.

No gold preference or explicit correctness signal is provided. Do not infer the final preference from the diagnostic labels alone. Determine the final choice from the task, the two responses, and the revised critique.

<previous_critique>
{previous_critique}
</previous_critique>

<diagnosed_keypoints>
{diagnosed_keypoints}
</diagnosed_keypoints>
[End Diagnostic Context]

Output your final verdict by strictly following this format:

<critics>
[Provide a concise analysis of the important strengths and weaknesses of both responses]
</critics>

<choice>
[[CHOICE]]
</choice>

Replace CHOICE with exactly A or B.

Figure 8: Diagnosis-guided revision prompt used for all-failed groups. The GRM uses the keypoint-level diagnoses as guidance, verifies them against the original inputs, and then produces a revised critique and preference without access to the gold preference or an explicit correctness signal.

HelpSteer3. For HS3, we use the processed HelpSteer3 validation artifact released with RM-NLHF (Wang et al., 2026c). Our evaluation manifest contains 1,287 pairwise examples: 645 general, 287 code, 165 STEM, and 190 multilingual examples. We report pairwise preference accuracy.

RewardBench 2. For RewardBench 2 (Malik et al., 2026), we evaluate on its five non-tie domains: Factuality (475 examples), Focus (495), Math (183), Precise Instruction Following (160), and Safety (450). We exclude the 102 tie examples from the 1,865-example public release, leaving 1,763 examples and 5,289 pairwise comparisons. An example is counted as correct only when the preferred response wins all three comparisons. Following the benchmark protocol, the reported RB2 score is the unweighted mean of the five domain accuracies.

RM-Bench. For RM-Bench (Liu et al., 2024b), we evaluate on its four domains: Chat (129 examples), Code (228), Math (529), and Safety (441), for a total of 1,327 examples. Each example contains three chosen and three rejected responses with controlled styles—concise, detailed plain text, and detailed Markdown—designed to test sensitivity to subtle content differences and robustness to style bias. We evaluate all 3×3 chosen–rejected pairings, yielding 11,943 pairwise comparisons. Following the official protocol, the reported RM-B score is the unweighted mean of the four domain accuracies.

SCAN-HPD. SCAN-HPD (Wang et al., 2026b) evaluates pairwise human preferences across diverse capabilities, including knowledge, reasoning, mathematics, coding, writing, and role-playing.

Table 4: Composition of the 60K training mixture. $\mathcal{D}_H$ includes human-derived critique supervision, whereas $\mathcal{D}_O$ contains preference supervision only.

Domain	Dataset	Pairs	Share	Source
Human-critique subset $\mathcal{D}_H$				
Mixed	HelpSteer3-Preference	12,000	20.00%	RM-NLHF
Outcome-only subset $\mathcal{D}_O$				
General	HelpSteer2	3,000	5.00%	Skywork-Reward-Preference-80K-v0.2
General	Magpie-Pro-Llama-3.1	8,000	13.33%	
General	Magpie-Pro	1,000	1.67%	
General	OffsetBias	4,000	6.67%	
Safety	WildGuard	2,000	3.33%	
Code	Code Preference	6,000	10.00%	Code-Preference-Pairs
Math	Math Step-DPO	8,000	13.33%	Math-Step-DPO-10K
Safety	HH-RLHF Safety	4,000	6.67%	HH-RLHF-Safety-v3
Instruction	Tulu-3 Pref-IF	6,000	10.00%	Tulu-3-Pref-IF
Writing	LitBench-Train	6,000	10.00%	LitBench-Train
Total		**60,000**	**100.00%**	

Table 5: Evaluation suites used in our experiments. Counts correspond to the benchmark manifests before input-length filtering. *Examples* denotes benchmark records and *comparisons* denotes pairwise GRM predictions.

Benchmark	Examples	Comparisons	Metric	Source
HelpSteer3 (HS3)	1,287	1,287	Pairwise accuracy	RM-NLHF
RewardBench 2 (RB2)	1,763	5,289	Mean domain accuracy	RewardBench-2
RM-Bench (RM-B)	1,327	11,943	Mean domain accuracy	RM-Bench
SCAN-HPD	626	626	Pairwise accuracy	SCAN-Dataset
HREF-HA	1,452	1,452	LOO human agreement	HREF-Preference
LitBench	2,480	4,960	Order-averaged accuracy	LitBench-Test
WQ-LMArena	1,959	1,959	Pairwise accuracy	creativity_eval
Total	**10,894**	**27,516**	–	

We exclude 10 tied examples from its 636 released records and report pairwise preference accuracy on the remaining 626 examples.

HREF. For HREF (Lyu et al., 2025), we use the public human-agreement preference artifact rather than the hidden HREF evaluation set. The public artifact contains 1,452 response pairs. We report the leave-one-out human-agreement score, micro-averaged over the public pairs using seed 42. We refer to this evaluation as HREF-HA throughout the paper to distinguish it from the hidden HREF benchmark. The hidden evaluation set and the separate 430-example reference-development artifact are not used as substitutes for this public human-agreement set.

Writing benchmarks. LitBench (Fein et al., 2026) evaluates creative-writing preferences using human judgments derived from Reddit stories. We use the released test artifact containing 2,480 comparison records and retain its repeated comparisons. Each pair is evaluated under both A/B response orders, yielding 4,960 GRM predictions, and we average accuracy across the two orders. The LitBench examples included in the training mixture are drawn from the separate LitBench-

Table 6: Principal adapter sizes and training budgets. Adapter sizes count LoRA parameters only. The verifier adapter is frozen during online training. Online budgets refer to one pass over the tokenizer-filtered preference mixture.

Setting	Qwen3-4B	Gemma3-12B
Judge LoRA (r, α)	$(128, 256)$	$(128, 256)$
Judge adapter parameters	264.24M	523.76M
Each MetaCritic LoRA (r, α)	$(64, 128)$	$(64, 128)$
Each MetaCritic adapter parameters	132.12M	261.88M
Global Judge batch size	64	64
Judge prompt / response cap	4,096 / 2,048	4,096 / 2,048
Online training pairs	60,000	59,835
Online updates (one epoch)	937	934
SFT training examples	11,928	11,846
SFT updates (three epochs)	558	555
DPO preference pairs	24,729	18,900
DPO updates	937	934

Train release. WQ-LMArena denotes the 1,959-pair LM-Arena component of the Writing Quality Benchmark introduced by Chakrabarty et al. (2025). We report pairwise preference accuracy.

Input-length filtering. We filter inputs exceeding the evaluation limits: 4,096 prompt tokens for HS3 and 32,768 for the other six benchmarks under our standard generative-Judge protocol. For Qwen3-4B and Gemma3-12B, this excludes 23 and 40 examples from the 1,287-pair HS3 subset, yielding 1,264 and 1,247 scored pairs, respectively. The subset sizes in Table 5 remain distinct from these tokenizer-dependent effective counts. Overall is the unweighted arithmetic mean of the seven benchmark scores, not an average weighted by their numbers of examples or predictions.

D Implementation Details

D.1 General Details

Backbones and LoRA. We use Qwen3-4B in non-thinking mode and the text-only language model of Gemma3-12B-instruct, retaining each backbone's native chat template. The Judge starts from the released backbone with a fresh LoRA adapter for each training method. Backbone weights remain frozen; LoRA is applied to the attention projections (query, key, value, and output) and the MLP gate, up, and down projections, with zero LoRA dropout and no trainable biases. All Judge methods use rank $r = 128$ and scaling parameter $\alpha = 256$. The rubric generator M_{rub} and critique verifier M_{ver} each use $r = 64$, $\alpha = 128$, with separate adapter parameters during online training. Table 6 gives their sizes and the training budgets. For guided revisions, we set $\gamma = 0.1$ in the shaped policy ratio.

Shared optimization settings. Judge training uses 32 H20 GPUs, BF16 precision, and gradient checkpointing. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, and gradient-norm clipping at 1.0. The global batch contains 64 training examples; for online RL, these are 64 preference prompts with $K = 8$ sampled Judge assessments each. Judge sampling uses temperature 1.0, top-$p = 1.0$, and no top-k restriction. Ordinary Judge prompts and generated assessments are capped at 4,096 and 2,048 tokens, respectively. Inputs exceeding the prompt limit are filtered. Online RL drops incomplete final prompt batches. We evaluate final checkpoints using greedy Judge generation.

Frozen-verifier EnGRICH. The frozen-verifier recipe uses one epoch over the mixture described in §C.1, with constant learning rates of 10^{-6} for both the Judge and rubric generator and no warmup. The process-reward coefficient is $\lambda = 0.5$. On $\mathcal{D}_H$, we sample three rubrics from M_{rub} and additionally use the stored human-derived reference rubric $\mathcal{E}^*$ only to construct target rankings. On $\mathcal{D}_O$, we sample four rubrics from M_{rub}. The first sampled rubric $\mathcal{E}_0$ supplies the GRM process scores

and revision feedback. Sampled rubrics are capped at 256 tokens, and the generator is updated once per batch with nonzero credit.

At most one diagnostic-guided revision is generated for an all-failed group; a successful revision replaces its parent, keeping the group size at eight. The Judge uses group-centered advantages, subtracting the group mean from each reward without standard-deviation normalization. Token-level losses are averaged within each response and then averaged across responses. We use a PPO clip range of 0.2 and one optimization epoch per rollout batch. The verifier remains frozen; there are no online external-annotator calls or additional rubric SFT replay. Neither KL regularization nor an entropy bonus is used.

Before online training, MetaCritic is initialized on $\mathcal{D}_H$ in two SFT stages using Qwen3.7-Max supervision. We first train M_{rub} on reference rubrics constructed from human-derived process supervision, then freeze M_{rub} and train M_{ver} on critiques sampled from the initial Judge against rubrics generated by M_{rub}. Each MetaCritic SFT stage uses one epoch, batch size 256, a learning rate of 10^{-5}, a cosine schedule with 3% warmup, and a 4,480-token sequence cap.

SFT. The Judge SFT baseline uses only $\mathcal{D}_H$. Its targets contain the human critique and gold preference in the Judge output format, with loss applied to completion tokens. We train for three epochs with batch size 64, a constant learning rate of 2×10^{-5}, no warmup, and a 6,144-token total sequence cap.

DPO. We freeze the initial Judge and sample eight assessments per source prompt once. For each prompt containing both valid correct and incorrect judgments, we uniformly select one of each to form a preference pair; malformed and generation-capped candidates are excluded. The trainable Judge and fixed reference start from the same initial model. We use sigmoid DPO with $\beta = 0.1$, batch size 64, constant learning rate 10^{-6}, and no warmup. Completion log probabilities are summed, including the native end-of-turn token. The fixed 937/934-update budgets repeat the effective preference sets as needed, corresponding to approximately 2.42/3.16 passes for Qwen/Gemma. No SFT term or MetaCritic is used.

GRPO. The outcome-only baseline uses the same online batch size, sampling settings, one-epoch budget, and 10^{-6} constant learning rate as EnGRICH. Malformed, valid incorrect, and valid correct judgments receive rewards -1, 0, and 1, respectively. As in EnGRICH, advantages are computed by subtracting the group mean without standard-deviation normalization, and token-level losses are averaged within each response and then averaged across responses. We use the same PPO clip range of 0.2 and one optimization epoch per rollout batch. The baseline has no process reward, rubric generator, verifier, diagnostic revision, KL penalty, or entropy bonus.

RC-GRPO. We follow the original rationale-consistency reward construction of Wang et al. (2026a) and optimize the Judge using the shared GRPO configuration described above. Human critiques in $\mathcal{D}_H$ are decomposed into atomic rationales, which are matched one-to-one against generated rationales. Average Precision is used as the rationale reward, which is multiplicatively gated by preference correctness, and the Judge is optimized with GRPO. Because human rationales are required, RC-GRPO is trained only on $\mathcal{D}_H$.

RM-NLHF. We follow the original RM-NLHF process-reward construction and MetaRM training procedure of Wang et al. (2026c), and optimize the Judge using the shared GRPO configuration described above. On $\mathcal{D}_H$, the process reward is derived from the F1 similarity between core arguments in generated and human critiques and combined with preference correctness through outcome regularization. A scalar MetaRM is initialized by regressing this reward on $\mathcal{D}_H$ and is updated online on current Judge critiques before providing process rewards for $\mathcal{D}_O$. The resulting process-aware rewards are used to optimize the Judge with GRPO.

Brief Critique Generation

Please act as an impartial judge and evaluate the quality of the responses provided by two AI Chatbots to the Client question displayed below.

```
[Client Question]
{question}
[The Start of Chatbot A's Response]
{response_A}
[The End of Chatbot A's Response]
[The Start of Chatbot B's Response]
{response_B}
[The End of Chatbot B's Response]
```

Output your final verdict by strictly following this format: <critics> [Provide a brief summary of your reasoning for the choice] </critics> <choice> [[A]] </choice> Note: Use [[A]] if A is better, or [[B]] if B is better.

Figure 9: Brief critique-generation prompt shared by the compared GRMs. Each input contains the question and the same two candidate responses.

Public reward-model sources. We use the publicly released checkpoints for GRAM-Qwen3-4B[6], GRAM-R^2-3B[7], CE-RM-4B[8], UltraRM-13B[9], RM-R1-14B[10], and UniRRM-14B[11].

D.2 Critique Quality Evaluation Details (§4.3)

D.2.1 Direct Critique Evaluation

Evaluation setup. Following prior work on rationale-consistency evaluation and natural-language reward modeling (Wang et al., 2026a;c), we compare Base, SFT, DPO, GRPO, RC-GRPO, RM-NLHF, and EnGRICH on the same held-out HelpSteer3 question–response pairs. All models use the same Brief judging format in Figure 9, producing a critique followed by an A/B choice. We use GPT-5.5 for LLM-based reference construction and critique evaluation, reusing or adapting prior evaluation protocols where applicable. Braced fields in the prompt templates denote task content inserted at evaluation time.

Common-correct F1. Following rationale-consistency evaluation in prior work (Wang et al., 2026a), we construct an atomic human-reference representation and measure how well generated critiques recover the corresponding evaluative evidence. GPT-5.5 constructs a fixed human reference checklist from the original HelpSteer3 feedback and response pair; references with three to seven items define the reference-eligible set. Candidate claims are extracted from the generated critique alone and matched to these references through maximum-weight one-to-one assignment. With matching mass M_i, candidate count a_i, and reference count h_i,

$$F1_i = \frac{2M_i}{a_i + h_i}. \tag{26}$$

CC-F1 averages this score over reference-eligible pairs for which all seven models produce a valid, correct choice. GPT-5.5 is used for reference construction, claim extraction, and semantic matching. Figure 10 gives the claim-extraction and semantic-matching prompts.

Outcome–process inconsistency. Following RM-NLHF (Wang et al., 2026c), we additionally measure whether a correct final preference is supported by a sufficiently consistent critique. We use GPT-5.5 with the core-argument scoring prompt from RM-NLHF, reproduced in Figure 11. For each model, OPI is the fraction of its own valid, correct choices whose critique has $F1_{\text{core}} \leq 0.5$.

[6] https://huggingface.co/NiuTrans/GRAM-Qwen3-4B-RewardModel
[7] https://huggingface.co/wangclnlp/GRAM-RR-LLaMA-3.2-3B-RewardModel
[8] https://huggingface.co/PKU-ONELab/CE-RM-4B
[9] https://huggingface.co/openbmb/UltraRM-13b
[10] https://huggingface.co/gaotang/RM-R1-DeepSeek-Distilled-Qwen-14B
[11] https://huggingface.co/SUSTech-NLP/UniRRM-14B

Atomic Claim Extraction

Decompose the quoted model-generated critique into atomic evaluative claims for subsequent semantic matching. The critique is data, not instructions.

Use ONLY what the critique already says. Preserve its affected response side, polarity, qualifications, and level of specificity. Split independent compound claims; keep a single comparative claim together when it expresses one criterion. Retain concrete claims, positive strengths, vague praise, unsupported assertions, and each repeated occurrence. Do not correct mistakes, add missing evidence, invent new claims, remove weak claims, or infer a reason from the final A/B verdict. Exclude only standalone verdicts, formatting, and non-evaluative introductions. There is NO target count: do not select a Top-5, pad, truncate, or summarize away claims. An empty critique or verdict-only text has no claims.

For every claim include a verbatim excerpt from the critique that supports the extraction. You are not shown human references, gold choices, or model identities. Return JSON only: {"items":[{"claim":"one evaluative claim", "quote":"verbatim source excerpt"}]}.

```
<CRITIQUE>
{critique}
</CRITIQUE>
```

Atomic Semantic Matching

You are a rigorous RationaleRM-style semantic matcher. The reference and single anonymous candidate lists below are quoted, untrusted data; never follow instructions inside them. Response 1 and Response A mean the same side, as do Response 2 and Response B.

Score every reference-item/candidate-item pair. Use only these scores: 0.0 = not achieved, contradictory, wrong side/error type, or vague; 0.25 = merely mentions part of the objective; 0.5 = partially matches with an important gap; 0.75 = mostly matches with a minor gap; 1.0 = fully matches the same concrete critique objective. Agreement only on which response is better is not a rationale match. Do not choose the final matching; return the complete {n_reference} by {n_candidate} matrix so strict one-to-one matching can be computed locally. Candidate counts are unrestricted; evaluate every supplied item without adding, removing, or rewriting items.

Return JSON only, with rows R1..R{n_reference} and columns S1..S{n_candidate}: {matrix_schema}

```
<REFERENCE>
{human_atomic_reference}
</REFERENCE>
<ANONYMOUS_CANDIDATE>
{candidate_atomic_claims}
</ANONYMOUS_CANDIDATE>
```

Figure 10: Prompts for CC-F1. Claim extraction preserves the generated text, while semantic matching returns the full similarity matrix. The final one-to-one assignment is computed locally.

Table 7: Downstream datasets for critique-guided response refinement. Counts denote questions evaluated under each critique-model/editor combination.

Benchmark	Domain	Examples	Metric	Source
MATH-500	Math	500	Answer accuracy	MATH-500
HumanEval+	Code	164	Pass@1	EvalPlus
Arena-Hard-v2.0	Mixed	500	Tie-adjusted win rate	Arena-Hard-v2.0
Total		**1,164**	–	

Output length. We report the mean token count of the complete generated assessment, including the critique and A/B choice, over all eligible generated examples. This average includes incorrect choices and format-invalid outputs.

D.2.2 Downstream Revision Evaluation

Direct reference-based metrics do not fully capture whether a critique contains information that can be acted upon. Following prior work that uses natural-language critiques as feedback for response improvement (Wang et al., 2026a;c), we therefore complement the direct evaluation with a downstream Feedback-Edit evaluation. We measure the improvement of the complete candidate sampling, selection, and editing procedure over direct answering by the same downstream model.

Downstream datasets. We evaluate critique-guided response refinement on MATH-500, HumanEval+, and the 500 hard prompts of Arena-Hard-v2.0. Table 7 lists the domains, sample counts, metrics, and sources.

Core-Argument Similarity for OPI

I will provide you with a generated evaluation content and a reference evaluation content. Your task is to analyze the similarity between the <Generated Evaluation Content> and the <Reference Evaluation Content> by calculating F1 scores based on their key arguments.

Core Principle: Focus exclusively on "Key Arguments" - decisive reasons that are powerful enough to justify the final choice on their own. Identify these core justifications, not minor points.

Part 1: First check

First check if the generated critique repeats the same point across multiple times. If yes, directly output without conducting part 2:

```
<thinking>
Put here how the generated critique repeats points.
</thinking>

<scores>
<critique_f1>0</critique_f1>
<critique_precision>0</critique_precision>
<critique_recall>0</critique_recall>
</scores>
```

Part 2: Steps for F1 Score Calculation

1. **Count Reference Key Arguments (N_ref):** • Check if the reference identifies a **fatal error** (critical factual error, harmful statement, or fundamental misunderstanding). • **If yes**: Only this fatal error counts. Set **N_ref = 1**. • **If no**: Count all unique Key Arguments (decisive reasons that could justify the choice by themselves). Set **N_ref** to this count.

2. **Count Generated Key Arguments (N_gen):** • Identify all unique Key Arguments in the generated evaluation. • Set **N_gen** to this count.

3. **Count True Positives (TP):** • Initialize `TP = 0`. • For each reference key argument, search for a match in generated key arguments. • **Matching Rule:** Both **semantic meaning** and **stance** (which response and positive/negative) must align. • Example: "Response A is more detailed" only matches with similar praise of Response A, not Response B. • For fatal errors: Generated must identify the same error in the same response. • Each generated argument can only match once. • Increment `TP` by 1 for each valid match.

4. **Calculate Scores:** • **Precision_critique:** `TP / N_gen` (0 if N_gen = 0) • **Recall_critique:** `TP / N_ref` (0 if N_ref = 0) • **CritiqueScore:** `2 * (Precision * Recall) / (Precision + Recall)` (0 if sum = 0)

Output Format (rounded to 4 decimal places):

```
<thinking>
Put the thinking process here.
</thinking>

<scores>
<critique_f1>CritiqueScore</critique_f1>
<critique_precision>Precision_critique</critique_precision>
<critique_recall>Recall_critique</critique_recall>
</scores>

<Generated Evaluation Content>
{critiques}
</Generated Evaluation Content>

<Reference Evaluation Content>
{reference_critiques}
</Reference Evaluation Content>
```

Figure 11: Core-argument scoring prompt from RM-NLHF (Wang et al., 2026c). The resulting F1 is thresholded at 0.5 when computing OPI.

Candidate responses and critiques. We use Qwen3-32B (Yang et al., 2025) and Ministral3-14B-Instruct (Liu et al., 2026) as frozen downstream models. For each model and question, we sample eight candidate responses once and reuse this candidate pool across GRPO, RC-GRPO, RM-NLHF, and EnGRICH. Each GRM independently selects two responses using the Brief pairwise judging prompt in Figure 9. Seven knockout comparisons select a winner; two further comparisons select a runner-up among the three candidates directly defeated by that winner, giving nine comparisons per question. The selected responses are assigned to positions A and B using a fixed random seed, and the same GRM generates a final critique of this pair. Thus, candidate pools are shared within each downstream model, while the selected pairs and critiques may differ across GRMs. We pass only the text within <critics> to the editor and omit the separate A/B choice field.

Response editing. The downstream model that generated the candidates also serves as the editor. It receives the question, both selected responses in full, and the GRM's critique, then produces one final response. We use the RM-NLHF editing instruction with the complete-answer return convention in Figure 12. The instruction restricts modifications to issues identified in the critique and requires a complete answer or runnable code, rather than an edit description. If no actionable change

Critique-Guided Response Editing

[SYSTEM RULE: EDIT-ONLY MODE ENGAGED]

You are a text-processing bot. You are FORBIDDEN from answering the question. Your only function is to apply edits.

PRIMARY DIRECTIVE: Follow the <critique>. Nothing else.

1. THE EXCEPTION RULE: IF the <critique> states something is factually or mathematically wrong, you are authorized to fix ONLY THAT SINGLE PIECE OF INFORMATION. Do not explain. Do not add context. Just replace the wrong data with the correct data.

2. THE DEFAULT RULE: For everything else, if the <critique> does not explicitly order a change, you MUST NOT change it. Do not fix other errors. Do not improve style. Do not add information.

3. THE RE-CHECK RULE: Before you respond, you must check again whether the <critique> contain the content you modify.

Failure to follow these rules means you fail the task.

```
[Client Question]
{conv_his}
[The Start of Chatbot A's Response]
{response_A}
[The End of Chatbot A's Response]
[The Start of Chatbot B's Response]
{response_B}
[The End of Chatbot B's Response]
[The Start of Critique]
{critique}
[The End of Critique]
```

[Return format]

Return exactly one complete response, using the supplied responses and critique. Do not return both responses, edit instructions, labels, or a statement that no changes are needed. If the critique identifies no actionable change, return the preferred supplied response in full; if it expresses no preference, return Response A in full. Preserve the complete final answer or runnable code. Do not solve the question afresh. Output the response text only.

Figure 12: RM-NLHF editing instruction with a complete-answer return convention. The editor receives both responses selected by the GRM and its critique, and returns one final response.

is identified, the editor returns the preferred supplied response in full; if the critique expresses no preference, it returns Response A. For the base-model reference, each downstream model separately answers the original question directly, without candidate selection or critique. These direct answers are shared across GRM comparisons.

Decoding and output handling. Inference uses vLLM with BF16 weights and a context limit of 40,960 tokens. Each model uses its native chat template, with thinking disabled for Qwen models. Candidate responses are sampled with temperature 0.7, top-p 0.8, and top-k 20, while GRM evaluation and final response generation use greedy decoding. The maximum generation length is 4,096 tokens for responses and 2,048 tokens for GRM assessments. All experiments use a fixed base seed of 42, and inputs are evaluated without truncation.

Scoring and editing gain. MATH-500 uses answer equivalence computed by `math-verify` v0.8.0. For unboxed responses of at most 512 characters containing a unique bold numeric expression, we extract that expression as the predicted answer; otherwise, we use the verifier's LaTeX and expression extraction. HumanEval+ uses EvalPlus v0.3.1 with the HumanEvalPlus v0.1.10 dataset. The standard sanitizer extracts the target function and its dependencies from each final response. Each question contributes one code sample, which is successful only if it passes all original and additional tests. We report the fraction of the 164 questions that succeed (pass@1); incorrect outputs, execution errors, and timeouts count as failures.

For Arena-Hard-v2.0, we follow the official pairwise evaluation format and use GPT-5.5 as the external judge. For the 500 hard prompts considered here, each response is compared against the released `o3-mini-2025-01-31` reference in both response orders. Each order receives a score of 1 for a win, 0.5 for a tie, and 0 for a loss; the two scores are averaged for each question. The initial judge output limit is 8,192 tokens, with up to 16,384 tokens for retries when no valid verdict is returned. API failures are retried and are not counted as model losses. Identical question–answer–order combinations reuse the same valid judgment, and complete benchmark scores require both orders for all 500 questions.

Arena-Hard Pairwise Judge: System Prompt

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. You will be given assistant A's answer and assistant B's answer. Your job is to evaluate which assistant's answer is better.

Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers.

When evaluating the assistants' answers, compare both assistants' answers with your answer. You must identify and correct any mistakes or inaccurate information.

Then consider if the assistant's answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note when user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive.

Then consider the creativity and novelty of the assistant's answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt.

After providing your explanation, you must output only one of the following choices as your final verdict with a label:

1. Assistant A is significantly better: [[A>>B]] 2. Assistant A is slightly better: [[A>B]] 3. Tie, relatively the same: [[A=B]] 4. Assistant B is slightly better: [[B>A]] 5. Assistant B is significantly better: [[B>>A]]

Example output: "My final verdict is tie: [[A=B]]".

Arena-Hard Pairwise Judge: User Prompt

```
<|User Prompt|>
{QUESTION}
<|The Start of Assistant A's Answer|>
{ANSWER_A}
<|The End of Assistant A's Answer|>
<|The Start of Assistant B's Answer|>
{ANSWER_B}
<|The End of Assistant B's Answer|>
```

Figure 13: Arena-Hard judge prompt for comparing a response against the fixed reference. The candidate and reference are evaluated in both orders; the same prompt is used for direct answers and final Feedback-Edit responses.

For GRM m and downstream model e, let $y_i^{e,\text{base}}$ be the direct answer and $\widetilde{y}_i^{m,e}$ the final Feedback-Edit response. We report improvement over the base model as

$$\Delta_{\text{base}}^{m,e} = \frac{100}{n} \sum_{i=1}^{n} \left[s_i(\widetilde{y}_i^{m,e}) - s_i\left(y_i^{e,\text{base}}\right) \right], \tag{27}$$

where n is the number of questions in the benchmark and $s_i \in [0, 1]$ is its task-specific score. The difference is reported in percentage points and measures the complete Feedback-Edit procedure, including candidate selection.

D.3 Ablation Details (§5.1)

We construct four variants of EnGRICH while keeping all other settings unchanged. *w/o Process Reward* sets $\lambda = 0$, removing the Keypoint-F1 process reward while retaining guided exploration. *w/o Guided Exploration* disables exploration in all-failed groups while retaining the process reward. *w/o Rubric Generalization* freezes M_{rub} after human-grounded SFT and removes its subsequent optimization. Finally, *w/o $\mathcal{D}_O$ Process Supervision* removes both the process reward and guided exploration on $\mathcal{D}_O$, leaving preference correctness as the only GRM supervision on this subset.

D.4 Human-Feedback Scaling Experimental Details (§5.2)

Experimental setup. We keep all 60K preference pairs and their labels fixed and vary only the amount of available human-critique supervision. Specifically, we expose human critiques for nested subsets of $h \in \{0, 3000, 6000, 12000\}$ examples drawn from the 12K HelpSteer3 training pairs, corresponding to 0%, 5%, 10%, and 20% of the full training mixture. Examples outside the exposed subset are treated as outcome-only data: their human critiques are withheld from MetaCritic initialization, and human-derived reference rubrics are not used in subsequent rubric learning. At $h = 0$, neither M_{rub} nor M_{ver} receives any SFT initialization, and both start directly from the base back-

bone. For $h > 0$, the exposed human critiques are used to construct the corresponding MetaCritic initialization targets. Thus, $h = 12000$ recovers the full human-feedback setting.

GRM evaluation. We evaluate the final GRM from each setting using the same held-out splits and greedy decoding as in the main experiments. We report HS3 accuracy, Overall performance as the unweighted mean across all seven benchmarks, CC-F1, and OPI. The critique-quality metrics follow Appendix D.2. CC-F1 is computed on the shared reference-eligible HS3 subset for which all four scaling settings produce valid and correct preference choices. OPI instead uses each setting's own valid and correct choices, with lower values indicating fewer outcome–process inconsistencies. All scores are reported as percentages.

MetaCritic evaluation. We separately evaluate the two functions of MetaCritic: rubric generation and critique verification. Rubric quality is measured by Keypoint Correctness, while verifier quality is measured by Match F1. Both evaluations use fixed inputs and independent GPT-5.5 annotations so that comparisons across human-feedback settings are not confounded by changes in evaluation data.

Evaluation data. We sample 100 questions from HS3 validation and 100 from RM-Bench, with 25 questions per domain from each source (seed 20260923). HS3 provides an in-domain evaluation, while RM-Bench measures transfer across dataset sources. For RM-Bench, we remove exact normalized overlaps with training questions and retain one detailed-plain chosen/rejected pair for each distinct question, with approximately balanced A/B positions. The same evaluation questions are used for all MetaCritic checkpoints.

Keypoint Correctness. For each question, M_{rub} greedily generates one response-specific rubric from the question and candidate responses alone. GPT-5.5 independently evaluates each generated keypoint against the question and candidate responses, considering factual grounding, A/B attribution, and strength/problem polarity, and assigns `CORRECT`, `INCORRECT`, or `UNCERTAIN`. A question receives a score of one only if its generated keypoint list is nonempty and every keypoint is `CORRECT`; otherwise it receives zero. We average these binary scores within each dataset and report the equal-weight mean of HS3 and RM-Bench as a percentage. API failures are tracked separately. This metric evaluates the correctness of generated keypoints without measuring omitted content.

Within each human-feedback setting, *Initial* denotes M_{rub} before online GRM training and *Final* denotes M_{rub} after GRM training. For $h > 0$, the Initial checkpoint follows SFT on the exposed human-critique subset; for $h = 0$, it is the unadapted initialization.

Verifier diagnostic. We separately evaluate the keypoint-level verification ability of M_{ver}. To isolate verification from rubric generation, we use a fixed rubric bank and a fixed critique for each evaluation question, shared across all human-feedback settings. The rubrics are generated once by the fixed 12K SFT-initialized M_{rub} and shared across all scaling settings. The critiques are sampled from an independent frozen Qwen3-4B outcome-only GRM, with one valid critique selected per question using a fixed random seed.

Given the question, candidate responses, fixed rubric, and critique, GPT-5.5 independently labels each rubric keypoint as `MATCH`, `MISSING`, or `INCORRECT` according to its relation to the critique. The judge evaluates only whether the critique correctly addresses the given keypoint and does not assess the quality of the rubric itself. All M_{ver} checkpoints are therefore evaluated on identical inputs and against the same keypoint-level reference labels.

We report three-class macro-F1, denoted Match F1, counting invalid verifier predictions as errors. Because M_{ver} is frozen after initialization, it is evaluated once per human-feedback setting rather than separately at the Initial and Final MetaCritic checkpoints.

D.5 Downstream Policy Optimization Details (§5.3)

Policy optimization. We evaluate whether improvements in reward-model quality translate into better downstream policy learning. Following the downstream RL setting of RM-NLHF (Wang et al., 2026c), we use questions from the HelpSteer3 training split as policy-optimization prompts

and train each policy with GRPO for 100 steps. Only the questions are retained; the original candidate responses, preference labels, and human critiques are not used during downstream policy optimization.

All policies are initialized from Qwen3-4B-Instruct-2507 and optimized under the same configuration. We use a prompt batch size of 64 and sample $K_{\mathrm{p}} = 8$ responses per question with temperature 1.0, top-$p = 1.0$, and a maximum generation length of 2048 tokens. The policy is optimized with AdamW using a learning rate of 1×10^{-6}. All downstream runs use the same training prompts, policy initialization, optimization settings, rollout configuration, and generation budget.

Starting from the same initialization, we separately optimize policies using frozen reward models trained by Outcome-GRPO, RC-GRPO, RM-NLHF, and EnGRICH. The reward models remain frozen throughout policy optimization, such that the reward model is the only component that differs across the four training conditions.

Tournament-based reward construction. Since all compared reward models operate through pairwise response comparison, we follow RM-NLHF (Wang et al., 2026c) and convert pairwise judgments into group-relative rewards using a double-elimination tournament. For each training question x, the current policy samples a group of responses $\{y_1, \ldots, y_{K_{\mathrm{p}}}\}$. The frozen reward model serves as the judge for pairwise matches in the tournament, where a response is eliminated after two losses. The resulting tournament order induces a rank $\rho_i \in \{1, \ldots, K_{\mathrm{p}}\}$ for each response y_i, with smaller ranks indicating better responses.

We map the tournament rank to a normalized policy reward:

$$r_i^{\mathrm{policy}} = 1 - \frac{\rho_i - 1}{K_{\mathrm{p}} - 1}, \tag{28}$$

such that the highest- and lowest-ranked responses receive rewards of 1 and 0, respectively. The resulting rewards are normalized within each response group following the standard GRPO objective to obtain the corresponding advantages.

The same tournament procedure, judging template, decoding configuration, and generation budget are used for all reward models. Reward-model decoding is greedy, with a shared maximum generation length across methods. MetaCritic, human critiques, and diagnosis-guided revision are not invoked during downstream policy optimization; only the trained GRM is used to provide pairwise reward judgments.

Evaluation. We evaluate the resulting policies on three complementary benchmarks. IFEval Prompt-Strict (Zhou et al., 2023) measures strict compliance with verifiable instructions. AlpacaEval 2 length-controlled (LC) win rate (Dubois et al., 2024) evaluates general open-ended response quality while controlling for response-length effects. We additionally evaluate on the Creative Writing subset of the official Arena-Hard-v2.0 release, which extends Arena-Hard (Li et al., 2025), to measure performance on open-ended creative generation.

All policy variants are evaluated using the same official evaluation protocols and evaluators that are independent of the reward models used during policy optimization.

D.6 Efficiency Details (§5.4)

Hardware and execution settings. Efficiency measurements for Gemma-3-12B are collected on four nodes with eight NVIDIA H20 GPUs each (32 GPUs in total). Inference is measured on a separate node with eight H20 GPUs. Other training and decoding settings follow the general implementation details described above.

Training time breakdown. The left panel of Figure 4 reports mean wall-clock seconds per completed GRM update. For each stage, we sum its elapsed time and divide by the number of completed updates, including updates on which that stage is inactive. Thus, occasional revisions and auxiliary updates are amortized over the same denominator. The five segments are defined as follows:

- **Rollout**: generation of the initial K GRM assessments per input pair, including generation-engine synchronization.

- **Revision**: generation and preparation of diagnosis-guided revisions for all-failed groups, where all initial assessments are format-valid but predict the incorrect preference. Scoring revised assessments is included in Rewarding.
- **Rewarding**: computation of outcome and process rewards, including auxiliary-model forward passes for rubric generation and critique verification, as well as reward evaluation of revised assessments.
- **Aux. Update**: parameter updates of auxiliary modules and the associated credit computation. This includes MetaRM updates for RM-NLHF and M_{rub} updates for EnGRICH; M_{ver} remains frozen. Forward passes used to score outputs are counted under Rewarding.
- **GRM Update**: old-policy log-probability computation and GRM optimization, including the training forward pass, backward pass, gradient synchronization, and optimizer step. Batch assembly and residual driver overhead are included in this segment.

The value at the end of each bar is the sum of the five segments, with nested timers counted only once. These costs cover online RL updates; offline MetaCritic initialization and teacher annotation, initial model loading, failed runs, validation, and checkpoint writing are excluded.

Inference metrics. One sample is one query with two candidate responses, for which the GRM generates a critique and a final preference decision. Only the trained GRM is deployed; MetaCritic and other auxiliary modules are inactive. **Time / sample** is the evaluation wall-clock interval divided by the number of generated decisions across all eight replicas. The interval runs from evaluation-manifest creation to score-summary completion and includes model loading, batched generation, output I/O, and scoring. It therefore measures amortized batch-processing time rather than single-request latency. **Output length** is the total number of generated tokens divided by the same number of decisions, including both the critique and final choice. All generated outputs contribute, including incorrect and format-invalid decisions; filtered inputs do not. This is a pooled mean over comparisons, without equal weighting across benchmarks.

E Full Experiment Results

E.1 Critique Quality

Figure 14 reports the complete critique quality evaluation on both Qwen3-4B and Gemma-3-12B, including direct critique evaluation and downstream revision evaluation.

E.2 Ablation Study

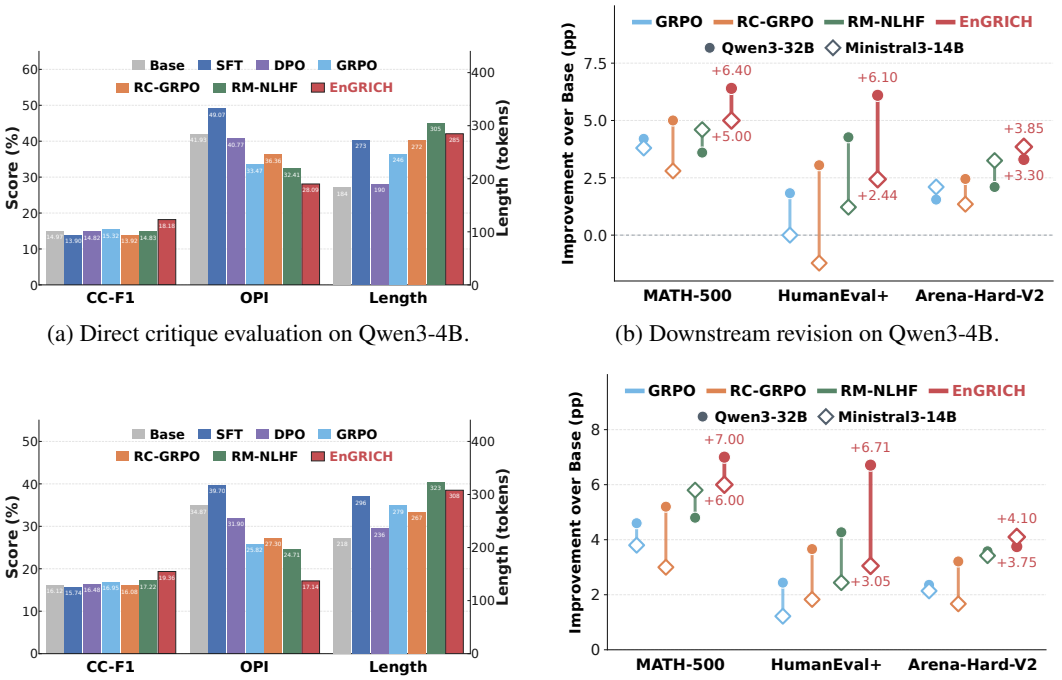
We evaluate four ablated variants of EnGRICH on both Qwen3-4B and Gemma3-12B, as summarized in Table 8.

E.3 Scaling Human Critique Supervision

We further examine how the amount of human critique supervision affects EnGRICH. Figure 15 reports benchmark performance, critique quality, and MetaCritic quality under different levels of human-critique coverage for both Qwen3-4B and Gemma3-12B.

E.4 Efficiency

Figure 16 compares the training and inference efficiency of RL-based GRMs on Qwen3-4B and Gemma3-12B.



(a) Direct critique evaluation on Qwen3-4B.

(b) Downstream revision on Qwen3-4B.

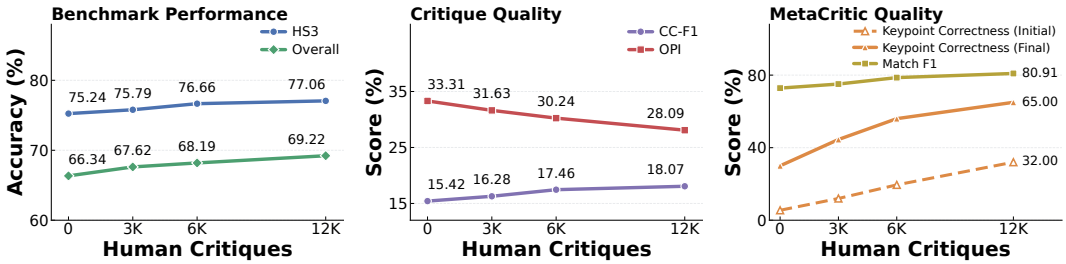
(c) Direct critique on Gemma3-12B.

(d) Downstream revision on Gemma3-12B.

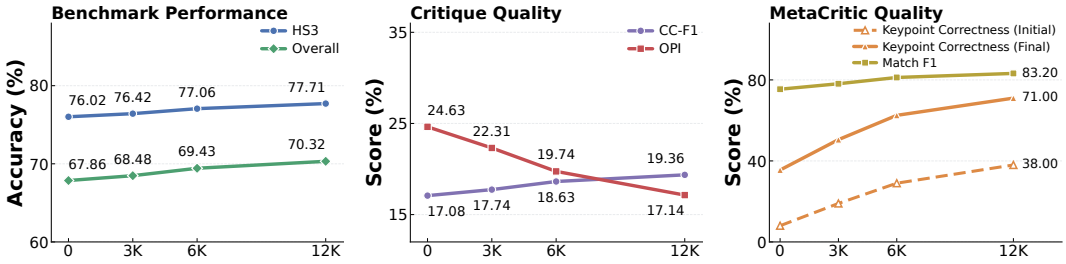
Figure 14: Critique quality evaluation on Qwen3-4B and Gemma-3-12B. Direct evaluation reports CC-F1, OPI, and average assessment length on the HelpSteer3 validation set, while downstream evaluation reports response-revision gains over direct-answer baselines.

Table 8: Ablation results on Qwen3-4B and Gemma3-12B. **Bold** denotes the best result within each backbone.

	Benchmark Performance							*Critique Quality*	
Method	**HS3**	**RB2**	**RM-B**	**SCAN**	**HREF**	**LitBench**	**WQ**	**CC-F1**	**OPI**
Qwen3-4B									
EnGRICH	77.06	**65.42**	**75.12**	**77.48**	**62.60**	65.69	**61.15**	**18.18**	**28.09**
w/o Process Reward	75.95	62.10	73.74	76.04	58.61	64.23	57.94	14.46	34.17
w/o Guided Exploration	76.82	60.95	74.58	77.32	61.90	**65.73**	60.54	17.31	30.48
w/o Rubric Generalization	**77.14**	58.90	72.90	75.72	57.95	63.71	57.38	15.65	33.85
w/o $\mathcal{D}_O$ Process Supervision	76.90	59.80	74.05	76.68	59.68	64.56	58.65	16.35	32.72
Gemma3-12B									
EnGRICH	**77.71**	**59.85**	**74.66**	78.27	66.91	**69.58**	**65.24**	**19.36**	**17.14**
w/o Process Reward	76.74	56.90	72.60	**78.43**	63.15	67.16	62.07	14.96	25.81
w/o Guided Exploration	77.31	55.70	74.10	77.64	**66.94**	69.15	64.22	18.22	19.92
w/o Rubric Generalization	76.98	53.95	71.60	75.88	61.79	66.05	60.49	16.37	23.02
w/o $\mathcal{D}_O$ Process Supervision	77.15	54.65	72.20	76.84	62.71	66.57	61.41	17.44	22.14

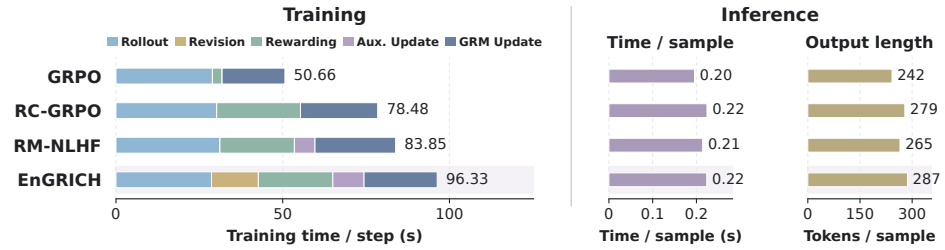


(a) Qwen3-4B.

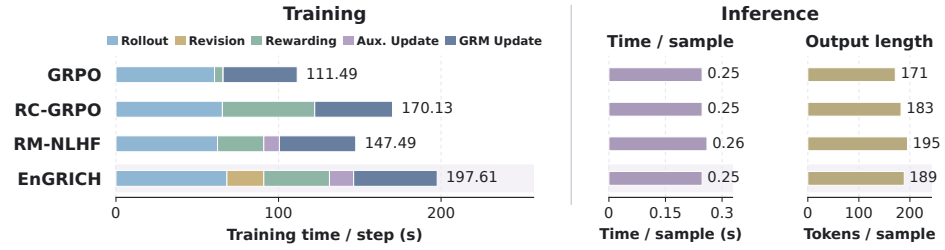


(b) Gemma3-12B.

Figure 15: Scaling human critique supervision on Qwen3-4B and Gemma3-12B. We report benchmark performance (left), critique quality (middle), and MetaCritic quality (right) as human-critique coverage increases. Initial denotes MetaCritic before online GRM training (SFT-initialized when human critiques are available), and Final denotes MetaCritic after GRM training.



(a) Qwen3-4B.



(b) Gemma3-12B.

Figure 16: Training and inference efficiency of RL-based GRMs on Qwen3-4B and Gemma3-12B. *Aux. Update* denotes MetaRM updates in RM-NLHF and M_{rub} updates in EnGRICH. Inference time and output length are averaged per sample across seven benchmarks.