SETGRPO: CALIBRATING GRADER-INDUCED POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubric-based reinforcement learning uses fine-grained grader feedback to train large language models on open-ended tasks, but grader randomness can alter GRPO updates in ways that score discrepancies alone cannot characterize. Because GRPO jointly normalizes rewards within each response group and combines the resulting training coefficients with response-level gradients, a scoring discrepancy can shift the coefficients of the entire group and induce policy-dependent update changes. Selective execution further affects the direction-reversal risk of retained updates. To address this, we propose **Set**-Valued **G**roup **R**elative **P**olicy **O**ptimization (**SetGRPO**). It comprises three modules: *(a) Joint Reward Set Calibration* constructs a joint uncertainty set for the training coefficients; *(b) Geometry-Guided Safe Recovery* maps candidate coefficients into the current policy's local update geometry and determines the largest safe recovery ratio for each response group; and *(c) Selective Risk-Gated Optimization* selects a global threshold through independent risk calibration to control the conditional direction-reversal risk of accepted updates while preserving the original GRPO objective. Under exchangeability, SetGRPO provides finite-sample marginal coverage for the joint coefficient set and simultaneous risk control across candidate thresholds. On ten open-ended benchmarks spanning three domains, SetGRPO achieves the highest macro average among all compared training methods on both Qwen3.5 and Llama3.2 model families. The code is available at https://anonymous.4open.science/r/SetGRPO-73EB.

1 INTRODUCTION

Nowadays, Rubric-based Reinforcement Learning (Rubric-based RL) has emerged as an important training paradigm for improving the performance of large language models on open-ended generation tasks (Li et al., 2026; Zhou et al., 2026; Kim et al., 2024; Ye et al., 2024). It uses fine-grained rubrics and LLM judges to evaluate open-ended responses across dimensions such as relevance, completeness, factuality, and style (Li et al., 2026; Zhou et al., 2026; Kim et al., 2024; Ye et al., 2024; Liu et al., 2023; Zheng et al., 2023). These dimension-level scores are aggregated into response-level rewards that serve as training signals for reinforcement learning (Ouyang et al., 2022; Stiennon et al., 2020), thereby accommodating grader's score discrepancies during training and enabling optimization of open-ended generation tasks.

Existing rubric-based RL studies build on GRPO (Shao et al., 2024; Guo et al., 2025) and examine reward over-optimization (Gao et al., 2023; Lambert et al., 2025), LLM-judge consistency and bias (Zheng et al., 2023; Wang et al., 2024a; Liu et al., 2023), fine-grained evaluation (Kim et al., 2024; Ye et al., 2024; Liu et al., 2023), rubric constructions and rubric-based training for open-ended tasks (Li et al., 2026; Zhou et al., 2026). However, they largely remain at the scoring level and do not explain how score discrepancies propagate through GRPO's within-group normalization and gradient aggregation to alter policy updates. Specifically, for a given response group, GRPO derives the training coefficients from the shared mean and scale of the group rewards and combines them with the response-level gradients of each response under the current policy to form a local policy update. We therefore conjecture that *a change in a single reward alters multiple training coefficients within the group through the shared statistics, and that different policy snapshots induce different update outcomes*.

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.

To verify this conjecture, we score fixed response groups twice with the same grader and compare the two scoring results pairwise (detailed in Section 2). As shown in Figure 1(a), we first compare score discrepancies with the coefficient residuals obtained after within-group normalization, testing whether shared statistics propagate score discrepancies to the entire group of coefficients. As shown in Figure 1(b), we further compare the local updates of the same response group under different policy snapshots, testing whether the effect of coefficient discrepancies depends on the gradient geometry of the current policy. The results show that shared normalization propagates score discrepancies to multiple coefficients within a group, whereas similar coefficient shifts can induce different update directions under different policy geometries, thereby changing which groups are retained by selective execution. These findings support our conjecture and motivate the following question: *how can we characterize the effect of the training coefficients used by the optimizer on the current policy update while preserving effective training progress and avoiding directional deviation?*

To address this problem, we propose **Set**-Valued **G**roup **R**elative **P**olicy **O**ptimization (**SetGRPO**), which models GRPO's within-group training coefficients as a joint uncertainty set and uses it to select response groups and construct policy updates. Concretely, SetGRPO comprises three components: (1) *Joint Reward Set Calibration*, which uses paired reward discrepancies and independent calibration data to characterize the joint coefficient shifts induced by within-group normalization; (2) *Geometry-Guided Safe Recovery*, which maps candidate coefficients through the current policy's response-level gradients and determines the largest direction-safe recovery ratio; and (3) *Selective Risk-Gated Optimization*, which selects a global threshold on an independent risk-calibration set to control the conditional direction-reversal risk of accepted updates. During training, only groups satisfying this threshold contribute their recovered coefficients to the original GRPO objective, preserving effective updates while limiting overall update risk.

Our main contributions are summarized as follows:

- We first investigate the propagation mechanism of rubric score discrepancies in GRPO: a score discrepancy couples the training coefficients of an entire group through the shared mean and scale, and then jointly with the response-level gradients of the current policy determines the local update, yielding different update magnitudes and directions.
- We formalize this propagation process as a training-coefficient-driven policy-update problem and propose SetGRPO, which conformally calibrates GRPO's joint coefficient space and combines policy-update geometry with selective risk gating to limit direction deviation while preserving effective training progress.
- We systematically evaluate SetGRPO on ten open-ended task benchmarks across three domains with the Qwen3.5 and Llama-3.2 model families. The results show that SetGRPO achieves the best performance among all compared training methods on both model families, confirming its effectiveness in improving the reliability of GRPO training under rubric score discrepancies.

2 PRELIMINARIES

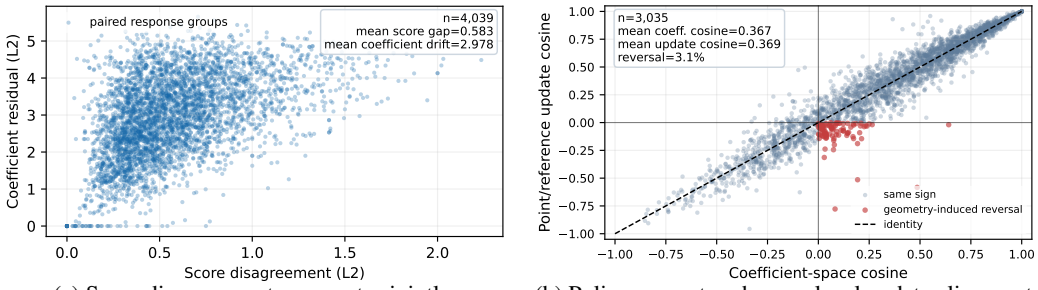
In rubric-based reinforcement learning, grader rewards are normalized within each GRPO response group to produce training coefficients, which are combined with the current policy's response-level gradients to form local updates. We therefore analyze score discrepancies along this reward-to-update chain, first formalizing the mapping and then evaluating its training consequences through paired pilot experiments.

2.1 RUBRIC-BASED GRPO

For a problem x, the rollout policy $\pi_{\boldsymbol{\theta}_{\text{old}}}$ generates a group of K responses, over which the grader produces response-level rewards according to the given rubric. Denote the response group, the reward vector, and the nominal training coefficients of GRPO as

$$\mathbf{Y}_x = (y_{x,1}, \ldots, y_{x,K}), \qquad \mathbf{r}_x = (r_{x,1}, \ldots, r_{x,K})^\top, \qquad \hat{\mathbf{a}}_x = T_{\text{train}}(\mathbf{r}_x), \tag{1}$$

where T_{train} denotes the training mapping of the original GRPO that group-centers and scale-normalizes a group of rewards, and $\hat{\mathbf{a}}_x$ denotes the training coefficients induced by the rewards produced by the grader. Because the responses within a group share the same center and scale, each $\hat{a}_{x,i}$ is determined jointly by the rewards of the entire group.



(a) Score disagreement propagates jointly. (b) Policy geometry changes local update alignment.

Figure 1: Illustration of score disagreement and its effect on local policy-update geometry.

These coefficients then enter the original GRPO objective as response-level advantages. We define the group-level objective as

$$\mathcal{J}_{\mathrm{GRPO}}(\boldsymbol{\theta}; \hat{\mathbf{a}}_x) = \mathbb{E}_x \left[\frac{1}{K} \sum_{i=1}^{K} \frac{1}{|y_{x,i}|} \sum_{t=1}^{|y_{x,i}|} \mathcal{L}_{\mathrm{GRPO}} \left(\omega_{x,i,t}(\boldsymbol{\theta}), \hat{a}_{x,i} \right) \right], \tag{2}$$

where $\omega_{x,i,t}(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(y_{x,i,t}|x, y_{x,i,<t})}{\pi_{\boldsymbol{\theta}_{\mathrm{old}}}(y_{x,i,t}|x, y_{x,i,<t})}$ is the token-level probability ratio, and $\mathcal{L}_{\mathrm{GRPO}}(\omega, a)$ denotes the token-level surrogate term retaining the probability ratio and the clipping mechanism.

These training coefficients determine each response's relative weight in the group objective but not the update direction, which also depends on the response-specific gradient directions and their mutual relations under the current policy. Let $\mathbf{g}_{x,i}$ denote the mean response log-probability gradient for $y_{x,i}$. Performing a first-order expansion around the current rollout policy and neglecting the learning rate and the common scaling across groups, the contribution of the coefficient vector $\mathbf{a}$ to the local update of the response group is

$$\mathbf{g}_{x,i} = \nabla_{\boldsymbol{\theta}} \left[\frac{1}{|y_{x,i}|} \sum_{t=1}^{|y_{x,i}|} \log \pi_{\boldsymbol{\theta}} \left(y_{x,i,t} \mid x, y_{x,i,<t} \right) \right], \quad \Delta\boldsymbol{\theta}_x(\mathbf{a}) \propto \sum_{i=1}^{K} a_i \mathbf{g}_{x,i}. \tag{3}$$

2.2 Preliminary Study on Grader-Induced Update Risk

The above equation reveals the propagation chain: *a score discrepancy changes the group's training coefficients, which modifies the local update through the current policy's gradients and ultimately affects which response groups are retained.* To assess this risk and motivate SetGRPO, we conduct pilot experiments asking whether (i) within-group normalization couples scoring errors, (ii) the effect of coefficient shifts depends on the policy state, and (iii) selective execution changes the aggregate direction-reversal risk.

Setup. We run original GRPO on Qwen3.5-9B with the RubricHub training set and collect response groups with their corresponding policy snapshots; training details are given in Section 4. For each group, we freeze the responses and their order, score it twice with the same grader, and map both scores through T_{train}. For the second scoring, we define $\mathbf{r}_x^\star$, $\mathbf{a}_x^\star = T_{\mathrm{train}}(\mathbf{r}_x^\star)$, $\boldsymbol{\delta}_x = \mathbf{a}_x^\star - \hat{\mathbf{a}}_x$. Using the same policy snapshot, we compare the updates induced by the two coefficient sets and analyze coefficient-residual dependence, policy-state dependence, and aggregate risk after group-wise screening.

Does Grader Discrepancy Remain Response-wise Independent? By Eq. 1 and Eq. 2, group normalization uses a shared center and scale. Therefore, as shown in Figure 1(a), since T_{train} normalizes with a center and scale shared over the whole group, a change in the score of a single response affects the entire coefficient vector. Moreover, the corresponding residuals appear as a within-group correlated holistic shift rather than mutually independent response-wise deviations. The uncertainty involved in training should thus be modeled for the entire coefficient vector, rather than for each response separately.

Can Coefficient Drift Alone Characterize Update Safety? Fig. 1(b) applies the coefficients from the two scorings to the response-level gradients of the same response group and compares the local updates under different policy snapshots. The results show that similar coefficient shifts may produce different update magnitudes and directions, and that the updates induced by the two scorings

may even have a negative inner product. Distance in coefficient space therefore cannot characterize update safety by itself, since the direction and magnitude of the update also depend on the gradient geometry of the current policy.

Can Group-wise Screening Control Aggregate Reversal Risk? The analyses in Fig. 1 jointly indicate that when the optimizer retains only the response groups satisfying a local criterion, the sample distribution entering optimization changes from all response groups to the acceptance set. Since different screening rules yield different acceptance sets, a per-group criterion only shows that a local condition holds and cannot directly provide control over the aggregate direction-reversal risk on the acceptance set. Risk assessment must therefore be based on conditional statistics over the response groups that are actually accepted.

In summary, group normalization turns score discrepancies into correlated shifts in training coefficients, whose update effects depend on the current policy's response-level gradient geometry. Selective execution further changes the accepted groups and their aggregate direction-reversal risk. SetGRPO therefore jointly models coefficient uncertainty, policy-dependent update geometry, and acceptance-conditioned reversal risk.

3 METHODOLOGY

As the main innovation, SetGRPO constructs a joint coefficient set in the training-coefficient space actually used by GRPO and propagates it into the update space of the current policy, so as to recover effective and directionally consistent training updates as much as possible under grader score discrepancies. To address the propagation chain revealed by the pilot experiments in Section 2, SetGRPO consists of three interconnected modules: (a) *Joint Reward Set Calibration, JRSC*, (b) *Geometry-Guided Safe Recovery, GGSR*, and (c) *Selective Risk-Gated Optimization, SRGO*. In the following, we introduce these three modules in turn.

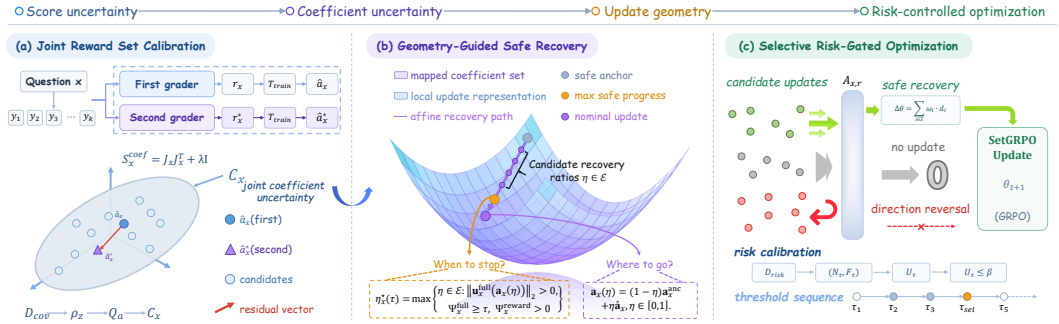


Figure 2: Overview of SetGRPO. *(a) Joint reward-set calibration* uses two grader scores on the same response group to construct a joint coefficient uncertainty set. *(b) Geometry-guided safe recovery* maps candidate coefficients through the current policy geometry and identifies the largest direction-preserving recovery ratio. *(c) Selective risk-gated optimization* selects a global threshold and applies the recovered coefficients only to updates satisfying the calibrated conditional-risk constraint.

3.1 JOINT REWARD SET CALIBRATION (JRSC)

As shown in Fig. 1(a), two grader scorings of the same response group produce correlated within-group shifts after the shared training mapping. We therefore propose *Joint Reward Set Calibration*, which models this coupled uncertainty as a joint coefficient set rather than independent response-wise errors. Specifically, for problem x, let $\mathbf{r}_x$ and $\mathbf{r}_x^\star$ denote the rewards from the first and second scorings. Using Eq. 1, $\hat{\mathbf{a}}_x = T_{\text{train}}(\mathbf{r}_x)$, $\mathbf{a}_x^\star = T_{\text{train}}(\mathbf{r}_x^\star)$, where T_{train} maps group rewards to within-group normalized coefficients. We define the paired residual $\boldsymbol{\delta}_x = \mathbf{a}_x^\star - \hat{\mathbf{a}}_x$, which calibrates the coverage radius of the subsequent joint set.

Subsequently, to capture how reward changes jointly perturb the coefficient vector, we linearize T_{train} around the first-scoring rewards $\mathbf{r}_x$ and define $\mathbf{J}_x = \nabla_{\mathbf{r}} T_{\text{train}}(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_x}$. Its j-th column describes how a perturbation in reward j affects all group coefficients, preserving the cross-response coupling induced by shared normalization. The resulting coefficient-uncertainty matrix under unit reward perturbations is

$$\mathbf{S}_x^{\text{coef}} = \mathbf{J}_x \mathbf{J}_x^\top + \lambda \mathbf{I}_K, \qquad \lambda \geq 0, \tag{4}$$

where $\mathbf{J}_x\mathbf{J}_x^\top$ captures the cross-coefficient effects of local reward perturbations and determines set's directional scales, while $\lambda\mathbf{I}_K$ stabilizes inversion and distance computation in near-degenerate cases.

However, $\mathbf{S}_z^{\text{coef}}$ specifies only the shape of the joint coefficient set, not its radius. We therefore calibrate a unified coverage radius from independent paired residuals. For each coverage-calibration group $z \in \mathcal{D}_{\text{cov}}$ [1], we compute the Mahalanobis score (Mahalanobis, 2018) $\rho_z = \sqrt{\boldsymbol{\delta}_z^\top (\mathbf{S}_z^{\text{coef}})^{-1}\boldsymbol{\delta}_z}$. This score accounts for directional scale differences and within-group correlations induced by reward normalization. Its geometric interpretation is discussed in Appendix A.3.

Next, let $n = |\mathcal{D}_{\text{cov}}|$, $\rho_{(j)}$ denote the j-th order statistic, and $\alpha \in (0,1)$ be the target miss rate. Under exchangeability of the calibration and target groups (i.e., identical data construction, response sampling, and scoring protocols with permutation-invariant joint distributions), we define

$$Q_\alpha = \rho_{(\lceil (n+1)(1-\alpha) \rceil)}, \quad \mathcal{C}_x = \left\{ \mathbf{a} \in \mathbb{R}^K : (\mathbf{a} - \hat{\mathbf{a}}_x)^\top (\mathbf{S}_x^{\text{coef}})^{-1} (\mathbf{a} - \hat{\mathbf{a}}_x) \le Q_\alpha^2 \right\}. \tag{5}$$

Then $\mathcal{C}_x$ is an ellipsoidal joint coefficient set centered at $\hat{\mathbf{a}}_x$ that contains the training coefficients of the second scoring with marginal probability at least $1-\alpha$. If $\lceil (n+1)(1-\alpha) \rceil = n+1$, no corresponding $(n+1)$-th order statistic exists in the calibration set. In this case we adopt the conservative convention of Appendix A.2 and set $Q_\alpha = +\infty$. A finite-sample coverage proof for this ellipsoidal set is given in Appendix A.2.

3.2 Geometry-Guided Safe Recovery (GGSR)

JRSC models the discrepancy between two grader scorings as the joint coefficient set $\mathcal{C}_x$. We then propose *Geometry-Guided Safe Recovery (GGSR)*, which maps $\mathcal{C}_x$ into the policy-update and reward spaces to identify the largest coefficient-recovery ratio whose update remains directionally consistent with the second scoring under the current policy.

Specifically, we first let $q \in \{\text{full}, \text{reward}\}$ denote the policy-update and reward spaces, respectively, with dimension D_q. Let $\mathcal{S}^q$ be a fixed low-dimensional linear map, $\mathbf{P}_x^q$ a linear transformation, and $\mathbf{h}_x^q$ a coefficient-independent offset. Notably, by full geometry we mean the complete local update geometry under a fixed projected representation, rather than the unprojected original parameter space. From the response-level gradients in Eq. 3, we define

$$\mathbf{G}_x^q = [\mathcal{S}^q(\mathbf{g}_{x,1}), \ldots, \mathcal{S}^q(\mathbf{g}_{x,K})] \in \mathbb{R}^{D_q \times K}, \quad \mathbf{u}_x^q(\mathbf{a}) = \mathbf{P}_x^q (\mathbf{G}_x^q \mathbf{a} + \mathbf{h}_x^q) \in \mathbb{R}^{D_q}, \tag{6}$$

where the i-th column of $\mathbf{G}_x^q$ is the projected gradient of response $y_{x,i}$, and $\mathbf{u}_x^q(\mathbf{a})$ is the local update induced by the coefficients $\mathbf{a}$ in the space q.

For a candidate coefficient $\mathbf{a}$, let $\mathbf{d} = \mathbf{u}_x^q(\mathbf{a})$ and define the worst-case projection over the joint coefficient set as $\Psi_x^q(\mathbf{d}) := \min_{\tilde{\mathbf{a}} \in \mathcal{C}_x} \mathbf{d}^\top \mathbf{u}_x^q(\tilde{\mathbf{a}})$. This quantity measures the candidate update's worst-case alignment and magnitude under grader uncertainty. If $\mathbf{a}_x^\star \in \mathcal{C}_x$, then $\Psi_x^{\text{full}}\left(\mathbf{u}_x^{\text{full}}(\mathbf{a})\right) \le \left\langle \mathbf{u}_x^{\text{full}}(\mathbf{a}), \mathbf{u}_x^{\text{full}}(\mathbf{a}_x^\star) \right\rangle$. Thus, a positive full-space projection rules out direction reversal, while a positive reward-space projection preserves reward-direction consistency. Because $\mathcal{C}_x$ is ellipsoidal and $\mathbf{u}_x^q$ is affine in $\mathbf{a}$, the projection has the closed form

$$\Psi_x^q(\mathbf{d}) = \mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x) - Q_\alpha \sqrt{\mathbf{d}^\top \mathbf{P}_x^q \mathbf{G}_x^q \mathbf{S}_x^{\text{coef}} (\mathbf{G}_x^q)^\top (\mathbf{P}_x^q)^\top \mathbf{d}}. \tag{7}$$

The derivation and directional-consistency interpretation are given in Appendix A.4.

However, the minimum-projection criterion evaluates a candidate coefficient but does not select one from $\mathcal{C}_x$. We therefore choose the feasible coefficient with the smallest local-update norm as an anchor, denoted as $\mathbf{a}_x^{\text{anc}} \in \arg\min_{\mathbf{a} \in \mathcal{C}_x} \frac{1}{2} \|\mathbf{u}_x^{\text{full}}(\mathbf{a})\|_2^2$. Since $\mathcal{C}_x$ is convex and centered at $\hat{\mathbf{a}}_x$, the path $\mathbf{a}_x(\eta) = (1-\eta)\mathbf{a}_x^{\text{anc}} + \eta \hat{\mathbf{a}}_x, \quad \eta \in [0,1]$, remains in $\mathcal{C}_x$. Here, η denotes the fraction of training recovered from the conservative anchor toward the first-scoring update. For a predetermined candidate set $\mathcal{E} \subseteq [0,1]$, the maximum acceptable ratio under threshold $\tau \ge 0$ is

$$\eta_x^\star(\tau) = \max \left\{ \eta \in \mathcal{E} : \left\| \mathbf{u}_x^{\text{full}}(\mathbf{a}_x(\eta)) \right\|_2 > 0,\ \Psi_x^{\text{full}}\left(\mathbf{u}_x^{\text{full}}(\mathbf{a}_x(\eta))\right) \ge \tau,\ \Psi_x^{\text{reward}}\left(\mathbf{u}_x^{\text{reward}}(\mathbf{a}_x(\eta))\right) > 0 \right\}. \tag{8}$$

If no η satisfies these conditions, group x produces no update. Appendix A.5 proves the existence of the anchor and the feasibility of the recovery path within $\mathcal{C}_x$.

[1] The paired groups are split by problem root into disjoint coverage- and risk-calibration sets. The former calibrates the set radius, while the latter selects the execution threshold; both are fixed before formal training.

3.3 Selective Risk-Gated Optimization (SRGO)

Equation 8 determines the maximum feasible recovery ratio for each response group at threshold τ. Because τ also controls whether a nonzero recovery exists, it induces the acceptance set $\mathcal{A}_\tau = \{x : A_{x,\tau} = 1\}$, where $A_{x,\tau} = 1$ iff the constraints admit a nonzero recovery ratio, and $A_{x,\tau} = 0$ otherwise. The aggregate direction-reversal risk therefore depends on the composition of $\mathcal{A}_\tau$, not only on the group-wise criterion. We thus introduce *Selective Risk-Gated Optimization*, which evaluates acceptance-conditioned risk for candidate thresholds on the independent risk-calibration set $\mathcal{D}_{\text{risk}}$ and selects a threshold that satisfies the risk constraint while retaining as much training signal as possible. The monotonicity of recovery feasibility, acceptance, and cumulative recovery is proved in Appendix A.7.

Specifically, for each candidate threshold $\tau \in \mathcal{T}$, we define the direction-reversal event for accepted groups as $Z_{x,\tau} = \mathbb{I}\left[\left\langle \mathbf{u}_x^{\text{full}}(\mathbf{a}_x(\eta_x^\star(\tau))), \mathbf{u}_x^{\text{full}}(\mathbf{a}_x^\star)\right\rangle < 0\right]$, and its acceptance-conditioned risk as $R_\tau = \Pr(Z_{x,\tau} = 1 \mid A_{x,\tau} = 1)$. On the independent risk-calibration set $\mathcal{D}_{\text{risk}}$, let N_τ and F_τ denote the numbers of accepted and direction-reversed groups, respectively. We estimate R_τ using the one-sided Clopper–Pearson upper bound $U_\tau = \text{CPUpper}\left(F_\tau, N_\tau; 1 - \frac{\gamma}{|\mathcal{T}|}\right)$, where β is the target risk level and γ is the family-wise failure probability across candidate thresholds. A threshold is feasible if $N_\tau > 0$ and $U_\tau \le \beta$; among feasible thresholds, we select the one maximizing cumulative recovery Y_τ, formualted as,

$$Y_\tau = \sum_{x \in \mathcal{D}_{\text{risk}}} A_{x,\tau}\eta_x^\star(\tau), \quad \tau_{\text{sel}} = \min\left(\arg\max_{\tau \in \mathcal{T}: N_\tau > 0,\ U_\tau \le \beta} Y_\tau\right). \tag{9}$$

If several thresholds tie, the smaller one is taken; if no threshold satisfies the risk constraint, τ_{sel} is undefined and the risk gating rule is disabled in subsequent training.

Once τ_{sel} is determined, SetGRPO retains the probability ratio, clipping, and token-level surrogate term of the original GRPO and replaces only the training coefficients with,

$$\mathcal{J}_{\text{SetGRPO}}(\boldsymbol{\theta}) = \mathcal{J}_{\text{GRPO}}(\boldsymbol{\theta}; \widetilde{\mathbf{a}}_x), \qquad \widetilde{\mathbf{a}}_x = \begin{cases} \mathbf{a}_x(\eta_x^\star(\tau_{\text{sel}})), & A_{x,\tau_{\text{sel}}} = 1, \\ \mathbf{0}_K, & A_{x,\tau_{\text{sel}}} = 0. \end{cases} \tag{10}$$

Accepted response groups use their recovered coefficients, while the remaining groups yield no policy update. We further evaluate the predetermined τ_{sel} across different training checkpoints to test its empirical stability under policy drift; details are given in Appendix F.4.

4 Experiments

4.1 Experimental Setup

Data and evaluation. We perform reinforcement learning training on the RubricHub training set (Li et al., 2026), whose data covers five task types: Chat, Instruction Following, Medical, Science, and Writing. After training, each model is evaluated on ten downstream benchmarks spanning three domains: healthcare, writing, and instruction following. The healthcare benchmarks include HealthBench, LLMEval-Med, MedQA, and MedMCQA; the writing benchmarks include WritingBench and Creative Writing v3; and the instruction following benchmarks include IFEval, IFBench, InfoBench, and MulDimIF. For benchmarks that require model scoring, we uniformly invoke the GPT-5.6-Luna API. Each benchmark uses its official evaluation metrics, and Macro Avg. is used only to aggregate the results of the ten benchmarks.

Models and baselines. We use Qwen3.5-4B, Qwen3.5-9B and Llama 3.2-3B as backbones and compare Base (the backbone without the training in this study) against four training methods: Rubric-based GRPO, RubricHub (Li et al., 2026), RuscaRL (Zhou et al., 2026), and SetGRPO. Rubric-based GRPO uses the same rewards obtained from the first scoring as SetGRPO and is the main control for differences in the training objective; RubricHub and RuscaRL are run following their publicly available training procedures, with identical data scale, training budget, and final evaluation protocol. The training details are given in Appendix C.

Training details. For each prompt, we generate $K = 8$ rollout responses with a maximum response length of 4096 tokens. All methods use the same training data, number of training steps, and generation budget, and are evaluated under the same protocol after training. For benchmarks requiring

Table 1: Main results on ten downstream benchmarks. HB: HealthBench; LEM: LLMEval-Med; MQA: MedQA; MCQA: MedMCQA; WB: WritingBench; CW: Creative Writing v3; IFE: IFEval; IFB: IFBench; IB: InfoBench; and MIF: MulDimIF. Our results report the mean across repeated runs, with absolute improvements over the corresponding Base model shown in parentheses.

Model	Medical				Writing		Instruction Following				Macro Avg.
	HB	LEM	MQA	MCQA	WB	CW	IFE	IFB	IB	MIF	
Closed-source Models											
OpenAI-o3	59.80	78.90	96.00	84.70	77.70	81.40	91.60	67.80	89.40	64.00	79.13
GPT-4.1	47.90	71.20	92.40	80.00	69.00	79.00	87.00	37.20	89.40	64.00	71.71
Gemini-3 Pro Preview	49.30	72.70	67.00	62.20	78.50	81.50	94.20	61.20	85.60	66.83	71.90
Open-sourced Models											
Qwen3.5-4B	37.09	26.24	77.77	63.26	50.20	27.60	46.40	17.35	78.22	41.25	46.54
+ GRPO	56.94 (+19.85)	43.29 (+17.05)	79.11 (+1.34)	63.55 (+0.29)	64.92 (+14.72)	29.98 (+2.38)	62.28 (+15.88)	25.09 (+7.74)	80.38 (+2.16)	48.55 (+7.30)	55.41 (+8.87)
+ RuscaRL	65.10 (+28.01)	45.30 (+19.06)	79.60 (+1.83)	63.30 (+0.04)	67.50 (+17.30)	31.80 (+4.20)	65.90 (+19.50)	24.40 (+7.05)	80.74 (+2.52)	47.80 (+6.55)	57.14 (+10.60)
+ RubricHub	67.18 (+30.09)	47.71 (+21.47)	78.06 (+0.29)	**64.22 (+0.96)**	69.62 (+19.42)	**32.80 (+5.20)**	67.88 (+21.48)	26.31 (+8.96)	**82.91 (+4.69)**	49.84 (+8.59)	58.65 (+12.11)
+ SetGRPO (Ours)	**69.64 (+32.55)**	**50.08 (+23.84)**	**80.55 (+2.78)**	63.84 (+0.58)	**72.14 (+21.94)**	32.23 (+4.63)	**70.45 (+24.05)**	**28.97 (+11.62)**	82.32 (+4.10)	**52.35 (+11.10)**	**60.26 (+13.72)**
Qwen3.5-9B	40.94	34.63	76.20	67.03	53.83	32.19	51.39	19.39	80.98	41.17	49.78
+ GRPO	61.06 (+20.12)	51.65 (+17.02)	77.68 (+1.48)	67.10 (+0.07)	68.84 (+15.01)	34.65 (+2.46)	67.61 (+16.22)	26.91 (+7.52)	83.02 (+2.04)	48.33 (+7.16)	58.69 (+8.91)
+ RuscaRL	69.00 (+28.06)	53.50 (+18.87)	78.35 (+2.15)	67.18 (+0.15)	71.70 (+17.87)	36.40 (+4.21)	71.30 (+19.91)	25.80 (+6.41)	82.86 (+1.88)	47.20 (+6.03)	60.33 (+10.55)
+ RubricHub	71.04 (+30.10)	55.62 (+20.99)	**79.86 (+3.66)**	67.28 (+0.25)	73.76 (+19.93)	34.21 (+2.02)	73.38 (+21.99)	28.24 (+8.85)	**85.42 (+4.44)**	**52.33 (+11.16)**	62.11 (+12.33)
+ SetGRPO (Ours)	**73.86 (+32.92)**	**58.15 (+23.52)**	79.32 (+3.12)	**67.34 (+0.31)**	**76.28 (+22.45)**	**36.91 (+4.72)**	**75.94 (+24.55)**	**30.57 (+11.18)**	84.83 (+3.85)	51.87 (+10.70)	**63.51 (+13.73)**
Llama 3.2-3B	14.37	15.96	57.18	48.74	30.63	32.57	71.08	23.67	44.93	35.69	37.48
+ GRPO	21.08 (+6.71)	17.42 (+1.46)	58.66 (+1.48)	49.32 (+0.58)	40.49 (+9.86)	35.41 (+2.84)	77.69 (+6.61)	25.58 (+1.91)	45.14 (+0.21)	76.42 (+40.73)	44.72 (+7.24)
+ RuscaRL	22.84 (+8.47)	17.96 (+2.00)	60.47 (+3.29)	51.05 (+2.31)	42.52 (+11.89)	40.28 (+7.71)	79.58 (+8.50)	27.76 (+4.09)	46.67 (+1.74)	79.35 (+43.66)	46.85 (+9.37)
+ RubricHub	24.61 (+10.24)	18.19 (+2.23)	61.34 (+4.16)	51.27 (+2.53)	47.33 (+16.70)	39.84 (+7.27)	80.93 (+9.85)	28.67 (+5.00)	47.44 (+2.51)	82.08 (+46.39)	48.17 (+10.69)
+ SetGRPO (Ours)	**26.72 (+12.35)**	**18.47 (+2.51)**	**62.18 (+5.00)**	**52.14 (+3.40)**	**45.66 (+15.03)**	**42.51 (+9.94)**	**83.47 (+12.39)**	**30.38 (+6.71)**	**49.46 (+4.53)**	**84.67 (+48.98)**	**49.57 (+12.09)**

model scoring, all methods use the same judge, prompt, and decoding configuration. The rubric rewards during training are provided by a Qwen3-32B grader. The online rewards of SetGRPO come from the first grader scoring; the second grader scoring is computed only for pre-frozen response groups, used for calibration and excluded from the training coefficients of online groups. The optimizer, batch size, learning rate, calibration set size, candidate thresholds, and hardware configuration are given in Appendix C.

4.2 Main Results

Overall performance. Table 1 reports results on ten benchmarks across three backbones. SetGRPO achieves the highest Macro Avg. on Qwen3.5-4B and Qwen3.5-9B, reaching 60.26 and 63.51, respectively. On Qwen3.5-9B, it improves over RubricHub by 2.82 points on HealthBench and 2.52 on WritingBench, and over RuscaRL by 4.86 on HealthBench and 4.65 on LLMEval-Med. Relative to RubricHub, its average gains across healthcare, writing, and instruction-following tasks are 1.22, 2.61, and 0.96 points, respectively, with consistent improvements across backbones. These results indicate that SetGRPO improves performance across different backbones and task domains by jointly modeling the training-coefficient shifts caused by score discrepancies and screening for actionable updates in combination with the update geometry of the current policy.

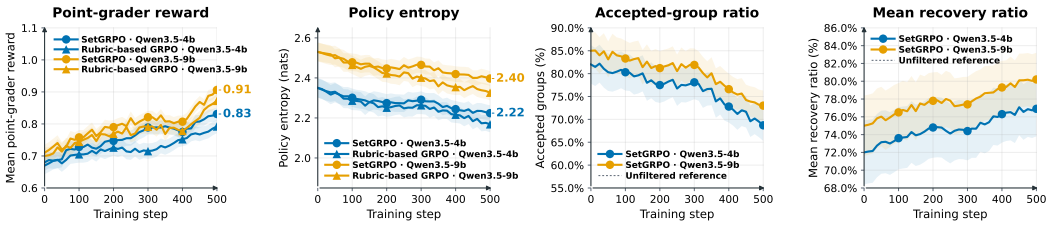


Figure 3: Training dynamics of SetGRPO, including the reward obtained from the first score, policy entropy, accepted-group ratio, and mean recovery ratio over the first 500 training steps.

Training dynamics. Figure 3 shows how the mean reward from the first scoring, policy entropy, proportion of accepted groups, and mean recovery ratio over accepted groups evolve during training. The mean reward from the first scoring rises steadily in the early phase and saturates later; policy entropy decreases gradually without an abrupt collapse. Compared with unfiltered training, the proportion of accepted groups is lower but remains nonzero in the later phase, and the mean recovery ratio over accepted groups stays within 76%–80%. This indicates that screening excludes response groups that fail the projection condition while nonzero training coefficients still enter the optimization. The steady rise of reward indicates that the retained updates bring training progress, and the stable decline of entropy indicates that the policy does not exhibit marked additional shrinkage.

5 Analysis and Discussion

This section reports the conditional direction-reversal risk of accepted updates and analyzes the effect of different design variants and the additional computational overhead on the final results.

5.1 Risk Calibration and Retained Updates

Figure 4 shows the empirical direction-reversal risk and its one-sided Clopper–Pearson upper confidence bound across candidate thresholds, together with the target risk β and the selected threshold $\tau_{\text{sel}} = 0.25$. As the screening condition tightens, both the empirical risk and its upper confidence bound decrease overall. At the selected threshold, the upper confidence bound remains below $\beta = 0.08$. The complete acceptance statistics, cumulative recovery, and formal-training results for each backbone are reported in Appendix C.5.

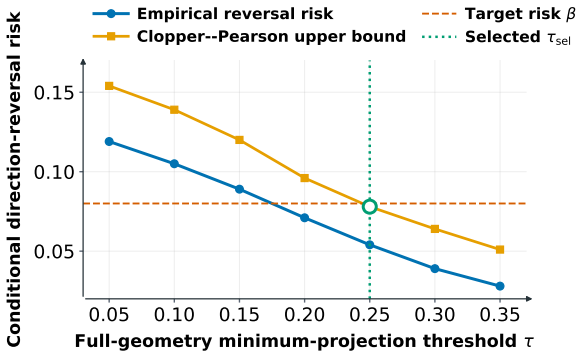


Figure 4: Risk calibration and retained updates at the selected threshold τ_{sel}.

5.2 Ablation Studies

To isolate the effects of joint coefficient modeling, policy-update geometry, coefficient recovery, and threshold selection, we evaluate three variant groups: (a) an isotropic coefficient set instead of the joint set; (b) variants removing the policy-space projection condition, coefficient recovery, or reward-space projection condition; and (c) a fixed threshold instead of risk-calibrated threshold selection. We also include a matched random-discard baseline that retains the same number of groups as SetGRPO to distinguish geometry-based screening from merely reducing updates.

Table 2: Ablation study of SetGRPO on Qwen3.5-4B. Δ denotes the change from Full SetGRPO in each metric's native units; upward arrows indicate the preferred direction. CR is normalized cumulative recovery and NUR is the nonzero update rate. Reversal risk, NUR, and CR lie in $[0, 1]$.

D#	Derivative	Macro Avg. ↑	Macro Avg. Δ	Reversal risk ↓	Reversal risk Δ	CR ↑	CR Δ	NUR ↑	NUR Δ
(a) Coefficient-set construction									
1	Spherical set	56.62	-3.64	0.096	+0.049	0.61	-0.19	0.512	-0.190
(b) Local geometry and recovery									
2	Identity geometry	55.08	-5.18	0.118	+0.071	0.51	-0.29	0.438	-0.264
3	Safe-anchor-only	56.94	-3.32	0.061	+0.014	0.36	-0.44	0.318	-0.384
4	Reward-only recovery	57.78	-2.48	0.103	+0.056	0.57	-0.23	0.566	-0.136
(c) Threshold selection and update control									
5	Fixed-threshold	58.11	-2.15	0.089	+0.042	0.60	-0.20	0.548	-0.154
6	Matched random discard	54.83	-5.43	0.154	+0.107	0.49	-0.31	0.702	0.000
SetGRPO (Ours)		**60.26**	**0.00**	**0.047**	**0.000**	**0.80**	**0.000**	**0.702**	**0.000**

Table 2 shows that all variants underperform the full setting (Macro Avg. 60.26). Replacing the joint coefficient set with a spherical set reduces the Macro Avg. to 56.72 and raises reversal risk to 0.096, indicating that this variant fails to preserve the within-group correlated coefficient shifts. Removing the projection condition of the policy-update space reduces the macro average to 55.18 and raises the risk to 0.118, indicating that screening in coefficient space alone is insufficient to determine the effect of coefficient changes on the current policy update. Keeping only the minimum-norm starting point gives a macro average of 57.04 with the recovery amount dropping to 0.36, indicating that not recovering the training coefficients costs training progress. Removing the minimum projection condition in the reward space gives a macro average of 57.88 with the risk rising to 0.103, indicating that the direction screening in the policy-update space cannot substitute for the positive projection condition in the reward space. The fixed threshold variant reaches 58.21 with a risk 0.042 higher than the full setting, indicating that risk calibration contributes to the selection of the execution threshold. Finally, matched random discard reaches only 54.93 despite a comparable acceptance rate, showing that the gains come from targeted screening rather than simply reducing updates.

5.3 Efficiency Comparison

Table 3: Efficiency comparison between GRPO and SetGRPO.

Method	Training-time breakdown					Resource usage		10-Benchmark Macro Avg.
	Single-step time (s)	Rollout time (s)	Grader time (s)	Jacobian time (s)	Coefficient recovery time (s)	Peak VRAM (GB)	Cumulative GPU-hours	
GRPO	82	48	18	0	0	61	22.8	55.41
SetGRPO	103	48	18	13	7	67	28.6	60.26

In Table 3, we compare GRPO and SetGRPO in computational cost and benchmark performance. SetGRPO increases per-step training time from 82 to 103 s, peak VRAM from 61 to 67 GB, and cumulative GPU hours from 22.8 to 28.6, corresponding to increases of 25.6%, 9.8%, and 25.4%, while rollout and grader times remain unchanged at 48 s and 18 s. The overhead mainly comes from Jacobian computation for coefficient-set calibration and per-group search for the largest projection-safe recovery ratio. In return, the ten-benchmark Macro Avg. increases from 55.41 to 60.26 (+4.75), indicating improved performance at the cost of additional computation.

6 Related Work

Reinforcement Learning for Large Language Models and Rubric Feedback. Alignment training for large language models has evolved from preference learning and supervised fine-tuning to human- or AI-feedback-based instruction alignment(Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022), giving rise to PPO and reward-model-free DPO(Schulman et al., 2017; Rafailov et al., 2023). For reasoning tasks, verifiers and process supervision provide fine-grained training signals(Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024), while DeepSeekMath and DeepSeek-R1 apply GRPO to large-language-model reasoning(Shao et al., 2024; Guo et al., 2025). For open-ended tasks without unique reference answers, G-Eval, FLASK, MT-Bench, and Prometheus use rubrics or LLM judges for structured evaluation(Liu et al., 2023; Ye et al., 2024; Zheng et al., 2023; Kim et al., 2024); RewardBench, RubricHub, and RuscaRL assess reward-model capability or use rubric feedback for reinforcement learning(Lambert et al., 2025; Li et al., 2026; Zhou et al., 2026). Despite evidence of reward overoptimization and systematic judge bias(Gao et al., 2023; Wang et al., 2024a; Zheng et al., 2023), prior work largely treats grader outputs as fixed training signals and does not trace how the score discrepancies propagate through GRPO's group normalization and response-level gradient aggregation to alter the actual policy update.

Conformal Prediction and Selective Risk Control. Conformal prediction exploits exchangeability to construct, from calibration samples, prediction sets with finite-sample coverage guarantees(Shafer & Vovk, 2008; Angelopoulos & Bates, 2023), and has extended from regression and classification coverage to input-dependent uncertainty and general risk functions(Lei et al., 2018; Romano et al., 2019; 2020; Bates et al., 2021; Angelopoulos et al., 2024). Selective prediction controls the risk on an accepted subset through abstention or sample screening(Geifman & El-Yaniv, 2019), and the Clopper–Pearson interval provides a classical one-sided upper confidence bound for finite-sample binomial risk(Clopper & Pearson, 1934). SetGRPO draws on these ideas of calibration and risk control, but shifts the object of calibration to the joint training coefficients actually used by GRPO and the policy updates they induce, and further controls the conditional direction-reversal risk over the set of accepted updates.

7 Conclusion

This paper proposed SetGRPO to address the effect of grader stochasticity on policy updates in rubric-based GRPO. Our pilot experiments showed that within-group normalization in GRPO converted grader stochasticity into correlated shifts in the training coefficients, while the response-level gradients of the current policy determined how these shifts affected local updates. Selective execution further changed the composition of the accepted response groups and their aggregate direction-reversal risk. Based on this observation, SetGRPO constructed a joint uncertainty set in training-coefficient space, calibrated possible coefficient shifts through JRSC, used GGSR to recover acceptable training signals in conjunction with the update geometry of the current policy, and finally calibrated and controlled the conditional direction-reversal risk of accepted updates through SRGO, while preserving the optimization structure of the original GRPO. Across ten open-ended task benchmarks in three domains, SetGRPO achieved the best macro-average performance among all compared training methods on the Qwen-3.5 and Llama-3.2 model families.

Ethics Statement SetGRPO uses paired rubric-grader scores and response-level gradients from the current policy to select and recover reinforcement-learning updates. These components may introduce several risks: grader judgments may be stochastic, biased, or subject to distribution shift; calibration data may contain sensitive information; and the coefficient calibration and selective update mechanisms may amplify artifacts present in the data or grader. The finite-sample guarantee in this work concerns only the conditional direction-reversal risk of accepted updates under the stated exchangeability assumption. It does not guarantee privacy, fairness, factuality, or harmlessness for individual responses. The training, coverage-calibration, and risk-calibration splits are separated by problem root, and downstream evaluation benchmarks are not used for training, calibration, or checkpoint selection. Any deployment should apply data minimization, appropriate authorization and access controls, and anonymization where applicable. It should also include regular audits for grader drift and subgroup disparities, as well as human review for high-risk applications. Updates should be suspended or reviewed when the exchangeability assumption, calibration conditions, or grader reliability are not satisfied.

AI Assistance Statement During the preparation of this manuscript, the authors used generative AI tools only to polish English writing, check grammar, and improve exposition. Generative AI tools were not used to design the method, derive the theory, conduct the experiments, or analyze the results. All AI-assisted text was reviewed by the authors, who take responsibility for the final content of this work.

Reproducibility Statement The SetGRPO method, execution procedure, local-update geometry, coefficient-recovery rule, and risk-selection rule are described in Section 3. Appendix C specifies the training data and splits, evaluation benchmarks, model and training configurations, grader protocol, and calibration settings. Appendix E provides the complete training algorithm, while Appendix A gives the proofs of the finite-sample coverage guarantee, the local projection formula, and simultaneous risk control. Additional quantitative and qualitative analyses are reported in Appendices D and F. Subject to the licenses and usage restrictions of the underlying datasets and models, we will release the implementation, configuration files, data-preparation scripts, and evaluation scripts. Results involving randomness should be regenerated using the reported data splits, training seeds, frozen calibration records, and evaluation checkpoints.

REFERENCES

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International conference on learning representations*, volume 2024, pp. 55198–55218, 2024.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

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

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip J.K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Survey Certification, Featured Certification.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

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

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pp. 2151–2159. PMLR, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Seungone Kim, Jay Shin, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Ryan Shin, Sungdong Kim, James Thorne, Minjoon Seo, et al. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, volume 2024, pp. 29927–29962, 2024.

Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 193–203, 2023.

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, 2025.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018.

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

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

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591, 2020.

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

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of machine learning research*, 9(3), 2008.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9440–9450, 2024a.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024b.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. Flask: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations*, volume 2024, pp. 55361–55414, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Chen Wei, Xie Yan, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for open-ended LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026.

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

Appendix

for ***"SetGRPO: Calibrating Grader-Induced Policy Optimization"***

Catalogs

Appendix A Proof

A.1 First-order Propagation of Reward Perturbations
A.2 Proof of Finite-Sample Coverage for the Joint Coefficient Set
A.3 Whitened Geometry and Comparison with Euclidean Distance
A.4 Derivation of the Local Projection Formula
A.5 Existence and Feasibility of the Recovery Path
A.6 Proof of Simultaneous Risk Control under Threshold Selection
A.7 Monotonicity of Thresholded Recovery

Appendix B Full Related Work

Appendix C Implementation Details

C.1 Training Data and Splits
C.2 Evaluation Benchmarks
C.3 Models and Training Configuration
C.4 Graders and Scoring Protocol
C.5 Calibration and Selection Configuration
C.6 Additional Method Specifications

Appendix D Additional Quantitative Analysis

D.1 Same-Backbone Comparison with RubricHub
D.2 Additional Recovery and Selection Diagnostics
D.3 Complete Ablation Results
D.4 Affine-Invariance Sanity Check
D.5 Local-Update Replay Validation

Appendix E Algorithm of Training Procedure

Appendix F Additional Qualitative Analysis

F.1 Backbone-Specific Preliminary Diagnostics
F.2 Joint Coefficient-Residual Structure
F.3 Empirical Coverage and Risk Selection
F.4 Checkpoint-wise Validation under Policy Drift
F.5 Threshold–Recovery Trade-off
F.6 Margin and Rejection Diagnostics
F.7 Projection Fidelity
F.8 Recovery Along the Coefficient Path
F.9 Representative Cases

A Proof

A.1 First-order Propagation of Reward Perturbations

The matrix $\mathbf{S}_x^{\mathrm{coef}}$ in the main text arises from the first-order propagation of reward perturbations through the training mapping. Assume that T_{train} is differentiable at $\mathbf{r}_x$, and let $\boldsymbol{\varepsilon}_x$ denote a sufficiently small reward perturbation. Its Taylor expansion gives

$$T_{\mathrm{train}}(\mathbf{r}_x + \boldsymbol{\varepsilon}_x) - T_{\mathrm{train}}(\mathbf{r}_x) = \mathbf{J}_x \boldsymbol{\varepsilon}_x + o(\|\boldsymbol{\varepsilon}_x\|_2), \qquad \mathbf{J}_x = \nabla_{\mathbf{r}} T_{\mathrm{train}}(\mathbf{r})\big|_{\mathbf{r}=\mathbf{r}_x}. \tag{1}$$

If the reference scale of the reward perturbation is specified by $\mathbb{E}[\boldsymbol{\varepsilon}_x] = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{\varepsilon}_x) = \mathbf{I}_K$, then the covariance of the coefficient perturbation under the first-order approximation is

$$\mathrm{Cov}[T_{\mathrm{train}}(\mathbf{r}_x + \boldsymbol{\varepsilon}_x) - T_{\mathrm{train}}(\mathbf{r}_x)] \approx \mathbf{J}_x \mathbf{J}_x^\top, \tag{2}$$

where diagonal entries characterize the local perturbation scales of the training coefficients, while the off-diagonal entries characterize the coupled effects of a shared reward perturbation on the coefficients of different responses. If the reward perturbation has a general covariance $\boldsymbol{\Sigma}_x^{\mathrm{reward}}$, the corresponding first-order shape matrix is $\mathbf{J}_x \boldsymbol{\Sigma}_x^{\mathrm{reward}} \mathbf{J}_x^\top$The choice $\mathbf{J}_x \mathbf{J}_x^\top$ in the main text corresponds to a unit-scale reference perturbation. The additional term $\lambda \mathbf{I}_K$ is included only to ensure stable matrix inversion and distance computation when the matrix is degenerate or nearly singular. It does not alter the coupled structure determined by the Jacobian. Thus, $\mathbf{S}_x^{\mathrm{coef}}$ defines the set shape, while conformal calibration of paired residuals determines its coverage radius.

A.2 Proof of Finite-Sample Coverage for the Joint Coefficient Set

This section proves the finite-sample coverage property of Eq. 5. The geometric construction of the set must be distinguished from its coverage guarantee. The center $\hat{\mathbf{a}}_x$ is obtained from the first score, the positive definite matrix $\mathbf{S}_x^{\mathrm{coef}}$ determines the ellipsoid's directions and relative scales, and the radius Q_α determines its coverage range. Here, exchangeability means that, under the same data construction, response sampling, and scoring protocol, the joint distribution of the calibration groups and the response groups to be covered is invariant under any permutation of the group indices. Consequently, scores computed under the same scoring rule have exchangeable ranks, which provides the finite-sample marginal coverage guarantee for Q_α.

This argument does not require different response groups to use the same center, Jacobian, or shape matrix. It is sufficient that $\hat{\mathbf{a}}_z$, $\mathbf{J}_z$, and $\mathbf{S}_z^{\mathrm{coef}}$ are computed by the same rule, fixed before coverage calibration and equivariant to permutations of the response-group indices. When the joint distribution of the group-level objects is permuted, the scores ρ_z are permuted accordingly. Thus, even when each response group has different local geometry, $\rho_1, \ldots, \rho_n, \rho_{n+1}$ remain exchangeable as required by the rank-statistic argument.

Because JRSC adds regularization to the shape matrix, we assume below that $\mathbf{S}_z^{\mathrm{coef}}$ is positive definite. Its inverse and the corresponding ellipsoidal geometry are therefore well defined. For any coverage calibration group or response group to be covered, let the difference between the training coefficients from the second and first scores be $\boldsymbol{\delta}_z = \mathbf{a}_z^\star - \hat{\mathbf{a}}_z$and define its Mahalanobis score as,

$$\rho_z = \sqrt{\boldsymbol{\delta}_z^\top (\mathbf{S}_z^{\mathrm{coef}})^{-1} \boldsymbol{\delta}_z}. \tag{3}$$

Thus, $\rho_z \le Q_\alpha$ means that $\mathbf{a}_z^\star$ lies in the joint coefficient set centered at $\hat{\mathbf{a}}_z$ with shape $\mathbf{S}_z^{\mathrm{coef}}$ and radius Q_α. It remains to show that this event has probability at least $1 - \alpha$.

Let the n coverage calibration scores be ordered as $\rho_{(1)} \le \cdots \le \rho_{(n)}$ and define $k = \lceil (n+1)(1-\alpha) \rceil$. When $k \le n$, the coverage radius in the main text is $Q_\alpha = \rho_{(k)}$. When $k = n+1$, the calibration scores do not contain the required order statistic. In this case, the implementation adopts the conservative convention $Q_\alpha = +\infty$, thereby retaining all possible coefficients.

Now include a response group to be covered that satisfies the same exchangeability condition as the coverage calibration groups. Denote its score by ρ_{n+1}, and let R_{n+1} denote its rank among $\rho_1, \ldots, \rho_{n+1}$. In the absence of ties, exchangeability implies that R_{n+1} is uniformly distributed on $1, \ldots, n+1$:

$$\Pr(R_{n+1} \le k) = \frac{k}{n+1} \ge 1 - \alpha. \tag{4}$$

When tied scores are handled using a conservative order statistic, the same coverage lower bound remains valid. If $R_{n+1} \le k$, the score of the group to be covered exceeds at most $k-1$ scores. Thus, when $k \le n$, $\rho_{n+1} \le \rho_{(k)} = Q_\alpha$When $k = n+1$, the same relation holds directly because $Q_\alpha = +\infty$. Therefore,

$$\Pr\left(\rho_{n+1} \le Q_\alpha\right) \ge 1 - \alpha. \tag{5}$$

Finally, $\rho_{n+1} \le Q_\alpha$ is exactly the condition that the reference coefficient of the response group to be covered belongs to its joint coefficient set:

$$\left(\mathbf{a}_{n+1}^\star - \hat{\mathbf{a}}_{n+1}\right)^\top \left(\mathbf{S}_{n+1}^{\mathrm{coef}}\right)^{-1} \left(\mathbf{a}_{n+1}^\star - \hat{\mathbf{a}}_{n+1}\right) \le Q_\alpha^2. \tag{6}$$

Hence, $\Pr(\mathbf{a}^{\star}_{n+1} \in \mathcal{C}_{n+1}) \geq 1-\alpha$. Therefore, under exchangeability, $\mathcal{C}_x$ covers the reference coefficient of the response group to be covered with marginal probability at least $1-\alpha$. Once formal training begins, Q_α is fixed using the coverage calibration set and is not updated based on the training process or test results.

A.3 Whitened Geometry and Comparison with Euclidean Distance

This section supplements the geometric relationship between the Mahalanobis score and Euclidean distance. The conclusion is conditional. The Mahalanobis score does not provide stronger coverage guarantees under every distribution. Its advantage arises when $\mathbf{S}_x^{\text{coef}}$ captures the joint uncertainty induced by group normalization, allowing the constructed set to match its directional scales and correlation structure. Let $\mathbf{S}_x^{\text{coef}}$ be positive definite and consider its spectral decomposition:

$$\mathbf{S}_x^{\text{coef}} = \mathbf{U}_x \operatorname{diag}(s_{x,1}, \ldots, s_{x,K}) \mathbf{U}_x^{\top}, \qquad s_{x,j} > 0. \tag{7}$$

For any candidate coefficient vector, let its residual be $\mathbf{e}_x(\mathbf{a}) = \mathbf{a} - \hat{\mathbf{a}}_x$, and define the whitened residual as,

$$\boldsymbol{\xi}_x(\mathbf{a}) = (\mathbf{S}_x^{\text{coef}})^{-1/2} \mathbf{e}_x(\mathbf{a}). \tag{8}$$

By the definition of the square root of a positive definite matrix, the Mahalanobis score in the main text satisfies,

$$\rho_x(\mathbf{a}) = \sqrt{\mathbf{e}_x(\mathbf{a})^{\top} (\mathbf{S}_x^{\text{coef}})^{-1} \mathbf{e}_x(\mathbf{a})} = \|\boldsymbol{\xi}_x(\mathbf{a})\|_2, \tag{9}$$

and therefore,

$$\mathcal{C}_x = \left\{ \mathbf{a} : \left\| (\mathbf{S}_x^{\text{coef}})^{-1/2} (\mathbf{a} - \hat{\mathbf{a}}_x) \right\|_2 \leq Q_\alpha \right\}. \tag{10}$$

Thus, the Mahalanobis ellipsoid is a Euclidean ball of radius Q_α in whitened coordinates. After mapping back to the original coefficient space, $\mathbf{S}_x^{\text{coef}}$ determines the scaling of this ball along different directions.

This can be seen directly from the residuals along the eigenvector directions. Let $\mathbf{e} = t\mathbf{u}_{x,j}$, where $\mathbf{u}_{x,j}$ is the jth eigenvector of $\mathbf{S}_x^{\text{coef}}$. Then

$$\rho_x(\mathbf{a}) = \frac{|t|}{\sqrt{s_{x,j}}}, \qquad \|\mathbf{e}\|_2 = |t|. \tag{11}$$

Thus, under the same threshold Q_α, the allowable physical displacement along this direction is $|t| \leq Q_\alpha \sqrt{s_{x,j}}$ Directions with smaller uncertainty receive stricter penalties, whereas directions with larger uncertainty permit larger shifts. When $\mathbf{S}_x^{\text{coef}}$ has nonzero off-diagonal entries, its eigenvectors also preserve the correlation structure among the coefficients of different responses. Euclidean distance cannot perform both corrections simultaneously.

We can also compare the set sizes required to cover the same ellipsoid. Let $s_{x,\max} = \max_j s_{x,j}$ and assume $Q_\alpha < \infty$. The smallest concentric Euclidean ball containing $\mathcal{C}_x$ has radius $Q_\alpha \sqrt{s_{x,\max}}$. If v_K denotes the volume of the unit Euclidean ball in $\mathbb{R}^K$, then

$$\operatorname{Vol}(\mathcal{C}_x) = v_K Q_\alpha^K \sqrt{\det(\mathbf{S}_x^{\text{coef}})}. \tag{12}$$

The ratio between the volume of this Euclidean ball and that of the ellipsoid is below,

$$\frac{s_{x,\max}^{K/2}}{\sqrt{\det(\mathbf{S}_x^{\text{coef}})}} = \prod_{j=1}^{K} \sqrt{\frac{s_{x,\max}}{s_{x,j}}} \geq 1. \tag{13}$$

Equality holds if and only if all eigenvalues are identical, that is, $\mathbf{S}_x^{\text{coef}} = c\mathbf{I}_K$. Therefore, when covering the anisotropic uncertainty described by $\mathbf{S}_x^{\text{coef}}$, a Euclidean ball expands all directions according to the widest direction, whereas a Mahalanobis ellipsoid expands each direction according to its corresponding scale. The latter therefore provides a more faithful set geometry and may offer better coverage efficiency.

Moreover, the Mahalanobis score is affine invariant. For any invertible matrix $\mathbf{A}$, let $\mathbf{b} = \mathbf{A}\mathbf{a} + \mathbf{c}$, $\hat{\mathbf{b}} = \mathbf{A}\hat{\mathbf{a}} + \mathbf{c}$, and $\mathbf{S}_b = \mathbf{A}\mathbf{S}_x^{\text{coef}}\mathbf{A}^{\top}$. Then

$$(\mathbf{b} - \hat{\mathbf{b}})^{\top} \mathbf{S}_b^{-1} (\mathbf{b} - \hat{\mathbf{b}}) = (\mathbf{a} - \hat{\mathbf{a}}_x)^{\top} (\mathbf{S}_x^{\text{coef}})^{-1} (\mathbf{a} - \hat{\mathbf{a}}_x). \tag{14}$$

Thus, the score is invariant to invertible linear reparameterizations of the coefficient coordinates, whereas ordinary Euclidean distance generally does not have this property.

Conditional Efficiency Under an Elliptical Residual Model. The potential efficiency of the Mahalanobis set relative to a Euclidean ball can be characterized under an explicit residual model. Assume that the reference coefficient residual satisfies,

$$\boldsymbol{\delta}_x = (\mathbf{S}_x^{\text{coef}})^{1/2}\boldsymbol{\xi}_x, \tag{15}$$

where $\boldsymbol{\xi}_x$ has a spherically symmetric density $f(\boldsymbol{\xi}) = h(\|\boldsymbol{\xi}\|_2)$, with h nonincreasing in the radius. If the radial distribution is continuous, let $q_{1-\alpha}$ satisfy $f(\boldsymbol{\xi}) = h(\|\boldsymbol{\xi}\|_2)$ and define

$$\mathcal{E}_M(q_{1-\alpha}) = \left\{\mathbf{a} : (\mathbf{a} - \hat{\mathbf{a}}_x)^\top (\mathbf{S}_x^{\text{coef}})^{-1}(\mathbf{a} - \hat{\mathbf{a}}_x) \le q_{1-\alpha}^2\right\}. \tag{16}$$

In whitened coordinates, $\mathcal{E}_M(q_{1-\alpha})$ is a Euclidean ball of radius $q_{1-\alpha}$ whose boundary is a level set of constant density. Because h is nonincreasing in the radius, this ball is a highest-density region for the specified probability mass. If a set removes part of the ball and adds an equal volume outside the ball, the added region has no higher density, so its coverage probability cannot increase. Consequently, under this elliptical residual model, any measurable set with coverage at least $1-\alpha$, including any Euclidean ball, satisfies

$$\text{Vol}(\mathcal{B}) \ge \text{Vol}(\mathcal{E}_M(q_{1-\alpha})) = v_K q_{1-\alpha}^K \sqrt{\det(\mathbf{S}_x^{\text{coef}})}, \tag{17}$$

where v_K is the volume of the unit Euclidean ball in $\mathbb{R}^K$. This result concerns minimum-volume efficiency at a fixed coverage level. It does not provide a stronger coverage guarantee than Euclidean distance. The Q_α used in this paper is a conformal order statistic calibrated under exchangeability and is used to guarantee finite-sample coverage of the joint coefficient set.

This comparison has an important applicability boundary. When $\mathbf{S}_x^{\text{coef}} = c\mathbf{I}_K$, the Mahalanobis score is merely a constant rescaling of Euclidean distance, so the two are equivalent. Under exchangeability, nonconformity scores based on Euclidean distance can likewise be conformally calibrated to obtain marginal coverage.

A.4 Derivation of the Local Projection Formula

This section provides a complete derivation of Eq. 7. We first transform the ellipsoidal coefficient set into a Euclidean ball, minimize the linear projection over this ball, and then express the result in the original local update space.

Given a candidate direction $\mathbf{d}$, $\Psi_x^q(\mathbf{d})$ is the minimum projection of the local update set along $\mathbf{d}$. Since $\mathcal{C}_x$ is an ellipsoid and $\mathbf{u}_x^q(\mathbf{a})$ is affine in $\mathbf{a}$, this minimization can be reduced to a linear optimization problem over a Euclidean ball. Let $\mathbf{S}_x = \mathbf{S}_x^{\text{coef}}$, $\mathbf{e} = \mathbf{a} - \hat{\mathbf{a}}_x$, and $\mathbf{z} = \mathbf{S}_x^{-1/2}\mathbf{e}$. Then $\mathbf{a} \in \mathcal{C}_x$ is equivalent to

$$\|\mathbf{z}\|_2 \le Q_\alpha, \qquad \mathbf{a} = \hat{\mathbf{a}}_x + \mathbf{S}_x^{1/2}\mathbf{z}. \tag{18}$$

The derivation below assumes that $\mathbf{S}_x^{\text{coef}}$ is positive definite. This condition is ensured by the regularization term or, alternatively, by the full-rank property of the Jacobian. With this positive-definiteness condition in place, substituting $\mathbf{a} = \hat{\mathbf{a}}_x + \mathbf{S}_x^{1/2}\mathbf{z}$ into Eq. 6 gives

$$\mathbf{d}^\top \mathbf{u}_x^q(\mathbf{a}) = \mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x) + \mathbf{d}^\top \mathbf{P}_x^q \mathbf{G}_x^q \mathbf{S}_x^{1/2}\mathbf{z} = \underbrace{\mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x)}_{\text{center projection}} + \underbrace{\left(\mathbf{b}_x^q(\mathbf{d})\right)^\top \mathbf{z}}_{\text{perturbation projection}}, \tag{19}$$

where $\mathbf{b}_x^q(\mathbf{d}) = \mathbf{S}_x^{1/2}(\mathbf{G}_x^q)^\top(\mathbf{P}_x^q)^\top\mathbf{d}$. The center projection is the projection onto $\mathbf{d}$ of the update induced by the center coefficient $\hat{\mathbf{a}}_x$. The perturbation projection represents the effect of coefficient perturbations within the ellipsoid on this projection. The vector $\mathbf{b}_x^q(\mathbf{d})$ is the coefficient vector obtained by mapping the candidate direction to the standardized variable $\mathbf{z}$. The remaining problem is therefore to minimize a linear function of $\mathbf{z}$ over a Euclidean ball of radius Q_α:

$$\Psi_x^q(\mathbf{d}) = \min_{\|\mathbf{z}\|_2 \le Q_\alpha} \left[\mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x) + \left(\mathbf{b}_x^q(\mathbf{d})\right)^\top \mathbf{z}\right]. \tag{20}$$

At this point, only the coefficient perturbation depends on $\mathbf{z}$. We therefore minimize the linear function $\left(\mathbf{b}_x^q(\mathbf{d})\right)^\top \mathbf{z}$ over the feasible region $\|\mathbf{z}\|_2 \le Q_\alpha$. By the Cauchy–Schwarz inequality, for any $\mathbf{z}$ satisfying this constraint,

$$\left(\mathbf{b}_x^q(\mathbf{d})\right)^\top \mathbf{z} \ge -\|\mathbf{b}_x^q(\mathbf{d})\|_2 \|\mathbf{z}\|_2 \ge -Q_\alpha \|\mathbf{b}_x^q(\mathbf{d})\|_2. \tag{21}$$

When $\mathbf{b}_x^q(\mathbf{d}) \neq \mathbf{0}$, set $\mathbf{z} = -Q_\alpha \mathbf{b}_x^q(\mathbf{d})/\|\mathbf{b}_x^q(\mathbf{d})\|_2$Then $\|\mathbf{z}\|_2 = Q_\alpha$, and the linear term attains its lower bound,

$$\left(\mathbf{b}_x^q(\mathbf{d})\right)^\top \mathbf{z} = -Q_\alpha \|\mathbf{b}_x^q(\mathbf{d})\|_2. \tag{22}$$

If $\mathbf{b}_x^q(\mathbf{d}) = \mathbf{0}$, the linear term is identically zero. Substituting the minimum value of the linear term into $\Psi_x^q(\mathbf{d})$ and expanding $\|\mathbf{b}_x^q(\mathbf{d})\|_2$ gives

$$\Psi_x^q(\mathbf{d}) = \mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x) - Q_\alpha \|\mathbf{b}_x^q(\mathbf{d})\|_2 = \underbrace{\mathbf{d}^\top \mathbf{u}_x^q(\hat{\mathbf{a}}_x)}_{\text{center projection}} - \underbrace{Q_\alpha \sqrt{\mathbf{d}^\top \mathbf{P}_x^q \mathbf{G}_x^q \mathbf{S}_x^{\text{coef}} (\mathbf{G}_x^q)^\top (\mathbf{P}_x^q)^\top \mathbf{d}}}_{\text{uncertainty penalty}}, \tag{23}$$

where the center projection is the projection of the update induced by the center coefficient $\hat{\mathbf{a}}_x$ onto direction $\mathbf{d}$. The worst-case uncertainty penalty is the maximum amount by which coefficient uncertainty within the set can reduce this projection. Since $\mathbf{h}_x^q$ does not depend on the coefficients, it appears only in the center update term and does not affect the worst-case uncertainty penalty. If $\mathbf{b}_x^q(\mathbf{d}) = \mathbf{0}$, the worst-case uncertainty penalty is zero, and the expression remains valid.

If the coverage event $\mathbf{a}_x^\star \in \mathcal{C}_x$ holds, then for any $\mathbf{d}$, $\Psi_x^q(\mathbf{d}) \leq \mathbf{d}^\top \mathbf{u}_x^q(\mathbf{a}_x^\star)$In the policy-update space, setting $\mathbf{d} = \mathbf{u}_x^{\text{full}}(\mathbf{a})$ gives,

$$\Psi_x^{\text{full}}\left(\mathbf{u}_x^{\text{full}}(\mathbf{a})\right) \leq \left\langle \mathbf{u}_x^{\text{full}}(\mathbf{a}), \mathbf{u}_x^{\text{full}}(\mathbf{a}_x^\star) \right\rangle. \tag{24}$$

Therefore, if the candidate update is nonzero and$\Psi_x^{\text{full}}(\mathbf{u}_x^{\text{full}}(\mathbf{a})) > 0$ its inner product with the reference update is positive, so no direction reversal occurs. In reward space, if $\Psi_x^{\text{reward}}(\mathbf{u}_x^{\text{reward}}(\mathbf{a})) > 0$ then every possible reward update in the set has a positive projection along the candidate direction.

A.5 Existence and Feasibility of the Recovery Path

Eq. 8 first finds the anchor with the minimum local update norm in the joint coefficient set $\mathcal{C}_x$, and then gradually recovers training progress along the line segment between this anchor and the training coefficients $\hat{\mathbf{a}}_x$. Since $\mathcal{C}_x$ is a nonempty compact convex set and the local update is continuous in the coefficients, the anchor optimization attains a solution. Moreover, because $\hat{\mathbf{a}}_x \in \mathcal{C}_x$, convexity ensures that the entire line segment remains in $\mathcal{C}_x$. Therefore, the recovery path and the maximum feasible recovery ratio in Eq. 8 are well defined.

Assume that $\mathbf{S}_x^{\text{coef}}$ is positive definite and $Q_\alpha < \infty$. By Eq. 5, $\mathcal{C}_x$ is a positive definite ellipsoid with finite radius. Positive definiteness ensures that the quadratic-form constraint defines a closed set, finite radius makes the set bounded, and the ellipsoid is convex. Hence, $\mathcal{C}_x$ is a nonempty compact convex set in $\mathbb{R}^K$. Eq. 6 maps a coefficient vector to the local update $\mathbf{u}_x^{\text{full}}(\mathbf{a})$. The objective used for anchor optimization is,

$$f_x(\mathbf{a}) = \frac{1}{2} \left\|\mathbf{u}_x^{\text{full}}(\mathbf{a})\right\|_2^2, \tag{25}$$

where $\mathbf{u}_x^{\text{full}}(\mathbf{a})$ is affine in $\mathbf{a}$, f_x is a continuous convex function. The anchor problem is therefore,

$$\mathbf{a}_x^{\text{anc}} \in \arg\min_{\mathbf{a} \in \mathcal{C}_x} f_x(\mathbf{a}). \tag{26}$$

A continuous function attains its minimum on a nonempty compact set, so this optimization problem has at least one solution. Thus, the anchor in Eq. 8 exists, and the recovery path can be defined within $\mathcal{C}_x$. The recovery step connects the conservative anchor to the point-reward training coefficients $\hat{\mathbf{a}}_x$ within $\mathcal{C}_x$. Since $\hat{\mathbf{a}}_x$ is the center of the ellipsoid in Eq. 5 and $\mathbf{a}_x^{\text{anc}} \in \mathcal{C}_x$, both points belong to the same convex set. By the definition of convexity, their line segment is,

$$\mathbf{a}_x(\eta) = (1-\eta)\mathbf{a}_x^{\text{anc}} + \eta \hat{\mathbf{a}}_x, \qquad \eta \in [0,1], \tag{27}$$

where $\mathbf{a}_x(\eta) \in \mathcal{C}_x, \quad \forall \eta \in [0,1]$. Thus, all coefficients along the recovery path remain within the calibrated joint coefficient set $\mathcal{C}_x$. Whether a point on the path can be written back into training requires substituting $\mathbf{a}_x(\eta)$ into Eq. 6 to obtain the local update and then checking the minimum projections in the policy-update space and reward space using Eq. 7. Hence, $\mathcal{C}_x$ specifies the range of training coefficients, while the two minimum projections determine which recovery ratios along the path can be written back into training.

A.6 Proof of Simultaneous Risk Control under Threshold Selection

This section describes how to control the conditional direction-reversal risk of the selected threshold when comparing multiple candidate thresholds simultaneously. Assume that the risk calibration groups are independent and identically distributed, and that the candidate threshold set $\mathcal{T}$ is fixed before risk calibration. For any threshold τ with $N_\tau > 0$, conditional on the number of accepted groups N_τ, the number of direction reversals satisfies,

$$F_\tau \mid N_\tau \sim \mathrm{Binomial}(N_\tau, R_\tau), \tag{28}$$

where $R_\tau = \Pr(Z_{x,\tau} = 1 \mid A_{x,\tau} = 1)$.

To compare all thresholds in $\mathcal{T}$ simultaneously, we allocate the total confidence failure budget γ evenly across thresholds. Let $\delta = \gamma/|\mathcal{T}|$ and define the confidence level for each threshold as $C = 1 - \delta$. The one-sided Clopper–Pearson upper confidence bound is,

$$U_\tau = \begin{cases} \mathrm{Beta}^{-1}\left(C; F_\tau + 1, N_\tau - F_\tau\right), & F_\tau < N_\tau, \\ 1, & F_\tau = N_\tau. \end{cases} \tag{29}$$

When $F_\tau = N_\tau$, every accepted group exhibits a direction reversal, so only the trivial upper bound $U_\tau = 1$ can be given. When $F_\tau < N_\tau$, the above expression is equivalent to solving

$$\sum_{j=0}^{F_\tau} \binom{N_\tau}{j} U_\tau^j (1 - U_\tau)^{N_\tau - j} = \delta. \tag{30}$$

Exact inversion of this tail probability gives $\Pr\left(R_\tau \le U_\tau \mid N_\tau\right) \ge 1 - \delta$.

The above inequality holds separately for each fixed threshold. Different thresholds may use the same risk calibration groups, so the upper confidence bounds need not be independent. Applying the union bound to the failure events yields

$$\Pr\left(\exists \tau \in \mathcal{T} : R_\tau > U_\tau\right) \le \sum_{\tau \in \mathcal{T}} \Pr\left(R_\tau > U_\tau\right) \quad \le |\mathcal{T}|\delta = \gamma. \tag{31}$$

Therefore,$\Pr\left(R_\tau \le U_\tau,\ \forall \tau \in \mathcal{T}\right) \ge 1 - \gamma$.In other words, with probability at least $1 - \gamma$, the true conditional risk of every candidate threshold does not exceed its corresponding upper confidence bound. On this event, every threshold satisfying $N_\tau > 0$ and $U_\tau \le \beta$ also satisfies $R_\tau \le U_\tau \le \beta$. Eq. 9 maximizes Y_τ only over these thresholds. Therefore, whenever the risk-feasible set is nonempty, the selected threshold satisfies $R_{\tau_{\mathrm{sel}}} \le \beta$. If no threshold satisfies these conditions, τ_{sel} is not defined and subsequent training is disabled.

A.7 Monotonicity of Thresholded Recovery

Eq. 8 restricts the feasible points along the recovery path using the full-space minimum projection threshold. This section examines how changes in this threshold affect the recovery ratio, acceptance event, and cumulative recovery of an individual response group. For a fixed response group x, let $\mathcal{F}_x(\tau)$ denote the set of recovery ratios satisfying all constraints in Eq. 8. Let $\tau_2 \ge \tau_1$, and take any $\eta \in \mathcal{F}_x(\tau_2)$. By definition,

$$\Psi_x^{\mathrm{full}}\left(\mathbf{u}_x^{\mathrm{full}}(\mathbf{a}_x(\eta))\right) \ge \tau_2. \tag{32}$$

Since $\tau_2 \ge \tau_1$, it follows directly that,

$$\Psi_x^{\mathrm{full}}\left(\mathbf{u}_x^{\mathrm{full}}(\mathbf{a}_x(\eta))\right) \ge \tau_1. \tag{33}$$

The nonzero-update condition and the reward-space minimum projection condition for η do not depend on the threshold. They therefore remain satisfied at τ_1. Hence, η satisfies every constraint defining $\mathcal{F}_x(\tau_1)$, for every $\tau_2 \ge \tau_1$,

$$\mathcal{F}_x(\tau_2) \subseteq \mathcal{F}_x(\tau_1). \tag{34}$$

When $\mathcal{F}_x(\tau)$ is nonempty, Eq. 8 selects its largest recovery ratio. When $\mathcal{F}_x(\tau)$ is empty, the response group has no nonzero recovery satisfying all constraints. We then set $\eta_x^\star(\tau) = 0$ and $A_{x,\tau} = 0$. Thus, the definition can be written uniformly as,

$$\eta_x^\star(\tau) = \begin{cases} \max\limits_{\eta \in \mathcal{F}_x(\tau)} \eta, & \mathcal{F}_x(\tau) \neq \varnothing, \\ 0, & \mathcal{F}_x(\tau) = \varnothing. \end{cases} \tag{35}$$

Because $\mathcal{F}_x(\tau_2) \subseteq \mathcal{F}_x(\tau_1)$ for $\tau_2 \geq \tau_1$, the feasible recovery ratio at the higher threshold cannot exceed the maximum feasible recovery ratio at the lower threshold. Therefore,$\eta_x^\star(\tau_2) \leq \eta_x^\star(\tau_1)$.

Furthermore, if a response group is accepted at the higher threshold τ_2, then $\mathcal{F}_x(\tau_2) \neq \varnothing$. Since $\mathcal{F}_x(\tau_2) \subseteq \mathcal{F}_x(\tau_1)$, $\mathcal{F}_x(\tau_1)$ must also be nonempty, and the group is therefore accepted at the lower threshold τ_1. Writing the acceptance event as,

$$A_{x,\tau} = \mathbb{I}!\left[\mathcal{F}_x(\tau) \neq \varnothing\right], \tag{36}$$

if $A_{x,\tau_2} = 1$, then $\mathcal{F}_x(\tau_2) \neq \varnothing$. Together with $\mathcal{F}_x(\tau_2) \subseteq \mathcal{F}_x(\tau_1)$, this implies $\mathcal{F}_x(\tau_1) \neq \varnothing$ and hence $A_{x,\tau_1} = 1$. Therefore,

$$A_{x,\tau_2} \leq A_{x,\tau_1}, \qquad \tau_2 \geq \tau_1. \tag{37}$$

Thus, increasing the full-space minimum projection threshold can only preserve or reduce the feasible recovery ratio. It cannot cause a response group with no nonzero feasible recovery ratio to re-enter training. Combining the two monotonicity relations with the recovery contribution of an individual response group, for $\tau_2 \geq \tau_1$ we have $A_{x,\tau_2} \leq A_{x,\tau_1}$ and $\eta_x^\star(\tau_2) \leq \eta_x^\star(\tau_1)$. Since recovery ratios are nonnegative, for every response group,

$$A_{x,\tau_2}\eta_x^\star(\tau_2) \leq A_{x,\tau_1}\eta_x^\star(\tau_1). \tag{38}$$

If the group remains accepted at τ_2, the inequality follows from the monotonicity of the recovery ratio. If it is rejected at τ_2, the left-hand side is zero, so the inequality still holds. Summing over all response groups in the risk calibration set and using the definition of Y_τ in Eq. 9 gives,

$$Y_{\tau_2} \leq Y_{\tau_1}, \qquad \tau_2 \geq \tau_1. \tag{39}$$

Therefore, the accepted-group count, nonzero update rate, and cumulative recovery remain unchanged or decrease as the threshold increases, consistent with the trend in Figure 14. This monotonicity applies only to recovery-related statistics. It does not imply that direction-reversal risk must also decrease monotonically, because the risk depends on the composition of the response groups entering training after the threshold changes. Accordingly, Eq. 9 does not directly select the largest candidate threshold. Instead, it selects the value with the largest cumulative recovery among candidate thresholds satisfying $N_\tau > 0$ and $U_\tau \leq \beta$.

B Full Related Work

Reinforcement Learning for Large Language Model Reasoning and GRPO. Alignment training for large language models initially followed a pipeline of human preference feedback, reward modeling, and policy optimization. Early work collected human preference data to train reward models and used reinforcement learning to optimize language-model policies (Christiano et al., 2017). Subsequent studies applied language-model fine-tuning and reinforcement learning from human feedback to large-scale instruction alignment (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). Constitutional AI further showed how AI feedback can reduce reliance on human annotation (Bai et al., 2022). For policy optimization, PPO provides a stable probability-ratio clipping mechanism for feedback-based language-model training (Schulman et al., 2017), whereas DPO reformulates preference optimization as a direct classification objective and thus avoids explicit reward-model training (Rafailov et al., 2023). However, reward modeling error and overoptimization remain important limitations of RLHF (Casper et al., 2023; Gao et al., 2023).

For mathematical and verifiable reasoning tasks, researchers typically use a verifier to determine whether candidate solutions are correct (Cobbe et al., 2021), or compare the effects of outcome supervision and process supervision on reasoning ability (Uesato et al., 2022). Process supervision further verifies intermediate reasoning steps at a fine granularity, providing a denser training signal for policy optimization than the final answer alone (Lightman et al., 2024; Wang et al., 2024b). Building on this direction, DeepSeekMath uses Group Relative Policy Optimization (GRPO) to compute relative training coefficients from a group of responses sampled for the same problem and optimize the policy without an independent value model (Shao et al., 2024). DeepSeek-R1 extends a similar mechanism to large-scale reasoning training (Guo et al., 2025). These studies mainly focus on reasoning performance, training stability, reward design, and computational efficiency, and generally treat grader or reward-model outputs as fixed training signals. In contrast, this paper studies the training consequences of score discrepancies after group centering, scale normalization,

and response-level gradient aggregation, with particular emphasis on how they alter an actual local policy update.

Rubric Feedback and LLM Graders. Open-ended generation tasks often lack a unique reference answer, and a single overall score cannot fully reflect response quality. Recent work has therefore used rubrics to decompose evaluation into multiple interpretable dimensions. G-Eval improves the consistency between automated evaluation and human judgment through a structured evaluation procedure and standardized prompts (Liu et al., 2023). FLASK evaluates responses at a finer granularity based on instruction-related skills (Ye et al., 2024). In translation and other generation tasks, LLMs have also been used to provide competitive automated evaluation results (Kocmi & Federmann, 2023).

LLM-as-a-Judge studies further examine the correspondence between model judgments and human preferences. MT-Bench evaluates model judgment through multi-turn dialogue and analyzes evaluation consistency when different models serve as judges (Zheng et al., 2023). Prometheus trains an open-source evaluator that provides fine-grained feedback according to customized rubrics (Kim et al., 2024). AlpacaFarm studies a training framework for instruction alignment using simulated feedback (Dubois et al., 2023). Other work, such as FairEval, shows that LLM judges may exhibit systematic biases related to response style, length, and form of expression (Wang et al., 2024a).

Beyond the evaluator itself, RewardBench assesses the discriminative ability and limitations of reward models across chat, reasoning, and safety scenarios (Lambert et al., 2025). RubricHub and RuscaRL further apply structured rubric feedback to reinforcement learning on open-ended tasks (Li et al., 2026; Zhou et al., 2026). Meanwhile, prior studies show that reward models may undergo overoptimization during policy optimization (Gao et al., 2023), and that LLM judgments can be affected by evaluation position, response length, and form of expression (Zheng et al., 2023; Wang et al., 2024a). These studies mainly measure scoring accuracy, consistency, fairness, or evaluation capability. They do not further trace how grader score discrepancies change the training coefficients of a response group through GRPO's group-wise normalization, or how these coefficient changes jointly determine local update directions with the response-level gradients of the current policy. This paper investigates precisely this propagation process at the training level.

Conformal Prediction and Selective Risk Control. Conformal prediction uses the exchangeability assumption to convert nonconformity scores from calibration samples into prediction sets or intervals, providing finite-sample coverage guarantees without specifying a particular distributional form (Shafer & Vovk, 2008; Angelopoulos & Bates, 2023). Early work focused mainly on regression and prediction intervals (Lei et al., 2018), and was later extended to classification prediction sets and adaptive variants (Romano et al., 2020). Conformalized Quantile Regression applies this idea to covariate-dependent prediction intervals, improving adaptability in heteroscedastic settings (Romano et al., 2019). Subsequent work further extended the calibration target from predictive coverage to general risk functions (Bates et al., 2021; Angelopoulos et al., 2024).

Unlike prediction sets, selective prediction focuses on controlling risk over a subset after accepting or abstaining on samples. SelectiveNet jointly trains a selection function and a prediction model, enabling control of the error rate on the selected subset at a given coverage level (Geifman & El-Yaniv, 2019). For finite-sample estimation of binomial risk, the Clopper–Pearson interval provides a classical exact confidence interval and can be used to construct a one-sided upper confidence bound for a risk probability (Clopper & Pearson, 1934). Together, these methods provide theoretical tools for uncertainty quantification, finite-sample calibration, and post-selection risk control.

SetGRPO draws on these ideas, but calibrates a different object from conventional conformal methods. Rather than constructing sets directly in the prediction-output space, label space, or task-level decision space, it calibrates grader score discrepancies in the joint training-coefficient space actually used by the GRPO optimizer, and then propagates the calibrated result to the policy-induced update space. SetGRPO therefore addresses the within-group correlation of score discrepancies, the geometric mapping from training coefficients to policy updates, and the conditional direction-reversal risk over the set of accepted updates after selective execution.

Table 4: Question-root counts in the training, coverage-calibration, and risk-calibration splits.

Split	Total question roots	Chat	Instruction Following	Medical	Science	Writing
Training	181528	9812	95173	29681	29418	17444
Coverage calibration	27229	1471	14275	4452	4412	2616
Risk calibration	27229	1471	14275	4452	4412	2616

C Implementation Details

C.1 Training Data and Splits

RubricHub consists of query-rubric pairs from five training domains, where each sample pairs a query with its corresponding task rubric. In the Science domain, RaR-science, ResearchQA, and MegaScience provide scientific question answering and research-oriented tasks. In Instruction Following, IFTRAIN provides instruction tasks with explicit constraints. In Writing, LongWriter, LongWriter-Zero, DeepWriting-20K, and LongAlign cover writing tasks with different formats and lengths. In Medical, II-medical provides medical question answering tasks. In Chat, WildChat-1M and LMSYS-Chat-1M provide open-ended multi-turn dialogue. Thus, rollout groups during training cover scientific reasoning, constraint following, writing, medical answering, and open-ended dialogue, with training scores provided by the corresponding rubrics.

We partition these data by problem root into three disjoint splits: training, coverage calibration, and risk calibration. The training split is used for reinforcement learning. The coverage calibration split determines the coverage range of the joint coefficient set. The risk calibration split evaluates post-screening risk and selects the execution threshold. The numbers of problem roots, domain-specific samples, and split assignments for the two backbones are reported in Table 4.

C.2 Evaluation Benchmarks

Evaluation covers ten benchmarks across the Medical, Writing, and Instruction Following domains.

The Medical domain includes four evaluations. HealthBench contains 500 open-ended health dialogues, each paired with a doctor-authored conversation-level rubric that evaluates the coverage, correctness, and safety of responses to specific health questions. The metric is a normalized rubric score. LLMEval-Med contains 667 evaluation rounds covering five task categories: medical knowledge, medical language understanding, medical reasoning, medical safety and ethics, and medical text generation. An LLM judge scores each response using a category checklist, and the metric is the proportion of responses scoring at least 4. MedQA uses 1,273 questions from the US four-option test split to evaluate medical knowledge and clinical problem solving, with accuracy as the metric. MedMCQA uses 4,183 multidisciplinary medical multiple-choice questions from the dev split to evaluate cross-disciplinary medical knowledge, also using accuracy.

The Writing domain includes two evaluations. WritingBench contains 1,000 real-world writing tasks, each paired with five instance-level evaluation criteria covering requirements such as content completeness, style, format, and length. We report the mean score across the five criteria, rescaled to 0–100. Creative Writing v3 contains 32 creative-writing prompts, with three generations per prompt and 96 responses in total. It evaluates creative-writing quality using a rubric score.

The Instruction Following domain includes four evaluations. IFEval contains 541 prompts with programmatically verifiable constraints. It tests whether explicit requirements concerning format, length, keywords, and other properties are satisfied simultaneously, using prompt-level strict accuracy as the metric. The local evaluation file for IFBench contains 294 prompts whose constraint types differ from the verifiable instructions commonly encountered during training. It evaluates generalization to unseen constraints and uses the same prompt-level strict accuracy metric. InfoBench contains 500 instructions and 2,250 decomposed requirements. It evaluates whether models satisfy the individual requirements of complex instructions, using the proportion of satisfied decomposed requirements, denoted as DRFR. MulDimIF contains 1,200 test samples that require models to sat-

isfy multiple verifiable constraints across dimensions. Its metric is prompt-level Overall accuracy, in which all requirements must be satisfied.

C.3 Models and Training Configuration

Table 5 lists the training configurations for the two backbones, including rollout group size, maximum prompt and response lengths, batch size, mini-batch size, optimizer, learning rate, weight decay, KL coefficient, rollout decoding parameters, number of training steps, and checkpoint frequency. All experiments were run on NVIDIA H100 GPUs. The software environment consisted of CUDA 12.8, PyTorch 2.10.0+cu128, vLLM 0.17.0, and verl 0.4.1.dev.

Table 5: Training and rollout configuration for the three backbones.

Configuration	**Qwen3.5-4B**	**Qwen3.5-9B**	**Llama-3.2-3B**
Rollout group size K	8	8	8
Max prompt length	2048	2048	2048
Max response length	4096	4096	4096
Batch size	64	64	64
Mini-batch size	32	32	32
Optimizer	AdamW	AdamW	AdamW
Learning rate	10^{-6}	10^{-6}	10^{-6}
Weight decay	0.01	0.01	0.01
KL coefficient	0.001	0.001	0.001
Rollout decoding (T / top-p / top-k)	1.0 / 1.0 / 20	1.0 / 1.0 / -1	1.0 / 1.0 / 20
Training steps	500	500	500
Checkpoint frequency	10	10	10

C.4 Graders and Scoring Protocol

Throughout the experiments, we use Qwen3-32B as the rubric grader and score each response twice. Both scores use the same decoding configuration: temperature $= 0.7$, top-$p = 0.8$, top-$k = 20$, and a maximum output length of 2048 tokens. During formal training, the first score is used to compute the training reward for the current rollout group. The second score is paired with the first by response ID and is used to determine the coverage range, evaluate risk, and select the threshold. During formal training, the first score is obtained for response groups generated by the current policy. During calibration, both scores are computed on frozen response groups.

C.5 Calibration and Selection Configuration

The paired response groups used for calibration are partitioned by problem root into two disjoint sets: the coverage calibration set $\mathcal{D}_{\text{cov}}$ and the risk calibration set $\mathcal{D}_{\text{risk}}$. The coverage calibration set determines the coverage radius Q_α of the joint coefficient set. The risk calibration set evaluates the acceptance-conditioned direction-reversal risk and cumulative recovery associated with each candidate threshold $\tau \in \mathcal{T}$, which are then used to select the execution threshold τ_{sel}. Here, τ is the threshold for the policy-update-space minimum projection in Eq. 8.

Before formal training, we fix $\alpha = 0.05$, $\beta = 0.08$, $\gamma = 0.05$, and the candidate threshold set $\mathcal{T} = \{0, 0.05, 0.10, 0.20, 0.25, 0.30, 0.50\}$.Calibration is performed separately for the two backbones. Table 6 summarizes the number of calibration response groups, Q_α, τ_{sel}, $N_{\tau_{\text{sel}}}$, $F_{\tau_{\text{sel}}}$, the empirical direction-reversal risk, and its one-sided Clopper–Pearson upper confidence bound for each backbone. During formal training, we use the corresponding backbone-specific values of Q_α and τ_{sel} obtained from calibration.

C.6 Additional Method Specifications

The training-coefficient mapping, joint coefficient set, local update geometry, and recovery search are implemented as follows. For each response group, point and reference rewards are transformed

Table 6: Frozen coverage- and risk-calibration records used during formal training. Each split contains 27,229 held-out question roots, with one frozen rollout group constructed for each root. The reported values are computed on these disjoint calibration groups. ECR denotes empirical conditional risk.

Backbone	Calibration groups		$\mathbf{Q}_\alpha$	Risk selection				
	Coverage calibration	Risk calibration		τ_{sel}	$N_{\tau_{\text{sel}}}$	$F_{\tau_{\text{sel}}}$	ECR	$U_{\tau_{\text{sel}}}$
Qwen3.5-4B	27,229	27,229	1.35	0.25	19,115	1,050	0.0549	0.0577
Qwen3.5-9B	27,229	27,229	1.42	0.25	19,142	1,060	0.0554	0.0582
Llama3.2-3B	27,229	27,229	1.36	0.25	19,122	1,053	0.0551	0.0574

Algorithm 1 Training-Coefficient Mapping

Require: Reward group $\mathbf{r}_x$, group size K, scale tolerance $\varepsilon_{\text{scale}}$
Ensure: Training coefficient $\mathbf{a}_x$
Construct $\mathbf{H}_K \leftarrow \mathbf{I}_K - K^{-1}\mathbf{1}_K\mathbf{1}_K^\top$
Center the rewards: $\mathbf{z}_x \leftarrow \mathbf{H}_K\mathbf{r}_x$
Compute the group scale: $s_x \leftarrow \sqrt{K^{-1}\|\mathbf{z}_x\|_2^2}$
if $s_x \le \varepsilon_{\text{scale}}$ or $\mathbf{z}_x$ contains a non-finite value **then**
    Return $\varnothing$
end if
Return $\mathbf{a}_x \leftarrow \mathbf{z}_x / s_x$

Algorithm 2 Joint Coefficient-Set Construction

Require: Point and reference rewards $\mathbf{r}_x, \mathbf{r}_x^\star$, scale tolerance $\varepsilon_{\text{scale}}$
Ensure: Joint coefficient set $\mathcal{C}_x$
Compute $\hat{\mathbf{a}}_x \leftarrow T_{\text{train}}(\mathbf{r}_x)$ and $\mathbf{a}_x^\star \leftarrow T_{\text{train}}(\mathbf{r}_x^\star)$
if either coefficient is invalid **then**
    Return $\varnothing$
end if
Compute the residual: $\boldsymbol{\delta}_x \leftarrow \mathbf{a}_x^\star - \hat{\mathbf{a}}_x$
Compute the Jacobian $\mathbf{J}_x$ of T_{train} at $\mathbf{r}_x$
Set $\mathbf{V}_x \leftarrow \mathbf{I}_K$ and $\lambda \leftarrow 10^{-8}$
Compute $\mathbf{S}_x^{\text{coef}} \leftarrow \mathbf{J}_x\mathbf{V}_x\mathbf{J}_x^\top + \lambda\mathbf{I}_K$
Construct $\mathcal{C}_x$ according to Eq. 5
Return $\mathcal{C}_x$

independently by T_{train}. The mapping first removes the group mean and then normalizes by the within-group root-mean-square scale:

$$\hat{\mathbf{a}}_x = T_{\text{train}}(\mathbf{r}_x) = \frac{\mathbf{H}_K\mathbf{r}_x}{\sqrt{K^{-1}\|\mathbf{H}_K\mathbf{r}_x\|_2^2}}, \qquad \mathbf{H}_K = \mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top. \tag{40}$$

The same mapping is applied to $\mathbf{r}_x^\star$ to obtain $\mathbf{a}_x^\star$. If the within-group scale is no larger than $\varepsilon_{\text{scale}}$, or if any intermediate value is non-finite, the group is discarded before coefficient-set construction. No clipping or truncation is applied to the coefficients.

For each valid group, the coefficient residual is $\boldsymbol{\delta}_x = \mathbf{a}_x^\star - \hat{\mathbf{a}}_x$. The Jacobian $\mathbf{J}_x$ is computed by differentiating T_{train} at $\mathbf{r}_x$. We use $\mathbf{V}_x = \mathbf{I}_K$ and $\lambda = 10^{-8}$:

$$\mathbf{S}_x^{\text{coef}} = \mathbf{J}_x\mathbf{V}_x\mathbf{J}_x^\top + \lambda\mathbf{I}_K. \tag{41}$$

The joint coefficient set $\mathcal{C}_x$ is then constructed according to Eq. 5. Its numerical construction uses the ellipsoid solver configured by the implementation tolerance ε_{qp} and maximum iteration count N_{qp}.

For $q \in \{\text{full}, \text{reward}\}$, response-level gradients are computed from the same frozen policy state and the sampled response group. They are transformed by the fixed linear map $\mathcal{S}^q$ into a $D = 256$ dimensional representation. The maps, together with their projection matrices, are initialized once using the fixed random seed s_{proj} and reused for all groups and training steps. In the reported

Algorithm 3 Local Update Geometry

Require: Response-level gradients, policy state, fixed maps $\mathcal{S}^{\text{full}}$ and $\mathcal{S}^{\text{reward}}$
Ensure: $\mathbf{G}_x^q$ and $\mathbf{u}_x^q(\mathbf{a})$
for $q \in \{\text{full}, \text{reward}\}$ **do**
  Apply the fixed map $\mathcal{S}^q$ to the response-level gradients
  Stack the transformed gradients into $\mathbf{G}_x^q \in \mathbb{R}^{D\times K}$, with $D = 256$
  Set $\mathbf{P}_x^q \leftarrow \mathbf{I}_D$ and $\mathbf{h}_x^q \leftarrow \mathbf{0}_D$
  Define $\mathbf{u}_x^q(\mathbf{a}) \leftarrow \mathbf{P}_x^q(\mathbf{G}_x^q\mathbf{a} + \mathbf{h}_x^q)$
end for
Return $\{\mathbf{G}_x^q, \mathbf{u}_x^q\}_{q\in\{\text{full},\text{reward}\}}$

Algorithm 4 Discrete Recovery-Ratio Search

Require: Joint coefficient set $\mathcal{C}_x$, anchor solver, local update mappings $\mathbf{u}_x^q$, threshold τ
Ensure: Recovery ratio $\eta_x^\star$ and recovered coefficient $\widetilde{\mathbf{a}}_x$
Solve the anchor problem in $\mathcal{C}_x$ and obtain $\mathbf{a}_x^{\text{anc}}$
Set $\mathcal{E} \leftarrow \{j/256 : j = 0, \ldots, 256\}$
Set $\eta_x^\star \leftarrow 0$
for $\eta \in \mathcal{E}$ **do**
  Compute $\mathbf{a}_x(\eta) \leftarrow (1-\eta)\mathbf{a}_x^{\text{anc}} + \eta\hat{\mathbf{a}}_x$
  Compute $\mathbf{u}_x^{\text{full}}(\mathbf{a}_x(\eta))$ and $\mathbf{u}_x^{\text{reward}}(\mathbf{a}_x(\eta))$
  Compute $\Psi_x^{\text{full}}(\eta)$ and $\Psi_x^{\text{reward}}(\eta)$ according to Eq. 7
  if the update is nonzero and $\Psi_x^{\text{full}}(\eta) \geq \tau$ and $\Psi_x^{\text{reward}}(\eta) > 0$ **then**
    $\eta_x^\star \leftarrow \eta$
  end if
end for
if $\eta_x^\star > 0$ **then**
  $\widetilde{\mathbf{a}}_x \leftarrow \mathbf{a}_x(\eta_x^\star)$
else
  $\widetilde{\mathbf{a}}_x \leftarrow \mathbf{0}_K$
end if
return $\eta_x^\star, \widetilde{\mathbf{a}}_x$

configuration, $\mathbf{P}_x^q = \mathbf{I}_D$ and $\mathbf{h}_x^q = \mathbf{0}$; hence

$$\mathbf{u}_x^q(\mathbf{a}) = \mathbf{P}_x^q\big(\mathbf{G}_x^q\mathbf{a} + \mathbf{h}_x^q\big). \tag{42}$$

The response-level gradients used here are computed after the same token-level clipping and KL terms as the training objective; the optimizer update itself is not included in the geometry calculation.

The safe anchor is obtained by solving the ellipsoid-constrained problem in Eq. 8, using the configured quadratic-program solver, tolerance ε_{qp}, and iteration limit N_{qp}. Recovery is searched over $\mathcal{E} = \{j/256 : j = 0, \ldots, 256\}$. The largest candidate ratio satisfying the nonzero-update condition, $\Psi_x^{\text{full}} \geq \tau$, and $\Psi_x^{\text{reward}} > 0$ is selected. If no candidate satisfies all conditions, the group contributes a zero update.

The implementation is decomposed into four procedures: the training-coefficient mapping in Algorithm 1, joint coefficient-set construction in Algorithm 2, local update geometry computation in Algorithm 3, and discrete recovery-ratio search in Algorithm 4.

D Additional Quantitative Analysis

D.1 Same-Backbone Comparison with RubricHub

In this section, we compare SetGRPO with RubricHub on Qwen3-4B and Qwen3-14B using the same backbone for each comparison. All other experimental settings follow those used for the Qwen3.5 and Llama3.2 experiments, including the data splits, rollout group size, decoding configuration, training steps, generation budget, grader configuration, and evaluation protocol. The backbone-specific implementation details are summarized separately in Tables 8 and 9. Consistent with the main text (detailed in Section 4.2), we report results on ten benchmarks and analyze them across the Medical, Writing, and Instruction Following domains.

Table 7: Same-backbone comparison between RubricHub and SetGRPO on Qwen3-4B and Qwen3-14B. RubricHub results were reproduced by our implementation using its publicly available training procedure. HB: HealthBench; LEM: LLMEval-Med; MQA: MedQA; MCQA: MedMCQA; WB: WritingBench; CW: Creative Writing v3; IFE: IFEval; IFB: IFBench; IB: InfoBench; and MIF: MulDimIF.

Backbone	Method	Medical				Writing		Instruction Following				Macro Avg.
		HB	LEM	MQA	MCQA	WB	CW	IFE	IFB	IB	MIF	
Qwen3-4B	RubricHub	64.34	45.12	**78.38**	**61.88**	66.51	28.47	64.21	24.08	**80.21**	47.35	56.06
	SetGRPO (Ours)	**66.82**	**47.57**	77.91	61.42	**68.97**	**30.18**	**66.83**	**26.49**	79.66	**49.72**	**57.56**
Qwen3-14B	Rubrichub	68.44	53.36	**77.53**	**64.76**	70.42	32.71	69.74	25.61	**82.88**	46.79	59.12
	SetGRPO (Ours)	**70.91**	**54.88**	76.94	64.18	**72.83**	**34.62**	**72.21**	**27.94**	82.17	**49.18**	**60.59**

On Qwen3-4B, SetGRPO achieves a Macro Avg. of 57.56, improving over RubricHub by 1.50 points. In the Medical domain, SetGRPO improves over RubricHub by 2.48 and 2.45 points on HealthBench and LLMEval-Med, respectively, while the differences on MedQA and MedMCQA are 0.47 and 0.46 points. Thus, the gain in the medical domain mainly comes from open-ended medical evaluation, while performance on multiple-choice evaluation remains close to that of RubricHub. SetGRPO represents the correlated training-coefficient shifts caused by shared normalization of within-group score discrepancies as a joint coefficient set. It then combines this set with the response-level gradients of the current policy to determine whether candidate coefficients induce policy updates consistent with the reference update, thereby reducing erroneous or reversed updates caused by score discrepancies.

In the Writing domain, SetGRPO improves over RubricHub by 2.46 and 1.71 points on WritingBench and Creative Writing v3, respectively. For writing tasks that jointly impose requirements on content, style, format, and length, as well as open-ended tasks emphasizing creative quality, SetGRPO retains effective score signals related to writing quality by recovering executable training progress that satisfies the projection conditions. It therefore improves performance on both evaluations. In the Instruction Following domain, SetGRPO improves over RubricHub by 2.62, 2.41, and 2.37 points on IFEval, IFBench, and MulDimIF, respectively, while the difference on InfoBench is 0.55 points. The gains on the first three evaluations indicate improved adherence to explicit, unseen, and multidimensional constraints. For instructions requiring multiple requirements to be satisfied simultaneously, the method first screens executable updates using the joint coefficient set and the projection conditions in the policy-update space, and then uses independent risk calibration to select a unified threshold. This reduces the damage that direction-reversed updates can cause to multi-constraint behavior.

Table 8: Training and rollout configuration for Qwen3-4B and Qwen3-14B.

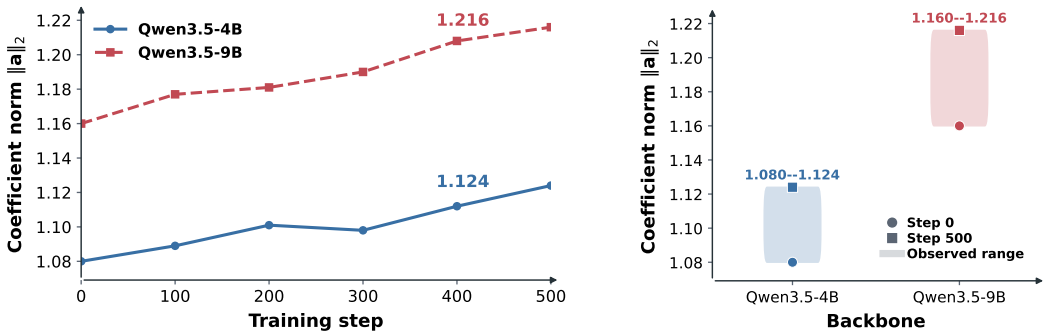
Configuration	Qwen3-4B	Qwen3-14B
Rollout group size K	8	8
Max prompt length	2048	2048
Max response length	4096	4096
Batch size	64	64
Mini-batch size	32	32
Optimizer	AdamW	AdamW
Learning rate	10^{-6}	10^{-6}
Weight decay	0.01	0.01
KL coefficient	0.001	0.001
Rollout decoding (T / top-p / top-k)	1.0 / 1.0 / 20	1.0 / 1.0 / 20
Training steps	500	500
Checkpoint frequency	10	10

On Qwen3-14B, SetGRPO improves the Macro Avg. over RubricHub by 1.47 points. After the joint coefficient set and execution threshold are calibrated separately for the two backbones, SetGRPO achieves consistent overall gains. This indicates that the improvement comes from its treatment of score discrepancies, policy updates, and risk screening rather than dependence on a particular backbone. Overall, SetGRPO jointly represents the training-coefficient shifts caused by within-group score discrepancies, uses the response-level gradients of the current policy to screen and recover executable training progress,

Table 9: Frozen coverage and risk calibration records for Qwen3-4B and Qwen3-14B. ECR denotes empirical conditional risk.

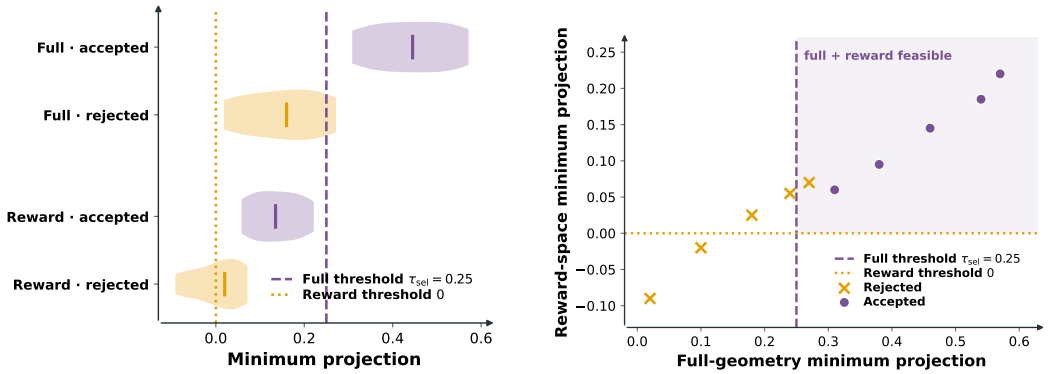
Backbone	Cov.	Risk	Q_α	τ_{sel}	N	F	ECR	U
Qwen3-4B	27,229	27,229	1.32	0.25	19113	1047	0.0548	0.575
Qwen3-14B	27,229	27,229	1.45	0.25	19137	1058	0.0554	0.580

and applies risk calibration to determine a unified execution threshold, thereby improving benchmark performance across backbones and task domains.



(a) Coefficient-norm trajectory. (b) Observed norm range.

Figure 5: Coefficient-norm dynamics during training. The left panel shows the coefficient-norm trajectories of the two backbones, while the right panel summarizes the observed norm range between training steps 0 and 500. The bounded ranges indicate that coefficient recovery does not lead to sustained numerical growth.



(a) Projection distributions. (b) Joint projection constraints.

Figure 6: Projection diagnostics for accepted and rejected updates. The left panel shows the distributions of full-geometry and reward-space minimum projections. The right panel shows their joint constraints; the shaded upper-right region satisfies both the full-geometry threshold τ_{sel} and the positive reward-space projection condition.

D.2 Additional Recovery and Selection Diagnostics

We report the main training curves in Figure 3. This section further reports the coefficient norm, full-space and reward-space minimum projections, and cumulative recovery to examine the scale of recovered coefficients, the projection conditions over the joint coefficient set, and the recovery retained during training.

1) Coefficient-Norm Dynamics. Figure 5 shows the norm of the recovered coefficients throughout training. The coefficient norm remains generally stable and changes moderately with the policy state, without a sustained upward trend. This result is consistent with the recovery range determined by the joint coefficient set and the recovery path from the minimum-update-norm anchor to the point-grader training coefficients. It indicates that the performance gain comes from screening and retaining executable recovery updates rather than continuously enlarging the training coefficients.

2) Minimum-Projection Diagnostics. Figure 6 shows the full-space and reward-space minimum projections. For accepted updates, the full-space minimum projection remains above the selected threshold τ_{sel}, while the reward-space minimum projection remains positive. Response groups that

Table 10: Ablation results for Qwen3.5-4B on ten downstream benchmarks. Parenthetical values show changes relative to Full SetGRPO.

D#	Derivative	Medical				Writing		Instruction Following				Macro Avg.
		HB	LEM	MQA	MCQA	WB	CW	IFE	IFB	IB	MIF	
(a) Coefficient-set construction												
1	Spherical set	64.73(-5.91)	45.49(-4.59)	79.50(-1.05)	63.30(-0.54)	65.79(-6.35)	30.62(-1.61)	64.43(-6.02)	25.17(-3.80)	80.88(-1.44)	46.29(-6.06)	56.62(-3.82)
(b) Local geometry and recovery												
2	Identity geometry	61.40(-8.24)	43.98(-6.10)	79.12(-1.43)	63.16(-0.68)	63.82(-8.32)	29.95(-2.28)	62.30(-8.15)	23.76(-5.21)	80.53(-1.79)	42.78(-9.57)	55.08(-5.18)
3	Safe-anchor-only	63.82(-5.82)	46.32(-3.76)	79.82(-0.73)	63.56(-0.28)	67.08(-5.06)	30.86(-1.37)	65.59(-4.86)	25.93(-3.04)	81.38(-0.94)	45.04(-7.31)	56.94(-3.32)
4	Reward-only recovery	65.92(-3.72)	47.20(-2.88)	80.02(-0.53)	63.62(-0.22)	67.80(-4.34)	31.41(-0.82)	66.34(-4.11)	26.71(-2.26)	81.61(-0.71)	47.17(-5.18)	57.78(-2.48)
(c) Threshold selection and update control												
5	Fixed-threshold	66.39(-3.25)	47.57(-2.51)	79.89(-0.66)	63.71(-0.13)	68.52(-3.62)	31.45(-0.78)	66.95(-3.50)	27.06(-1.91)	81.65(-0.67)	47.91(-4.44)	58.11(-2.15)
6	Matched random discard	61.10(-8.62)	43.88(-6.28)	78.94(-1.69)	63.15(-0.77)	63.31(-8.91)	29.73(-2.58)	62.06(-8.47)	23.92(-5.13)	80.15(-2.24)	42.06(-10.36)	54.83(-5.43)
	SetGRPO (Ours)	**69.64 (+0.00)**	**50.08 (+0.00)**	**80.55 (+0.00)**	**63.84 (+0.00)**	**72.14 (+0.00)**	**32.23 (+0.00)**	**70.45 (+0.00)**	**28.97 (+0.00)**	**82.32 (+0.00)**	**52.35 (+0.00)**	**60.26 (+0.00)**

produce no recovery update fail at least one of these conditions. This result shows that screening jointly checks the policy-update projection under the joint coefficient set $\mathcal{C}_x$ and the reward-space projection, thereby keeping retained updates consistent with the local updates induced by the reference training coefficients $\mathbf{a}_x^\star$ while maintaining a positive projection in reward space.

3) Cumulative Recovery. Figure 7 shows cumulative recovery as a function of training steps. Cumulative recovery continues to increase, but its growth rate varies across training stages, reflecting differences in the number of response groups satisfying the recovery conditions and in their maximum recovery ratios. The curve shows that response groups continue to satisfy the recovery conditions and write back nonzero recovered coefficients throughout formal training. Thus, the retained training progress is not concentrated in only a few training steps.

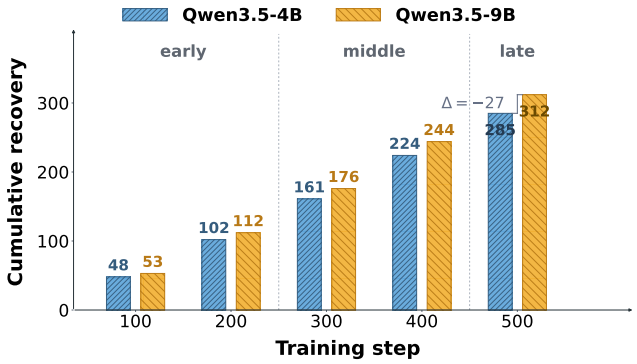


Figure 7: Cumulative recovery over the first 500 training steps for Qwen3.5-4B and Qwen3.5-9B. Shaded regions distinguish the early, middle, and late stages; the bracket reports the final cumulative-recovery gap.

D.3 Complete Ablation Results

This section provides the complete results for the variants in Table 2. All ablations use the same data splits, response-generation configuration, training steps, and evaluation protocol, and report Macro Avg. over ten benchmarks. As shown in Table 10, we first report the scores of each variant on the ten benchmarks and then analyze the results from three perspectives: joint coefficient-set construction, projection and recovery in the policy-update space, and risk-threshold selection.

1) Joint coefficient-set construction. Replacing the joint coefficient set with a uniform-radius set reduces Macro Avg. to 56.80, with lower scores than the full method on all ten benchmarks. After within-group rewards are normalized using the shared center and scale, score discrepancies simultaneously affect the training coefficients of the entire group. The uniform-radius set does not preserve these within-group correlated coefficient shifts and therefore changes the subsequent screening of candidate coefficients. This result shows that the joint coefficient set must preserve the within-group coupled effects reflected by the Jacobian of the training mapping.

2) Strategy-update space and recovery. Removing the current policy's policy-update space reduces the Macro Avg. of Identity geometry to 55.14, below the full method on all ten benchmarks. Without response-level gradients to construct the local update mapping, distances in coefficient space cannot reflect how candidate coefficients affect the current policy update. When only the anchor is retained and the training coefficients are not recovered along the recovery path, Macro Avg. is 57.10, indicating that using only a conservative starting point reduces the retained training progress. When only the reward-space condition is retained, Reward-only recovery achieves a Macro Avg. of 57.90, showing that positive projection in reward space cannot replace the projection

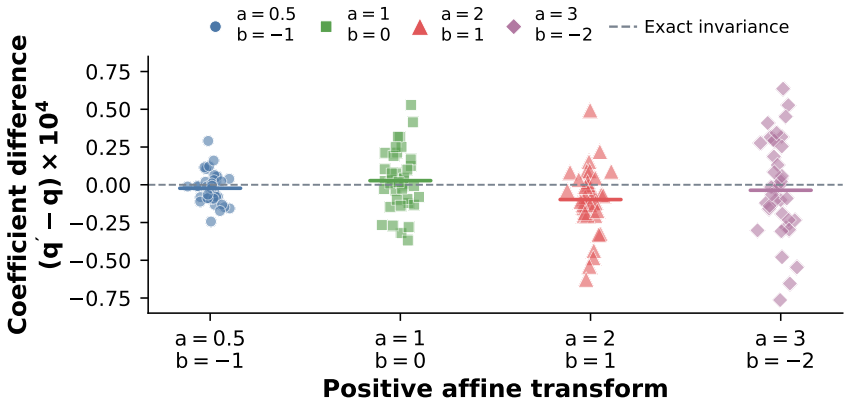


Figure 8: Positive-affine invariance under frozen-policy control. The left plot reports coefficient differences between the original and transformed standardized coefficients. The right table summarizes coefficient cosine, update cosine, decision flips, and the absolute recovery-ratio difference for each transform $\mathbf{r}'_x = a\mathbf{r}_x + b\mathbf{1}_K$ with $a > 0$.

condition in the policy-update space. Together, these variants show that the response-level gradients of the current policy, the recovery path, and the full-space and reward-space minimum projections jointly determine the executable recovery ratio.

3) Threshold selection and matched selection. When risk calibration is replaced with a fixed threshold, Fixed-threshold achieves a Macro Avg. of 58.25, below the full method. A fixed threshold does not evaluate this tradeoff on the risk calibration set and therefore cannot identify a threshold that both satisfies the risk constraint and retains substantial training progress. By contrast, risk calibration selects the threshold with the largest cumulative recovery among the candidate thresholds satisfying the risk upper confidence bound constraint, enabling the full method to achieve higher final performance. Even when Matched random discard retains a similar number of accepted groups, its Macro Avg. is 5.51 points lower. Similar numbers of updates therefore do not yield the same performance, and the gain of the full method comes from screening based on the joint coefficient set and policy-update projection conditions.

Overall, the full method addresses three layers of the problem in sequence. The joint coefficient set represents coefficient shifts caused by normalized within-group score discrepancies. The minimum projections in the policy-update space and the recovery path determine the executable training progress. Risk calibration selects the execution threshold satisfying the direction-reversal risk constraint. The ablations show that removing any one of these components harms performance, risk control, or training progress retention.

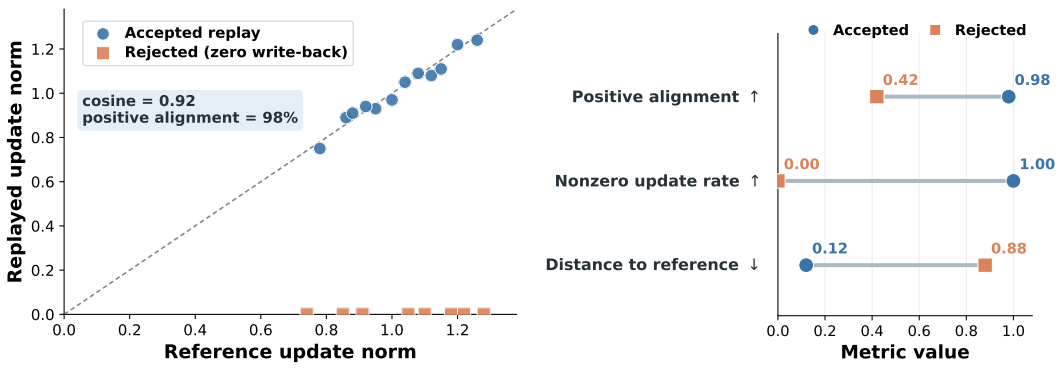
D.4 Affine-Invariance Sanity Check

Training coefficients are determined by the centering and scale normalization of rewards within each group. To test this property, we apply a positive-affine transformation to frozen response groups and policy snapshots:

$$\mathbf{r}'_x = a\mathbf{r}_x + b\mathbf{1}_K, \qquad a > 0. \tag{43}$$

Because a positive-affine transformation does not change within-group standardized rewards, the training coefficients and joint coefficient sets before and after transformation should remain unchanged. Under the same response-level gradients, $\mathbf{G}_x^q$, and $\mathbf{u}_x^q(\mathbf{a})$, with the calibrated risk rule fixed, we compare the local updates, maximum recovery ratios, and response groups entering training under the two reward representations.

Figure 8 shows the coefficient residuals $(\mathbf{a}'_x - \mathbf{a}_x) \times 10^4$ under four positive-affine transformations. The point clouds are centered around zero, and the within-group means are close to zero, indicating that the standardized training coefficients differ only within numerical tolerance. Table 11 reports the corresponding directional and recovery metrics. The coefficient cosine and update cosine range from 0.9997 to 1.0000 and from 0.9995 to 1.0000, respectively. None of the four transformations



(a) Uupdate norm versus reference update norm.

(b) Replay diagnostics by decision status.

Figure 9: Frozen-policy local-update replay validation. The left subfigure compares the norm of the recomputed update with the reference update; the dashed identity line indicates exact agreement, while rejected groups have no nonzero write-back. The right subfigure uses dumbbell comparisons to summarize replay consistency and write-back diagnostics by decision status.

produces a decision flip, and the largest difference in recovery ratio is only 0.8×10^{-3}. Thus, the training coefficients, local updates, and recovery ratios remain free of systematic changes under positive-affine transformations, indicating that the screening results are stable to reward translation and positive rescaling.

Table 11: Positive-affine invariance diagnostics.

Transform	Coeff. cosine	Update cosine	Flip (%)	$\lvert\Delta\eta^\star\rvert$ (10^{-3})
$a = 0.5,\ b = -1$	0.9998	0.9996	0.00	0.6
$a = 1,\ b = 0$	1.0000	1.0000	0.00	0.0
$a = 2,\ b = 1$	0.9999	0.9998	0.00	0.4
$a = 3,\ b = -2$	0.9997	0.9995	0.00	0.8

This invariance follows from the elimination of positive-affine transformations by within-group normalization. For $\mathbf{r}'_x = a\mathbf{r}_x + b\mathbf{1}_K$ with $a > 0$, the centered and scale-normalized rewards remain unchanged. Consequently, the training coefficients obtained from Eq. 1 remain unchanged, and the joint coefficient set constructed from the paired coefficient residuals does not change materially. With the response-level gradients, $\mathbf{G}_x^q$, and $\mathbf{u}_x^q(\mathbf{a})$ of the current policy fixed, the updates and their minimum projections in the policy-update space and reward space also remain stable. Along the same recovery path, $\eta_x^\star$ changes only at the level of floating-point error.

This experiment shows that the training coefficients, local updates, minimum projections, and recovery ratios in SetGRPO are determined by the relative structure of within-group rewards. Translating or positively rescaling all rewards in a group does not systematically change these quantities. In contrast, transformations that do not preserve the within-group reward ordering, such as negative rescaling or response-specific non-affine perturbations, may alter the normalized rewards and consequently change the training coefficients, minimum projections, and recovery ratios.

D.5 Local-Update Replay Validation

To verify whether the updates written back after screening are consistent with the local updates induced by the training coefficients from the second grader score under the frozen policy, we rerun the local-update mapping on the frozen policy and response groups using the saved training coefficients. This process involves neither online training nor additional grader calls. For each response group, we compute the candidate update from Eq. 6 and the saved response-level gradients, then compare it with the local update induced by the training coefficients $\mathbf{a}_x^\star$ obtained from the second grader score. We record the reference cosine, positive reference alignment, nonzero update rate, and distance to reference. This check evaluates the consistency between the recomputed updates after the coefficient mapping and the final write-back state.

Figure 9(a) compares the norms of the recomputed updates and the comparison updates induced by the second grader score. Accepted groups are concentrated near $y = x$, indicating that the retained

training coefficients produce update magnitudes close to those of the comparison updates when Eq. 6 is rerun under the frozen policy. Groups not accepted lie on the horizontal axis because their final write-back coefficients are $\mathbf{0}_K$ and therefore produce no nonzero updates. The accepted groups have a mean reference cosine of 0.92 and positive reference alignment of 98%, whereas the nonzero update rate for groups not accepted is 0. Figure 9(b) shows local-update metrics under different write-back states. Accepted groups have higher reference cosine and positive reference alignment, with a distance to reference of 0.12. For groups not accepted, positive reference alignment is 0.42 and distance to reference is 0.88. Because their final write-back update is zero, the reference cosine is undefined for these groups and is not reported.

These results follow the computational order of the method. Eq. 6 constructs local updates from the training coefficients and the current policy's response-level gradients. Eq. 7 computes the full-space and reward-space minimum projections over the joint coefficient set. Eq. 8 then selects the maximum recovery ratio among candidate ratios satisfying the nonzero-update and projection conditions. Groups satisfying these conditions write back the corresponding recovered coefficients, while all other groups write back $\mathbf{0}_K$. Thus, this replay check supports numerical consistency among the coefficient mapping, projection-based screening, and final write-back.

E Algorithm of Training Procedure

Algorithm 5 presents the SetGRPO training procedure. Formal training consists of two stages: calibration-rule construction and formal optimization. First, we obtain two grader scores for frozen calibration response groups and pair them by response ID. Each score is mapped to training coefficients, from which we construct paired coefficient residuals and the joint coefficient set. We then compare candidate thresholds on an independent risk calibration split, select the risk threshold, and fix Q_α and τ_{sel} before formal training. During formal optimization, training coefficients and response-level gradients are recomputed for each current rollout group. Under the fixed joint coefficient set and risk rule, the maximum feasible recovery ratio is determined, and the recovered coefficients satisfying the screening conditions are written back into the original GRPO objective.

F Additional Qualitative Analysis

F.1 Backbone-Specific Preliminary Diagnostics

In the Preliminary analysis (detailed in Section 2), we showed on Qwen3.5-9B how score discrepancies propagate through the training-coefficient mapping and further affect local update directions. To examine whether this phenomenon also occurs with a smaller backbone, we conduct the same analysis on Qwen3.5-4B. Using the same two grader scores and local-update computation procedure, we analyze score discrepancies, coefficient shifts, and the relationship between coefficient changes and local updates under the current policy.

Figure 10(a) compares the discrepancy between the two scores with the coefficient residual induced by the training mapping on Qwen3.5-4B. The figure contains 4,039 valid response groups. The mean score disagreement is 0.583, and the mean coefficient residual norm is 2.966. The two quantities show an overall positive association, but the points do not collapse onto a single curve, indicating that similar score discrepancies can correspond to coefficient shifts of different magnitudes. Training-coefficient changes therefore also depend on each response's relative position in the within-group reward distribution. This result is consistent with the observation on Qwen3.5-9B. Because T_{train} normalizes using the shared center and scale of the entire reward group, a score change for one response can simultaneously affect the training coefficients of other responses in the same group. Score discrepancies should therefore be modeled as joint shifts of the group-level training coefficients, and the training-coefficient mapping in Eq. 1 must operate at the group level.

Figure 10(b) further compares the coefficient-space cosine with the local-update cosine. Among the 3,038 response groups with valid local updates, the mean coefficient-space cosine is 0.369 and the mean local-update cosine is 0.372, with a direction-reversal rate of 1.9%. This shows that proximity in coefficient space alone does not determine the local update direction. After score discrepancies pass through the response-level gradients of the current policy, they may still produce a direction opposite to the reference local update. According to Eq. 3, response-level gradients determine how coefficient changes are converted into actual policy updates. Thus, the same coefficient change

Algorithm 5 SetGRPO Training Procedure

Input: Training, coverage-calibration, and risk-calibration splits $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{cov}}$, $\mathcal{D}_{\text{risk}}$; initial policy $\boldsymbol{\theta}_0$.
Parameters: $K, \alpha, \beta, \gamma, \mathcal{T}, \mathcal{E}$.
Output: Trained policy $\boldsymbol{\theta}$.

Warm-up calibration
Generate one frozen response group $\mathbf{Y}_z^0$ for each $z \in \mathcal{D}_{\text{cov}} \cup \mathcal{D}_{\text{risk}}$ and obtain two scores $(\mathbf{r}_z, \mathbf{r}_z^\star)$ for the same responses.
for $z \in \mathcal{D}_{\text{cov}} \cup \mathcal{D}_{\text{risk}}$ **do**
  Compute $\hat{\mathbf{a}}_z \leftarrow T_{\text{train}}(\mathbf{r}_z)$, $\mathbf{a}_z^\star \leftarrow T_{\text{train}}(\mathbf{r}_z^\star)$, and $\boldsymbol{\delta}_z \leftarrow \mathbf{a}_z^\star - \hat{\mathbf{a}}_z$.
  Compute $\mathbf{J}_z \leftarrow \nabla_{\mathbf{r}} T_{\text{train}}(\mathbf{r}_z)$ and $\mathbf{S}_z^{\text{coef}} \leftarrow \mathbf{J}_z \mathbf{V}_z \mathbf{J}_z^\top + \lambda \mathbf{I}_K$.
  Compute $\rho_z \leftarrow \sqrt{\boldsymbol{\delta}_z^\top (\mathbf{S}_z^{\text{coef}})^{-1} \boldsymbol{\delta}_z}$.
  Compute the current response-level gradients and construct $\mathbf{G}_z^q$ and $\mathbf{u}_z^q(\mathbf{a})$ for $q \in \{\text{full}, \text{reward}\}$.
end for
Set Q_α to the $\lceil (|\mathcal{D}_{\text{cov}}| + 1)(1-\alpha) \rceil$-th order statistic of $\{\rho_z : z \in \mathcal{D}_{\text{cov}}\}$.
Risk calibration
for $z \in \mathcal{D}_{\text{risk}}$ **do**
  Construct $\mathcal{C}_z$ using Q_α, compute the safe anchor $\mathbf{a}_z^{\text{anc}}$, and define $\mathbf{a}_z(\eta) \leftarrow (1-\eta)\mathbf{a}_z^{\text{anc}} + \eta \hat{\mathbf{a}}_z$ for $\eta \in [0, 1]$.
end for
for $\tau \in \mathcal{T}$ **do**
  for $z \in \mathcal{D}_{\text{risk}}$ **do**
    Compute $\eta_z^\star(\tau)$ using Eq. 8. Set $A_{z,\tau} = 1$ if a nonzero feasible recovery exists; otherwise set $A_{z,\tau} = 0$.
    if $A_{z,\tau} = 1$ **then**
      Compute $Z_{z,\tau} \leftarrow \mathbb{I}\big[\big\langle \mathbf{u}_z^{\text{full}}\big(\mathbf{a}_z(\eta_z^\star(\tau))\big), \mathbf{u}_z^{\text{full}}(\mathbf{a}_z^\star)\big\rangle < 0\big]$.
    end if
  end for
  Compute $N_\tau \leftarrow \sum_z A_{z,\tau}$, $F_\tau \leftarrow \sum_z A_{z,\tau} Z_{z,\tau}$, and $Y_\tau \leftarrow \sum_z A_{z,\tau} \eta_z^\star(\tau)$.
  Compute $U_\tau \leftarrow \text{CPUpper}(F_\tau, N_\tau; 1 - \gamma/|\mathcal{T}|)$.
end for
Select $\tau_{\text{sel}} \in \arg\max_{\tau \in \mathcal{T}} Y_\tau$ subject to $N_\tau > 0$ and $U_\tau \le \beta$.
Freeze Q_α and τ_{sel}.
Formal optimization
for $t = 1, \ldots, T$ **do**
  Sample $\mathcal{B}_t \subset \mathcal{D}_{\text{train}}$ and generate $\mathbf{Y}_x \sim \pi_{\boldsymbol{\theta}_t}^K(\cdot \mid x)$ for $x \in \mathcal{B}_t$.
  for $x \in \mathcal{B}_t$ **do**
    Compute $\mathbf{r}_x$ and $\hat{\mathbf{a}}_x \leftarrow T_{\text{train}}(\mathbf{r}_x)$.
    Compute the current response-level gradients and construct $\mathbf{G}_x^q$ and $\mathbf{u}_x^q(\mathbf{a})$ for $q \in \{\text{full}, \text{reward}\}$.
    Construct $\mathcal{C}_x$ using the frozen Q_α and compute the safe anchor and $\eta_x^\star(\tau_{\text{sel}})$ using Eq. 8.
    Set $A_{x,\tau_{\text{sel}}} = 1$ if a nonzero feasible recovery exists and set $A_{x,\tau_{\text{sel}}} = 0$ otherwise.
    Set

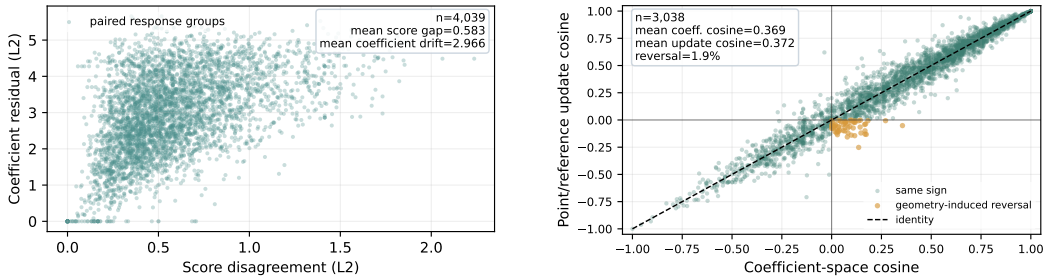
$$\widetilde{\mathbf{a}}_x \leftarrow \begin{cases} \mathbf{a}_x(\eta_x^\star(\tau_{\text{sel}})), & A_{x,\tau_{\text{sel}}} = 1, \\ \mathbf{0}_K, & A_{x,\tau_{\text{sel}}} = 0. \end{cases}$$

    Write $\widetilde{\mathbf{a}}_x$ into the original GRPO coefficient interface and apply the response-token mask.
  end for
  Update $\boldsymbol{\theta}_t$ with the original GRPO objective using the written-back coefficients.
end for

can produce different update directions under different policy states. Accordingly, SetGRPO maps candidate coefficients into the policy-update space and reward space using Eq. 6, and computes the minimum projection over the joint coefficient set using Eq. 7. The direction reversals observed on 4B show that the impact of score discrepancies on training updates must be assessed together with the response-level gradients of the current policy. This phenomenon is not limited to a single backbone.

F.2 Joint Coefficient-Residual Structure

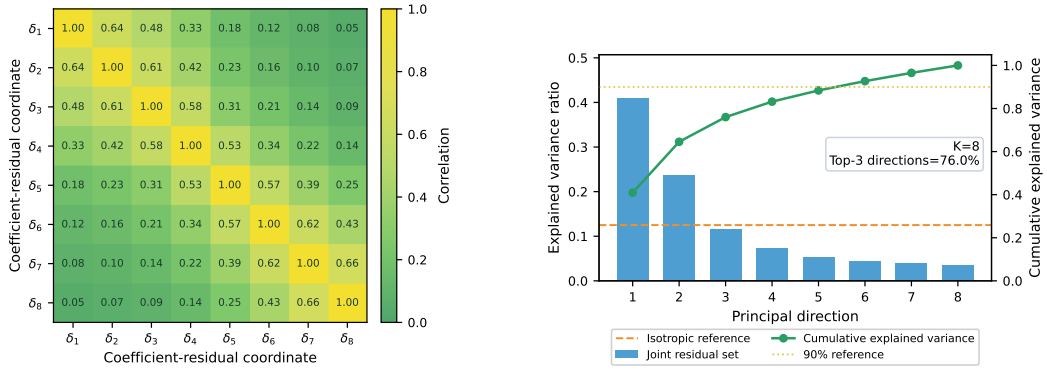
The preliminary analysis (detailed in Section 2) shows that score discrepancies change the training coefficients after group normalization and consequently affect local updates. This section further analyzes the paired coefficient residuals in frozen response groups, $\boldsymbol{\delta}_x = \mathbf{a}_x^\star - \hat{\mathbf{a}}_x$, to examine whether



(a) Score disagreement and coefficient-residual drift.

(b) Coefficient-space and policy-update alignment.

Figure 10: Backbone-specific preliminary diagnostics for Qwen3.5-4B.



(a) Coefficient-residual correlation matrix.

(b) Spectrum and cumulative variance.

Figure 11: Joint structure of coefficient residuals for $K = 8$. The left subfigure reports the coefficient-residual correlation matrix, while the right subfigure shows the explained-variance spectrum and cumulative explained variance.

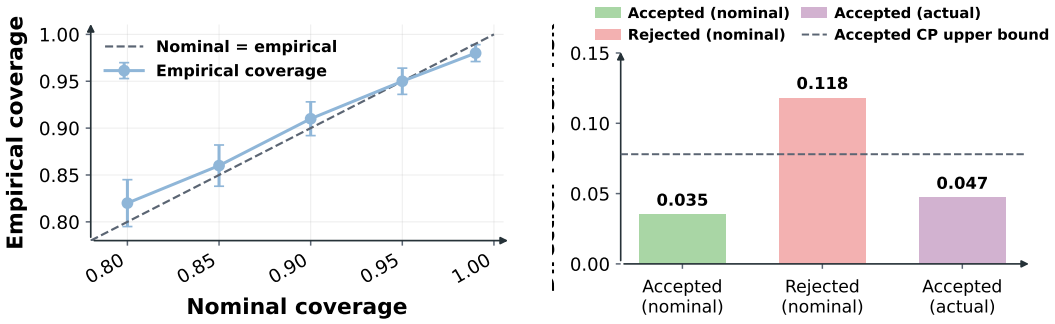
residuals associated with different responses vary jointly. Because T_{train} uses the shared center and scale of the entire reward group, a score change for one response may also affect the training coefficients of other responses in the same group. The training coefficient residuals may therefore exhibit within-group dependence, and score uncertainty should be represented jointly across the entire group.

Figure 11(a) shows the correlation matrix of the $K = 8$ training coefficient residual components. Most off-diagonal correlations are positive, ranging from 0.05 to 0.66, with stronger correlations between residuals associated with adjacent responses. This indicates that score discrepancies within a response group can affect multiple training coefficient components simultaneously, so the residuals cannot be treated as independent across responses.

Figure 11(b) shows the eigenvalue spectrum of the residual correlation matrix. The three largest principal directions jointly explain approximately 76.0% of the residual variation, whereas the remaining directions explain substantially less. Together with the nonzero correlations in Figure 11(a), this result indicates that the residual vectors exhibit within-group association and cannot be fully described component by component. Treating coefficient changes for each response separately would therefore fail to capture how score discrepancies propagate through the training coefficients of the entire group. A joint coefficient set is consequently needed in the training-coefficient space to preserve this joint variation.

F.3 Empirical Coverage and Risk Selection

Figure 12 reports the coverage performance of the joint coefficient set and the post-screening direction-reversal risk. For each predefined nominal coverage level, we compute Q_α on the cov-



(a) Coverage of the joint coefficient set.

(b) Risk separation at the selected threshold.

Figure 12: Empirical coverage of the joint coefficient set and direction-reversal risk separation at the selected threshold.

erage calibration groups and compare nominal and empirical coverage on independent risk calibration groups. Empirical coverage varies with the target level and remains close to it overall. Under the final setting, both nominal and empirical coverage are 0.95, indicating that the joint coefficient set constructed from coefficient residuals standardized by $\mathbf{S}_x^{\mathrm{coef}}$ covers the reference training coefficients. This result is consistent with Eq. 5: when the calibration groups and the response groups to be covered satisfy exchangeability, Q_α provides the corresponding finite-sample marginal coverage guarantee.

Figure 12(b) compares the directional relationship between local updates induced by the training coefficients from the two scores for response groups entering and not entering training under τ_{sel}. According to Eq. 9, the direction-reversal event $Z_{x,\tau}$ and its conditional risk R_τ are defined only for response groups with nonzero recovery updates. For groups not entering training, we report only the comparison indicator between the local updates induced by the training coefficients from the two scores:

$$Z_x^{\mathrm{nom}} = \mathbb{I}!\left[\left\langle \mathbf{u}_x^{\mathrm{full}}(\hat{\mathbf{a}}_x), \mathbf{u}_x^{\mathrm{full}}(\mathbf{a}_x^\star)\right\rangle < 0\right]. \tag{44}$$

In the figure, response groups entering training satisfy $\Pr(Z_x^{\mathrm{nom}} = 1 \mid A_{x,\tau_{\mathrm{sel}}} = 1) = 0.035$, whereas the corresponding proportion is $\Pr(Z_x^{\mathrm{nom}} = 1 \mid A_{x,\tau_{\mathrm{sel}}} = 0) = 0.118$ for groups not entering training. For groups that actually enter training, the empirical direction-reversal risk is 0.047, with a one-sided Clopper–Pearson upper confidence bound of $U_{\tau_{\mathrm{sel}}} = 0.078$, satisfying $U_{\tau_{\mathrm{sel}}} \le \beta$. Thus, the screened training groups exhibit a lower direction-reversal rate. This is because Eq. 8 retains only groups with a nonzero feasible recovery ratio that satisfy $\Psi_x^{\mathrm{full}} \ge \tau_{\mathrm{sel}}$ and $\Psi_x^{\mathrm{reward}} > 0$. The first condition requires the candidate update to maintain sufficient policy-update projection under the joint coefficient set, while the second requires a positive projection in reward space. Therefore, risk-calibrated screening changes the composition of groups entering training and keeps the acceptance-conditioned direction-reversal risk within the target range.

F.4 Checkpoint-wise Validation under Policy Drift

Figure 13 presents checkpoint validation of the fixed calibration results along the training trajectory. During calibration, we determine Q_α and τ_{sel}, and then rerun SRGO on the same response groups at global steps $t \in \{0, 100, 250, 500\}$. At each checkpoint, the projected local updates, Ψ_x^{full}, Ψ_x^{reward}, and recovery ratios are computed solely from the response-level gradients of the current policy. The values of Q_α and τ_{sel} remain unchanged across checkpoints.

As training proceeds, the local update geometry of the current policy changes, but the screening behavior does not exhibit abrupt changes. In Figure 13(a), the accepted-group proportion decreases slightly from the initial to the intermediate checkpoint and then stabilizes. In Figure 13(b), the mean recovery ratio of accepted groups remains positive, indicating that policy changes do not eliminate feasible recovery paths. Meanwhile, the empirical conditional direction-reversal rate of accepted groups remains low, and its one-sided upper confidence bound shows no substantial increase at any checkpoint. Rejected groups consistently exhibit higher direction-reversal rates. These results indi-

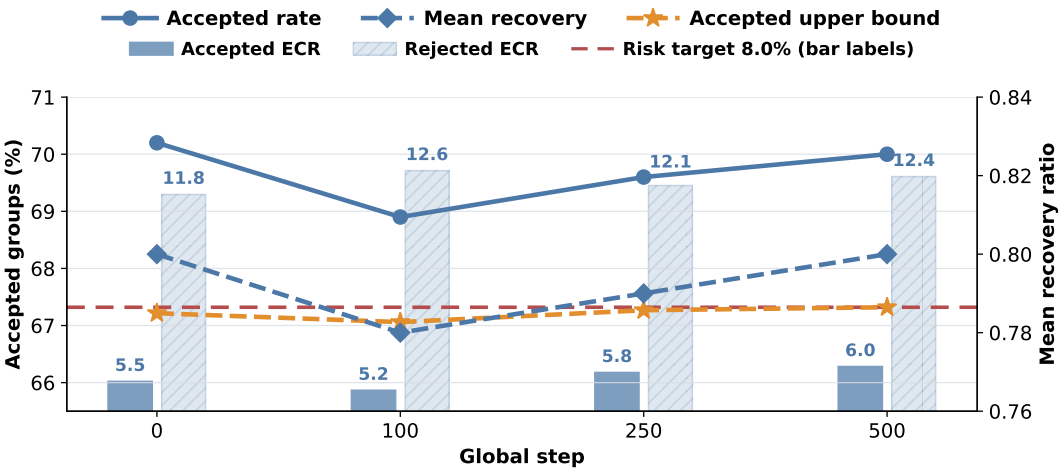


Figure 13: Checkpoint-wise validation of the fixed SRGO rule under policy drift on Qwen3.5-4B. Curves show the accepted-group rate, mean recovery ratio, and accepted-group upper confidence bound; bars show empirical direction-reversal rates for accepted and rejected groups. The red dashed line denotes $\beta = 0.08$.

cate that policy drift mainly changes the positions of response groups relative to the joint coefficient set, without disrupting the screening relation defined by $\Psi_x^{\text{full}} \geq \tau_{\text{sel}}$ and $\Psi_x^{\text{reward}} > 0$.

This result is consistent with the joint coefficient set and risk-selection threshold fixed during calibration. During formal training, Q_α and τ_{sel} remain fixed, while each checkpoint recomputes Ψ_x^{full}, Ψ_x^{reward}, and the recovery ratios using only the local gradients at that checkpoint. Thus, differences across checkpoints reflect the effect of changing local update geometry on recovery ratios rather than re-estimation of the calibration quantities. Response groups satisfying $\Psi_x^{\text{full}} \geq \tau_{\text{sel}}$ and $\Psi_x^{\text{reward}} > 0$ retain nonzero recovered coefficients, whereas the remaining groups stop recovery and write back zero coefficients. The results show that policy drift changes projection values and recovery ratios, while acceptance and rejection continue to be determined by the same projection conditions. Accepted groups retain lower direction-reversal risk and positive recovery ratios, indicating that the screening rule continues to distinguish recoverable updates from updates that fail the conditions throughout the training trajectory.

F.5 Threshold–Recovery Trade-off

In this section, we analyzes how the candidate threshold τ affects recoverable training progress and direction-reversal risk, as well as threshold selection under the risk constraint. Figure 14 shows the direction-reversal risk and cumulative recovery under different candidate thresholds. Each bubble corresponds to a candidate threshold τ. The horizontal axis shows the empirical direction-reversal risk, the vertical axis shows cumulative recovery Y_τ, bubble size indicates the number of accepted groups, and color indicates the nonzero update rate. Smaller thresholds appear in the upper-right region, with higher cumulative recovery, more accepted groups, and a higher nonzero update rate, but also higher empirical direction-reversal risk. As τ increases, the bubbles generally move toward the lower-left, indicating lower risk together with fewer accepted groups, a lower nonzero update rate, and reduced cumulative recovery. The orange ring marks the selected threshold $\tau_{\text{sel}} = 0.25$, whose cumulative recovery is approximately 400.

This behavior follows from the requirement in Eq. 8 that the full-space minimum projection along the recovery path be at least τ. Increasing the threshold tightens this condition, causing some response groups to lose feasible nonzero recovery ratios. For groups that remain feasible, the maximum recovery ratio $\eta_x^\star(\tau)$ may also decrease. Consequently, the number of accepted groups and cumulative recovery decrease as the threshold increases, while the empirical direction-reversal risk of accepted updates after screening generally declines. This trend is consistent with the selection rule in Eq. 9: among candidate thresholds with nonempty accepted groups and a one-sided upper confidence bound on direction-reversal risk satisfying the target constraint, the threshold with the

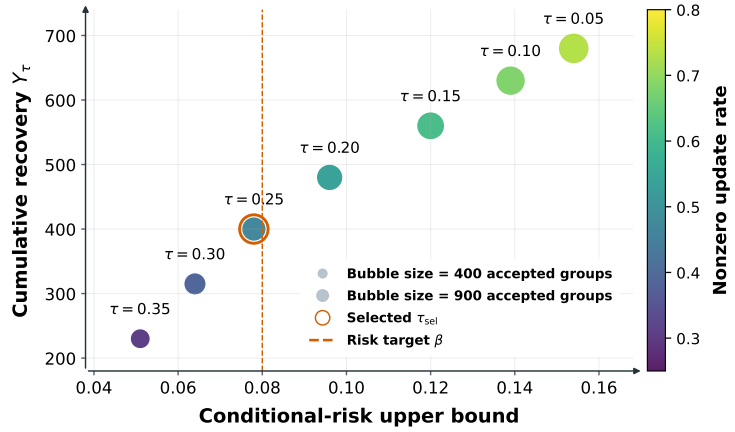
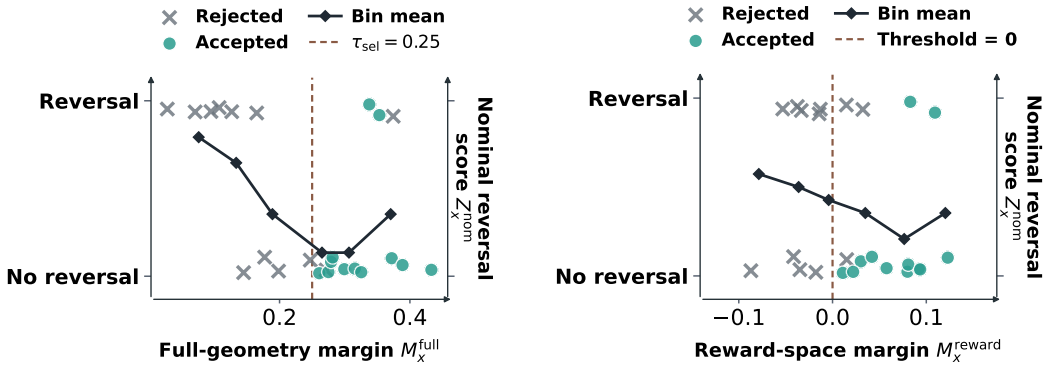


Figure 14: Risk–recovery trade-off across candidate thresholds. Bubble size indicates the number of accepted groups, color indicates the nonzero-update rate, and the highlighted point is the selected threshold τ_{sel}.



(a) Full-geometry margin

(b) Reward-space margin

Figure 15: Nominal reversal versus full-geometry and reward-space margins.

largest cumulative recovery is selected. Among the candidate thresholds satisfying the risk upper confidence bound constraint, $\tau_{\text{sel}} = 0.25$ retains the largest cumulative recovery. It therefore satisfies the risk requirement while preserving as much executable training progress as possible.

F.6 Margin and Rejection Diagnostics

Following Eqs. 6 and 7, for $q \in \text{full}, \text{reward}$ we define $M_x^q = \Psi_x^q(\mathbf{u}_x^q(\hat{\mathbf{a}}_x))$. To compare response groups that enter training with those that do not, we define the directional comparison indicator between the local updates induced by the training coefficients from the two scores,

$$Z_x^{\text{nom}} = \mathbb{I}!\left[\left\langle \mathbf{u}_x^{\text{full}}(\hat{\mathbf{a}}_x), \mathbf{u}_x^{\text{full}}(\mathbf{a}_x^\star)\right\rangle < 0\right]. \tag{45}$$

Figure 15 shows the relationship between M_x^{full}, M_x^{reward}, and the directional relationship between the updates induced by the training coefficients from the two scores. In the left panel, smaller M_x^{full} values correspond to higher direction-reversal rates, which generally decrease as M_x^{full} increases. The direction-reversal rate is 0.20 for groups entering training, compared with 0.64 for groups not entering training. The right panel shows no comparably clear trend between M_x^{reward} and the direction-reversal rate. This indicates that the full-space minimum projection in the policy-update space is more closely related to directional consistency between the two induced updates, whereas the reward-space minimum projection only reflects the projection in reward space and cannot alone determine the policy-update direction. M_x^q is the value of Ψ_x^q along the direction $\mathbf{u}_x^q(\hat{\mathbf{a}}_x)$. A smaller M_x^{full} indicates that the joint coefficient set $\mathcal{C}_x$ contains coefficients that produce a smaller

projection along this direction, making it more sensitive to coefficient variation within the set. By contrast, M_x^{reward} reflects the minimum projection in reward space and cannot replace the full-space minimum projection in the policy-update space.

Figure 16 summarizes the first condition that fails during recovery. Zero local update, insufficient full-space minimum projection, insufficient reward-space minimum projection, and non-finite values account for 25%, 44%, 24%, and 7%, respectively. Insufficient full-space minimum projection is the most common rejection reason, indicating that the projection condition in the policy-update space is the primary unmet condition for these response groups. This distribution is consistent with the screening conditions in Eq. 8: a response group has a feasible recovery ratio and writes back recovered coefficients only when its local update is nonzero, its full-space minimum projection reaches the threshold, and its reward-space minimum projection is positive.

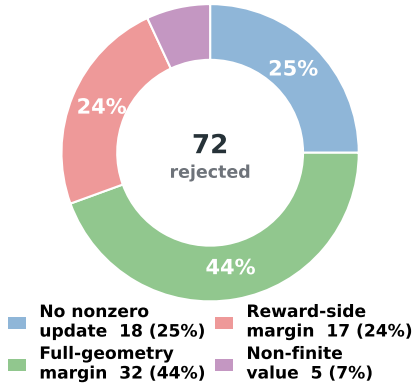


Figure 16: First rejection reason among rejected groups, categorized by the primary constraint preventing an executable update.

F.7 Projection Fidelity

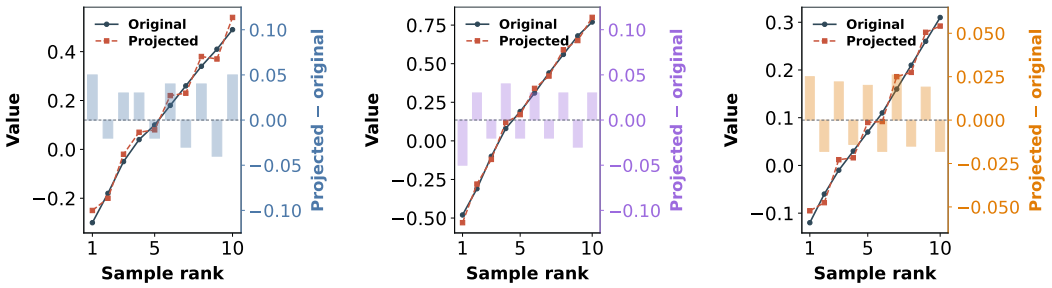
Figure 17 examines whether the fixed low-dimensional linear map $\mathcal{S}^q$ in Eq. 6 preserves the update relationships used in Eq. 7. For the same response groups, we compute the inner product, cosine similarity, and minimum projection between the updates induced by the training coefficients from the two scores, both before and after applying the map, and plot the corresponding identity scatter plots. The three metrics are concentrated near $y = x$, with correlation coefficients of 0.998, 0.999, and 0.997, respectively. The errors are centered at zero, with a sign agreement rate of 100%. These results show that, for the response groups evaluated in this experiment, the local update metrics after fixed low-dimensional projection are highly consistent with their counterparts in the original local update space.

This consistency results from applying the same fixed linear map $\mathcal{S}^q$ to all response-level gradients within each group. The updates induced by the training coefficients from the two scores are therefore computed under the same map, so the resulting differences mainly appear as numerical deviations. Eq. 6 constructs the local update from the projected $\mathbf{G}_x^q$, and Eq. 7 then computes the minimum projection over the joint coefficient set in that space. Thus, Figure 17 shows that the fixed low-dimensional representation preserves the local update relationships and minimum projection structure required by SetGRPO, providing a stable representation for subsequent geometric screening and recovery computation.

F.8 Recovery Along the Coefficient Path

Figure 18 samples response groups by quantiles of their final $\eta_x^\star(\tau_{\text{sel}})$ and plots Ψ_x^{full}, Ψ_x^{reward}, and $\|\mathbf{u}_x^{\text{full}}\|_2^2$ along the path defined by Eq. 8. Each trajectory starts from the minimum-update-norm anchor $\mathbf{a}_x^{\text{anc}}$ and interpolates toward the training coefficients $\hat{\mathbf{a}}x$ obtained from the first scoring, with the final recovery ratio marked. As η increases, the local update and its minimum projections vary along the path. Once a candidate ratio no longer satisfies $\Psi_x^{\text{full}} \geq \tau\text{sel}$ or $\Psi_x^{\text{reward}} > 0$, it is no longer feasible. The final recovery ratios of the illustrated response groups are $\eta_x^\star = 0.82$, 0.61, and 0.43, respectively. The remaining groups reach their maximum feasible values at smaller candidate ratios. The different locations of $\eta_x^\star$ show that recoverable progress depends on the current response-level gradients and the relation between the joint coefficient set and the corresponding update space.

This behavior is consistent with Eqs. 6–7. As the coefficients vary along the interpolation path, the local update $\mathbf{u}_x^q(\mathbf{a})$ and its minimum projection over $\mathcal{C}_x$ also change. Because $\mathcal{C}_x$ is convex, all coefficients on the path remain within the set. Eq. 8 therefore selects, from the fixed candidate ratio set $\mathcal{E}$, the largest η that simultaneously yields a nonzero update, satisfies $\Psi_x^{\text{full}} \geq \tau_{\text{sel}}$, and maintains $\Psi_x^{\text{reward}} > 0$. Thus, $\eta_x^\star$ represents the first-scoring training progress retained in the current policy-update space.



(a) Inner product. (b) Cosine. (c) Minimum projection.

Figure 17: Projection fidelity across three local-geometry quantities. Each panel compares the original and projected values along the same sample ranking. The bars show the signed projection error, defined as projected minus original, on the secondary vertical axis.

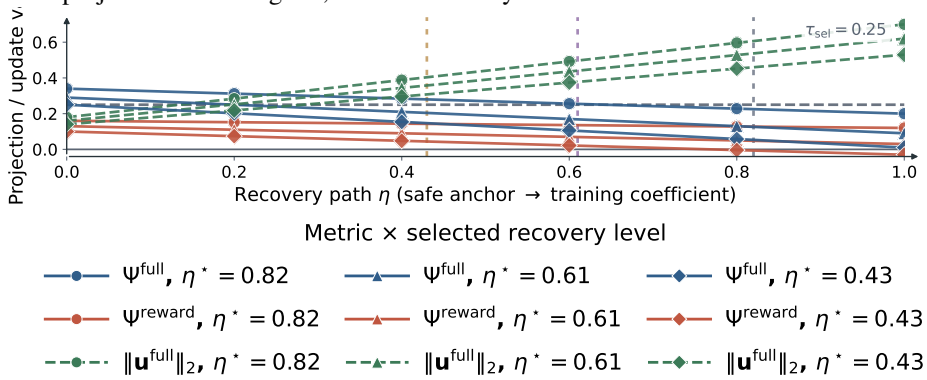


Figure 18: Recovery trajectories along the path from the safe anchor to the training coefficient. Line colors identify the full-geometry margin, reward-space margin, and update norm, while marker shapes identify the selected recovery levels. The dashed horizontal line marks the selected full-geometry projection threshold $\tau_{\text{sel}} = 0.25$, the dashed vertical lines mark the selected recovery levels $\eta_x^\star \in \{0.43, 0.61, 0.82\}$.

F.9 Representative Cases

Figures 19–21 present three types of response groups: accepted, rejected, and recovery-infeasible, with three representative cases for each type. Each case reports the two scores, the corresponding training coefficients, the minimum projections in the policy-update space and reward space, the recovery ratio $\eta_x^\star$, and the final decision rationale. For the accepted cases, consider WritingBench #4 as an example. Its two minimum projections are 0.84 and 0.81, satisfying $\Psi_x^{\text{full}} \geq \tau_{\text{sel}}$ and $\Psi_x^{\text{reward}} > 0$, with a maximum feasible recovery ratio of $\eta_x^\star = 0.82$. The rejected cases all have nonzero candidate updates but fail at least one projection condition. For example, HealthBench 920ca4e9 has $M^{\text{full}} = 0.29$, satisfying the full-space condition, but $M^{\text{reward}} = -0.06$, which fails the positive reward-space projection condition. Its recovery ratio is therefore $\eta^\star = 0.00$. The recovery-infeasible cases have no nonzero feasible recovery ratio satisfying Eq. 8. For example, IFEval #2035 has $M^{\text{full}} = 0.11$ and $M^{\text{reward}} = -0.03$, neither of which satisfies the corresponding condition, and is therefore marked as infeasible. The two scores do not vary monotonically with the decision outcome across the three types of cases. In contrast, the minimum projections and maximum feasible recovery ratio directly correspond to the screening conditions in Eq. 8.

The decision differences across cases arise because the same score discrepancy induces different local updates under different response-level gradients. SetGRPO represents the joint uncertainty of the resulting training coefficients by $\mathcal{C}_x$, then combines it with the response-level gradients of the current policy to evaluate the minimum projections of candidate updates in the policy-update space and reward space and determine the maximum recovery ratio that can be written back for each response group. Thus, response groups satisfying the projection conditions retain nonzero training updates, whereas groups that fail these conditions do not write back an update.

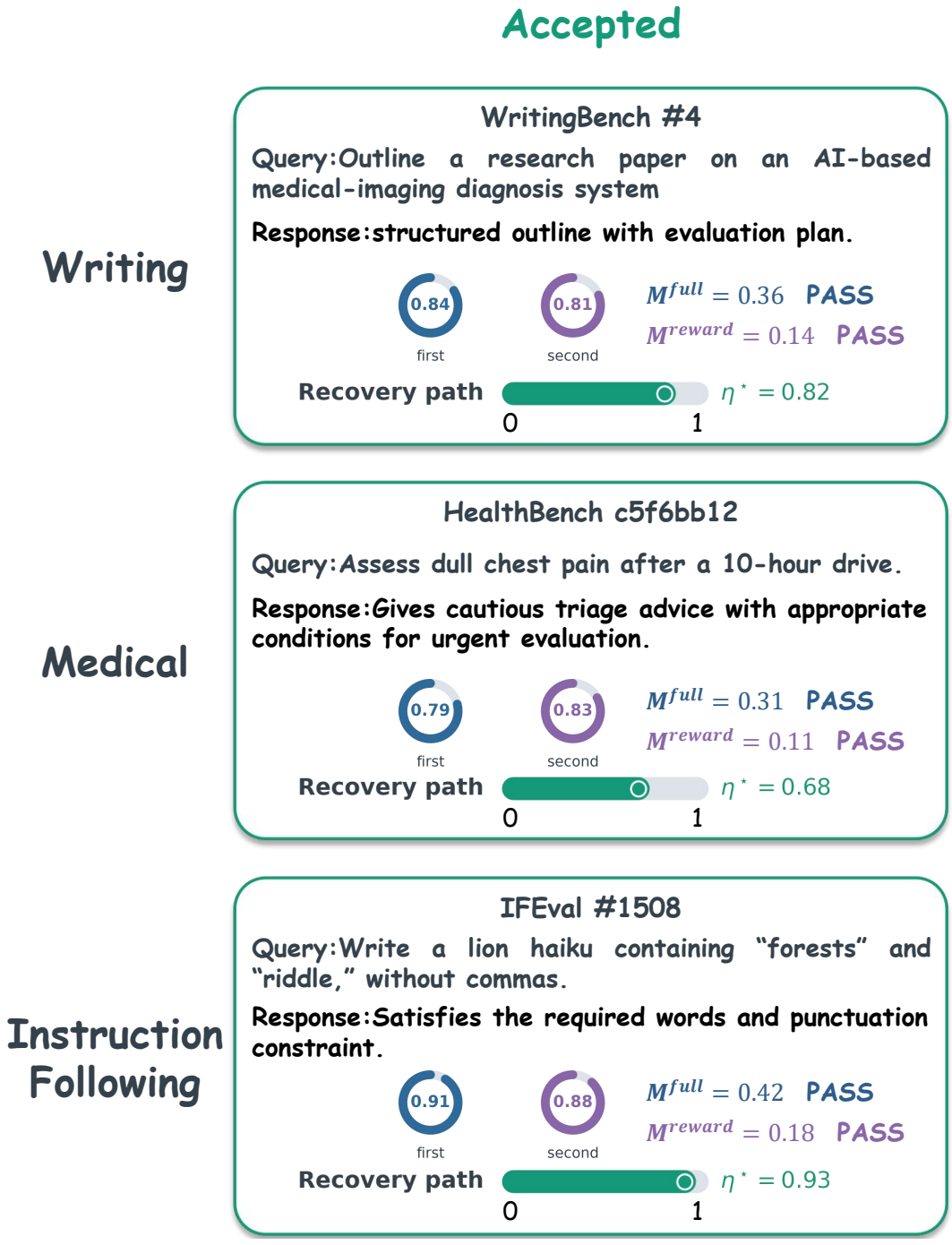


Figure 19: Representative accepted cases. Each case shows the anonymized query–response record, point and reference scores, full-geometry and reward-space projection checks, the recovery path from $\eta = 0$ to $\eta = 1$, the recovered training coefficient $\hat{\mathbf{a}}_x$, and the final decision.

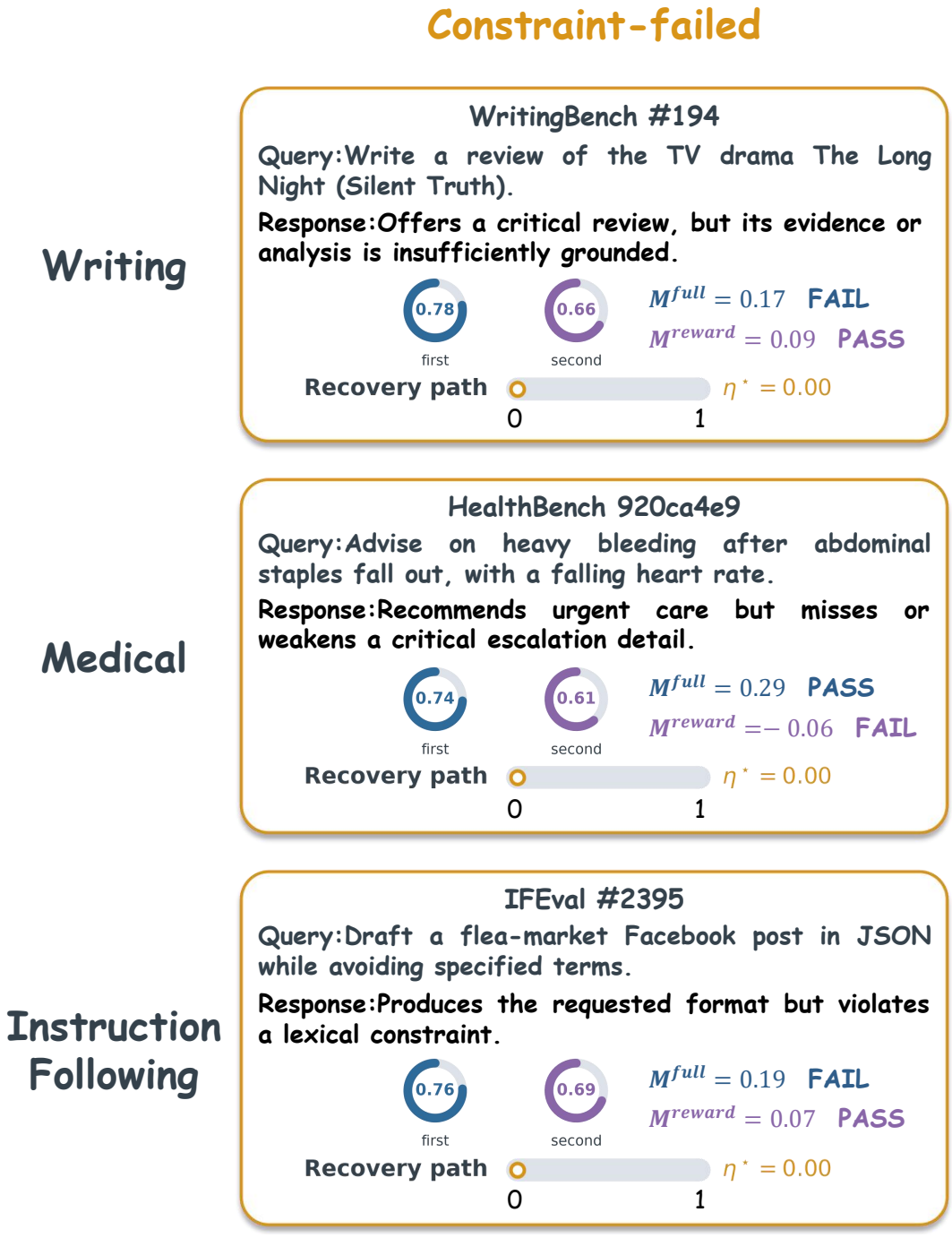


Figure 20: Representative rejected cases. Each case shows the anonymized query–response record, point and reference scores, full-geometry and reward-space projection checks, the recovery path from $\eta = 0$ to $\eta = 1$, the recovered training coefficient $\hat{\mathbf{a}}_x$, and the final decision.

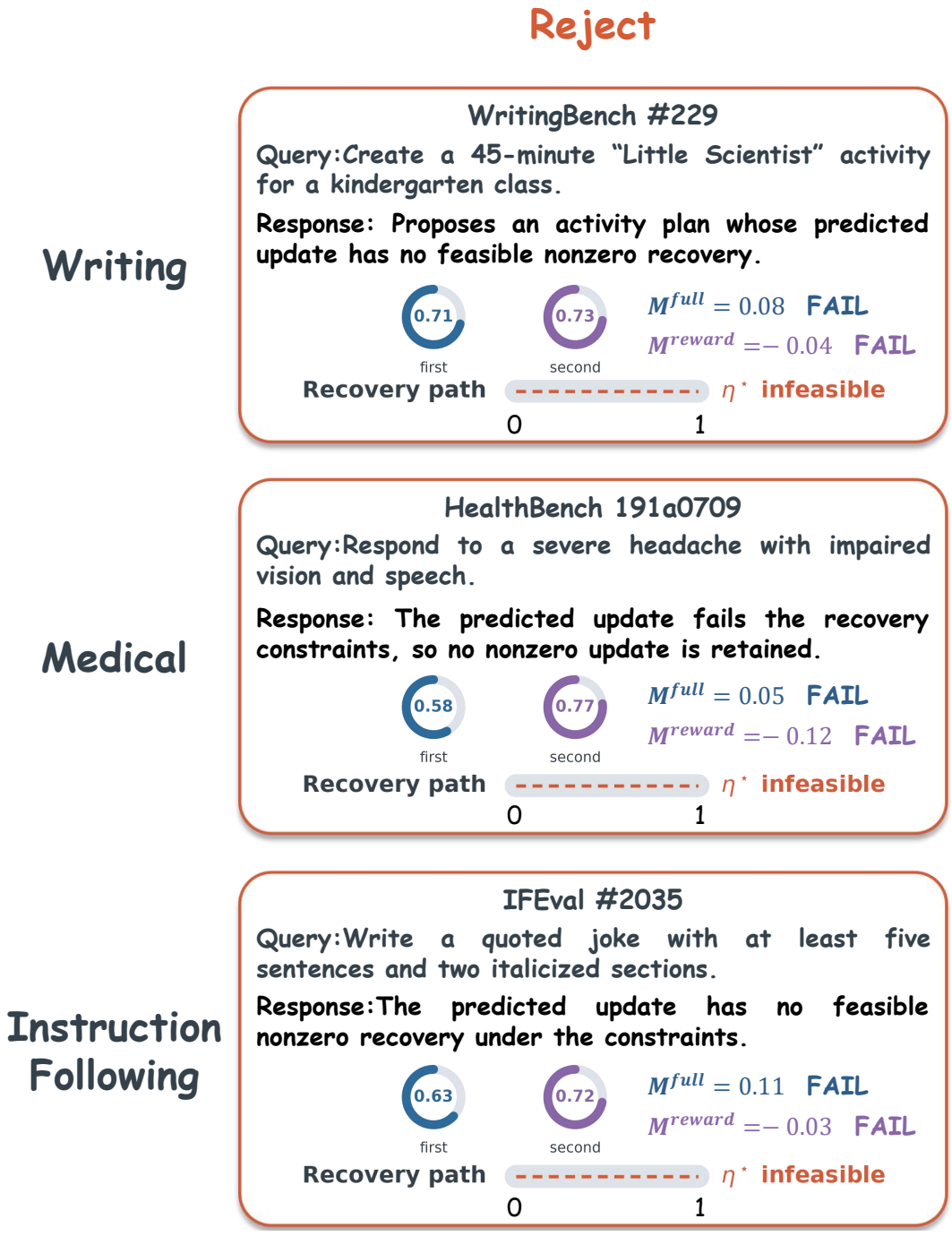


Figure 21: Representative recovery-infeasible cases. Each case shows the anonymized query–response record, point and reference scores, full-geometry and reward-space projection checks, the recovery path from $\eta = 0$ to $\eta = 1$, the recovered training coefficient $\hat{\mathbf{a}}_x$, and the final decision.