INCORRECT ANSWERS CAN BE REINFORCED MORE: RELATIVE-REINFORCEMENT POLICY OPTIMIZATION FOR LEARNING FROM VERIFIABLE REWARDS

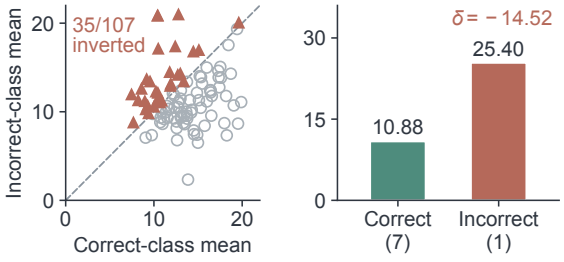
Anonymous authors
Paper under double-blind review

ABSTRACT

Group relative policy optimization (GRPO) trains reasoning models from verifiable rewards by favoring correct responses within each sampled group. However, we observe that this local preference does not always carry over to what training accumulates: measured against the initial policy, the incorrect responses to a question can be reinforced more than its correct ones. We call this phenomenon *reinforcement inversion* and attribute it to the local nature of GRPO's signal, which ranks responses within the current group but does not constrain how accumulated change is divided between correct and incorrect responses. To address this, we propose **R**elative-**R**einforcement **P**olicy **O**ptimization (**R^{2}PO**), which adds a group-level loss on the reinforcement gap between the two classes, weighted by class balance and shaped by a logistic function. R^{2}PO reuses GRPO's rollouts, verifier labels and reference log-probabilities, requiring no additional sampling. Experiments on mathematical and multimodal reasoning tasks across two backbones show that R^{2}PO consistently improves over GRPO and outperforms alternative policy optimization methods across models and datasets.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) improves the reasoning of language models using feedback on their final answers (Shao et al., 2024; DeepSeek-AI, 2025). Group relative policy optimization (GRPO) samples several responses to each question and assigns advantages by normalizing their rewards within the group (Shao et al., 2024). These advantages favor correct responses over incorrect ones in the current group. A natural question is whether this local preference is also reflected in what training accumulates: *are correct responses reinforced more than incorrect responses to the same question?*



(a) Class means. (b) Inverted group.

Figure 1: **Reinforcement inversion in GRPO.** Qwen3.5-4B on Geometry3K, with relative reinforcement in 10^{-3} nats per token. (a) Class means in the 107 mixed groups, central range shown; triangles mark inverted groups. (b) Class means in one inverted group.

To answer this question, we measure the *relative reinforcement* of each response, the change in its mean token log-probability from the initial policy to the trained policy. Applying this measure to a GRPO run of Qwen3.5-4B on Geometry3K (Lu et al., 2021), we find that the answer is often no: in about one third of the questions with both correct and incorrect responses, the incorrect responses are reinforced more than the correct ones (Figure 1(a)). Figure 1(b) shows an example, where a response with an arithmetic error is reinforced more than twice as much as the seven correct responses on average. We refer to this phenomenon as *reinforcement inversion*.

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.

We attribute reinforcement inversion to the local nature of GRPO's signal: its advantages rank the responses within each sampled group, but nothing in the objective constrains how the change accumulated over training is divided between correct and incorrect responses. Since the two often share most of their reasoning, updates that favor one can also raise the other, and the accumulated preference between them is never optimized directly.

A direct remedy is to optimize the *reinforcement gap*, the difference between the mean relative reinforcement of correct and incorrect responses to a question. Doing so raises two challenges. Firstly, groups differ in how reliably they measure this gap: a group with a single incorrect response offers a much noisier contrast than an evenly split one. Secondly, groups differ in how much correction they need: inverted groups should be pushed, whereas groups that already favor correct responses gain little from further pushing.

To this end, we propose *Relative-Reinforcement Policy Optimization* (R^2PO), which augments GRPO with a group-level loss on the reinforcement gap. R^2PO weights each group by its class balance, which arises naturally from the covariance between correctness and relative reinforcement, and adopts a logistic loss that pushes inverted groups harder and eases off once the correct responses lead. The resulting update assigns each response a single coefficient, whose direction, allocation and strength are set by its correctness, its group's composition and the gap. R^2PO reuses the rollouts, verifier labels and reference log-probabilities already computed by GRPO, requiring no additional sampling.

Experiments on mathematical and multimodal reasoning with two backbones show that R^2PO consistently improves on GRPO and outperforms alternative policy optimization methods across models and datasets. Further analyses show that each design choice contributes, that the logistic gap weight adapts to each group, and that R^2PO reduces reinforcement inversion.

Our contributions are:

- **(i)** We identify *reinforcement inversion* in GRPO training, where incorrect responses can be reinforced more than correct responses to the same question, and attribute it to the local nature of GRPO's advantages.
- **(ii)** We propose R^2PO, a balance-weighted logistic loss on the reinforcement gap that complements GRPO without additional sampling.
- **(iii)** We show that R^2PO consistently improves on GRPO across three reasoning tasks and two backbones, and analyze its components and its effect on reinforcement inversion.

2 Related Work

RLVR and group-relative policy optimization. Verifiers check final answers or intermediate steps (Cobbe et al., 2021; Lightman et al., 2024), and RLVR turns such checks into rewards for language (Shao et al., 2024; DeepSeek-AI, 2025) and multimodal reasoning (Zhang et al., 2025), building on policy gradients (Williams, 1992; Schulman et al., 2017; Ouyang et al., 2022) with sampled baselines (Ahmadian et al., 2024; Li et al., 2024). GRPO variants refine the per-step update through clipping and normalization (Yu et al., 2025; Liu et al., 2025b), importance ratios (Zheng et al., 2025; Gao et al., 2025; MiniMax, 2025), negative-sample updates (Deng et al., 2025) or classification losses (Zhai et al., 2026). R^2PO leaves this update unchanged and adds a loss on what it accumulates.

Learning from correct and incorrect responses. Correct responses can be reused as training data (Zelikman et al., 2022; Singh et al., 2024) or provided as privileged context (Tan & Hong, 2026), while preference objectives contrast preferred and dispreferred responses (Rafailov et al., 2023; Azar et al., 2024; Meng et al., 2024; Ethayarajh et al., 2024). R^2PO instead applies a single logistic loss to the gap between class means within each verifier-labeled group.

What RLVR changes. Prior analyses ask whether RLVR sharpens sampling or expands coverage (Yue et al., 2025; Liu et al., 2025a), and show that preference optimization can lower the probability of preferred responses (Razin et al., 2025). We instead examine how accumulated change is divided between correct and incorrect responses to the same question.

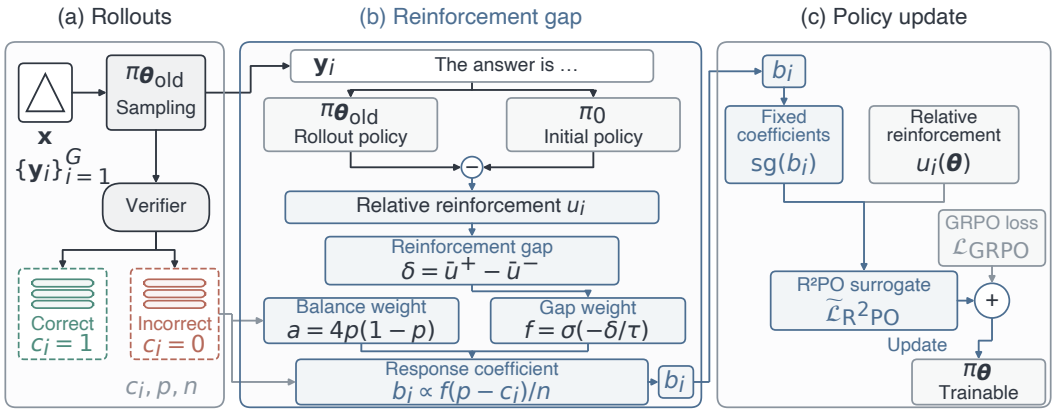


Figure 2: **Overview of R²PO.** (a) GRPO samples a group of responses, and a verifier labels each one. (b) Relative reinforcement against the initial policy yields the reinforcement gap, which, with the balance and gap weights, sets each response coefficient. (c) The fixed coefficients define a surrogate added to the GRPO loss.

3 Preliminaries

A reasoning policy $\pi_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ generates a response $\mathbf{y}$ to a question $\mathbf{x}$, which may include an image. Each generation batch contains B questions, indexed by g, and G responses per question, indexed by i. A verifier assigns each response a correctness label $c_{gi} \in \{0, 1\}$, and the reward r_{gi} is based on correctness and may also include a format term. GRPO samples the responses from the rollout policy $\pi_{\boldsymbol{\theta}_{\text{old}}}$ and normalizes their rewards within each group (Shao et al., 2024):

$$A_{gi} = \frac{r_{gi} - \bar{r}_g}{s_g + \varepsilon_{\text{std}}}, \qquad \bar{r}_g = \frac{1}{G}\sum_{i=1}^{G} r_{gi}, \tag{1}$$

where s_g is the standard deviation of the group's rewards and $\varepsilon_{\text{std}} > 0$ avoids division by zero. The GRPO loss $\mathcal{L}_{\text{GRPO}}$ combines these advantages with clipped policy ratios (Schulman et al., 2017) and a KL penalty toward the initial policy $\pi_0 = \pi_{\boldsymbol{\theta}_0}$, which stays fixed throughout training. Because the advantages are recomputed from fresh responses at every step, they compare the responses of the current group but carry no information about how training has changed these responses relative to π_0.

4 Method

R²PO augments GRPO with a loss on the reinforcement gap between correct and incorrect responses. As shown in Figure 2, it reuses GRPO's rollouts and verifier labels (a), measures the relative reinforcement of each response against the initial policy and turns each group's gap into response coefficients (b), and adds the resulting gradient to the GRPO update (c).

4.1 Relative reinforcement and the reinforcement gap

For each group, let $\mathcal{E}_g \subseteq \{1, \ldots, G\}$ index the responses used by R²PO and $n_g = |\mathcal{E}_g|$; we exclude responses cut off before their final answer and responses with no scored tokens (Appendix B.1). For $i \in \mathcal{E}_g$, let $m_{gi,t} \in \{0, 1\}$ mark the scored tokens of $\mathbf{y}_{gi}$ and $T_{gi} = \sum_t m_{gi,t}$ be their number. The relative reinforcement of a response is its mean token log-ratio between the current and initial policies:

$$u_{gi}(\boldsymbol{\theta}) = \frac{1}{T_{gi}} \sum_{t=1}^{|\mathbf{y}_{gi}|} m_{gi,t} \log \frac{\pi_{\boldsymbol{\theta}}(y_{gi,t} \mid \mathbf{x}_g, \mathbf{y}_{gi,<t})}{\pi_0(y_{gi,t} \mid \mathbf{x}_g, \mathbf{y}_{gi,<t})}. \tag{2}$$

It is positive when training has made the response more likely than under the initial policy, and averaging over tokens makes responses of different lengths comparable.

Algorithm 1 Relative-Reinforcement Policy Optimization

Require: Initial parameters $\boldsymbol{\theta}_0$, question batches of size B, group size G, $\lambda \geq 0$, $\tau > 0$
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$; freeze $\pi_0 \leftarrow \pi_{\boldsymbol{\theta}_0}$
for each question batch $\{\mathbf{x}_g\}_{g=1}^{B}$ **do**
  $\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}$; sample G responses per question from $\pi_{\boldsymbol{\theta}_{\text{old}}}$
  Compute correctness c_{gi}, rewards r_{gi}, masks $m_{gi,t}$ and eligibility sets $\mathcal{E}_g$
  Initialize $b_{gi} \leftarrow 0$ for all g, i; form the set of mixed groups $\mathcal{M}$
  for each $g \in \mathcal{M}$ **do**
    Compute p_g, $u_{gi}(\boldsymbol{\theta}_{\text{old}})$ and δ_g^{old} using Eqs. (2)–(3)
    $b_{gi} \leftarrow \dfrac{4\lambda}{Bn_g}\sigma(-\delta_g^{\text{old}}/\tau)(p_g - c_{gi})$ for $i \in \mathcal{E}_g$; detach b_{gi}
  end for
  Form $\mathcal{L}_{\text{train}}$ from $\mathcal{L}_{\text{GRPO}}$ and the surrogate in Eq. (10)
  Accumulate gradients across the batch; take one optimizer step on $\boldsymbol{\theta}$
end for
return $\pi_{\boldsymbol{\theta}}$

Let $p_g = n_g^{-1} \sum_{i \in \mathcal{E}_g} c_{gi}$ be the fraction of correct responses. We call a group *mixed* if it contains both correct and incorrect responses, $0 < p_g < 1$, and denote the set of mixed groups by $\mathcal{M}$. For $g \in \mathcal{M}$, the class means and the reinforcement gap are

$$\bar{u}_g^+ = \frac{\sum_{i \in \mathcal{E}_g} c_{gi} u_{gi}}{n_g p_g}, \qquad \bar{u}_g^- = \frac{\sum_{i \in \mathcal{E}_g} (1 - c_{gi}) u_{gi}}{n_g (1 - p_g)}, \qquad \delta_g = \bar{u}_g^+ - \bar{u}_g^-. \tag{3}$$

A group is inverted when $\delta_g < 0$, that is, when its incorrect responses have been reinforced more on average (Figure 1). As a difference of class means, δ_g is unaffected by changes shared by all responses in the group, such as a common drift in style or format, and isolates the contrast between correctness classes. Averaging within each class further smooths out variation specific to individual responses, such as their length or phrasing.

4.2 Weighting groups by class balance

The first challenge is that groups measure the gap with different reliability. The empirical covariance between correctness c and relative reinforcement u over the responses in $\mathcal{E}_g$ makes this dependence explicit. With $\bar{u}_g = n_g^{-1} \sum_{i \in \mathcal{E}_g} u_{gi}$,

$$\text{Cov}_g(c, u) = \frac{1}{n_g} \sum_{i \in \mathcal{E}_g} (c_{gi} - p_g)(u_{gi} - \bar{u}_g) = p_g(1 - p_g)\, \delta_g. \tag{4}$$

The covariance factors into the reinforcement gap and the class balance $p_g(1-p_g)$, which is largest for an even split and vanishes as either class disappears. Following this factorization, we weight each mixed group by the normalized balance $a_g = 4p_g(1-p_g)$, a form also used by SSOPD (Tan & Hong, 2026). It equals one for an even split and approaches zero as a group loses one of its classes, so groups with a well-supported contrast contribute more; groups outside $\mathcal{M}$ receive zero weight.

4.3 A logistic loss on the reinforcement gap

The second challenge is that groups need different amounts of correction. A loss linear in the gap, such as the negative covariance, pushes each group at a rate that does not depend on its current gap, even when the group is already correctly ordered. We instead apply a logistic loss to the gap:

$$\mathcal{L}_{\text{R}^2\text{PO}}(\boldsymbol{\theta}) = \frac{\lambda}{B} \sum_{g \in \mathcal{M}} a_g \tau \left[\text{softplus}\left(-\frac{\delta_g(\boldsymbol{\theta})}{\tau} \right) - \log 2 \right], \quad \lambda \geq 0, \quad \tau > 0, \tag{5}$$

where $\text{softplus}(z) = \log(1 + e^z)$, λ sets the weight of the loss and the temperature τ sets its scale. Subtracting $\log 2$ sets the loss to zero at $\delta_g = 0$ without changing its gradient, and normalizing by all B groups lets a batch with fewer mixed groups contribute less. The derivative with respect to the gap is

$$-\frac{\partial \mathcal{L}_{\text{R}^2\text{PO}}}{\partial \delta_g} = \frac{\lambda a_g}{B} f_g, \qquad f_g = \sigma\left(-\frac{\delta_g}{\tau} \right), \tag{6}$$

where $\sigma(z) = (1 + e^{-z})^{-1}$. We call f_g the *gap weight*. It exceeds $1/2$ for inverted groups, equals $1/2$ at $\delta_g = 0$ and decreases toward zero as the correct class leads. For the same balance weight, inverted groups are therefore pushed harder than correctly ordered ones, and unlike a linear loss, the logistic loss is bounded below. With one response per class, the loss takes the logistic form of DPO (Rafailov et al., 2023) applied to mean-token log-ratios; with more responses, it acts on the gap between class means rather than averaging pairwise losses. We set τ on the scale of the reinforcement gaps, around 10^{-3}, so that f_g varies across groups (Section 5.5).

Fixing $f_g = 1/2$ instead recovers a loss linear in the covariance,

$$\mathcal{L}_{\text{lin}} = -\frac{2\lambda}{B} \sum_{g \in \mathcal{M}} \text{Cov}_g(c, u), \tag{7}$$

which we use as the *linear gap loss* in Section 5.3 to isolate the effect of the logistic form.

4.4 Training with shared rollouts

By the chain rule, the loss assigns each response in a mixed group the coefficient

$$\frac{\partial \mathcal{L}_{\text{R}^2\text{PO}}}{\partial u_{gi}} = \frac{4\lambda}{B n_g} f_g (p_g - c_{gi}). \tag{8}$$

Correct responses receive negative coefficients and incorrect responses receive positive ones, so the loss favors a larger gap between the two classes. We compute $b_{gi} = \partial \mathcal{L}_{\text{R}^2\text{PO}} / \partial u_{gi}$ at $\boldsymbol{\theta}_{\text{old}}$ once per batch and hold it fixed during the update; responses outside mixed groups receive $b_{gi} = 0$. Within a group, the coefficients sum to zero and the two classes receive equal total magnitude,

$$\underbrace{-\sum_{i \in \mathcal{E}_g :\, c_{gi}=1} b_{gi}}_{\text{correct class}} = \underbrace{\sum_{i \in \mathcal{E}_g :\, c_{gi}=0} b_{gi}}_{\text{incorrect class}} = \frac{\lambda a_g}{B} f_g^{\text{old}}, \tag{9}$$

where f_g^{old} is the gap weight at $\boldsymbol{\theta}_{\text{old}}$. The balance weight thus sets each group's total contribution, while a smaller class receives a larger coefficient per response.

Proposition 1 (Properties of the R^{2}PO coefficients)**.** *For a fixed batch, labels and masks, the coefficients of every mixed group g satisfy: (i) $\sum_{i \in \mathcal{E}_g} b_{gi} = 0$, so the loss is unaffected by a change shared by all responses in the group; (ii) correct responses receive nonpositive and incorrect responses nonnegative coefficients, with equal class totals given by Eq. (9); (iii) $|b_{gi}| \le 4\lambda(n_g - 1)/(B n_g^2)$, and no smaller constant holds for all gaps.*

Appendix A gives the proofs. Property (i) means that R^{2}PO acts only on the contrast between the two classes, and property (iii) bounds the coefficient of any single response, even when a class contains only one response.

Relation to the GRPO update. When the reward reduces to correctness, the GRPO advantage in Eq. (1) is proportional to $c_{gi} - p_g$, and so is the coefficient in Eq. (8). Within a group, R^{2}PO therefore reinforces correct responses and penalizes incorrect ones in the same relative proportions as GRPO; what it changes is how much each group contributes, through a weight proportional to $\lambda f_g s_g$. This weight grows with the class balance and, through f_g, with how strongly the group is inverted relative to the initial policy, information that GRPO discards by normalizing the advantages of every mixed group to unit scale.

We implement the update with the surrogate

$$\widetilde{\mathcal{L}_{\text{R}^2\text{PO}}}(\boldsymbol{\theta}) = \sum_g \sum_{i \in \mathcal{E}_g} \text{sg}(b_{gi}) \big[u_{gi}(\boldsymbol{\theta}) - \text{sg}\big(u_{gi}(\boldsymbol{\theta})\big) \big], \qquad \mathcal{L}_{\text{train}} = \mathcal{L}_{\text{GRPO}} + \widetilde{\mathcal{L}_{\text{R}^2\text{PO}}}, \tag{10}$$

where sg denotes stop-gradient. The surrogate has zero value and gradient $\sum_{g,i} b_{gi} \nabla_{\boldsymbol{\theta}} u_{gi}$, which equals the gradient of Eq. (5) at $\boldsymbol{\theta}_{\text{old}}$. Algorithm 1 summarizes the complete procedure; Appendix B.1 describes its distributed implementation, which computes group statistics on the full batch so that the coefficients are identical across devices.

Table 1: **Main results.** pass@k (%) on three reasoning tasks with two backbones; Δ: gain over GRPO. Bold: best; underline: second best within each backbone.

	Geometry3K			MATH500			M3CoT		
Method	pass@1	pass@4	pass@8	pass@1	pass@4	pass@8	pass@1	pass@4	pass@8
Qwen3.5-4B									
GRPO	74.33	83.30	86.67	<u>89.80</u>	91.56	92.80	83.39	90.28	<u>93.05</u>
Dr. GRPO	72.00	83.01	87.33	88.60	91.23	93.00	83.66	90.64	92.69
DAPO	68.00	82.76	87.33	85.80	87.31	90.00	82.67	90.82	92.69
GSPO	74.33	82.47	86.00	86.60	90.03	91.40	84.21	90.13	92.51
SAPO	<u>76.33</u>	82.88	86.67	89.00	<u>91.72</u>	<u>93.40</u>	83.84	<u>90.84</u>	<u>93.05</u>
CISPO	<u>76.33</u>	<u>83.79</u>	<u>88.33</u>	88.60	91.53	<u>93.40</u>	84.39	90.40	92.87
NTHR	72.33	77.71	83.33	86.40	86.08	87.40	82.04	87.41	90.70
REAL	73.00	82.92	85.67	89.00	91.26	93.00	<u>84.48</u>	90.30	92.33
R^{2}PO (ours)	**77.67**	**84.15**	**89.00**	**91.00**	**92.19**	**93.80**	**85.11**	**90.96**	**94.22**
Δ vs. GRPO	**+3.34**	**+0.85**	**+2.33**	**+1.20**	**+0.63**	**+1.00**	**+1.72**	**+0.68**	**+1.17**
Qwen3-VL-4B-Thinking									
GRPO	61.00	71.18	75.67	79.20	84.76	86.20	72.83	80.41	83.30
Dr. GRPO	58.67	70.78	75.67	79.00	<u>86.33</u>	88.40	72.56	80.42	83.30
DAPO	<u>62.33</u>	<u>71.79</u>	76.33	**81.00**	<u>86.33</u>	88.20	73.38	<u>81.03</u>	<u>84.12</u>
GSPO	58.67	70.95	76.67	80.00	86.22	<u>88.60</u>	72.29	79.50	82.40
SAPO	59.33	71.66	<u>77.00</u>	80.00	86.29	88.00	72.20	80.30	83.21
CISPO	58.00	70.54	75.00	<u>80.20</u>	85.05	86.80	<u>73.47</u>	80.55	83.48
NTHR	55.00	68.50	73.67	77.60	83.75	86.00	70.85	79.79	83.21
REAL	61.00	69.89	74.67	79.00	86.07	87.80	71.84	80.38	83.57
R^{2}PO (ours)	**63.00**	**73.40**	**79.00**	**81.00**	**87.57**	**89.00**	**74.01**	**81.04**	**84.21**
Δ vs. GRPO	**+2.00**	**+2.22**	**+3.33**	**+1.80**	**+2.81**	**+2.80**	**+1.18**	**+0.63**	**+0.91**

5 Experiments

We evaluate whether R^{2}PO improves accuracy over GRPO and alternative methods (Section 5.2), which of its components contribute (Section 5.3), whether it acts through the intended mechanism (Section 5.4), and how sensitive and efficient it is (Section 5.5).

5.1 Experimental setup

Tasks and models. We train Qwen3.5-4B (Qwen Team, 2026) and Qwen3-VL-4B-Thinking (Bai et al., 2025) (Qwen3-VL-4B for short) separately on three reasoning tasks: diagram-based geometry with Geometry3K (Lu et al., 2021), textual mathematics with MATH (Hendrycks et al., 2021), and multimodal reasoning with M3CoT (Chen et al., 2024). We evaluate on the Geometry3K validation set, MATH500 (Lightman et al., 2024) and the M3CoT validation set.

Evaluation metrics. A symbolic verifier checks each final answer against the reference. We report pass@1 from one deterministic response per question and pass@k for $k \in \{4, 8\}$ with the unbiased estimator of Chen et al. (2021) from eight samples per question, following prior RLVR studies (Yue et al., 2025); pass@8 is the fraction of questions solved by at least one of the eight samples.

Baselines. We compare with GRPO variants that change its normalization and clipping, Dr. GRPO (Liu et al., 2025b) and DAPO (Yu et al., 2025); its importance ratios, GSPO (Zheng et al., 2025), SAPO (Gao et al., 2025) and CISPO (MiniMax, 2025); its treatment of negative samples, NTHR (Deng et al., 2025); and its loss, REAL (Zhai et al., 2026). All methods share the backbone, training data and evaluation protocol of each setting and are reported with their best hyperparameters; Appendix C.1 describes their implementation.

Table 2: **Ablation on design choices.** pass@1 (%) of R^{2}PO variants on Qwen3.5-4B. Bold: best.

Variant	Geometry3K	MATH500	M3CoT
GRPO	74.33	89.80	83.39
R^{2}PO (full)	**77.67**	**91.00**	**85.11**
w/o balance weight ($a_g = 1$)	73.00	88.20	84.30
w/ linear gap loss ($f_g = 1/2$)	74.00	89.00	82.58
w/o initial-policy reference	74.67	89.00	84.03

Implementation details. R^{2}PO and GRPO share the same configuration: eight responses per question, 32 responses per generation batch, one optimizer update per batch, and one epoch at a learning rate of 10^{-6} on eight NVIDIA A100 GPUs; we evaluate the final checkpoint. The reward combines correctness with a format term weighted by 0.1, while the R^{2}PO loss uses correctness alone. We select τ around 10^{-3} and use a loss weight of $\lambda = 1$ on Geometry3K, $\lambda = 1.75$ on M3CoT and $\lambda = 3$ on MATH for Qwen3.5-4B, and $\lambda = 1$, 1.5 and 3 on the same tasks for Qwen3-VL-4B. Appendix B lists the full configuration.

5.2 Main results

R^{2}PO consistently improves on GRPO. Table 1 compares R^{2}PO with GRPO and seven alternatives in six model–task settings. We highlight three observations. Firstly, R^{2}PO improves on GRPO in every setting and on every metric, raising pass@1 by 1.18–3.34 percentage points on both text-only and multimodal tasks. Secondly, the gains extend to pass@4 and pass@8, where R^{2}PO is the best of the nine methods in all six settings; by raising pass@8, R^{2}PO increases the fraction of questions solved by at least one of eight samples, rather than only changing which answer the policy selects. On Qwen3-VL-4B, the gains in pass@4 and pass@8 on Geometry3K and MATH500 (2.22–3.33 points) exceed those in pass@1. Thirdly, R^{2}PO attains the best or tied-best pass@1 in every setting, whereas the alternatives that modify the per-step update help in some settings and hurt in others; DAPO, for example, is among the strongest methods with Qwen3-VL-4B but the weakest on Geometry3K with Qwen3.5-4B. Because R^{2}PO keeps the GRPO update unchanged, these gains are attributable to the loss on the reinforcement gap.

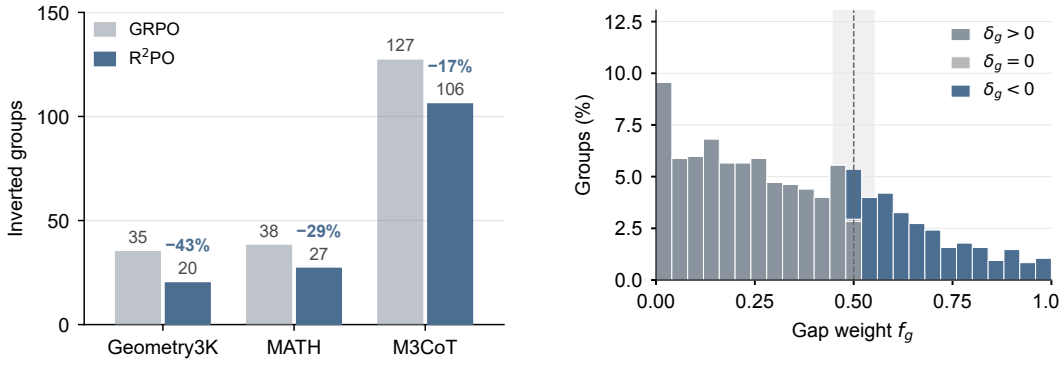
Alternatives trade off one task against another. On Qwen3.5-4B, methods that change the importance ratio (GSPO, SAPO and CISPO) raise pass@1 on M3CoT but fall below GRPO on MATH500, and those that change normalization and clipping (Dr. GRPO and DAPO) fall below GRPO on Geometry3K and MATH500. NTHR, which reduces the penalty on negative samples, is below GRPO on every task and metric. On Qwen3.5-4B, no alternative improves pass@1 on all three tasks, whereas R^{2}PO does so with both backbones.

5.3 Ablation on design choices

We verify that each component of R^{2}PO contributes by changing one at a time and keeping the configuration of Table 1: we set $a_g = 1$ for every mixed group, replace the logistic loss with the linear gap loss of Eq. (7), or replace the relative reinforcement with the mean token log-probability under the current policy while retaining GRPO's KL penalty.

Class-balance weighting is essential. Without the balance weight, pass@1 drops by 4.67 and 2.80 points on Geometry3K and MATH500, falling below GRPO, and by 0.81 points on M3CoT (Table 2). Weighting groups by how reliably they measure the gap therefore matters beyond the sign of the gap.

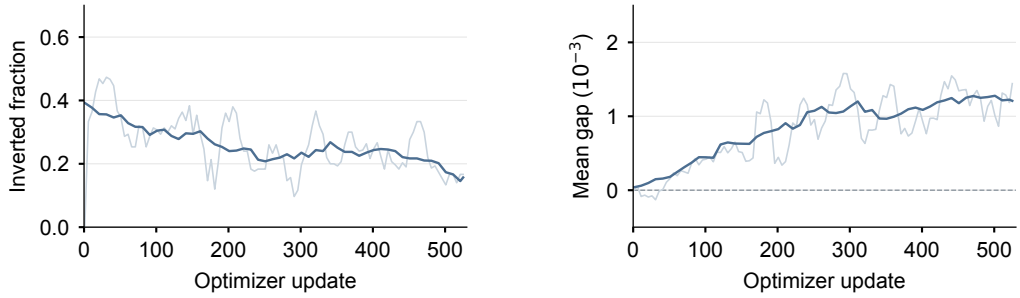
The logistic loss and the initial-policy reference both contribute. Replacing the logistic loss with the linear gap loss lowers pass@1 by 3.67, 2.00 and 2.53 points, below GRPO on all three tasks, and removing the initial-policy reference lowers it by 3.00, 2.00 and 1.08 points. Section 5.4 shows that the logistic weight indeed separates inverted from correctly ordered groups.



(a) Inverted groups after training.

(b) Gap weights during training.

Figure 3: **Mechanism analysis on Qwen3.5-4B.** (a) Inverted groups of GRPO and R^{2}PO on each task, re-scoring the response groups saved during training under each final checkpoint and the initial policy; percentages give the reduction. (b) Gap weights f_g of the 952 mixed groups in the 525 updates of R^{2}PO on Geometry3K, colored by the sign of the gap; the shaded band marks $0.45 \leq f_g \leq 0.55$.



(a) Fraction of inverted groups.

(b) Mean reinforcement gap.

Figure 4: **Training dynamics on Geometry3K.** (a) Fraction of inverted mixed groups and (b) mean reinforcement gap (10^{-3} nats per token) during R^{2}PO training; thin lines: 25-update trailing means, thick lines: 101-update centered averages.

5.4 Mechanism validation

We verify that R^{2}PO improves reasoning through the intended mechanism, i.e., it reduces reinforcement inversion, and its logistic gap weight treats inverted and correctly ordered groups differently.

R^{2}PO reduces reinforcement inversion. If R^{2}PO works as designed, fewer groups should end up with their incorrect responses reinforced more than their correct ones. We test this by applying the diagnostic of Figure 1 to the final checkpoints of GRPO and R^{2}PO with Qwen3.5-4B and re-scoring the response groups saved during training under each final checkpoint and the initial policy. As Figure 3(a) shows, R^{2}PO reduces the number of inverted groups from 35 to 20 on Geometry3K, from 38 to 27 on MATH and from 127 to 106 on M3CoT, that is, by 43%, 29% and 17%. The largest reduction, on Geometry3K, coincides with the largest pass@1 gain on Qwen3.5-4B in Table 1. The reduction also appears during training: on Geometry3K, the fraction of mixed groups that are inverted falls from about 0.45 in the first 25 updates to about 0.17 in the last 25, while the mean reinforcement gap increases and stays positive (Figure 4).

The gap weight adapts to each group. Figure 3(b) shows the gap weights that R^{2}PO assigns during training on Geometry3K. With $\tau = 10^{-3}$, f_g spreads over the whole interval $(0, 1)$ rather than staying near $1/2$: inverted groups receive $f_g > 1/2$ and are pushed harder, whereas groups whose correct class already leads receive $f_g < 1/2$, a sizeable share of them close to zero. Groups

Table 3: **Sensitivity to τ.** pass@k (%) of R^{2}PO on Geometry3K with Qwen3.5-4B. Bold: best.

τ	pass@1	pass@4	pass@8
GRPO	74.33	83.30	86.67
10^{-4}	73.33	81.07	85.67
3×10^{-4}	74.33	83.09	86.33
10^{-3}	**77.67**	83.42	**89.00**
3×10^{-3}	75.67	**84.15**	87.00
10^{-2}	76.00	83.33	86.00
3×10^{-2}	74.33	83.15	86.67

whose correct class clearly leads therefore contribute little, and the update concentrates on inverted and weakly ordered groups. Most weights fall outside the band $0.45 \leq f_g \leq 0.55$, in which the logistic loss behaves like the linear gap loss, consistent with the gap between the two losses in Table 2.

5.5 Parameter sensitivity and efficiency

R^{2}PO adds little training overhead. R^{2}PO reuses GRPO's rollouts, verifier labels and reference log-probabilities, and its group statistics add only $O(N_{\text{tok}}+BG)$ operations for N_{tok} scored tokens. As Table 4 shows, it adds 0.12%, 0.27% and 3.55% to the median time per update on Geometry3K, MATH and M3CoT, and it adds no parameters or inference-time cost.

Table 4: **Training cost** on Qwen3.5-4B: median seconds per update.

Task	GRPO	R^{2}PO	Change
Geometry3K	22.60	22.62	+0.12%
MATH	21.77	21.83	+0.27%
M3CoT	23.13	23.95	+3.55%

The temperature should match the scale of the gap. Table 3 varies τ on Geometry3K with Qwen3.5-4B. Because relative reinforcement is averaged over tokens, reinforcement gaps are of order 10^{-3} nats per token, and τ determines how strongly the gap weight f_g varies across groups. R^{2}PO improves pass@1 over GRPO for τ between 10^{-3} and 10^{-2}, with the best pass@1 and pass@8 at $\tau = 10^{-3}$. A much larger τ keeps f_g close to $1/2$, and pass@1 returns to that of GRPO, as with the linear gap loss; a much smaller τ turns f_g into a nearly binary weight on the sign of the gap, and pass@1 falls to or below that of GRPO.

6 Conclusion

We have shown that the local preference encoded by GRPO's advantages does not always carry over to what training accumulates: measured against the initial policy, the incorrect responses to a question can be reinforced more than its correct ones. R^{2}PO addresses this reinforcement inversion with a group-level logistic loss on the reinforcement gap, weighted by class balance and computed from GRPO's existing rollouts. It consistently improves on GRPO in pass@1 and pass@k across mathematical and multimodal reasoning tasks with two backbones at little training cost, and reduces the reinforcement inversion it targets. In our experiments, it reduces the number of inverted groups by 17–43% and adds at most 3.6% to the training time per update. Since R^{2}PO only adds a loss and leaves the per-step update unchanged, it can be combined with the modifications made by the alternatives in Table 1. More broadly, our results suggest that examining what training accumulates, and not only what each update rewards, is a useful lens for designing RLVR objectives.

7 Limitations and Future Work

Our experiments cover two 4B-parameter backbones from the Qwen family on three reasoning tasks. Extending the evaluation to other model families, larger models and datasets from more domains, and combining R^{2}PO with the alternatives in Table 1, are left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to help draft, edit and polish the manuscript. We also used them to assist with the preparation of figures and tables, literature searches and the verification of bibliographic information. We have not used generative AI tools for conducting experiments, generating training data or experimental results, originating the core research idea, or making final scientific conclusions. We have reviewed and verified all AI-assisted work, including manually checking the generated text, figures and cited references for correctness and consistency. We take full responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We make efforts to ensure the reproducibility of our work by providing detailed descriptions of the proposed method and experimental protocols. The main paper specifies the R^2PO loss in Equation (5), its implementation with shared rollouts in Equation (10), and the complete training procedure in Algorithm 1. Appendix B provides the training configuration, hyperparameters, data statistics, prompts, decoding settings, evaluation protocol and training cost for every task, and Appendix C.1 describes the implementation of every alternative method. The source code will be released upon acceptance to facilitate further research and reproduction of our results.

ETHICS STATEMENT

This work studies a training objective for reasoning models and does not involve human subjects or personal data. All experiments use publicly available datasets (Geometry3K, MATH and M3CoT) and open-weight models under their respective licenses. More capable reasoning models inherit the general risks of language models, but our method does not introduce risks specific to this work.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455, 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8199–8221, 2024. doi: 10.18653/v1/2024.acl-long.446. URL https://aclanthology.org/2024.acl-long.446/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948v1.

Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2475. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6adaf0cbeba11705d4ea67a62044f63d-Abstract-Conference.html.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12634–12651, 2024. URL https://proceedings.mlr.press/v235/ethayarajh24a.html.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025. URL https://arxiv.org/abs/2511.20347v2.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. *arXiv preprint arXiv:2309.06180*, 2023. URL https://arxiv.org/abs/2309.06180.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29128–29163, 2024. URL https://proceedings.mlr.press/v235/li24cd.html.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025a. URL https://arxiv.org/abs/2505.24864.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b. URL https://arxiv.org/abs/2503.20783.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786, 2021. doi: 10.18653/v1/2021.acl-long.528. URL https://aclanthology.org/2021.acl-long.528/.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124198–124235, 2024. doi: 10.52202/079017-3946. URL

https://proceedings.neurips.cc/paper_files/paper/2024/hash/e099c1c9699814af0be873a175361713-Abstract-Conference.html.

MiniMax. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025. URL https://arxiv.org/abs/2506.13585.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

Qwen Team. Qwen3.5: Towards native multimodal agents. Official model card, February 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional Unalignment: Likelihood Displacement in Direct Preference Optimization. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.08847.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=lNAyUngGFK.

Zhiquan Tan and Yinrong Hong. Self-supervised on-policy distillation for reasoning language models. *arXiv preprint arXiv:2605.17497*, 2026. URL https://arxiv.org/abs/2605.17497.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/537d5aa768c2d534016a4d06f87bc8fb-Abstract-Conference.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Zepeng Zhai, Meilin Chen, Jiaxuan Zhao, Junlang Qian, Lei Shen, and Yuan Lu. Rewards as labels: Revisiting RLVR from a classification perspective. *arXiv preprint arXiv:2602.05630*, 2026. URL https://arxiv.org/abs/2602.05630v4.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-VL: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1859–1869, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_R1-VL_Learning_to_Reason_with_Multimodal_Large_Language_Models_via_ICCV_2025_paper.html.

Yuze Zhao, Jintao Huang, Jinghan Hu, et al. SWIFT:A Scalable lightWeight Infrastructure for Fine-Tuning. *arXiv preprint arXiv:2408.05517*, 2024. URL https://arxiv.org/abs/2408.05517.

Chujie Zheng, Shixuan Liu, Mingze Li, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071v2.

A Derivations

All results below condition on a fixed generation batch, correctness labels, response sets $\mathcal{E}_g$ and masks, and hold the initial policy fixed; we do not differentiate through sampling or verification. For a mixed group, $n_g \geq 2$ and both classes are nonempty.

A.1 Covariance and response coefficients

Let $\hat{\mathbb{E}}_g$ denote the uniform empirical average over the n_g responses in $\mathcal{E}_g$. Since correctness is binary,

$$\hat{\mathbb{E}}_g[cu] = p_g \bar{u}_g^+, \qquad \hat{\mathbb{E}}_g[u] = p_g \bar{u}_g^+ + (1-p_g)\bar{u}_g^-. \tag{11}$$

Subtracting $\hat{\mathbb{E}}_g[c]\,\hat{\mathbb{E}}_g[u]$ from $\hat{\mathbb{E}}_g[cu]$ gives $\mathrm{Cov}_g(c,u) = p_g(1-p_g)(\bar{u}_g^+ - \bar{u}_g^-)$, which proves Eq. (4) for the $1/n_g$ normalization. The unbiased sample covariance, normalized by $1/(n_g-1)$, would introduce an additional factor $n_g/(n_g-1)$.

For a correct response, $\partial\delta_g/\partial u_{gi} = 1/(n_g p_g)$; for an incorrect response, it is $-1/[n_g(1-p_g)]$. Differentiating the logistic loss gives $\partial\mathcal{L}_{\mathrm{R^2PO}}/\partial\delta_g = -(\lambda a_g/B)\,\sigma(-\delta_g/\tau)$. Substituting $a_g = 4p_g(1-p_g)$ and the two class derivatives yields Eq. (8). The coefficients sum to zero because $\sum_i (p_g - c_{gi}) = n_g p_g - \sum_i c_{gi} = 0$. For the $n_g p_g$ correct responses, the coefficients sum to $-n_g p_g\,[4\lambda f_g^{\mathrm{old}}(1-p_g)/(Bn_g)] = -\lambda a_g f_g^{\mathrm{old}}/B$, and the $n_g(1-p_g)$ incorrect responses give the same magnitude with the opposite sign, which proves Eq. (9). At the token level, the chain rule multiplies each response coefficient by $m_{gi,t}/T_{gi}$, so equal coefficients within a class do not imply equal parameter gradients.

A.2 A tight bound on the response coefficients

For fixed $n_g \geq 2$ and $\lambda > 0$, every mixed group satisfies

$$\left|\frac{\partial\mathcal{L}_{\mathrm{R^2PO}}}{\partial u_{gi}}\right| \leq \frac{4\lambda(n_g-1)}{Bn_g^2}. \tag{12}$$

Let k be the number of correct responses. For a correct response, $|p_g - c_{gi}| = (n_g-k)/n_g \leq (n_g-1)/n_g$; for an incorrect response, it is $k/n_g \leq (n_g-1)/n_g$. Since $0 < f_g < 1$, the bound follows. It is tight: in a group with exactly one correct response whose relative reinforcement tends to $-\infty$ relative to the incorrect class, the coefficient of that response approaches the right-hand side, and the symmetric construction uses exactly one incorrect response. The bound concerns the response coefficients; the parameter gradient additionally depends on $\nabla_{\boldsymbol{\theta}} u_{gi}$.

A.3 Surrogate gradient and the linear gap loss

With the coefficients b_{gi} held fixed, differentiating Eq. (10) gives

$$\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}_{\mathrm{R^2PO}}} = \sum_g \sum_{i\in\mathcal{E}_g} b_{gi}\nabla_{\boldsymbol{\theta}} u_{gi}. \tag{13}$$

At $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{old}}$, the coefficients equal those obtained by applying the chain rule to Eq. (5), which proves the gradient identity in Section 4.4. Away from $\boldsymbol{\theta}_{\mathrm{old}}$, the logistic loss would use an updated gap weight, so the zero value of the surrogate is not the value of the loss.

Setting $f_g = 1/2$ in Eq. (8) gives the derivative of the linear gap loss in Eq. (7). Equivalently, the first-order expansion of the logistic loss about $\delta_g = 0$ is $-\delta_g/2$, so the weighted loss becomes $-(\lambda/2B)\sum_{g\in\mathcal{M}} a_g\delta_g = -(2\lambda/B)\sum_{g\in\mathcal{M}}\mathrm{Cov}_g(c,u)$. The initial-policy term of each u_{gi} does not depend on $\boldsymbol{\theta}$ and therefore contributes no gradient to the linear gap loss.

B Training and Evaluation Details

B.1 Update procedure

Algorithm 1 gives the complete update, which inherits the optimizer, learning-rate schedule, clipping and KL settings of GRPO. After sampling, responses cut off before their final answer and responses

with no scored tokens are excluded from $\mathcal{E}_g$. We gather group statistics across all devices before computing b_{gi} and normalize by all B groups, including those that are not mixed. Each microbatch then uses its detached coefficients, and we multiply its surrogate by the data-parallel world size and the gradient-accumulation factor so that averaging across devices and accumulation steps recovers the global sum. One optimizer update follows accumulation over the full generation batch. R^{2}PO adds no parameters or inference-time inputs; its extra training cost consists of computing relative reinforcement and the group coefficients.

B.2 Configuration and data

Table 5 lists the configuration of the main Qwen3.5-4B runs, which R^{2}PO and GRPO share except for the R^{2}PO loss. Class means are arithmetic means within each correctness class, and the initial policy stays fixed throughout training. The format term contributes to the reward but does not affect membership in $\mathcal{E}_g$, and relative reinforcement is computed over the completion tokens.

Table 5: **Training configuration.** Configuration of the main Qwen3.5-4B runs. All settings are shared by R^{2}PO and GRPO except for the R^{2}PO loss.

Setting	Value
Model and objective	
Backbone / reference	Qwen3.5-4B / fixed initial policy
Reward	Correctness $+0.1\times$ format
KL coefficient / ratio clipping	0.001 / 0.2
Loss weight λ / temperature τ	1 (Geometry3K), 1.75 (M3CoT), 3 (MATH) / around 10^{-3} (Table 3)
Optimization	
Tuning / precision / epochs	Full parameters / bfloat16 / 1
Learning rate / schedule / warmup	10^{-6} / constant / 0
GPUs	$8\times$ NVIDIA A100
Microbatch per device / accumulation	1 / 4
Seed	27182
Generation and evaluation	
Responses per group / groups / batch	8 / 4 / 32
Thinking / completion / answer reserve	1888 / 2048 / 160 tokens
Training length / generation context	4096 / 6144 tokens
Sampling temperature / top-p / top-k	1.0 / 0.95 / 20
Presence penalty	1.5
Evaluation	pass@1: temperature 0; pass@k: 8 samples per question

Dynamic sampling and sequence packing are disabled, the trainer's overlong filter is enabled, and the rollout engine runs in colocated mode. Training uses ms-swift 4.4.2 (Zhao et al., 2024) with Transformers 5.9.0 and TRL 0.29.1, and rollouts are generated with vLLM 0.18.1 (Kwon et al., 2023). Answer verification uses mathruler 0.1.0 and SymPy 1.14.0.

Training lasts 525, 1,874 and 1,965 optimizer updates on Geometry3K, MATH and M3CoT, with 2,101, 7,497 and 7,863 training questions and 300, 500 and 1,108 evaluation questions, respectively. MATH500 is a 500-question evaluation subset of MATH. All M3CoT evaluation questions include images, and exact matching on normalized text and images finds no overlap between the training and evaluation sets of MATH and M3CoT.

B.3 Prompts and decoding

Each question is followed by the instruction:

> Think briefly; you have a budget of about 1888 thinking tokens and exceeding it is penalized. Stop thinking as soon as you can answer. Then give exactly one final answer between `<answer>` and `</answer>`, and write nothing after the closing tag.

R^{2}PO and GRPO use the same instruction. Evaluation uses the same chat template, image inputs and thinking and answer budgets as training. The pass@1 metric uses one deterministic completion (temperature 0) per question. For pass@k, we draw $n = 8$ independent samples per question and use the unbiased estimator (Chen et al., 2021)

$$\text{pass@}\,k = \frac{1}{|\mathcal{Q}|} \sum_{j \in \mathcal{Q}} \left[1 - \binom{n - h_j}{k} \Big/ \binom{n}{k} \right], \tag{14}$$

where $\mathcal{Q}$ is the set of evaluation questions and h_j is the number of correct samples for question j; for $k = n$, it is the fraction of questions with at least one correct sample.

B.4 Training cost

Table 4 in Section 5.5 reports the median wall-clock time per optimizer update of R^{2}PO and GRPO over all updates of the Qwen3.5-4B runs in Table 1, measured on the same hardware. Both methods perform 525, 1,874 and 1,965 updates on Geometry3K, MATH and M3CoT, and relative changes use unrounded medians.

C Additional Results

C.1 Baseline implementations

All alternative methods share the configuration in Table 5 except where noted below. GSPO, SAPO and CISPO collect 128 responses and then perform four optimizer updates of 32 responses each, so later updates use responses sampled by an earlier policy.

GSPO uses sequence-level importance ratios with lower and upper clipping thresholds 3×10^{-4} and 4×10^{-4} (Zheng et al., 2025). SAPO uses smooth token-level gates with positive and negative temperatures 1 and 1.05 (Gao et al., 2025). CISPO clips a detached importance weight while keeping the token log-probability gradient (MiniMax, 2025); we cap the weight at 5, following the framework default since the MiniMax-M1 paper does not specify it, and keep our common KL and optimizer settings instead of the full MiniMax recipe. For CISPO and DAPO, the loss is normalized by the number of valid tokens in the current optimizer window, accounting for averaging across devices and accumulation steps. DAPO filters groups by accuracy with at most three resampling retries and keeps a homogeneous group if all retries fail. REAL follows the released objective with complete-group gathering across devices, binary accuracy labels and temperature 0.5 (Zhai et al., 2026). NTHR reduces negative-gradient penalties on tokens associated with decreases in correct-response likelihood (Deng et al., 2025); we disable dynamic sampling and use a learning rate of 5×10^{-7}.