PC-GRPO: PAIRWISE CONFIDENCE GROUP RELATIVE POLICY OPTIMIZATION FOR POST-TRAINING REASONING LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) is an effective paradigm for post-training reasoning large language models (LLMs), and Group Relative Policy Optimization (GRPO) is a widely used critic-free optimizer in this setting. However, standard GRPO relies on group-wise reward normalization for advantage estimation, which provides coarse credit assignment and is particularly limited under sparse binary rewards. As a result, trajectories with different degrees of partial correctness often receive indistinguishable supervision. We propose **Pairwise Confidence Group Relative Policy Optimization (PC-GRPO)**, which improves GRPO with two components: a pairwise confidence-based advantage estimator and a fine-grained verifiable reward. The first replaces coarse group-normalized advantages with pairwise comparisons that capture relative reward differences and policy confidence. The second provides denser supervision for incorrect trajectories using step-level overlap with reference reasoning traces, without introducing learned reward models. PC-GRPO remains critic-free and preserves the verifiable nature of RLVR. Across mathematical, scientific, knowledge-intensive, and coding benchmarks, PC-GRPO consistently outperforms standard GRPO and other strong baselines across multiple model scales.

1 INTRODUCTION

Post-training with reinforcement learning (RL) (Guo et al., 2025; Ouyang et al., 2022) has advanced the reasoning capabilities of large language models (LLMs). Chain-of-thought prompting and self-consistency (Wei et al., 2022; Wang et al., 2022) improve multi-step reasoning, while test-time scaling further enhances reasoning performance by allocating additional computation during inference (Snell et al., 2024; Muennighoff et al., 2025; OpenAI, 2024). For tasks with verifiable outcomes, reinforcement learning with verifiable rewards (RLVR) uses deterministic feedback, such as exact answer checking (Cobbe et al., 2021), reducing reliance on human preference annotations (Bai et al., 2022). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) avoids training a separate critic by normalizing rewards within groups of responses sampled for the same prompt.

Recent work on RL-based reasoning has explored diversity-aware policy optimization (Yao et al., 2025). A complementary issue is credit assignment in standard GRPO, whose group-wise normalization lacks nonlinear reward-gap transformations and confidence-dependent pairwise reweighting. Under binary rewards, incorrect responses within a group receive identical advantages despite differences in intermediate reasoning quality. Preference optimization methods obtain relative learning signals by comparing candidate responses (Rafailov et al., 2023; Azar et al., 2024; Meng et al., 2024; Hong et al., 2024), motivating the incorporation of reward differences and policy confidence into group-based advantage estimation.

Outcome-only supervision provides limited information about intermediate reasoning progress (Uesato et al., 2022): binary verification evaluates final-answer correctness without assigning partial credit to useful intermediate steps. Learned process reward models provide denser supervision (Lightman et al., 2024), but may introduce reward misspecification, overoptimization, or reward gaming (Gao et al., 2023; Skalse et al., 2022; Casper et al., 2023). These concerns motivate reference-

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

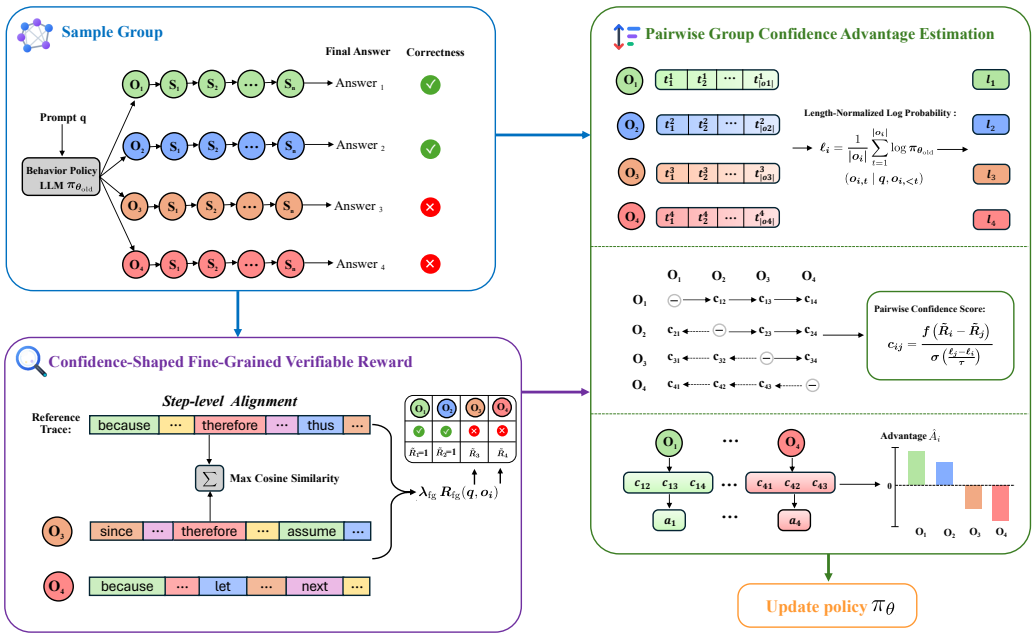


Figure 1: Overview of our PC-GRPO method.

guided partial-credit rewards that distinguish incorrect responses without an additional learned reward model.

To address these limitations, we propose **Pairwise Confidence Group Relative Policy Optimization (PC-GRPO)** (Figure 1). PC-GRPO augments GRPO with a *pairwise confidence-aware advantage estimator* that combines reward differences with behavior-policy confidence, and a *fine-grained verifiable reward* that assigns partial credit to incorrect responses through step-level overlap with reference reasoning traces. These components provide more informative intra-group supervision while retaining a critic-free framework without an additional learned reward model.

Our contributions are threefold. First, we formulate confidence-aware pairwise comparisons for group-based advantage estimation. Second, we construct bounded, reference-guided rewards that distinguish incorrect responses while preserving the higher reward of correct responses. Third, experiments on mathematical, scientific, knowledge-intensive, and coding benchmarks demonstrate improvements over strong RLVR baselines across multiple model scales, with ablations supporting both components.

2 Related Work

Reasoning LLM Post-Training Reasoning LLMs have advanced through chain-of-thought prompting (Wei et al., 2022), instruction tuning, distillation, and reinforcement learning (Guo et al., 2025). RL enables exploration of solution trajectories using task-level feedback. Complementary directions include reasoning-oriented pretraining (Azerbayev et al., 2023), tool-augmented reasoning data (Gou et al., 2023), and mathematical instruction data (Tang et al., 2024). Recent work improves RL-based reasoning by reusing rollout experiences (Zhan et al., 2025) and learning from incorrect responses through negative reinforcement (Zhu et al., 2025).

RLVR and GRPO-family Methods RLVR uses verifiable outcomes for policy optimization (Chen et al., 2025a). GRPO replaces the learned value networks used in PPO-style methods (Schulman et al., 2017; Ziegler et al., 2019) with group reward normalization. DAPO introduces decoupled clipping and dynamic sampling (Yu et al., 2025), GSPO uses sequence-level likelihood ratios (Zheng et al., 2025), and DRA-GRPO incorporates diversity-aware reward adjustments (Chen et al., 2025b). FastGRPO

uses speculative decoding (Zhang et al., 2025b), S-GRPO introduces early-exit strategies (Dai et al., 2025), and Scaf-GRPO provides reasoning scaffolds (Zhang et al., 2025a). Dr. GRPO removes response-length and reward-standard-deviation normalization to address optimization biases (Liu et al., 2025). CoDaPO adapts update weights and sampling to confidence and difficulty (Zhou et al., 2026). AVSPO mitigates advantage collapse by augmenting reward-normalization statistics with virtual rewards, without additional rollouts (He et al., 2026). PC-GRPO combines confidence-aware pairwise advantage estimation with reference-guided partial-credit rewards.

Process Supervision and Fine-Grained Rewards Outcome-only rewards provide limited feedback on intermediate reasoning (Uesato et al., 2022), motivating step-level process supervision (Lightman et al., 2024). Recent approaches include automated supervision in Math-Shepherd (Wang et al., 2024a) and process reinforcement through implicit rewards (Cui et al., 2025). Related self-improvement methods train verifiers on self-generated correct and incorrect solutions (Hosseini et al., 2024), or refine policies through reinforced self-training on model-generated samples (Gulcehre et al., 2023). However, optimizing learned rewards can lead to overoptimization and exploitation (Gao et al., 2023; Coste et al., 2023). PC-GRPO provides denser supervision through step-level overlap with reference reasoning traces, without training an additional reward model.

3 PRELIMINARIES

3.1 RLVR SETUP AND GROUP RELATIVE POLICY OPTIMIZATION

We consider post-training a reasoning language model under the RLVR framework. Let $q \sim P(Q)$ denote a prompt sampled from the prompt distribution, and let $o \sim \pi_\theta(\cdot \mid q)$ denote a response generated by the autoregressive policy π_θ, where $\pi_\theta(o \mid q)$ denotes the sequence probability of the sampled response. In tasks such as mathematical reasoning and code generation, the quality of a response can often be evaluated by a deterministic verifier, giving a reward $R(q, o)$ that is typically binary (Cobbe et al., 2021; Chen et al., 2025a). The standard RLVR objective is therefore to maximize the expected reward:

$$J(\theta) = \mathbb{E}_{q\sim P(Q),\, o\sim\pi_\theta(\cdot|q)} \left[R(q, o)\right]. \tag{1}$$

GRPO (Shao et al., 2024) is a critic-free optimization method that replaces a learned value function with within-group reward normalization. For each prompt q, GRPO samples a group of G responses $\{o_1, \ldots, o_G\}$ independently from the behavior policy $\pi_{\theta_{\text{old}}}(\cdot \mid q)$. Let $r_i = R(q, o_i)$ be the verifier reward for response o_i. For the t-th token of response o_i, define the token-level importance ratio as

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{2}$$

GRPO estimates the advantage of each response by normalizing rewards within the sampled group:

$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G) + \delta}, \tag{3}$$

where $\delta > 0$ is a small constant for numerical stability. With this group-normalized advantage, GRPO applies a PPO-style clipped objective over the generated tokens, together with a Kullback-Leibler (KL) regularization term (Stiennon et al., 2020) against a reference policy π_{ref} to limit excessive policy drift:

$$\begin{aligned} J_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\Big(\rho_{i,t}(\theta)\hat{A}_i,\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_i\Big) \\ - \beta D_{\text{KL}}(\pi_\theta(\cdot \mid q) \,\|\, \pi_{\text{ref}}(\cdot \mid q)) \Bigg]. \end{aligned} \tag{4}$$

This formulation provides a relative training signal within each sampled group while avoiding the cost of training an additional critic.

3.2 Limitations of GRPO under Binary Verifiable Rewards

Standard GRPO has two limitations under binary verifiable rewards. First, although reward centering admits a linear pairwise representation, its normalized advantages do not incorporate nonlinear reward-gap transformations or confidence-dependent pairwise weights. Second, all incorrect responses within a group receive the same binary reward and therefore the same advantage, regardless of partially correct intermediate reasoning. This prevents outcome-only supervision from distinguishing meaningful progress among unsuccessful responses. These limitations motivate a richer intra-group advantage estimator and a fine-grained reward that assigns partial credit while preserving the priority of correct responses.

4 Method

4.1 Overview of PC-GRPO

As shown in Figure 1, PC-GRPO extends GRPO with two modifications aimed at improving credit assignment under sparse verifiable rewards. For each prompt q, we sample a response group $\{o_i\}_{i=1}^{G}$ from the behavior policy, as in GRPO. We combine binary verifier rewards with bounded reference-guided partial credit for incorrect responses, then compare each response with all others in the group to obtain confidence-aware advantages. These advantages enter the same PPO-style clipped objective as GRPO. In this way, PC-GRPO remains critic-free, preserves deterministic supervision, and provides a more informative intra-group training signal than reward normalization alone.

4.2 Pairwise Group Confidence Advantage Estimation

Standard GRPO estimates advantages from group-level reward normalization. While simple and effective, this design can be coarse when many sampled responses receive the same binary outcome reward (Yu et al., 2025). We instead model the relative quality of sampled responses through explicit pairwise comparisons within each group. This design is related in spirit to pairwise ranking and preference-optimization methods (Meng et al., 2024; Hong et al., 2024; Yuan et al., 2023; Song et al., 2024), but the supervision signal here is induced by verifiable rewards rather than human preference labels.

Let $\tilde{R}_i$ denote the received reward of response o_i given prompt q. To avoid length bias in sequence-level probabilities, we use the length-normalized log-probability of each response under the behavior policy:

$$\ell_i = \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \log \pi_{\theta_{\text{old}}} \left(o_{i,t} \mid q, o_{i,<t} \right), \tag{5}$$

where $|o_i|$ denotes the number of generated tokens in response o_i. Since $\pi_{\theta_{\text{old}}}$ is fixed during the current policy update, this confidence signal is treated as a fixed scalar and does not introduce an additional gradient path through the advantage estimator. For each pair (o_i, o_j) with $i \neq j$, we define a pairwise confidence score:

$$c_{ij} = \frac{f(\tilde{R}_i - \tilde{R}_j)}{\sigma\left(\frac{\ell_j - \ell_i}{\tau}\right)}, \tag{6}$$

where $f(\cdot)$ is a smooth transformation of the reward gap (implemented as GELU), $\sigma(\cdot)$ is the sigmoid function, and $\tau > 0$ is a temperature hyperparameter. The numerator reflects the relative reward difference between two responses, while the denominator introduces a policy-dependent confidence reweighting based on their length-normalized log-probability gap under the behavior policy. The positive confidence factor amplifies an individual comparison when the target response has higher behavior-policy support than its counterpart. It strengthens positive comparisons for higher-reward responses and negative comparisons for lower-reward responses, while lower relative support produces less amplification. Confidence therefore adjusts comparison strength without reversing the reward-gap sign. These effects concern individual comparisons; final advantages additionally depend on aggregation and group centering.

We aggregate pairwise comparisons over the group to obtain a pre-normalized advantage signal for each response:

$$a_i = \frac{1}{G-1} \sum_{j \neq i} c_{ij}. \tag{7}$$

We then normalize these aggregated scores across the sampled group to obtain the final advantage:

$$\hat{A}_i = \frac{a_i - \text{mean}(a_1, \ldots, a_G)}{\text{std}(a_1, \ldots, a_G) + \delta}, \tag{8}$$

where $\delta > 0$ is a small constant for numerical stability. The normalized advantage $\hat{A}_i$ directly replaces the standard GRPO reward-normalized advantage in the PPO-style surrogate objective, while preserving the overall critic-free training pipeline. The formal algorithm for PC-GRPO is given by Algorithm 1.

4.3 Confidence-Shaped Fine-Grained Verifiable Reward

The second component of PC-GRPO addresses the sparsity of binary outcome supervision. Under outcome-only verification, incorrect responses receive the same verifier reward even when some contain partially correct intermediate reasoning steps. To provide denser supervision without introducing a learned reward model, we assign an additional fine-grained verifiable reward only to incorrect responses and keep the reward for correct responses unchanged.

For each sampled response o_i, let $R_{\text{acc}}(q, o_i) \in \{0, 1\}$ denote the binary outcome reward from the verifier, and let $R_{\text{fg}}(q, o_i) \in [0, 1]$ denote a bounded fine-grained reward computed from step-level overlap with a reference reasoning trace. We define the combined reward as

$$\tilde{R}(q, o_i) = \begin{cases} 1, & \text{if } R_{\text{acc}}(q, o_i) = 1, \\ \lambda_{\text{fg}}\, R_{\text{fg}}(q, o_i), & \text{if } R_{\text{acc}}(q, o_i) = 0, \end{cases} \tag{9}$$

where $\lambda_{\text{fg}} \in (0, 1)$ controls the contribution of the fine-grained signal. For notational convenience, we write $\tilde{R}_i = \tilde{R}(q, o_i)$ in the rest of this section. Because $R_{\text{fg}}(q, o_i) \in [0, 1]$ and $\lambda_{\text{fg}} < 1$, any incorrect response receives a combined reward strictly smaller than that of a correct response. PC-GRPO supports diverse correct reasoning paths by assigning every verifier-accepted response the full reward of 1, regardless of its similarity to the reference trace. Reference-based partial credit applies only to responses with incorrect final answers.

This reward design follows three principles. First, the auxiliary signal should be deterministic and reference-guided, so that it remains compatible with the RLVR setting and does not require training an additional reward model. Second, the reward should be bounded, ensuring that partial correctness cannot dominate exact correctness. Third, the reward should be local enough to distinguish responses that make meaningful progress from those that are entirely uninformative.

We consider four instantiations of the fine-grained reward R_{fg} to quantify the alignment between a generated reasoning trace $\hat{\mathcal{T}}$ and a reference trace $\mathcal{T}$. First, **continuous semantic similarity** evaluates contextual embedding overlap between the generated and reference traces using a frozen encoder, following BERTScore-style similarity (Zhang et al., 2019). Second, **discrete string-level similarity** captures lexical overlap by using normalized edit distance to measure sequence-level alignment (Levenshtein et al., 1966). Third, **formula-level alignment** isolates mathematical structure by extracting expressions from both traces and computing their Jaccard overlap to assign credit based on exact formula matching.

Finally, **marker-based decomposition** partitions the generated and reference traces into M and N segments using predefined logical discourse markers (e.g., "because", "therefore", "thus", "assume", "so", "hence"). Let $z_{ij} = [1 + \cos(\phi(\hat{s}_i), \phi(s_j))]/2$, where ϕ is a fixed text embedding function, and assign each generated segment to $j_i = \min \arg\max_j z_{ij}$. We retain only the highest valid score for each reference segment:

$$R_{\text{fg}} = \frac{1}{N} \sum_{j=1}^{N} \max(\{z_{ij} : j_i = j,\ z_{ij} \geq \eta\} \cup \{0\}), \tag{10}$$

where $\eta \in (0, 1]$ is a matching threshold and unmatched reference segments receive zero credit. Repeated copies do not increase the reward; a single segment receives at most $1/N$. Appendix A.1 gives the details and proof.

Marker-based decomposition achieves the best downstream performance and training stability among the evaluated variants, so we adopt it as the default R_{fg}. Algorithm 2 gives the implementation.

4.4 Property Discussion

PC-GRPO has several useful design properties. First, it preserves correctness ordering: by construction, any correct response receives reward 1, whereas any incorrect response receives at most $\lambda_{\text{fg}} < 1$. Second, it increases intra-group sensitivity: even when binary rewards assign the same outcome to many sampled responses, the fine-grained reward and pairwise aggregation can still induce a nontrivial ranking signal within the group. Third, it preserves a critic-free training pipeline without introducing a learned value function or an online learned reward model. Together, these properties allow PC-GRPO to provide more informative supervision than standard GRPO in sparse-reward reasoning tasks.

Proposition 4.4.1. Pairwise representation of GRPO normalization. *Let $c_{ij}^{\text{lin}} = \kappa(\tilde{R}_i - \tilde{R}_j)$ be linear, uniformly weighted pairwise scores with $\kappa > 0$. For $G \geq 2$ and $\delta > 0$,*

$$a_i^{\text{lin}} = C(\tilde{R}_i - \bar{R}), \qquad \hat{A}_i^{\text{lin}} = \frac{\tilde{R}_i - \bar{R}}{s_R + \delta/C}, \qquad C = \frac{\kappa G}{G - 1},$$

where $\bar{R}$ and s_R are the group reward mean and population standard deviation.

This identity provides a pairwise representation of GRPO normalization. PC-GRPO retains the critic-free, group-relative framework while introducing nonlinear reward-gap transformation and confidence-dependent weighting (Appendix A.2).

Proposition 4.4.2. Pairwise sign preservation. *For exact GELU, finite rewards and log-confidences, and $\tau > 0$,*

$$\text{sign}(c_{ij}) = \text{sign}(\tilde{R}_i - \tilde{R}_j).$$

Confidence weighting adjusts only the strength of each individual pairwise comparison, without changing the sign determined by the reward difference. See Appendix A.3 for the proof.

5 Experiments

5.1 Datasets

We evaluate PC-GRPO on a diverse benchmark suite spanning mathematical reasoning, knowledge-intensive reasoning, scientific question answering, and code generation. Our evaluation benchmarks include Math-500 (Lightman et al., 2024), AIME-24 (Zhang & Math-AI, 2024), AIME-25 (Zhang & Math-AI, 2025), MMLU-Pro (Wang et al., 2024b), GPQA (Rein et al., 2023), TheoremQA (Chen et al., 2023), and LeetCodeDataset (Xia et al., 2025). These benchmarks are chosen to cover different forms of verifiable reasoning difficulty: Math-500 and the AIME sets emphasize multi-step mathematical problem solving, MMLU-Pro and GPQA evaluate broad and expert-level knowledge reasoning, TheoremQA focuses on scientific and theorem-related question answering, and LeetCodeDataset measures program synthesis under execution-based verification. For RLVR post-training, we use the Mixture-of-Thoughts dataset (Hugging Face, 2025), which contains approximately 350K verified reasoning traces distilled from DeepSeek-R1. We use this corpus only for training, while all reported benchmark results are obtained on the corresponding held-out evaluation sets. Dataset descriptions are provided in Appendix B.

We report **pass@4** (%), the fraction of problems with at least one correct solution among four samples, averaged over ten independent evaluation runs for AIME-24, AIME-25, and GPQA, and five for other benchmarks. Per-run scores are rounded to one decimal before averaging, and means are rounded likewise. Unless otherwise stated, methods share benchmark-specific prompting formats and decoding budgets.

Table 1: Overall performance on the reasoning benchmarks.

Method	Math-500	AIME-24	AIME-25	MMLU-Pro	GPQA	TheoremQA	LeetCodeDataset	Avg.
Qwen-2.5 1.5B Instruct								
IO	55.0	3.0	0.0	13.4	18.5	20.7	0.0	15.8
CoT	76.4	14.6	10.4	20.9	34.4	47.0	14.9	31.2
GRPO	77.8	38.3	34.3	55.4	36.0	52.5	45.5	48.5
DAPO	78.0	41.0	35.7	55.9	36.6	52.8	46.4	49.5 (+0.9)
GSPO	80.4	38.7	34.3	56.3	38.0	53.2	47.2	49.7 (+1.2)
Dr.GRPO	78.8	39.0	35.3	55.7	37.7	53.5	48.2	49.7 (+1.2)
CoDaPO	81.8	40.7	35.7	56.1	37.2	53.9	48.4	50.5 (+2.0)
AVSPO	81.0	39.3	36.7	56.2	38.6	52.7	48.5	50.4 (+1.9)
PC-GRPO	**82.2**	**42.3**	**37.6**	**56.6**	**38.9**	**54.6**	**49.4**	**51.7** (+3.1)
Qwen-2.5 3B Instruct								
CoT	79.8	24.3	20.3	40.7	50.0	54.9	39.9	44.3
GRPO	89.2	41.7	38.3	69.2	59.7	71.0	61.1	61.5
PC-GRPO	**92.6**	**47.0**	**41.6**	**73.7**	**62.4**	**73.4**	**65.4**	**65.2** (+3.7)
Qwen-2.5 7B Instruct								
CoT	84.0	39.0	35.6	52.0	67.0	62.8	60.4	57.3
GRPO	90.4	53.0	47.7	80.9	75.4	79.9	72.7	71.4
PC-GRPO	**93.2**	**59.6**	**56.3**	**84.0**	**78.9**	**82.6**	**75.2**	**75.7** (+4.3)

5.2 BASELINES

We compare **PC-GRPO** with both inference-only baselines and RLVR-based post-training baselines. As prompting baselines, we include **IO**, i.e., direct answer-only prompting without explicit intermediate reasoning, and **Chain-of-Thought (CoT)** prompting (Wei et al., 2022), which elicits step-by-step reasoning in the output. These baselines provide a reference for the gain obtained from RL-based post-training beyond inference-time prompting alone.

For RLVR baselines, we use **GRPO** (Shao et al., 2024) as the primary controlled baseline, since PC-GRPO is built directly on the GRPO training framework. We additionally compare against several strong open RL post-training methods, including **DAPO** (Yu et al., 2025), which focuses on stabilizing optimization under long-horizon reasoning, and **GSPO** (Zheng et al., 2025), which performs sequence-level group policy optimization. We further include **Dr. GRPO** (Liu et al., 2025), **CoDaPO** (Zhou et al., 2026), and **AVSPO** (He et al., 2026) in an extended 1.5B comparison across all seven benchmarks (Table 1). We report the most comprehensive baseline comparison on the 1.5B scale, and use GRPO as the main matched-budget baseline at larger scales. The introductions to the baselines are given in Appendix C.

5.3 EXPERIMENT SETUP

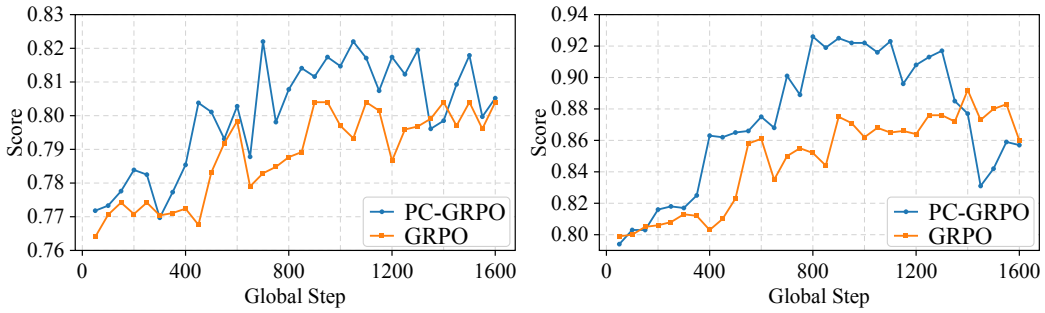
Hardware. All experiments are conducted on a single server with four NVIDIA A800 GPUs (80GB each).

Backbone models. We evaluate Qwen-2.5 Instruct models at 1.5B, 3B, and 7B parameters (Qwen Team, 2024). We also report results on LLaMA-3.2 1B Instruct[1] and SmolLM2-1.7B-Instruct.[2] Following prior RLVR work, we apply RL post-training directly to the instruction-tuned checkpoints without introducing an additional task-specific supervised fine-tuning stage.

Training setup. All experiments are implemented in the OPEN-R1 framework (Hugging Face, 2025), using BF16 training with FSDP and gradient checkpointing. Unless otherwise specified, we follow the default RLVR configuration from Tina (Wang et al., 2025), keeping shared settings fixed across methods for fair comparison. We optimize the policy with AdamW using a learning rate of 1×10^{-6}. We include an explicit KL penalty to a frozen reference policy π_{ref} with coefficient $\beta = 0.04$. Following prior work on stabilized GRPO-style training (Yu et al., 2025), we use asymmetric clipping thresholds $(\epsilon^+, \epsilon^-) = (0.28, 0.20)$. Unless stated otherwise, each run uses 1,000 optimization steps, with eight prompts and four sampled responses per prompt at each step.

[1] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[2] https://huggingface.co/HuggingFaceTB/SmolLM2-1.7B-Instruct



(a) Performance dynamics on Qwen2.5 1.5B (b) Performance dynamics on Qwen2.5 3B

Figure 2: Test-set performance dynamics during RL post-training at different training steps.

The maximum prompt and completion lengths are 2,048 and 4,096 tokens, respectively. Sampling temperatures are 0.7 during training and 0.6 during evaluation, with top-p fixed at 0.95. We save and evaluate checkpoints on a held-out validation split every 50 steps, reporting test results from the best validation checkpoint. For PC-GRPO, we use the marker-based step decomposition reward (Section 4.3) with $\lambda_{\text{fg}} = 0.5$ and confidence temperature $\tau = 1.5$. Evaluation standard deviations are reported in Appendix D.1.

Extended baseline comparison. For Dr. GRPO, CoDaPO, and AVSPO, we use the same Open-R1 pipeline, backbone, training data, prompts, verifier, group size, rollout budget, and optimization settings described above. The Dr. GRPO and CoDaPO ports follow the authors' public implementations and method-specific settings; CoDaPO uses the same total number of generated trajectories.

5.4 Main Results

Table 1 reports the main results across math, knowledge, science, and coding benchmarks. At the Qwen-2.5 1.5B scale, PC-GRPO achieves the best results across all seven tasks, substantially outperforming IO and CoT prompting on strictly verifiable datasets. Compared to standard GRPO, PC-GRPO yields consistent gains: $+4.4$, $+4.0$, $+3.3$, $+1.2$, $+2.9$, $+2.1$, and $+3.9$ points on Math-500, AIME-24, AIME-25, MMLU-Pro, GPQA, TheoremQA, and LeetCodeDataset, respectively. Furthermore, PC-GRPO surpasses DAPO, GSPO, Dr. GRPO, CoDaPO, and AVSPO across all seven benchmarks at the 1.5B scale. The extended comparison includes methods targeting normalization bias, confidence- and difficulty-based training allocation, and advantage collapse.

These improvements scale effectively to larger models. On Qwen-2.5 3B, PC-GRPO consistently outperforms GRPO, notably on Math-500 ($89.2 \rightarrow 92.6$), AIME-24 ($41.7 \rightarrow 47.0$), and LeetCodeDataset ($61.1 \rightarrow 65.4$). At the 7B scale, the largest gains emerge in competition-level math, adding $+6.6$ and $+8.6$ points on AIME-24 and AIME-25. Notably, the most significant improvements occur on difficult tasks with high failure rates under sparse binary verification. This confirms that the benefits of PC-GRPO are scale-agnostic and directly validate our pairwise confidence-aware advantage estimation and fine-grained verifiable rewards.

Figure 2 shows test-set pass@4 on Math-500 during RL post-training for Qwen-2.5 1.5B and 3B Instruct. Across training checkpoints, PC-GRPO generally maintains a stronger trajectory than GRPO, suggesting that the improvements in Table 1 are not solely due to checkpoint selection. These runs extend to 1,600 steps to examine training dynamics; the main experiments use 1,000 steps.

5.5 Analysis and Ablations

Unless otherwise stated, our component ablations use Qwen-2.5 3B Instruct on Math-500, MMLU-Pro, and TheoremQA.

Ablation on the PC-GRPO framework. Table 2 compares eight controlled variants of **PC-GRPO**. **PC-GRPO-1** uses standard GRPO group-normalized advantages while keeping the fine-grained reward unchanged. **PC-GRPO-2**, **PC-GRPO-3**, and **PC-GRPO-4** retain pairwise compar-

isons but replace division-based confidence coupling with addition, subtraction, and multiplication, respectively. **PC-GRPO-5** uses only binary outcome rewards, removing the fine-grained component. Finally, **PC-GRPO-6**, **PC-GRPO-7**, and **PC-GRPO-8** replace the default marker-based decomposition reward with the other fine-grained reward instantiations described in Section 4.3, namely continuous semantic similarity, discrete string-level similarity, and formula-level alignment. These variants isolate whether the performance gain comes from the pairwise advantage estimator, the confidence coupling form, or the particular construction of the fine-grained verifiable reward.

Table 2: Ablation studies on the PC-GRPO framework.

Method	Math-500 pass@4	MMLU-Pro pass@4	TheoremQA pass@4
PC-GRPO	**92.6**	**73.7**	**73.4**
PC-GRPO-1	91.4	72.9	71.9
PC-GRPO-2	92.0	73.1	72.3
PC-GRPO-3	91.6	73.2	71.7
PC-GRPO-4	92.0	73.3	72.8
PC-GRPO-5	91.6	72.8	72.6
PC-GRPO-6	92.0	73.1	72.9
PC-GRPO-7	91.8	72.9	72.7
PC-GRPO-8	92.2	73.4	73.0

The results show that each component contributes to the full method. Replacing the pairwise confidence-aware advantage estimator with the standard GRPO estimator in **PC-GRPO-1** reduces performance from 92.6 to 91.4 on Math-500, from 73.7 to 72.9 on MMLU-Pro, and from 73.4 to 71.9 on TheoremQA, indicating that the improvements cannot be attributed solely to the fine-grained reward. Removing the fine-grained verifiable reward in **PC-GRPO-5** also leads to consistent degradation, with scores of 91.6, 72.8, and 72.6 on the three benchmarks, respectively, confirming the value of partial-credit supervision under sparse outcome rewards. The alternative coupling variants, **PC-GRPO-2/3/4**, remain below the full model, suggesting that division-based coupling outperforms simpler additive, subtractive, or multiplicative forms. Among the alternative fine-grained rewards, formula-level alignment in **PC-GRPO-8** is the strongest substitute, but it still underperforms the default marker-based decomposition. Overall, the full PC-GRPO configuration achieves the best results on all three benchmarks.

Hyperparameter sensitivity. We evaluate the sensitivity of PC-GRPO to the fine-grained reward coefficient λ_{fg} and the confidence temperature τ on Qwen2.5-1.5B-Instruct by varying one parameter at a time. Across the tested settings, PC-GRPO remains above GRPO on all four benchmarks, while the default configuration $(\lambda_{\text{fg}}, \tau) = (0.5, 1.5)$ achieves the highest observed mean scores. Detailed results are provided in Appendix D.2, Table 4.

Generalization across heterogeneous LLM backbones. To examine whether the gains of PC-GRPO extend beyond the Qwen-2.5 family, we further evaluate it on two instruction-tuned backbones with different model families and parameter scales: Llama-3.2-1B-Instruct[1] and SmolLM2-1.7B-Instruct. We use the same RLVR pipeline, training budget, and evaluation protocol as in the main experiments. As reported in Table 5 in Appendix D, PC-GRPO consistently improves over GRPO on all three evaluated benchmarks for both backbones. On Llama-3.2-1B-Instruct, PC-GRPO improves Math-500, MMLU-Pro, and TheoremQA from 66.8 to 69.0, 47.9 to 50.3, and 51.4 to 54.2, respectively. On SmolLM2-1.7B-Instruct, it further improves the corresponding scores from 76.8 to 79.4, 59.5 to 61.1, and 59.9 to 63.9. These results support generalization beyond the Qwen-2.5 family, with gains varying across models and tasks.

6 Conclusion

Conclusion. We presented PC-GRPO, which combines confidence-aware pairwise advantage estimation with reference-guided partial-credit rewards for reasoning LLM post-training. The method retains a critic-free framework without training an additional reward model. Experiments across multiple reasoning domains and model scales show improvements over GRPO and other evaluated RLVR baselines, while ablations support the contributions of both components.

Limitations. PC-GRPO targets tasks with deterministically verifiable outcomes. Extending it to open-ended generation or subjective alignment may require learned or preference-based rewards, with associated risks of reward over-optimization and exploitation.

AI USE STATEMENT

Large language models were used to assist in drafting selected sections of the manuscript and polishing the text for clarity, readability, and consistency. Their use included developing initial drafts from author-provided material, rephrasing sentences, checking grammar, and refining paragraph structure and transitions. These tools also helped identify repetitive expressions and improve the precision and coherence of the writing while preserving the intended meaning. The authors retain responsibility for assessing the accuracy and appropriateness of AI-assisted content.

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, 2023.

Xingwu Chen, Tianle Li, and Difan Zou. On the mechanism of reasoning pattern selection in reinforcement learning for language models. *arXiv e-prints*, pp. arXiv–2506, 2025a.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Dra-grpo: Exploring diversity-aware reward adjustment for r1-zero-like training of large language models. *arXiv preprint arXiv:2505.09655*, 2025b.

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.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025.

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

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

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.

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. *arXiv preprint arXiv:2605.21125*, 2026.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, 2024.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pp. 707–710. Soviet Union, 1966.

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.

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*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

OpenAI. Learning to reason with LLMs, 2024. URL `https://openai.com/index/learning-to-reason-with-llms/`.

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.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

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.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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

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.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18990–18998, 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.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*, 2024.

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, 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, 2024a.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via lora. *arXiv preprint arXiv:2504.15777*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. *arXiv preprint arXiv:2504.14655*, 2025.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950, 2023.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F Wong, and Yu Cheng. Exgrpo: Learning to reason from experience. *arXiv preprint arXiv:2510.02245*, 2025.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*, 2025a.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yizhou Zhang, Ning Lv, Teng Wang, and Jisheng Dang. Fastgrpo: Accelerating policy optimization via concurrency-aware speculative decoding and online draft learning. *arXiv preprint arXiv:2509.21792*, 2025b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Zhanke Zhou, Xiangyu Lu, Chentao Cao, Brando Miranda, Tongliang Liu, Bo Han, and Sanmi Koyejo. The easy, the hard, and the learnable: Confidence and difficulty-adaptive policy optimization for llm reasoning. *arXiv preprint arXiv:2606.07950*, 2026.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

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

A Appendix for Methods

The algorithm to implement the Pairwise Group Confidence Advantage Estimation mechanism is given in Algorithm 1. The algorithm to implement the marker-based decomposition method is given in Algorithm 2.

A.1 Duplicate Suppression and Reference Coverage

Equation (10) assigns each generated segment to one reference segment and retains only the highest valid score for each reference segment. The index j_i is chosen deterministically: ties in normalized cosine similarity are resolved by the smallest reference index. The threshold η applies to $z_{ij} \in [0, 1]$; an equivalent threshold on the original cosine similarity is $2\eta - 1$. A reference segment with no assigned match above the threshold receives zero credit. Algorithm 2 returns zero if either segmented trace is empty.

For $M, N > 0$, write

$$c_j = \max(\{z_{ij} : j_i = j,\ z_{ij} \geq \eta\} \cup \{0\}), \qquad K = \big|\{j : c_j > 0\}\big|.$$

Since $0 \leq c_j \leq 1$ and each generated segment is assigned to only one reference segment, $K \leq \min(M, N)$. Hence

$$0 \leq R_{\text{fg}} = \frac{1}{N}\sum_{j=1}^{N} c_j \leq \frac{K}{N} \leq \frac{\min(M, N)}{N} \leq 1.$$

Now consider adding an exact copy of an already segmented generated segment. The fixed embedding function and deterministic tie rule give the copy the same similarities and reference assignment as the original. Updating the corresponding credit with the same score leaves its maximum unchanged, so the reward does not increase. A trace consisting of one segment and any number of its exact copies can credit at most one reference segment; its reward is therefore at most $1/N$, strictly below 1 when $N > 1$. If a new segment provides a higher-scoring match to an already credited reference segment, it may improve that segment's retained score, but the scores are never added together. Thus normalization by the full reference length accounts for unmatched segments while preventing repeated matches from accumulating credit.

A.2 Connection between Linear Pairwise Aggregation and GRPO

This subsection gives the detailed statement and proof of Proposition 4.4.1. We fix a prompt and a group of $G \geq 2$ responses. Throughout this comparison, both estimators use the same finite rewards $\tilde{R}_1, \ldots, \tilde{R}_G$ and the same standard-deviation convention. Write

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} \tilde{R}_i, \qquad s_R = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(\tilde{R}_i - \bar{R})^2}. \tag{11}$$

We consider a linear comparison variant obtained by setting $f(x) = x$ and replacing the inverse confidence denominator by a constant $\kappa > 0$ shared by all ordered pairs. Thus,

$$c_{ij}^{\text{lin}} = \kappa(\tilde{R}_i - \tilde{R}_j), \qquad a_i^{\text{lin}} = \frac{1}{G-1}\sum_{j \neq i} c_{ij}^{\text{lin}}. \tag{12}$$

For example, equal log-confidences $\ell_i = \ell_j$ give a sigmoid denominator of $1/2$ and hence $\kappa = 2$, provided that the reward-gap transformation is also replaced by the linear map.

Expanded statement of Proposition 4.4.1. Under the assumptions above, let $C = \kappa G/(G-1) > 0$. Then

$$a_i^{\text{lin}} = C(\tilde{R}_i - \bar{R}), \qquad \bar{a}^{\text{lin}} = 0, \qquad s_a = C s_R, \tag{13}$$

where $\bar{a}^{\text{lin}}$ and s_a are the group mean and population standard deviation of the aggregated scores. With stabilizer $\delta > 0$, their normalized advantages satisfy

$$\hat{A}_i^{\text{lin}} = \frac{a_i^{\text{lin}} - \bar{a}^{\text{lin}}}{s_a + \delta} = \frac{\tilde{R}_i - \bar{R}}{s_R + \delta/C}. \tag{14}$$

Proof. First, expand the sum of reward differences for a fixed response i:

$$\begin{aligned}\sum_{j\neq i}(\tilde{R}_i-\tilde{R}_j) &= (G-1)\tilde{R}_i-\sum_{j\neq i}\tilde{R}_j\\ &= (G-1)\tilde{R}_i-(G\bar{R}-\tilde{R}_i)\\ &= G(\tilde{R}_i-\bar{R}).\end{aligned} \tag{15}$$

Multiplying by $\kappa/(G-1)$ yields $a_i^{\text{lin}} = C(\tilde{R}_i-\bar{R})$. Its group mean is

$$\bar{a}^{\text{lin}} = \frac{C}{G}\sum_{i=1}^{G}(\tilde{R}_i-\bar{R}) = \frac{C}{G}(G\bar{R}-G\bar{R}) = 0. \tag{16}$$

Using this zero mean and $C>0$, its standard deviation is

$$\begin{aligned}s_a &= \sqrt{\frac{1}{G}\sum_{i=1}^{G}(a_i^{\text{lin}}-\bar{a}^{\text{lin}})^2}\\ &= \sqrt{\frac{C^2}{G}\sum_{i=1}^{G}(\tilde{R}_i-\bar{R})^2} = Cs_R.\end{aligned} \tag{17}$$

Substituting these expressions into the normalization gives

$$\hat{A}_i^{\text{lin}} = \frac{C(\tilde{R}_i-\bar{R})}{Cs_R+\delta} = \frac{\tilde{R}_i-\bar{R}}{s_R+\delta/C}, \tag{18}$$

which proves the claimed identity. □

Role of the numerical stabilizer. If $s_R>0$ and the stabilizer is omitted, Equation (14) is exactly $(\tilde{R}_i-\bar{R})/s_R$, the standardized reward used by GRPO. With a nonzero stabilizer δ_{GRPO} in GRPO, exact equality is obtained by using $C\delta_{\text{GRPO}}$ in the linear pairwise normalization. If instead both estimators use the same positive constant δ, their denominators are $s_R+\delta/C$ and $s_R+\delta$, respectively; they should not be described as numerically identical. If all rewards are equal, then $s_R=0$ and all centered rewards and pairwise scores are zero, so both stabilized estimators return zero advantages. Normalization without a stabilizer is undefined in that degenerate case.

Scope of the connection. The result identifies reward centering as a special case of linear, uniformly weighted pairwise aggregation. It does not assert that the actual GELU-based, confidence-dependent estimator equals GRPO. In particular, equal confidence weights alone do not make GELU linear. The comparison also requires identical reward inputs: applying the identity to the combined rewards $\tilde{R}_i$ recovers GRPO-style normalization of those combined rewards, rather than necessarily the normalization of the original binary outcome rewards. Finally, this algebraic identity concerns advantage construction and is not a convergence or performance guarantee for the full training procedure.

A.3 Sign Preservation of Pairwise Confidence Scores

We now give the detailed statement and proof of Proposition 4.4.2. Fix a pair of responses with finite combined rewards and finite length-normalized behavior-policy log-probabilities. Define

$$d_{ij} = \tilde{R}_i-\tilde{R}_j, \qquad z_{ij} = \frac{\ell_j-\ell_i}{\tau}, \qquad w_{ij} = \sigma(z_{ij}), \tag{19}$$

where $\tau>0$. For the exact GELU transformation, $f(x)=x\Phi(x)$, where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}\,dt \tag{20}$$

is the standard normal cumulative distribution function. The pairwise score can therefore be written as

$$c_{ij} = \frac{d_{ij}\Phi(d_{ij})}{w_{ij}}. \tag{21}$$

Expanded statement of Proposition 4.4.2. For finite $\tilde{R}_i, \tilde{R}_j, \ell_i, \ell_j$ and $\tau > 0$, the GELU-based pairwise score satisfies

$$\operatorname{sign}(c_{ij}) = \operatorname{sign}(\tilde{R}_i - \tilde{R}_j). \tag{22}$$

In particular, $c_{ij} > 0$ when $\tilde{R}_i > \tilde{R}_j$, $c_{ij} < 0$ when $\tilde{R}_i < \tilde{R}_j$, and $c_{ij} = 0$ when $\tilde{R}_i = \tilde{R}_j$.

Proof. Because z_{ij} is finite, the sigmoid satisfies

$$w_{ij} = \frac{1}{1 + e^{-z_{ij}}} \in (0, 1). \tag{23}$$

For every finite d_{ij}, the normal cumulative distribution function also satisfies $0 < \Phi(d_{ij}) < 1$. Consequently, the factor

$$b_{ij} = \frac{\Phi(d_{ij})}{w_{ij}} \tag{24}$$

is strictly positive. Equation (21) gives $c_{ij} = b_{ij} d_{ij}$. Multiplication by a strictly positive scalar preserves the sign: if $d_{ij} > 0$, then $c_{ij} > 0$; if $d_{ij} < 0$, then $c_{ij} < 0$; and if $d_{ij} = 0$, then $\mathrm{GELU}(0) = 0$ and $c_{ij} = 0$. These three cases establish Equation (22). □

Correct versus incorrect responses. The combined reward definition gives $\tilde{R}_i = 1$ for a correct response and $\tilde{R}_j = \lambda_{\text{fg}} R_{\text{fg}}(q, o_j) \in [0, \lambda_{\text{fg}}]$ for an incorrect response, with $0 < \lambda_{\text{fg}} < 1$. Hence

$$d_{ij} = 1 - \lambda_{\text{fg}} R_{\text{fg}}(q, o_j) \geq 1 - \lambda_{\text{fg}} > 0. \tag{25}$$

Proposition 4.4.2 therefore implies $c_{ij} > 0$ and $c_{ji} < 0$, regardless of which response has greater behavior-policy confidence. The confidence term changes the magnitude of the comparison but cannot reverse this reward preference. More generally, the same reasoning applies whenever the reward-gap transformation has the form $f(x) = xh(x)$ with $h(x) > 0$ over the reward-gap range.

Scope of the guarantee. This is a statement about the sign of an individual pairwise score. The aggregated value a_i combines comparisons with multiple responses, and the final advantage additionally subtracts the group mean; the proposition does not establish global reward ordering of the final advantages or guarantee their signs for every response. It also does not imply that reversed scores have equal magnitudes, that score magnitudes are bounded independently of confidence, or that policy optimization converges. If all combined rewards in a group are identical, every d_{ij} and c_{ij} is zero, and the stabilized advantages are zero as well: confidence reweighting alone cannot create a reward preference in that case.

Algorithm 1 Pairwise Group Confidence Advantage Estimation

Require: Prompt q; behavior policy $\pi_{\theta_{\text{old}}}$; responses $\{o_i\}_{i=1}^{G}$ sampled from $\pi_{\theta_{\text{old}}}(\cdot \mid q)$ with $G \geq 2$; corresponding combined rewards $\{\tilde{R}_i = \tilde{R}(q, o_i)\}_{i=1}^{G}$; temperature τ; numerical constant δ; reward-gap transformation $f(\cdot)$

Ensure: Pairwise confidence-aware advantages $\{\hat{A}_i\}_{i=1}^{G}$

for $i = 1$ to G **do**
  Compute the length-normalized behavior-policy log-confidence:

$$\ell_i = \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \log \pi_{\theta_{\text{old}}}\left(o_{i,t} \mid q, o_{i,<t}\right).$$

  Treat ℓ_i and $\tilde{R}_i$ as fixed scalars during the current policy update.
end for
for $i = 1$ to G **do**
  Initialize the aggregated pairwise score $a_i \leftarrow 0$.
  for $j = 1$ to G **do**
    if $j \neq i$ **then**
      Compute the reward-gap signal:

$$d_{ij} = f(\tilde{R}_i - \tilde{R}_j).$$

      Compute the confidence reweighting term:

$$w_{ij} = \sigma\left(\frac{\ell_j - \ell_i}{\tau}\right).$$

      Compute the pairwise confidence score:

$$c_{ij} = \frac{d_{ij}}{w_{ij}}.$$

      Accumulate the pairwise comparison:

$$a_i \leftarrow a_i + c_{ij}.$$

    end if
  end for
  Average over all pairwise comparisons:

$$a_i \leftarrow \frac{a_i}{G-1}.$$

end for
Compute the group mean and standard deviation of aggregated scores:

$$\mu_a = \frac{1}{G}\sum_{i=1}^{G} a_i, \qquad \sigma_a = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(a_i - \mu_a)^2}.$$

for $i = 1$ to G **do**
  Normalize the aggregated pairwise score to obtain the final advantage:

$$\hat{A}_i = \frac{a_i - \mu_a}{\sigma_a + \delta}.$$

end for
return $\{\hat{A}_i\}_{i=1}^{G}$.

Algorithm 2 Marker-Based Decomposition for Fine-Grained Verifiable Reward

Require: Generated reasoning trace $\hat{\mathcal{T}}$; reference reasoning trace $\mathcal{T}$; predefined marker set $\mathcal{M}$; fixed embedding function $\phi(\cdot)$; normalized-similarity threshold $\eta \in (0, 1]$
Ensure: Fine-grained reward $R_{\text{fg}} \in [0, 1]$

Define the marker set:

$$\mathcal{M} = \{\text{because}, \text{therefore}, \text{thus}, \text{assume}, \text{so}, \text{hence}, \dots\}.$$

Segment the generated trace by markers:

$$\hat{\mathcal{S}} = \text{SplitByMarkers}(\hat{\mathcal{T}}, \mathcal{M}) = \{\hat{s}_1, \dots, \hat{s}_M\}.$$

Segment the reference trace by the same markers:

$$\mathcal{S} = \text{SplitByMarkers}(\mathcal{T}, \mathcal{M}) = \{s_1, \dots, s_N\}.$$

Remove empty or invalid segments from $\hat{\mathcal{S}}$ and $\mathcal{S}$.
Reindex the remaining segments and set $M \leftarrow |\hat{\mathcal{S}}|$ and $N \leftarrow |\mathcal{S}|$.
if $M = 0$ or $N = 0$ **then**
    return 0.
end if
for $j = 1$ to N **do**
    Compute $e_j \leftarrow \phi(s_j)$ and initialize reference credit $c_j \leftarrow 0$.
end for
for $i = 1$ to M **do**
    Compute $\hat{e}_i \leftarrow \phi(\hat{s}_i)$.
    for $j = 1$ to N **do**
        Compute $z_{ij} \leftarrow [1 + \cos(\hat{e}_i, e_j)]/2$.
    end for
    Select $j_i \leftarrow \min \arg\max_{1 \le j \le N} z_{ij}$ ▷ Break ties by the smallest reference index
    if $z_{ij_i} \ge \eta$ **then**
        $c_{j_i} \leftarrow \max(c_{j_i}, z_{ij_i})$ ▷ Retain only the best valid match
    end if
end for
Average credits over all reference segments, including unmatched ones:

$$R_{\text{fg}} = \frac{1}{N} \sum_{j=1}^{N} c_j.$$

Clip the reward to ensure boundedness:

$$R_{\text{fg}} \leftarrow \min\big(1, \max(0, R_{\text{fg}})\big).$$

return R_{fg}.

B Appendix: Datasets

We provide additional details on the evaluation benchmarks and the RLVR training corpus used in our experiments.

Math-500 (Lightman et al., 2024). Math-500 is a 500-problem mathematical reasoning benchmark derived from the MATH dataset. It covers diverse topics, including algebra, geometry, number theory, counting, probability, and other high-school to early undergraduate mathematics. Each problem is evaluated by extracting the model's final answer and matching it against the reference answer in a standardized format.

AIME-24 and AIME-25 (Zhang & Math-AI, 2024; 2025). AIME-24 and AIME-25 are constructed from the 2024 and 2025 American Invitational Mathematics Examination, respectively. The problems require multi-step competition-style mathematical reasoning, and each has an integer answer in the range $[0, 999]$. This answer format supports exact-match evaluation after final-answer extraction.

MMLU-Pro (Wang et al., 2024b). MMLU-Pro is a more challenging extension of MMLU for evaluating advanced knowledge and reasoning across professional and academic domains. It consists of multiple-choice questions spanning subjects such as mathematics, physics, chemistry, engineering, law, health, and computer science. We use it to assess broad knowledge-intensive reasoning beyond purely mathematical problem solving.

GPQA (Rein et al., 2023). GPQA is a graduate-level scientific question-answering benchmark covering biology, physics, and chemistry. In our experiments, we evaluate on the GPQA-Diamond subset and report the results under the GPQA column. This subset contains particularly difficult questions designed to require expert-level domain knowledge and multi-step reasoning.

TheoremQA (Chen et al., 2023). TheoremQA evaluates theorem-grounded reasoning over university-level STEM problems. Solving these questions requires identifying and applying relevant mathematical or scientific theorems. The benchmark includes multiple answer types, such as numerical, symbolic, Boolean, and multiple-choice answers.

LeetCodeDataset (Xia et al., 2025). We use the 228-problem test split of LeetCodeDataset (v0.3.1), a code-generation benchmark built from Python-compatible LeetCode problems. Each instance provides the problem specification and associated evaluation resources, such as starter code and test cases. Generated programs are assessed through execution-based verification against the provided tests.

Mixture-of-Thoughts (Hugging Face, 2025). For RLVR post-training, we use Mixture-of-Thoughts as the training corpus rather than as an evaluation benchmark. The full corpus contains verified reasoning traces distilled from DeepSeek-R1 and covers mathematics, coding, and scientific reasoning. In our experiments, we sample 40K mathematical traces, 40K coding traces, and 50K scientific traces for training, and reserve a balanced validation set of 1,500 examples for checkpoint selection.

C Appendix: Baselines

IO. IO denotes direct-answer prompting, where the model is asked to produce the final answer without an explicit instruction to generate intermediate reasoning steps. This baseline measures the performance of the instruction-tuned backbone under minimal reasoning-oriented prompting.

CoT (Wei et al., 2022). Chain-of-Thought prompting asks the model to generate intermediate reasoning before providing the final answer. It changes only the prompting format and does not involve any additional supervised or reinforcement-learning post-training.

GRPO (Shao et al., 2024). GRPO is the primary controlled RLVR baseline in our experiments. It is a critic-free policy optimization method that estimates advantages by normalizing verifier rewards within a group of responses sampled for the same prompt, thereby avoiding the need to train a separate value model.

DAPO (Yu et al., 2025). DAPO is a GRPO-family post-training method designed to improve the stability and efficiency of long-horizon reasoning optimization. It introduces refinements such as decoupled clipping, dynamic sampling, token-level loss design, and overlong-response control.

GSPO (Zheng et al., 2025). GSPO performs group policy optimization at the sequence level by applying importance weighting and clipping to complete responses rather than individual tokens. This design aims to reduce optimization instability for long generated trajectories.

Dr. GRPO (Liu et al., 2025). Dr. GRPO (GRPO Done Right) addresses the length and difficulty biases induced by response-length averaging and group reward-standard-deviation normalization. It removes these normalization factors while retaining the critic-free policy-gradient framework.

CoDaPO (Zhou et al., 2026). Confidence and Difficulty-Adaptive Policy Optimization assigns a question-level value using rollout confidence and empirical difficulty. It uses this value for update weighting and resampling, concentrating training effort on questions that remain learnable. Our comparison matches the total number of generated trajectories.

AVSPO (He et al., 2026). Adaptive Virtual Sample Policy Optimization addresses advantage collapse when responses within a group receive identical rewards. It monitors the Advantage Collapse Rate and adaptively injects virtual reward samples into the normalization statistics, yielding non-zero advantages for the real sampled responses. The virtual samples are synthetic reward values rather than generated responses, so the intervention requires no additional model rollouts.

D Appendix: Experiments

D.1 Variability across Evaluation Runs

Table 3 reports evaluation variability for the methods and benchmarks in Table 1.

D.2 Sensitivity to Hyperparameters

We study the fine-grained reward coefficient λ_{fg} and the confidence temperature τ on Qwen2.5-1.5B-Instruct. We vary one factor at a time while keeping the remaining settings fixed: $\lambda_{\text{fg}} \in \{0.25, 0.50, 0.75\}$ with $\tau = 1.5$, and $\tau \in \{0.5, 1.5, 3.0\}$ with $\lambda_{\text{fg}} = 0.50$. The shared default setting is included only once in Table 4. Evaluation follows the main protocol: all scores are pass@4 (%), with ten evaluation runs for AIME-24 and five for Math-500, MMLU-Pro, and TheoremQA. Each run's score is rounded to one decimal place before averaging; the means are also reported to one decimal place.

The default setting achieves the highest observed mean score on each benchmark among the configurations tested. Relative to this setting, the largest decreases are 1.2, 1.0, 0.6, and 0.9 percentage points on Math-500, AIME-24, MMLU-Pro, and TheoremQA, respectively. Every tested configuration remains above the corresponding GRPO scores in Table 1. These results indicate limited variation over the tested one-factor ranges; the reported means alone do not establish statistical significance or characterize interactions between the two hyperparameters.

D.3 Generalization across LLM Backbones

The performance on other LLM backbones is reported in Table 5.

Table 3: Sample standard deviations of pass@4 scores (percentage points) across repeated evaluations of the same selected checkpoint. GPQA uses ten runs; the other benchmarks use five.

Method	Math-500	MMLU-Pro	GPQA	TheoremQA	LeetCode
Qwen2.5-1.5B-Instruct					
IO	0.2	0.1	0.4	0.3	0.0
CoT	0.3	0.2	0.3	0.2	0.3
GRPO	0.4	0.1	0.5	0.3	0.4
DAPO	0.2	0.1	0.3	0.2	0.4
GSPO	0.3	0.2	0.4	0.3	0.4
Dr.GRPO	0.3	0.2	0.5	0.1	0.5
CoDaPO	0.3	0.2	0.4	0.1	0.2
AVSPO	0.4	0.2	0.3	0.3	0.4
PC-GRPO	0.2	0.2	0.3	0.2	0.2
Qwen2.5-3B-Instruct					
CoT	0.5	0.3	0.4	0.3	0.3
GRPO	0.3	0.1	0.4	0.2	0.4
PC-GRPO	0.4	0.2	0.3	0.2	0.3
Qwen2.5-7B-Instruct					
CoT	0.4	0.1	0.5	0.3	0.4
GRPO	0.2	0.2	0.4	0.3	0.2
PC-GRPO	0.4	0.2	0.4	0.1	0.4

Table 4: Sensitivity to λ_{fg} and τ on Qwen2.5-1.5B-Instruct. All scores are mean pass@4 (%); AIME-24 uses ten evaluation runs and the other benchmarks use five. One parameter is varied at a time. The default configuration is shown in bold.

λ_{fg}	τ	Math-500	AIME-24	MMLU-Pro	TheoremQA
0.25	1.5	81.4	41.3	56.1	53.9
0.50	**1.5**	**82.2**	**42.3**	**56.6**	**54.6**
0.75	1.5	81.8	41.7	56.3	54.1
0.50	0.5	81.6	41.7	56.0	54.0
0.50	3.0	81.0	41.7	56.4	53.7

Table 5: Performance on other LLM backbones.

Method	Math-500 pass@4	MMLU-Pro pass@4	TheoremQA pass@4
LLaMA-3.2 1B Instruct			
CoT	49.2	30.7	23.3
GRPO	66.8	47.9	51.4
PC-GRPO	69.0	50.3	54.2
SmolLM2-1.7B-Instruct			
CoT	60.4	47.9	41.6
GRPO	76.8	59.5	59.9
PC-GRPO	79.4	61.1	63.9

E APPENDIX: IMPACT STATEMENT

This work presents PC-GRPO, an enhanced reinforcement learning framework for post-training reasoning language models. By relying strictly on verifiable rewards, our approach reduces the need for expensive human preference annotations and mitigates the alignment risks associated with learned reward model hacking. While advancing the reasoning and problem-solving capabilities of language models holds significant potential for scientific, mathematical, and educational applications, we acknowledge that stronger reasoning models inherently carry dual-use risks. Enhanced logic and planning capabilities could be misused to automate deceptive content generation or facilitate harmful coding tasks. Therefore, we encourage future deployments of models optimized with our method to incorporate task-appropriate safeguards, transparent evaluation, and robust access controls.