

Stop Rolling the Dice: Active Group Relative Policy Optimization via Intrinsic Metacognition

Zheng Gong¹ and Ying Sun^{2*} and Hao Peng³ and Philip S. Yu⁴

¹Thrust of AI, Hong Kong University of Science and Technology (Guangzhou)

²The 63rd Research Institute, National University of Defense Technology

³School of Cyber Science and Technology, Beihang University

⁴Computer Science, University of Illinois Chicago

Correspondence: sunying@nudt.edu.cn

Abstract

Group Relative Policy Optimization (GRPO) is an attractive reinforcement learning paradigm for Large Reasoning Models because its group-wise relative advantages eliminate the need for a value network, but its efficiency is severely limited by zero-variance queries whose rollouts are uniformly correct or uniformly incorrect and therefore consume expensive Chain-of-Thought generation without providing useful learning signal. Existing remedies typically rely on offline difficulty labels or external capability trackers, which either fail to follow the model’s rapidly evolving competence or compromise the end-to-end simplicity of GRPO. We propose Active-GRPO, a prompt-free extension that equips the policy with intrinsic fuzzy metacognition via a lightweight confidence head attached to the final prefill representation. The head predicts ordinal confidence bins and is jointly calibrated against empirical rollout outcomes through a dual-objective optimization framework. It further enables a multi-criterion voting mechanism that filters uninformative queries before rollout generation, thereby concentrating compute on samples near the learning boundary with negligible overhead and without perturbing the reasoning distribution. Experiments on mathematical and logical reasoning benchmarks show that Active-GRPO reduces training computation while improving both reasoning accuracy and intrinsic calibration quality.

1 Introduction

Large Language Models (LLMs) (Guo et al., 2025; Jaech et al., 2024) have demonstrated strong reasoning ability in domains such as mathematics (Shao et al., 2024) and coding (Yang et al., 2024), and reinforcement learning (RL) (Arulkumaran et al., 2017; Kaelbling et al., 1996) has become the dominant post-training mechanism for eliciting and

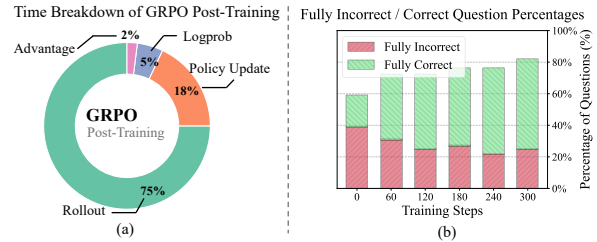


Figure 1: Compute and data inefficiency in GRPO post-training. Left: rollout generation dominates wall-clock cost (75%), far exceeding policy update (18%), log-probability computation (5%), and advantage estimation (2%). Right: fully incorrect and fully correct questions together across training steps, implying that many rollout groups exhibit zero reward variance and yield limited learning signal.

sharpening these capabilities. Among recent RL algorithms, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is especially attractive because it estimates groupwise relative advantages across multiple rollouts for the same prompt and thereby removes the memory-intensive *Value* network, yielding a scalable recipe for training Large Reasoning Models.

This efficiency, however, is undermined by a severe sample-selection problem. GRPO produces useful gradients only when rollout rewards within a prompt group exhibit sufficient variation. If a query is uniformly solved or uniformly failed, the intra-group reward variance collapses, causing the resulting update to become negligible. As shown in Figure 1, rollout generation dominates GRPO post-training cost, while fully correct and fully incorrect questions constitute a substantial share of the training stream across optimization steps. Consequently, a large fraction of computation is spent on zero-variance samples that contribute little or no meaningful learning signal (Guo et al., 2025).

Existing attempts to alleviate this inefficiency typically rely on data filtering (Yu et al., 2025) or adaptive sampling (Zhang et al., 2026). Some

*Corresponding author.

methods use static offline difficulty labels (Liu et al., 2024), whereas others introduce external state-tracking models (Wan et al., 2026), such as Hidden Markov Models (Mao et al., 2026), to estimate whether a question lies near the model’s current competence frontier. These strategies are limited for complementary reasons: offline labels cannot track the rapid non-stationarity of RL training, whereas external trackers sacrifice the architectural simplicity of GRPO and introduce additional modeling and system-level overhead.

We address this problem with Active-GRPO, an end-to-end framework that internalizes sample selection as intrinsic fuzzy metacognition. Before generating reasoning tokens, the policy predicts an ordinal confidence distribution from its final prefill representation through a lightweight confidence head. A dual-objective loss jointly optimizes standard GRPO rewards while calibrating this confidence against empirical rollout outcomes. In addition, a multi-criterion voting mechanism leverages the calibrated signal to discard uninformative extreme-confidence queries while retaining samples near the learning boundary. The resulting framework requires no prompt engineering, offline labels, or external tracking models, preserves the generative distribution of the base policy, and redirects computation toward the most informative training instances.

The main contributions of our work are summarized as follows:

- We propose Active-GRPO, a unified GRPO variant that embeds prompt-free metacognitive self-assessment into the policy backbone through a lightweight confidence head, thereby eliminating the need for external reward models or state trackers.
- We develop a dual-objective calibration scheme and a confidence-guided voting mechanism that identify informative near-boundary queries before rollout generation, substantially reducing computation wasted on zero-variance samples.
- We show on mathematical and logical reasoning benchmarks that Active-GRPO improves training efficiency, downstream reasoning performance, and calibration quality.

2 Preliminaries

In the standard Group Relative Policy Optimization (GRPO) paradigm, we consider a policy model π_θ and a given question (or prompt) q . During training, the policy generates a group of N independent responses (rollouts), denoted as $\{y_1, y_2, \dots, y_N\}$. An environment or rule-based verifier then assigns a raw reward r_i to each rollout y_i . To eliminate the need for a separate Value network, GRPO normalizes the rewards within each group to compute the intra-group advantage A_i :

$$A_i = \frac{r_i - \mu(r)}{\sigma(r)}, \quad (1)$$

where $\mu(r)$ and $\sigma(r)$ are the mean and standard deviation of the rewards $\{r_1, \dots, r_N\}$, respectively. The standard GRPO loss is then computed by clipping the policy ratio $\rho_i = \frac{\pi_\theta(y_i|q)}{\pi_{old}(y_i|q)}$ to maximize the advantage while incorporating a KL-divergence regularization term to prevent the model from drifting excessively far from a reference model π_{ref} :

$$\mathcal{J}_q(\theta) = \frac{1}{N} \sum_{i=1}^N \left[\min(\rho_i A_i, \text{clip}(\rho_i, 1 - \epsilon, 1 + \epsilon) A_i) - \beta \text{KL}(\pi_\theta(y_i|q) \parallel \pi_{ref}(y_i|q)) \right]. \quad (2)$$

The efficiency of GRPO heavily relies on the variance $\sigma(r)$. If a question q is extremely easy (the model solves all N rollouts correctly, $r_i = 1$) or extremely difficult (all rollouts fail, $r_i = 0$), the standard deviation $\sigma(r)$ approaches zero. Consequently, the advantages A_i become zero (or undefined), leading to near-zero gradients. Because generating N long Chain-of-Thought (CoT) rollouts requires substantial computational resources, processing these zero-variance samples leads to a severe waste of training computation.

3 The Active-GRPO Framework

In this section, we present the Active-GRPO framework in detail. We first introduce our core architectural innovation: a lightweight *confidence head* that enables embedding-level fuzzy metacognition without requiring any prompt engineering or output format constraints (§3.1). We then formulate a dual-objective optimization scheme that calibrates the confidence head’s predictions through categorical cross-entropy alignment with empirical performance (§3.2). Finally, we present a multi-criterion

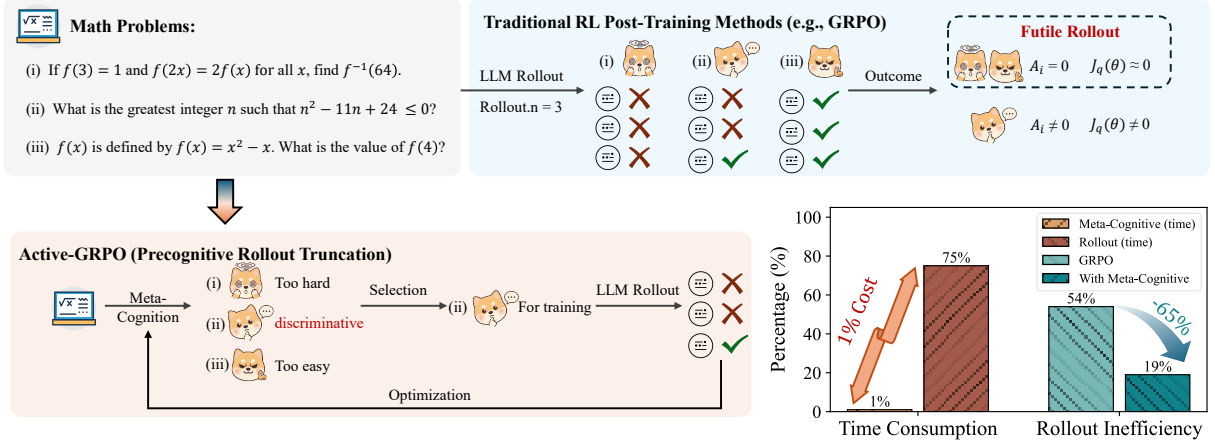


Figure 2: Overview of the Active-GRPO framework. Given a question q , the policy model performs a prefill forward pass, after which the *confidence head*, a lightweight MLP attached to the final token embedding, produces a categorical confidence distribution over five fuzzy ordinal bins. A multi-criterion voting mechanism then determines whether to commit compute to full rollouts. For selected samples, the model generates N rollouts, and the resulting rewards jointly optimize both the GRPO reasoning loss and the categorical calibration loss.

voting mechanism that leverages the calibrated confidence signals to filter uninformative samples *before* committing computational resources to expensive rollouts (§3.3). The complete procedure is summarized in Algorithm 1.

3.1 Fuzzy Metacognition via Latent Confidence Probing

A fundamental prerequisite for compute-aware active sample selection is the ability to estimate, *before* committing resources to full rollouts, whether a given query lies near the model’s learning boundary. Prior self-evaluation approaches (Kadavath et al., 2022; Lin et al., 2022; Ren et al., 2023) typically rely on explicit prompting, instructing the model to verbalize a numerical confidence score before reasoning. Such methods require careful prompt template design, output format parsing, and, in many cases, continued instruction-following fine-tuning. We argue that such prompting-based metacognition introduces unnecessary complexity and carries non-trivial risks of *task interference*. In particular, the additional formatting constraints and instruction-following requirements can perturb the model’s autoregressive generation distribution, potentially degrading reasoning performance or inducing mode collapse in the primary task.

Instead, we propose a purely *architectural* approach that probes the model’s latent confidence directly from its internal representations. During the standard prefill forward pass of the question prompt q through the policy model π_θ , the model computes hidden states at every token position. We

extract the hidden representation at the terminal position:

$$\mathbf{h}_q = \mathbf{H}_\theta(q)|_q \in \mathbb{R}^d, \quad (3)$$

where $\mathbf{H}_\theta(q) \in \mathbb{R}^{|q| \times d}$ denotes the full hidden-state matrix produced by the prefill pass and d is the model’s hidden dimension. This terminal representation $\mathbf{h}_q$ serves as a compressed encoding of the model’s holistic “*first impression*” of the query, implicitly capturing the relationship between the query’s cognitive demands and the model’s current capabilities, before any reasoning tokens are generated.

We augment the policy model with a lightweight *confidence head*, a small multi-layer perceptron (MLP) parameterized by ϕ , that projects this latent representation into a categorical confidence prediction over K ordinal fuzzy bins:

$$\begin{aligned} \mathbf{z}_q &= \text{MLP}_\phi(\mathbf{h}_q) \in \mathbb{R}^K, \\ \mathbf{p}_q &= \text{softmax}(\mathbf{z}_q) = [p_0, p_1, \dots, p_{K-1}], \end{aligned} \quad (4)$$

where $K = 5$ categories discretize the success probability spectrum into the following ordinal fuzzy bins:

Class	Range	Semantics
$\mathcal{A}$ (0)	[0%, 20%)	Very Low Confidence
$\mathcal{B}$ (1)	[20%, 40%)	Low Confidence
$\mathcal{C}$ (2)	[40%, 60%)	Moderate / Boundary
$\mathcal{D}$ (3)	[60%, 80%)	High Confidence
$\mathcal{E}$ (4)	[80%, 100%]	Very High Confidence

This design philosophy of *fuzzy categorical* metacognition, as opposed to prompting-based verbalization or continuous numerical regression, confers three distinctive advantages:

Architectural Transparency with few Task Interference. Because the confidence head operates directly on the latent representation $\mathbf{h}_q$ without modifying the input prompt or imposing output format constraints, there is no need for auxiliary prompt templates, output format parsers, or instruction-following fine-tuning. This design entirely eliminates the risk of distributional shift or mode collapse in the primary reasoning pipeline, which is a critical concern when the meta-cognitive mechanism is trained jointly with the policy.

Calibration-Appropriate Granularity. A continuous regression target predicting the exact success probability $\hat{c}(q) \in [0, 1]$, would impose a disproportionately heavy optimization burden on a lightweight auxiliary head. Critically, the empirical success rate derived from a finite group of N rollouts is inherently a noisy estimator of the true underlying success probability, particularly when N is small. Requiring precise numerical calibration against such noisy supervision signals risks *overfitting to stochastic fluctuations* and may induce optimization instability in the confidence head. In contrast, our five-bin fuzzy discretization provides sufficient discriminative resolution for the downstream task of sample selection. It clearly distinguishes questions that the model will almost certainly solve (class $\mathcal{E}$), those it will almost certainly fail (class $\mathcal{A}$), and those located near the critical learning boundary (classes $\mathcal{B}$ – $\mathcal{D}$), while remaining substantially more robust to label noise and easier to optimize with minimal additional parameters.

Computational Negligibility. Obtaining $\mathbf{p}_q$ requires only the standard prefill forward pass through the question tokens, which would be executed regardless as the first step of any generation, plus a single MLP projection of negligible cost. The total computational overhead is $\mathcal{O}(|q| + d \cdot K)$, compared to $\mathcal{O}(N \cdot L)$ for generating N full reasoning rollouts of average length L , where in practice $N \cdot L \gg |q|$.

3.2 Ordinal Calibration via Dual-Objective Optimization

The confidence head’s categorical predictions are useful for active sampling only if they faithfully reflect the model’s true problem-solving competence. Without explicit supervision, MLP_ϕ may produce arbitrary or systematically biased confidence estimates, particularly during the early stages of training when the model’s reasoning capabilities

are evolving rapidly. We therefore introduce a *calibration objective* that anchors the confidence predictions in empirical evidence, optimized jointly with the standard GRPO loss.

Empirical-to-Categorical Target Construction.

For each question q selected for full rollout generation, we first compute the empirical success rate from the N sampled responses:

$$\hat{c}(q) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(r_i > 0), \quad (5)$$

where r_i denotes the reward for the i -th rollout and $\mathbb{I}(\cdot)$ is the indicator function. This continuous rate is then mapped to a discrete categorical supervision target via ordinal binning:

$$y^*(q) = \min(\lfloor K \cdot \hat{c}(q) \rfloor, K - 1), \quad (6)$$

where $K = 5$ is the number of confidence categories. For example, an empirical success rate of $\hat{c}(q) = 0.35$ maps to target class $y^*(q) = 1$ (class $\mathcal{B}$: Low Confidence), while $\hat{c}(q) = 0.90$ maps to $y^*(q) = 4$ (class $\mathcal{E}$: Very High Confidence). This binning aligns the supervision signal with the same fuzzy granularity used by the confidence head, ensuring consistency between prediction and calibration.

Categorical Cross-Entropy Calibration. We optimize the confidence head to minimize the cross-entropy between its predicted categorical distribution and the empirical target:

$$\mathcal{L}_{\text{Calib}}(\theta, \phi) = -\log p_{y^*(q)}, \quad (7)$$

where $p_{y^*(q)}$ is the probability assigned by the confidence head (Eq. 4) to the ground-truth category $y^*(q)$. By adopting categorical cross-entropy over K ordinal bins instead of mean squared error on a continuous regression target, the optimization landscape aligns naturally with the fuzzy categorical design. The loss penalizes the model for misclassifying the difficulty *level* rather than for imprecise numerical estimation, resulting in more stable gradients and faster convergence of the confidence head.

Joint Dual-Objective Optimization. The total training objective combines the standard GRPO policy gradient loss with the calibration loss:

$$\mathcal{L}_{\text{Total}}(\theta, \phi) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda \mathcal{L}_{\text{Calib}}(\theta, \phi), \quad (8)$$

where $\lambda > 0$ is a hyperparameter controlling the relative importance of calibration. Crucially, although $\mathcal{L}_{\text{Calib}}$ directly supervises the confidence head parameters ϕ , its gradients also propagate through the shared backbone parameters θ via the hidden representation $\mathbf{h}_q$ (Eq. 3), encouraging the model’s internal representations to become more informative about question difficulty. This induces a *virtuous feedback cycle*: as the backbone representations become increasingly more difficulty-aware, the confidence head produces better-calibrated predictions, enabling more effective sample selection, which in turn further accelerates policy learning.

3.3 Filtering Before Thinking: Active Sampling via Multi-Criterion Voting

With a calibrated confidence head available at negligible computational overhead, we now describe how to leverage its output for dynamic sample filtering *before* the expensive rollout generation phase. The core insight is that questions classified with high confidence into extreme categories ($\mathcal{A}$ or $\mathcal{E}$) correspond precisely to the zero-variance samples that dominate computational waste in standard GRPO: they are either too easy or too difficult for the current model to generate diverse intra-group rewards.

Rather than relying on a single scalar threshold, which may be brittle in the face of noisy or miscalibrated early-stage predictions, we design a robust *multi-criterion voting* mechanism that aggregates multiple complementary signals derived from the confidence head’s output distribution. Given the logit vector $\mathbf{z}_q$ and the corresponding probability vector $\mathbf{p}_q = [p_0, \dots, p_{K-1}]$ for a candidate question q , we define three binary voting criteria:

Criterion I: Argmax Extremity. The predicted class $\hat{y} = \arg \max_k z_{q,k}$ falls into an extreme category:

$$v_1(q) = \mathbb{I}(\hat{y} \in \{0, K-1\}). \quad (9)$$

This criterion detects whether the model’s point estimate of question difficulty lands at one of the two extremes of the confidence spectrum, serving as the most direct indicator of a potential zero-variance sample.

Criterion II: Extreme Probability Mass. The maximum probability assigned to the two extreme classes exceeds a threshold $\delta_p \in (0, 1)$:

$$v_2(q) = \mathbb{I}(\max(p_0, p_{K-1}) > \delta_p). \quad (10)$$

This criterion captures cases where a substantial proportion of the predictive mass concentrates on an extreme class, providing a soft magnitude-based signal that complements the hard argmax decision of Criterion I.

Criterion III: Extreme-to-Adjacent Margin.

The probability gap between an extreme class and its adjacent sub-extreme class exceeds a margin threshold $\delta_m > 0$:

$$v_3(q) = \mathbb{I}(\max(p_0 - p_1, p_{K-1} - p_{K-2}) > \delta_m) \quad (11)$$

This criterion verifies that an extreme-class prediction is *sharply peaked* rather than diffused into neighboring bins. A large gap between $\mathcal{A}$ and $\mathcal{B}$ (or between $\mathcal{E}$ and $\mathcal{D}$) indicates that the model confidently places the question at the tail of the difficulty spectrum, rather than hedging across the extreme and sub-extreme categories, a distinction that Criteria I and II alone cannot capture.

Consensus-Based Skip Decision. A candidate question q is *skipped*, excluded from the expensive full rollout generation, only if the vote count reaches a minimum consensus threshold $V_{\min}$:

$$\text{skip}(q) = \mathbb{I}\left(\sum_{j=1}^3 v_j(q) \geq V_{\min}\right) \quad (12)$$

The active training batch is then constructed as:

$$\mathcal{B}_{\text{active}} = \{q \in \mathcal{B}_{\text{cand}} \mid \text{skip}(q) = 0\} \quad (13)$$

By requiring consensus among multiple complementary signals, this voting mechanism achieves a favorable precision–recall trade-off in the filtering decision. It is conservative to avoid mistakenly discarding informative boundary samples, which would otherwise impair learning, while remaining aggressive enough to reliably identify and exclude the genuinely uninformative extreme samples that dominate computational waste. Crucially, this entire filtering procedure occurs *before* any rollout generation, ensuring that the expensive $\mathcal{O}(N \cdot L)$ generation budget is exclusively allocated to questions with genuine learning potential.

4 Experiments

In this section, we conduct experiments to evaluate Active-GRPO. We first describe the setup (§4.1), then present the main results on reasoning benchmarks (§4.2), followed by analyses of computational efficiency (§4.3), calibration quality (§4.4), and ablation studies (§4.5). We include additional **datasets** and **backbones** in Appendix A.2.

4.1 Experimental Setup

Training Datasets. Aligned with prior work (Mao et al., 2026; Sheng et al., 2025), we conduct RL post-training on the training split of MATH (Hendrycks et al., 2021), which contains 7,500 competition-style mathematical problems. This dataset provides a standardized and sufficiently challenging training source for studying sample efficiency in reasoning RL.

Evaluation Benchmarks. We evaluate on a diverse suite of mathematical and logical reasoning benchmarks: (1) **MATH-500** (Hendrycks et al., 2021), a held-out test set of 500 competition-level math problems; (2) **GSM8K** (Cobbe et al., 2021), 1,319 test problems requiring multi-step arithmetic; (3) **Minerva Math** (Lewkowycz et al., 2022), a collection of quantitative reasoning problems from science and mathematics; (4) **OlympiadBench** (He et al., 2024), a challenging benchmark of olympiad-level problems; (5) **AMC23**, a held-out set of American Mathematics Competition problems that emphasizes concise contest-style reasoning.

Backbones. We use Qwen2.5-1.5B and Qwen2.5-7B as policy backbones to cover two representative model scales. The 1.5B model serves as a compute-constrained regime in which zero-variance samples are especially frequent, while the 7B model tests whether the proposed active selection strategy remains effective for stronger policies. Unless otherwise specified, all compared methods share the same prompt format, reward design, tokenizer.

Baselines. We compare Active-GRPO against the following methods:

- **GRPO** (Shao et al., 2024): the standard GRPO algorithm without any sample filtering.
- **GSPO** (Zheng et al., 2025) uses a different policy update rule via the sequence loglikelihood.
- **DAPO** (Yu et al., 2025) incorporates data filtering to remove zero-variance samples based on post-hoc reward statistics.
- **DPS** (Mao et al., 2026) uses Hidden Markov Models to estimate model capability evolution for proactive sample selection.

Our Variants. We instantiate the proposed active filtering mechanism on top of two group-based RL objectives. **Active-GRPO** augments GRPO with the confidence head, dual-objective calibration loss, and pre-rollout voting mechanism. **Active-GSPO**

applies the same active selection strategy to GSPO.

Implementation Details. All experiments are conducted using the veRL framework (Sheng et al., 2025) on 8×H800 (80GB) GPUs. We set the rollout group size $N = 8$ and the maximum generation length to 2,048 tokens. The confidence head is a two-layer MLP with hidden dimension 256, adding fewer than 0.5M parameters ($<0.01\%$ of the base model). We use the AdamW optimizer with a learning rate of 1×10^{-6} for the policy and 5×10^{-5} for the confidence head. The calibration weight is set to $\lambda = 0.1$. For the multi-criterion voting mechanism, we use $\delta_p = 0.6$, $\delta_m = 0.3$, and $V_{\min} = 2$. To mitigate cold-start effects, we use a 200-step warm-up stage during which all candidate queries are retained and full rollout groups are generated; confidence-guided filtering is activated only afterwards. Training runs for 3 epochs with a cosine learning rate schedule. We use a KL penalty coefficient $\beta = 0.04$ and clipping parameter $\epsilon = 0.2$. All reported evaluation metrics are averaged over eight different seeds.

4.2 Main Results

Table 1 reports the main comparison. Across both model scales, Active-GRPO consistently outperforms vanilla GRPO while requiring substantially fewer rollouts. On Qwen2.5-1.5B, Active-GRPO increases the average score from 40.7 to 41.5, with especially clear gains on the more difficult benchmarks, including Minerva (+0.8), OlympiadBench (+0.8), and AMC23 (+0.5). At the same time, it reduces the rollout volume from 737k to 311k, corresponding to a 57.8% reduction, and shortens runtime from 20h to 14h. On Qwen2.5-7B, the gains are even more pronounced: Active-GRPO improves the average score from 56.8 to 58.0 and surpasses both DAPO and DPS, while cutting rollout volume by 29.3% and runtime by 28.0% relative to standard GRPO. These results directly support our central claim that reallocating compute away from high-confidence extreme samples and toward boundary samples can improve training efficiency and reasoning quality. Moreover, Active-GSPO achieves the best overall average on both backbones, indicating that the proposed active sampling mechanism transfers beyond a single policy objective and remains effective across group-based RL variants.

Base Model	Method	MATH-500	GSM8K	Minerva	Olympiad	AMC23	↑Avg.	↓Rollout	↓Runtime
Qwen2.5-1.5B	no RL	49.4	69.4	9.6	15.9	22.2	33.3	-	-
	GRPO	56.7	76.6	13.0	23.3	34.1	40.7	737k	20h
	GSPO	58.6	76.9	13.6	23.6	34.5	41.4	737k	20h
	DAPO	<u>59.1</u>	<u>77.3</u>	13.4	23.7	34.3	41.6	737k	20h
	DPS	58.2	76.3	12.9	23.6	29.4	40.1	737k	22h
	Active-GRPO	58.3	76.8	<u>13.8</u>	<u>24.1</u>	<u>34.6</u>	41.5	311k	14h
	Active-GSPO	59.5	77.9	14.5	24.5	35.3	42.3	<u>327k</u>	<u>15h</u>
Qwen2.5-7B	no RL	72.7	91.2	22.7	37.2	48.8	54.5	-	-
	GRPO	76.4	91.8	23.2	38.9	53.8	56.8	287k	25h
	GSPO	76.8	91.8	23.7	39.5	55.7	57.5	287k	25h
	DAPO	<u>77.5</u>	92.5	23.5	39.5	55.6	57.7	287k	25h
	DPS	76.8	92.0	23.1	39.0	54.2	57.0	287k	32h
	Active-GRPO	77.2	<u>92.7</u>	<u>24.6</u>	<u>39.7</u>	<u>56.0</u>	<u>58.0</u>	203k	18h
	Active-GSPO	77.9	93.2	24.9	40.3	56.1	58.5	<u>210k</u>	<u>18h</u>

Table 1: Main results on mathematical reasoning benchmarks. Active variants consistently improve or preserve downstream accuracy while substantially reducing the number of rollouts and end-to-end wall-clock training time. The last two columns report the cumulative rollout count and total runtime under the same training budget.

4.3 Computational Efficiency Analysis

Because rollout generation constitutes the primary computational cost in GRPO training, reducing the number of retained questions yields near-direct practical speedups. As shown in Table 1, Active-GRPO reduces cumulative rollouts from 737k to 311k on Qwen2.5-1.5B and from 287k to 203k on Qwen2.5-7B, with similar trends observed for Active-GSPO. These reductions translate into wall-clock savings of 6 hours for the 1.5B model and 7 hours for the 7B model under identical optimization budgets. The scale-dependent filtering ratio is also informative. The smaller 1.5B model benefits from more aggressive filtering because it contains a higher proportion of uniformly failed or uniformly solved questions, making extreme-confidence predictions easier to identify. In contrast, the stronger 7B model retains a larger share of informative boundary samples, resulting in a lower but still substantial skip rate. Notably, Active-GRPO remains more efficient than DPS despite eschewing external trackers. DPS preserves the full rollout budget and incurs additional HMM-related overhead, extending runtime to 22h and 32h on the two backbones, respectively. This comparison underscores that pre-rollout filtering offers greater computational efficiency.

4.4 Calibration Quality

Table 2 evaluates whether the confidence head is not merely discriminative, but genuinely calibrated. Active-GRPO reduces the Expected Calibration Error from 0.163 to 0.108 and improves

Method	ECE (↓)	Brier Score (↓)
DPS	0.163	0.252
Active-GRPO	0.108	0.198

Table 2: Calibration quality on MATH-500 using Qwen2.5-7B. ECE: Expected Calibration Error. Lower is better for both metrics.

the Brier score from 0.252 to 0.198 on Qwen2.5-7B. This result indicates that the dual-objective training scheme successfully aligns the predicted ordinal confidence with empirical rollout outcomes. This property is crucial for reliable active sampling. If the confidence head were systematically overconfident, informative boundary samples could be discarded prematurely. Conversely, if it were underconfident, the method would fail to suppress zero-variance queries effectively.

4.5 Ablation Studies

Component Ablation. Table 3 shows that removing the multi-criterion voting mechanism and replacing it with a single argmax threshold consistently degrades performance across all reported benchmarks. This result suggests that the effectiveness of Active-GRPO does not stem from confidence prediction alone; rather, robust skip decisions require combining multiple complementary views of the predictive distribution.

Number of Confidence Bins. The number of ordinal confidence bins also matters. Using only $K = 3$ bins reduces performance because the confidence space becomes too coarse to clearly distinguish boundary, and confidently solvable questions.

Variant	MATH-500	GSM8K	AMC23
Active-GRPO	58.3	76.8	34.6
w/o Voting (argmax only)	57.6	76.2	33.8
$K = 3$ bins	57.9	76.4	34.0
$K = 5$ bins (default)	58.3	76.8	34.6
$K = 10$ bins	57.8	76.3	33.9
$V_{\min} = 1$	57.7	76.3	33.9
$V_{\min} = 2$ (default)	58.3	76.8	34.6
$V_{\min} = 3$	58.0	76.5	34.2

Table 3: Ablation studies on Qwen2.5-1.5B. The top block removes key components; the middle block varies the number of confidence bins K ; the bottom block varies the minimum vote threshold $V_{\min}$.

Increasing the granularity to $K = 10$ likewise underperforms relative to the default setting, likely because empirical success rates estimated from a finite rollout group are too noisy to support overly fine-grained supervision.

Minimum Vote Threshold. The minimum vote threshold $V_{\min}$ governs the aggressiveness of filtering. Setting $V_{\min} = 1$ is overly aggressive: a sample is skipped whenever any single criterion fires, which increases the risk of discarding genuinely informative boundary questions and harms all three benchmarks. At the other extreme, $V_{\min} = 3$ is too conservative and therefore leaves more zero-variance samples in the training stream. The default configuration $V_{\min} = 2$ achieves the best balance between selectivity and recall, showing that partial consensus among the three criteria is sufficient to suppress uninformative extremes without sacrificing useful learning signal.

5 Related Work

RL for LLM Reasoning. RL has become a central paradigm for enhancing the reasoning capabilities of LLMs during post-training. Early efforts such as RLHF (Ouyang et al., 2022; Bai et al., 2022) demonstrated that RL from human feedback can substantially improve model alignment and instruction following. More recently, the focus has shifted toward leveraging RL to unlock deeper reasoning abilities. GRPO (Shao et al., 2024) and DeepSeek-R1 (Guo et al., 2025) eliminates the value network through intra-group advantage estimation. Subsequent work has explored the interplay between supervised fine-tuning and RL (Chu et al., 2025; Wei et al., 2025), and whether RL genuinely extends reasoning beyond the base model’s latent capabilities (Yue et al., 2025). Despite these

advances, the sample efficiency of GRPO remains a critical bottleneck, as zero-variance rollout groups yield no learning signal, motivating the need for intelligent sample selection strategies.

Confidence Calibration and Self-Evaluation in LLMs. Calibration refers to the consistency between a model’s perceived confidence and its actual accuracy. While LLMs demonstrate a degree of internal self-knowledge—allowing them to distinguish answerable questions from unanswerable ones (Kadavath et al., 2022)—research has shown that this uncertainty can be effectively elicited through verbal expressions (Lin et al., 2022). Such self-evaluation mechanisms are crucial for selective generation, enabling models to abstain from answering when uncertain (Ren et al., 2023). However, a notable paradox exists: while Chain-of-Thought (CoT) reasoning enhances task performance, it often degrades calibration by inducing overconfidence (Xiong et al., 2024). This mismatch has led to recent efforts in designing advanced prompting strategies to extract more reliable uncertainty estimates from these models (Tian et al., 2023).

Efficient Sample Selection. The high cost of generating long chain-of-thought rollouts has spurred research into efficient sample selection for RL training. Recent works have explored adaptive rollout adjustments (Zhang et al., 2026) and off-policy replay buffers (Wan et al., 2026) to minimize generation redundancy. Moving toward predictive modeling, Mao et al. (2026) employ Hidden Markov Models to anticipate capability evolution for proactive selection. In contrast to these approaches, our method derives selection signals directly from the model’s own calibrated confidence, bypassing the need for external models, offline labels, or complex auxiliary tracking systems.

6 Conclusion

We presented Active-GRPO, an end-to-end framework that addresses the computational inefficiency caused by zero-variance rollouts in GRPO. By augmenting the policy model with a lightweight confidence head that probes the final token embedding, our method enables fuzzy metacognition at negligible cost and without perturbing the reasoning distribution. A dual-objective optimization jointly calibrates this confidence signal against empirical rollout outcomes, while a multi-criterion voting

mechanism filters uninformative samples before expensive generation. Experiments on mathematical reasoning benchmarks demonstrate that Active-GRPO significantly reduces wasted computation, and accelerates training convergence.

Acknowledgements

This work is partly supported by the Science Research Project of Hebei Higher Education Institutions under grant CYZD2026005, Shijiazhuang Science and Technology Plan Project through grant 2511301807A.

Limitations

Due to computational constraints, our experiments are conducted on models up to a moderate parameter scale. While the architectural design of the confidence head is scale-agnostic and the computational overhead remains negligible at any model size, verifying the effectiveness of Active-GRPO on larger models (e.g., 70B+ parameters) is an important direction for future investigation.

References

- Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE signal processing magazine*, 34(6):26–38.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In *International Conference on Machine Learning*, pages 10818–10838. PMLR.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.
- Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. 2026. Dynamics-predictive sampling for active rl finetuning of large reasoning models. *arXiv preprint arXiv:2603.10887*.

- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models. In *Proceedings on*, pages 49–64. PMLR.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.
- Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.
- Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. 2026. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. *arXiv preprint arXiv:2602.20722*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.
- Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. 2025. First SFT, second RL, third UPT: Continual improving multi-modal LLM reasoning via unsupervised post-training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2026. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186.
- John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhi Zhang, Zhen Han, Costas Mavromatis, Qi Zhu, Yunyi Zhang, Sheng Guan, Dingmin Wang, Xiong Zhou, Shuai Wang, Soji Adeshina, and 1 others. 2026. Train less, learn more: Adaptive efficient rollout optimization for group-based reinforcement learning. *arXiv preprint arXiv:2602.14338*.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Appendix

A.1 Active-GRPO Algorithm

Algorithm 1 summarizes the full training procedure of Active-GRPO. Each iteration consists of four stages. First, the model performs a lightweight prefill pass and uses the confidence head to predict a categorical confidence distribution for each candidate question. Second, a multi-criterion voting rule retains only samples that do not exhibit sufficiently strong extreme-confidence signals, thereby avoiding expensive rollouts on questions that are likely to be uniformly solved or uniformly failed. Third, for the retained questions, the policy generates a rollout group, computes verifier rewards, and converts the empirical success rate into an ordinal calibration target. Finally, the model jointly updates the policy parameters and the confidence head using the standard GRPO objective together with the auxiliary calibration loss. Compared with vanilla GRPO, the key difference is that Active-GRPO moves sample selection ahead of full rollout generation, allowing training compute to be concentrated on boundary samples that are more likely to yield informative learning signals.

A.2 Additional Experimental Results

Evaluation on General Reasoning Tasks. To further assess the generality of Active-GRPO beyond mathematical reasoning, we extend the evaluation to two widely used general reasoning benchmarks, **ARC-c** (Clark et al., 2018) and **MMLU-Pro** (Wang et al., 2024) with the Qwen2.5-1.5B backbone. Our evaluation settings follow Yan et al. (2026). The backbones all also trained on the training split of MATH dataset. This design isolates the effect of the proposed active selection mechanism and allows us to examine whether the benefits of Active-GRPO transfer to out-of-domain reasoning tasks that are not directly aligned with the RL training distribution.

The results are reported in Table 4. Active-GRPO and Active-GSPO consistently shows improvements over baseline methods. These results indicate that the proposed active selection mechanism remains effective beyond mathematical evaluation benchmarks, and can adapt to general reasoning tasks without increasing training cost.

Evaluation with the Llama Backbone. We also examine whether the proposed method transfers across backbone families by replacing Qwen with

Method	ARC-c	MMLU-Pro	↑Avg.	↓Rollout
Qwen2.5-1.5B	34.2	15.8	25.0	-
+GRPO	36.8	16.4	26.6	737k
+GSPO	37.2	16.5	<u>26.9</u>	737k
+DPS	36.6	16.4	26.5	737k
+Active-GRPO	37.1	16.7	26.9	311k
+Active-GSPO	37.8	17.0	27.4	<u>327k</u>

Table 4: General reasoning evaluation on ARC-c and MMLU-pro datasets with Qwen2.5-1.5B backbone.

Llama-3.2-3B-Instruct. To ensure a controlled comparison, we follow the same training pipeline used for the Qwen models, including the same RL training dataset, prompt format, reward construction, rollout group size, and optimization protocol. The model is then evaluated on the same two general reasoning benchmarks, ARC-c and MMLU-Pro. This setting allows us to test whether the gains of Active-GRPO depend on a particular architecture family or instead reflect a more general improvement in sample allocation during group-based RL post-training.

Table 5 presents corresponding results for the Llama backbone. Consistent improvements in this setting provide additional evidence that the proposed confidence-guided filtering mechanism is architecture-agnostic and can be integrated into different instruction-tuned backbones with minimal modification.

Method	ARC-c	MMLU-Pro	↑Avg.	↓Rollout
Llama-3.2-3B	36.1	17.2	26.7	-
+GRPO	38.0	17.9	28.0	737k
+GSPO	37.8	17.5	27.7	737k
+DPS	38.1	17.9	28.0	737k
+Active-GRPO	38.5	18.0	<u>28.3</u>	359k
+Active-GSPO	38.4	18.3	28.4	<u>370k</u>

Table 5: General reasoning evaluation on ARC-c and MMLU-pro datasets with Llama-3.2-3B-Instruct backbone. The training procedure is same as Qwen2.5.

A.3 Additional Ablation Results

To further examine the robustness of the proposed filtering mechanism, we report additional ablations from the rebuttal experiments. Unless otherwise specified, these experiments use the same training and evaluation protocols as the main experiments.

Warm-up for the Confidence Head. Table 6 compares training with and without the 200-step warm-up on Qwen2.5-7B. The warm-up improves both accuracy and calibration, supporting the use of full rollout supervision before confidence-guided filtering is enabled.

Setting	MATH-500	GSM8K	Minerva	Olympiad	AMC23	↑Avg.	↓ECE	↓Brier
No warm-up	76.1	92.4	24.5	38.6	54.4	57.2	0.127	0.224
200-step warm-up (Ours)	77.2	92.7	24.6	39.7	56.0	58.0	0.108	0.198

Table 6: Effect of the warm-up stage on mathematical tasks with Qwen2.5-7B and Active-GRPO.

Hyperparameter Sensitivity. Table 7 evaluates the sensitivity to the voting thresholds and calibration weight on Qwen2.5-1.5B. The results remain stable around the default configuration ($\delta_p = 0.6$, $\delta_m = 0.3$, and $\lambda = 0.1$).

δ_p	δ_m	λ	MATH-500	GSM8K	Minerva	Olympiad	AMC23	↑Avg.	↓Rollouts
0.6	0.3	0.1	58.3	76.8	13.8	24.1	34.6	41.5	311k
0.8	0.3	0.1	58.5	76.8	14.1	23.8	34.2	41.5	347k
0.6	0.5	0.1	58.2	76.5	14.0	24.0	34.5	41.4	335k
0.6	0.3	0.5	57.9	76.4	13.7	23.6	34.3	41.2	307k

Table 7: Sensitivity to δ_p , δ_m , and λ on mathematical tasks with Qwen2.5-1.5B and Active-GRPO.

Separate Filtering of Extreme-Confidence Queries. Table 8 separately evaluates low-confidence and high-confidence filtering. Filtering either type alone yields intermediate performance, whereas bidirectional filtering provides the best accuracy with the fewest rollouts.

Strategy	MATH-500	GSM8K	Minerva	Olympiad	AMC23	↑Avg.	↓Rollouts
No filtering (GRPO)	56.7	76.6	13.0	23.3	34.1	40.7	737k
Filter low-confidence only	57.8	76.4	13.2	23.8	34.2	41.1	595k
Filter high-confidence only	57.4	76.8	13.0	23.6	34.0	41.0	483k
Bidirectional filtering (Ours)	58.3	76.8	13.8	24.1	34.6	41.5	311k

Table 8: Effects of filtering low-confidence and high-confidence queries separately on mathematical tasks with Qwen2.5-1.5B and Active-GRPO.

Prompt-Based Baseline. We additionally compare Active-GRPO with a prompting-based confidence baseline that asks the model to output a confidence score before reasoning. As shown in Table 9, Active-GRPO achieves higher accuracy and

Variant	MATH-500	GSM8K	AMC23	↓Rollouts	↓ECE	↓Brier
$K = 3$	57.9	76.4	34.0	426k	0.096	0.185
$K = 5$	58.3	76.8	34.6	311k	0.108	0.198
$K = 10$	57.8	76.3	33.9	482k	0.127	0.224

Table 10: Effect of the number of confidence bins K on Qwen2.5-1.5B.

substantially better calibration while using fewer rollouts.

Strategy	MATH-500	GSM8K	Minerva	Olympiad	AMC23	↑Avg.	↓Rollouts	↓ECE	↓Brier
Prompt-based	56.7	75.2	12.5	22.6	33.7	40.1	352k	0.253	0.329
Active-GRPO	58.3	76.8	13.8	24.1	34.6	41.5	311k	0.108	0.198

Table 9: Comparison with a prompting-based confidence baseline using Qwen2.5-1.5B.

Number of Confidence Bins. Table 10 supplements the existing accuracy ablation with rollout and calibration metrics. The default $K = 5$ provides a favorable balance between calibration quality and sample efficiency.

B LLM Usage

The LLM was employed solely for the purpose of linguistic polishing and enhancing the readability of the manuscript. All intellectual contributions pertaining to the central ideas, methodological design, experimental work, and analytical insights originate entirely from the authors.

Algorithm 1 Active Group Relative Policy Optimization (Active-GRPO)

Require: Policy model π_θ , Confidence head MLP ϕ , Reference model π_{ref}

Require: Question pool $\mathcal{D}$, Rollout group size N , Learning rate η , Number of categories $K=5$

Require: Calibration weight λ , Vote thresholds δ_p, δ_m , Minimum votes $V_{\min}$

Ensure: Optimized policy π_{θ^*} and calibrated confidence head MLP ϕ^*

```
1: while not converged do
2:   Sample candidate batch  $\mathcal{B}_{\text{cand}} \sim \mathcal{D}$ 
3:    $\mathcal{B}_{\text{active}} \leftarrow \emptyset$ 
   {Phase 1: Latent Confidence Probing & Multi-Criterion Voting}
4:   for each question  $q \in \mathcal{B}_{\text{cand}}$  do
5:     Extract terminal hidden state:  $\mathbf{h}_q = \mathbf{H}_\theta(q)_{|q|}$  ▷ Standard prefill pass
6:     Compute confidence logits and distribution:  $\mathbf{z}_q = \text{MLP}_\phi(\mathbf{h}_q)$ ,  $\mathbf{p}_q = \text{softmax}(\mathbf{z}_q)$ 
7:      $\hat{y} \leftarrow \arg \max_k z_{q,k}$ 
8:     Compute votes:  $v_1 = \mathbb{I}(\hat{y} \in \{0, K-1\})$ ,  $v_2 = \mathbb{I}(\max(p_0, p_{K-1}) > \delta_p)$ ,  $v_3 =$   

        $\mathbb{I}(\max(p_0 - p_1, p_{K-1} - p_{K-2}) > \delta_m)$ 
9:     if  $v_1 + v_2 + v_3 < V_{\min}$  then
10:       $\mathcal{B}_{\text{active}} \leftarrow \mathcal{B}_{\text{active}} \cup \{q\}$  ▷ Retain for full rollout
11:    end if
12:  end for
   {Phase 2: Full Rollout Generation & Reward Computation}
13:  for each question  $q \in \mathcal{B}_{\text{active}}$  do
14:    Generate  $N$  rollouts  $\{y_1, y_2, \dots, y_N\} \sim \pi_\theta(\cdot | q)$ 
15:    Compute rewards  $\{r_1, r_2, \dots, r_N\}$  via environment verifier
16:    Empirical success rate:  $\hat{c}(q) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(r_i > 0)$ 
17:    Categorical target:  $y^*(q) = \min(\lfloor K \cdot \hat{c}(q) \rfloor, K-1)$ 
18:    Intra-group advantages:  $A_i = (r_i - \mu(r))/\sigma(r)$ 
    {Phase 3: Dual-Objective Loss Computation}
19:    Compute GRPO loss  $\mathcal{L}_{\text{GRPO}}(\theta)$  using  $A_i$  and  $\text{KL}(\pi_\theta \| \pi_{\text{ref}})$ 
20:    Compute calibration loss  $\mathcal{L}_{\text{Calib}}(\theta, \phi) = -\log p_{y^*(q)}$ 
21:     $\mathcal{L}_{\text{Total}}^{(q)} = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda \mathcal{L}_{\text{Calib}}(\theta, \phi)$ 
22:  end for
   {Phase 4: Joint Policy & Confidence Head Update}
23:   $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{(\theta, \phi)} \left( \frac{1}{|\mathcal{B}_{\text{active}}|} \sum_{q \in \mathcal{B}_{\text{active}}} \mathcal{L}_{\text{Total}}^{(q)} \right)$ 
24: end while
25: return  $\pi_{\theta^*}, \text{MLP}_{\phi^*}$ 
```
