DENOISING-GRPO: CRITIC-FREE CONFIDENCE-GATED CONSENSUS REWARD CORRECTION FOR NOISY REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) is a practical critic-free method for reasoning reinforcement learning (RL), yet it struggles with noisy supervision, such as inconsistencies between prompts and expected labels. Such noise compromises group-relative advantage estimation, slowing convergence and, in the worst case, steering the model toward degenerate solutions. To address this bottleneck, we propose DENOISING-GRPO, a highly practical and scalable drop-in reward-side extension of GRPO. DENOISING-GRPO converts within-group answer agreement into a structured correction target. To prevent the policy from being overridden by unstable or incorrect majorities during early training, it uses the model's internal generation confidence to apply this correction only when the group is deemed reliable. Because it operates entirely on the reward construction side, our method avoids heavier auxiliary estimators or external reward models and leaves the efficient critic-free GRPO outer loop unchanged. To thoroughly benchmark this capability, we construct noisy *training* variants of textual **MATH** and multimodal **Geo3K**, simulating realistic corruption scenarios during supervision. All reported test and benchmark results are evaluated on clean held-out data with correct ground-truth targets. We evaluate DENOISING-GRPO after both low-noise and high-noise training, and test transfer on GSM8K, held-out same-corpus MATH500, Minerva-Math, MathVista, MathVerse, and MMK12. On clean in-domain tests after high-noise training, DENOISING-GRPO improves over GRPO by up to +10.5 on MATH and +3.0 on Geo3K, and also improves clean benchmark performance (e.g., +10.1 on GSM8K). Mechanism diagnostics further show that the confidence gate reduces harmful consensus overrides while recovering most of the gap to clean-supervision GRPO. These results indicate that reasoning models can self-denoise corrupted training supervision and still generalize to clean evaluation data whenever within-group agreement is sufficiently reliable.

1 INTRODUCTION

Reinforcement learning has become a central component of post-training for reasoning-oriented language models, especially in mathematics, where supervised fine-tuning alone often struggles to produce reliable long-horizon reasoning (Shao et al., 2024; Rafailov et al., 2023; Yu et al., 2025). Among recent formulations, Group Relative Policy Optimization (GRPO) is highly attractive because it removes the explicit critic and estimates the learning signal directly from within-group reward statistics (Shao et al., 2024; Guo et al., 2025). However, the group-relative normalization in GRPO still struggles to handle noisy supervision.

This vulnerability is a critical bottleneck in realistic mathematical pipelines, where supervision can easily be corrupted by extraction errors, mislabeled datasets, think-answer mismatches, or cross-modal perturbations (Qiu et al., 2025; Mansouri et al., 2025; Shen et al., 2025; Liu et al., 2025a). In standard RL, such inconsistencies might only degrade the affected samples. In GRPO, however, corrupted rewards distort the group mean and variance, thereby compromising the advantage estimation for the entire rollout group. This broadcast of noise slows convergence and can push the model toward degenerate reasoning solutions.

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.

Existing work has only partially addressed this issue. Optimization-oriented studies have sought to improve clipping behavior, scaling properties, or engineering stability in GRPO-style training (Liu et al.; Yu et al., 2025; Lin et al., 2025; Yang et al., 2025a; Liu et al., 2026). Conversely, robustness-oriented studies often resort to heavy external interventions, focusing on auxiliary noise estimators, reweighting networks, or alternative learning objectives (Qiu et al., 2025; Mansouri et al., 2025; Kim, 2026; Zhu et al., 2025; Zhang & Stratos, 2026). A simpler and more scalable question remains underexplored: can GRPO be protected against degenerate solutions by directly correcting the reward target itself, while strictly preserving the efficient, critic-free outer loop?

We answer this question with DENOISING-GRPO, a **Confidence-Gated Consensus Reward Correction** framework. The key idea is to intervene before noisy rewards are reused by group-relative normalization. While converting majority agreement into a learning signal is heavily explored in recent test-time RL frameworks (Zuo et al., 2025; Wang et al., 2025b), we uniquely adapt this concept to construct a structured correction target for *training-time* denoising. Furthermore, to decide when this target should be trusted, we introduce a confidence gate. Although concurrent works actively leverage intrinsic confidence to stabilize GRPO updates or gate external reward models (Wang et al., 2025a; Zhu et al., 2026), our specific architectural formulation uses confidence strictly to gate the internal consensus. This yields two tightly coupled components: **Consensus Reward Correction**, which denoises the target through majority predicted-answer agreement, and **Confidence-Gated Activation**, which prevents unreliable early consensus from overriding the raw reward.

A key empirical finding is that the confidence source itself is a meaningful design choice. Full-response confidence often becomes informative earlier because it aggregates over the whole response, whereas answer-level confidence is more localized to the final decision and can become more decisive once answer extraction stabilizes. We therefore study **gated full-confidence consensus** and **gated answer-confidence consensus** as two principled instantiations of the same reward-side framework.

DENOISING-GRPO preserves the simplicity of GRPO while substantially changing its behavior under noisy supervision. To rigorously test this, we inject controlled text-only and cross-modal corruption only into the *training* splits of textual **MATH** and multimodal **Geo3K**; the in-domain test splits and all external benchmarks remain clean. The gains are largest after training on the hardest noise regimes, and they transfer across both language-only and multimodal mathematical reasoning. The method is not intended to solve every possible consensus failure: if a rollout group becomes confidently but incorrectly aligned, any purely internal consensus mechanism can still be misled. Our design therefore treats confidence gating as a conservative reliability filter rather than as an oracle. Our main contributions are:

- We propose **Consensus Reward Correction**, a group-level denoising mechanism that converts majority predicted-answer agreement into a structured reward target for noisy GRPO training.
- We propose **Confidence-Gated Activation**, which enables consensus correction only when the rollout group is sufficiently reliable, reducing harmful override in uncertain groups.
- We add direct mechanism evidence and a probabilistic reliability analysis showing when consensus correction is expected to be safer than raw noisy labels, and when it should be blocked.
- We show across text-only and multimodal reasoning that this reward-side design delivers large and consistent robustness gains, especially in high-noise regimes, and that full-response and answer-level confidence act as complementary gate signals.

2 RELATED WORK

2.1 REASONING MODELS AND GRPO

Large-scale reasoning models have progressed from text-only solvers to multimodal systems that combine symbolic inference with visual evidence (Wang et al., 2024b; Xu et al., 2025b; Jaech et al., 2024; Yang et al., 2025b; Bai et al., 2025; Wang et al., 2024a). This progress relies on post-training: PPO/TRPO established stable policy improvement (Schulman et al., 2015; 2017), DPO avoids online rollouts (Rafailov et al., 2023), and GRPO makes critic-free group-normalized RL practical for mathematical reasoning (Shao et al., 2024; Guo et al., 2025). Subsequent GRPO-style work refines clipping, scaling, and engineering stability (Liu et al.; Yu et al., 2025; Lin et al., 2025; Yang et al., 2025a); our focus is instead the reliability of reward statistics consumed by the same optimizer.

2.2 Robustness, Consensus, and Reward Reliability

Noisy supervision is now recognized as a major obstacle for reasoning RL. Prior work studies Bayesian denoising, noise-corrected objectives, mean-normalization fragility, think-answer mismatch, value regularization, and multimodal rollout corruption (Qiu et al., 2025; Mansouri et al., 2025; Kim, 2026; Shen et al., 2025; Zhu et al., 2025; Zhang & Stratos, 2026; Liu et al., 2025a; Xu et al., 2025a; Liu et al., 2025b; Sun et al., 2025; Yan et al., 2025). Majority voting and test-time RL also exploit agreement among generations (Zuo et al., 2025; Wang et al., 2025b). In contrast, DENOISING-GRPO uses consensus for *training-time* reward construction, where wrong agreement can affect future policy updates; confidence gating is therefore essential.

3 Methodology

3.1 Preliminaries: GRPO under Noisy Supervision

For a prompt x_g, GRPO samples a rollout group $\{y_{g,1}, \ldots, y_{g,K}\}$ from the rollout policy π_{old}. We define the usual policy ratio $\rho_{g,i}(\theta) = \pi_\theta(y_{g,i} \mid x_g)/\pi_{\text{old}}(y_{g,i} \mid x_g)$. In its clipped form, GRPO optimizes a PPO-style surrogate,

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_g \left[\frac{1}{K} \sum_{i=1}^{K} \min\Big(\rho_{g,i}(\theta) A_{g,i}, \text{clip}(\rho_{g,i}(\theta), 1-\epsilon_{clip}, 1+\epsilon_{clip}) A_{g,i} \Big) \right] - \beta \mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \tag{1}$$

where $A_{g,i}$ is an advantage estimate and π_{ref} is a reference policy. Standard GRPO uses within-group reward normalization to define this advantage. Let $r_{g,i}$ denote the observed scalar reward for sample i in group g. One computes

$$z_{g,i} = \frac{r_{g,i} - \mu_g}{\sigma_g + \epsilon_{norm}}, \quad \mu_g = \frac{1}{K} \sum_{i=1}^{K} r_{g,i}, \quad \sigma_g = \sqrt{\frac{1}{K} \sum_{i=1}^{K} (r_{g,i} - \mu_g)^2}, \tag{2}$$

and sets $A_{g,i} = z_{g,i}$. This estimator is appealing because it is simple and critic-free, but under noisy supervision both μ_g and σ_g can become unreliable. A corrupted reward therefore perturbs not only the update of one sample, but the normalization shared by the entire group.

3.2 Problem Formulation

To formalize this issue, we distinguish between an unobserved latent reward $r^*_{g,i} \in [0,1]$, which represents the clean reasoning quality we would ideally optimize, and an observed reward $r_{g,i} \in [0,1]$, which may be corrupted by answer extraction errors, mislabeled supervision, think-answer mismatch, or multimodal corruption. The latent reinforcement learning objective is

$$\max_\theta \ \mathbb{E}_{x_g \sim \mathcal{D},\, y_{g,1:K} \sim \pi_\theta(\cdot|x_g)} \left[\frac{1}{K} \sum_{i=1}^{K} r^*_{g,i} \right]. \tag{3}$$

In practice, however, optimization proceeds using $r_{g,i}$ rather than $r^*_{g,i}$. This induces a central *reward-target error*: the observed reward may be a poor proxy for the latent objective. Because GRPO reuses the same noisy reward inside group-relative normalization, corruption then propagates to the policy update of the entire group. Our method is therefore designed to improve the reward signal before it is reused by the standard GRPO estimator.

3.3 Overview of Denoising-GRPO

Under noisy supervision, the main bottleneck in GRPO is often the reward target rather than the optimizer. By leveraging the natural agreement within a rollout group, we can construct a more robust training signal to mitigate this issue. Thus, our method modifies GRPO exclusively on the reward side through two tightly coupled components:

- **Consensus Reward Correction.** We replace the raw observed reward with a consensus-based signal that more faithfully tracks the latent true reward.

- **Confidence-Gated Activation.** We apply this correction only when the rollout group exhibits sufficient reliability under a chosen confidence metric.

3.4 Consensus Reward Correction

For each sample we assume access to two quantities produced by the training pipeline: the raw reward $r_{g,i} \in [0, 1]$ and the extracted predicted answer label $a_{g,i}$. Among valid predicted answer labels in the group, we compute the consensus answer

$$\hat{a}_g = \mathrm{mode}\{a_{g,1}, \ldots, a_{g,K}\}. \tag{4}$$

The corresponding group-level correction signal is

$$p_{g,i} = \begin{cases} \mathbb{I}[a_{g,i} = \hat{a}_g], & \hat{a}_g \text{ is not null,} \\ r_{g,i}, & \text{otherwise.} \end{cases} \tag{5}$$

Here, mode means the unique most frequent extracted answer among valid boxed answers. It need not be a strict $> K/2$ majority, but $\hat{a}_g$ evaluates to null if no valid boxed answers can be extracted from the rollout group or if the top frequency is tied; in either case the method defaults back to the raw reward. Consensus reward correction is stronger than scalar posterior smoothing because it uses *answer identity* rather than only group-average correctness. It therefore injects a structured correction signal that can separate agreement from disagreement inside the same rollout group.

To implement this interpolation stably, we employ a linear time-based warmup schedule

$$\lambda_t = \max(\lambda_{\min}, 1 - t/T_w), \tag{6}$$

where t is the current training step and T_w is the reward warmup horizon. This schedule acts as an *implicit curriculum*: early updates remain anchored to the raw reward, while later updates increasingly trust stabilized group consensus. This avoids prematurely amplifying "confidently wrong" internal agreement while still denoising supervision once the policy becomes more reliable. The final corrected reward is

$$\tilde{r}_{g,i} = \lambda_t r_{g,i} + (1 - \lambda_t) p_{g,i}. \tag{7}$$

Eq. equation 7 makes the method conservative by construction: early training remains anchored to the observed reward, while later training places increasing trust in reliable group-level structure.

Error decomposition. Relative to the latent reward, the corrected reward admits the decomposition

$$\tilde{r}_{g,i} - r^*_{g,i} = \lambda_t (r_{g,i} - r^*_{g,i}) + (1 - \lambda_t)(p_{g,i} - r^*_{g,i}), \tag{8}$$

which immediately yields

$$|\tilde{r}_{g,i} - r^*_{g,i}| \leq \lambda_t |r_{g,i} - r^*_{g,i}| + (1 - \lambda_t)|p_{g,i} - r^*_{g,i}|. \tag{9}$$

This bound clarifies the role of the method: whenever the consensus-based estimator is more reliable than the raw reward on trustworthy groups, the corrected reward inherits that gain while remaining anchored to the original supervision through λ_t.

3.5 Confidence-Gated Activation

Consensus correction is useful only if the group agreement itself is trustworthy. We therefore introduce a second component that controls *when* consensus is allowed to override the raw reward.

For each sample we assume access to a confidence score $c_{g,i} \in [0, 1]$. In practice, the training pipeline supports two main choices:

$$c^{full}_{g,i} \in [0, 1], \qquad c^{ans}_{g,i} \in [0, 1], \tag{10}$$

where $c^{full}_{g,i}$ denotes full-response confidence and $c^{ans}_{g,i}$ denotes answer-level confidence. We aggregate either source to the group level:

$$\bar{c}_g = \frac{1}{K} \sum_{i=1}^{K} c_{g,i}. \tag{11}$$

The gate then decides whether consensus override is activated:

$$m_g = \mathbb{I}[\bar{c}_g \geq \gamma], \tag{12}$$

where γ is the confidence gate threshold. The gated correction signal is

$$p_{g,i}^{gate} = \begin{cases} r_{g,i}, & m_g = 0 \text{ or } \hat{a}_g \text{ does not exist,} \\ \mathbb{I}[a_{g,i} = \hat{a}_g], & m_g = 1 \text{ and } \hat{a}_g \text{ exists.} \end{cases} \tag{13}$$

Hence, low-confidence groups fall back to the raw observed reward, while high-confidence groups use answer consensus as a structured correction signal. This design can be viewed as a selective prior over group agreement: consensus is trusted only when the rollout group has sufficient reliability.

The final *gated* corrected reward used for optimization replaces $p_{g,i}$ with $p_{g,i}^{gate}$:

$$\tilde{r}_{g,i} = \lambda_t r_{g,i} + (1 - \lambda_t) p_{g,i}^{gate}. \tag{14}$$

Confidence definition. The bounded confidence scores are derived from length-normalized generation probabilities. Full-response confidence uses all response tokens, $c_{g,i}^{full} = \exp\left(\frac{1}{|y_{g,i}|}\sum_{t=1}^{|y_{g,i}|} \log P(y_{g,i}^t \mid x_g, y_{g,i}^{<t})\right)$, while answer-level confidence applies the same computation only to extracted final-answer tokens. If extraction fails, $c_{g,i}^{ans} = 0$ and the sample is invalid for consensus construction; the group can still fall back to the raw reward through Eq. equation 13.

Why compare full-response and answer-level confidence? Full-response confidence is smoother because it averages over more tokens; answer-level confidence is closer to the final decision but more sensitive to short spans and extraction instability. We therefore treat them as complementary gate signals.

3.6 Optimization Objective of the Proposed Method

The methodology above modifies the reward statistic while leaving the GRPO optimizer itself unchanged. Concretely, once the gated corrected reward $\tilde{r}_{g,i}$ has been constructed, the method reuses the standard group-relative normalization

$$\tilde{z}_{g,i} = \frac{\tilde{r}_{g,i} - \tilde{\mu}_g}{\tilde{\sigma}_g + \epsilon_{norm}}, \qquad \tilde{\mu}_g = \frac{1}{K}\sum_{i=1}^{K} \tilde{r}_{g,i}, \qquad \tilde{\sigma}_g^2 = \frac{1}{K}\sum_{i=1}^{K} (\tilde{r}_{g,i} - \tilde{\mu}_g)^2. \tag{15}$$

The resulting policy objective is therefore the standard GRPO clipped surrogate with corrected rewards:

$$\mathcal{L}_{\text{DENOISING-GRPO}}(\theta) = \mathbb{E}_g\left[\frac{1}{K}\sum_{i=1}^{K} \min\Big(\rho_{g,i}(\theta)\tilde{z}_{g,i}, \text{clip}(\rho_{g,i}(\theta), 1-\epsilon_{clip}, 1+\epsilon_{clip})\tilde{z}_{g,i}\Big)\right] - \beta \mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \tag{16}$$

where $\rho_{g,i}(\theta) = \pi_\theta(y_{g,i} \mid x_g)/\pi_{\theta_{\text{old}}}(y_{g,i} \mid x_g)$. The optimizer itself is therefore unchanged: the contribution of DENOISING-GRPO is a reward-construction layer that replaces the statistics entering the standard GRPO objective while leaving the outer training loop intact.

3.7 Theoretical Properties

Proposition 1 (Bounded reward correction)**.** *If $r_{g,i} \in [0,1]$, $p_{g,i}^{gate} \in [0,1]$, and $\lambda_{\min} \in [0,1]$, then the corrected reward $\tilde{r}_{g,i}$ in Eq. equation 14 also lies in $[0,1]$ for all g, i, t.*

Failure probability of gated consensus. The previous propositions are design guarantees; they do not by themselves quantify when a majority should be trusted. We therefore use a simple reliability model to make the role of K, confidence gating, and consensus margin explicit. Let a_g^* be the latent correct answer for group g, and define

$$p_{\text{eff}}(\gamma) = \Pr(a_{g,i} = a_g^* \mid m_g = 1,\ a_{g,i} \text{ is valid}),$$

the effective per-rollout correctness after conditioning on the gate opening. Under conditional independence inside a reliable group, the probability that a strict majority of K rollouts is correct is

$$P_{\text{maj}}(p_{\text{eff}}, K) = \sum_{j=\lceil (K+1)/2 \rceil}^{K} \binom{K}{j} p_{\text{eff}}^j (1 - p_{\text{eff}})^{K-j}. \tag{17}$$

If $q_\gamma = \Pr(m_g = 1)$ is the gate-open rate, then the probability that the method applies an incorrect strict-majority correction is bounded by

$$\Pr(\text{incorrect correction}) \leq q_\gamma \left(1 - P_{\text{maj}}(p_{\text{eff}}(\gamma), K)\right). \tag{18}$$

This binomial model captures the intended safety mechanism. Increasing γ usually reduces q_γ while increasing $p_{\text{eff}}(\gamma)$; increasing K sharpens majority reliability when $p_{\text{eff}} > 0.5$ but amplifies majority failure when $p_{\text{eff}} < 0.5$. Because our implementation accepts a unique mode rather than requiring a strict majority, Eq. equation 17 should be read as a conservative analysis of the high-margin subset. The empirical margin diagnostics in Appendix F confirm this picture: unanimous groups have low harmful-override rates, whereas bare $5/8$ majorities are unreliable and should be filtered by the gate.

Proposition 2 (Gate as a safeguard)**.** *If $\bar{c}_g < \gamma$, then the reward correction module falls back to the raw reward and does not allow group consensus to override low-confidence supervision.*

Proposition 3 (No-worse interpolation under better consensus)**.** [1] *Suppose that on reliable groups the gated consensus-based estimator is at least as accurate as the raw reward, i.e.,*

$$\mathbb{E}[|p_{g,i}^{gate} - r_{g,i}^*| \mid m_g = 1] \leq \mathbb{E}[|r_{g,i} - r_{g,i}^*| \mid m_g = 1].$$

Then the corrected reward satisfies

$$\mathbb{E}[|\tilde{r}_{g,i} - r_{g,i}^*| \mid m_g = 1] \leq \mathbb{E}[|r_{g,i} - r_{g,i}^*| \mid m_g = 1].$$

4 Experimental Setup and Main Results

This section evaluates the central claim of the paper: reward-side correction alone can make GRPO substantially more robust under noisy supervision. We first establish the main baseline result, then test transfer, isolate ablations, and analyze training dynamics.

4.1 Datasets

Training datasets. We train on two noisy supervision corpora, one textual and one multimodal:

- **MATH.** We construct *low-noise* and *high-noise* variants from a standardized MATH-style training corpus (Hendrycks et al.). Low noise introduces distractors, reordering, and mild ambiguity; high noise introduces severe omission, contradictory modifications, and answer corruption.
- **Geo3K.** We construct low- and high-noise training variants from a multimodal geometry corpus (Lu et al., 2021). Text corruption follows the MATH split, while the multimodal setting adds image degradation at low noise and semantic mismatch at high noise.

Together, they test whether reward correction remains effective when the *training supervision* contains text-only or multimodal corruption. Full construction details are deferred to Appendix C and Table 4.

Evaluation protocol. We report both *clean in-domain test performance after noisy training* and clean benchmark evaluation. In-domain evaluation uses clean MATH and Geo3K test splits. Importantly, the noise is strictly injected into the *training data*; every reported test or benchmark split remains clean, fully solvable, and paired with the original correct ground-truth target. This ensures that performance measures true reasoning capability after learning from noisy supervision, rather than accuracy on corrupted test items. The benchmark suite contains GSM8K, MATH500, Minerva-Math, MathVista, MathVerse, and MMK12. MATH500 is reported as a held-out same-corpus benchmark rather than as a fully out-of-distribution transfer set, while GSM8K, Minerva-Math, and the multimodal benchmarks provide broader cross-dataset evidence. The main paper emphasizes high-noise training because it is the clearest stress test, while low-noise training is reported for completeness. See Appendices D.5 and E for benchmark details.

[1] **Remark:** Detailed derivations are deferred to Appendix A.5. These propositions formalize the intended inductive bias of the method. Reward correction is conservative in uncertain groups, improves whenever reliable consensus is more accurate than raw supervision, and remains bounded by construction. The theoretical emphasis is therefore on *when* group consensus should be trusted and *how* it should be blended with the original reward, rather than on redesigning the GRPO optimizer itself.

Table 1: Broad comparison against RL baselines. Low/high denote the training-noise level; all reported MATH and Geo3K accuracies are measured on clean in-domain test splits. Subscripts denote improvements over the GRPO baseline; Appendix D reports multi-seed stability for the high-noise in-domain settings.

Family	Method	MATH-L	MATH-H	Geo3K-L	Geo3K-H	Avg.
Classical / GRPO-family	PPO	45.3	21.8	–	–	–
	CPPO	47.1	35.1	33.4	27.3	35.7
	DCPO	55.2	40.7	33.5	32.8	40.6
	GRPO	54.4	43.4	33.8	32.0	40.9
	DRGRPO	55.3	46.7	35.5	34.1	42.9
Robust RL	VRPO	40.2	25.4	–	–	–
	NC-GRPO	54.2	24.7	33.7	31.7	36.1
	NOISYGRPO	54.8	45.7	34.4	30.4	41.3
	MC-GRPO	55.4	44.5	35.3	34.3	42.4
Gated Consensus + AnswerConf		$54.9_{(+0.5)}$	$52.6_{(+9.2)}$	$35.6_{(+1.8)}$	$\mathbf{35.0}_{(+3.0)}$	$44.5_{(+3.6)}$
Gated Consensus + FullConf		$\mathbf{56.0}_{(+1.6)}$	$\mathbf{53.9}_{(+10.5)}$	$\mathbf{36.2}_{(+2.4)}$	$34.6_{(+2.6)}$	$\mathbf{45.2}_{(+4.3)}$

4.2 PRIMARY BASE MODELS AND EXPERIMENTAL SETUP

The main text uses two compact backbones: Qwen3-0.6B for text-only mathematics and Qwen3-VL-2B-Instruct for multimodal mathematics. They allow controlled comparisons while still exposing clear differences among RL methods.

Extended model suite. The supplement extends these backbones with Qwen3-4B, Qwen2.5-Math-7B, Qwen3-VL-4B-Instruct, and Qwen3-VL-8B-Instruct using the same evaluation template, so it serves as a scaling extension rather than a separate experimental story; see Appendix D.

Training and evaluation protocol. Reported comparable runs use the same GRPO-style pipeline, reward interface, rollout group size $K=8$, optimization budget, prompt format, and decoding protocol. The supplement keeps this setup fixed within each comparable setting and only broadens the backbone set and reporting detail. Full hyperparameters and prompt/verifier details are provided in Appendix D.

Comparison methods. The comparison set covers the strongest relevant alternatives for noisy reasoning RL without becoming a catalogue of minor variants.

- **Classical / GRPO-family baselines:** PPO, GRPO, CPPO, DCPO, and DRGRPO.
- **Robustness-oriented baselines:** VRPO, NC-GRPO, NOISYGRPO, and MC-GRPO.

Among the robust RL baselines, VRPO reintroduces a heavy auxiliary Critic for value regularization, while NOISYGRPO and NC-GRPO rely on Bayesian noise estimators to correct the reward signal. Within each comparable reported setting, methods follow the same training budget and evaluation protocol. The comparison isolates one question: under noisy supervision, is robustness better improved by optimizer-side GRPO variants, external-module baselines, or lightweight reward-side confidence-gated consensus correction?

4.3 MAIN RESULTS AGAINST PUBLISHED RL BASELINES

We begin with a comparison against representative published RL baselines. Table 1 reports clean-test endpoint performance under four noisy-training conditions, testing whether improving reward construction is enough to produce a stronger policy than modifying the optimization loop.

Main observations. Both gated variants improve upon the GRPO baseline across all training-noise conditions, on both text-only (MATH) and math-oriented multimodal (Geo3K) reasoning. The gains are largest after severe training corruption (*high-noise* training), reaching +10.5 on the clean MATH test split after MATH-high training and +3.0 on the clean Geo3K test split after Geo3K-high training. The gap over existing optimizer-side variants and robust-RL algorithms is equally clear in the comparable settings: under MATH-high training our best variant (53.9) surpasses the strongest baselines, DRGRPO (46.7) and NOISYGRPO (45.7), and similar trends hold under Geo3K-high training. Correcting the reward target therefore buys more robustness than reworking the optimization loop.

The critic-free design also comes at a favorable cost. Robust baselines such as VRPO and NOISY-GRPO rely on heavy external value models or noise estimators, whereas DENOISING-GRPO obtains its robustness purely through lightweight self-correction on internal confidence and group consensus, keeping the computational profile of vanilla GRPO essentially unchanged.

4.4 DIRECT MECHANISM DIAGNOSTICS

The main results improve endpoint accuracy for the intended reason: the gate converts reliable consensus into cleaner rewards while blocking many harmful majorities. Appendix F provides the full diagnostic tables. To summarize their content, GATED-CONSENSUS+FULLCONF improves clean MATH accuracy by +16.6 under answer-only corruption and by +4.7 even when the training prompt itself is degraded, and it recovers 10.5 of the 13.0 points lost by noisy-training GRPO relative to clean-supervision GRPO—closing roughly 81% of the gap without oracle labels. On the safety side, confidence gating reduces early harmful overrides from 46% under ungated consensus to 24%, directly supporting the design choice in Eq. equation 13.

5 COMPARATIVE ANALYSIS

5.1 GENERALIZATION TO STANDARD BENCHMARKS

Appendix Table 9 shows that robustness gains from *high-noise* training carry over to clean benchmarks, avoiding overfitting to the noisy distribution. AnswerConf improves over GRPO across all six benchmarks, while FullConf yields especially large gains on textual tasks (+10.1 on GSM8K and +12.2 on the held-out same-corpus MATH500 benchmark). Multimodal transfer gains are modest but positive, including +5.3 on MMK12, indicating that reward correction improves downstream generalization alongside in-domain robustness within math-oriented evaluation.

5.2 ABLATION STUDY ON CONSENSUS RELABELING

Table 2 ablates our method by separating ungated consensus relabeling from the final confidence-gated variants. (see Appendix B for extended comparisons).

Ablation study. Table 2 isolates the contribution of each component under high-noise training while keeping evaluation clean. Ungated consensus relabeling on its own already improves over GRPO (40.6 vs. 37.7 on average), confirming that the consensus target carries useful information. The confidence gate is where the largest jump occurs, however, lifting GATED-CONSENSUS+ANSWERCONF to 52.6 and GATED-CONSENSUS+FULLCONF to 53.9 on the clean MATH test split after MATH-high training—clear evidence that filtering unreliable majorities is the decisive step rather than relabeling alone. Removing the implicit curriculum ("w/o curri.") produces a mild but consistent decline relative to the scheduled variant, indicating that a gradual transition helps stabilize consensus injection. The three components thus play complementary roles: consensus supplies the target, the gate decides when to trust it, and the curriculum paces the handover.

Table 2: Ablation study of the method design under high-noise training, evaluated on clean in-domain test splits.

Method	MATH	Geo3K	Avg.
GRPO	43.4	32.0	37.7
CONSENSUS+ANSWERCONF (w/o curri.)	46.4	33.6	40.0
CONSENSUS+ANSWERCONF (w curri.)	47.3	33.8	40.6
GATED-CONSENSUS+ANSWERCONF (w curri.)	52.6	**35.0**	43.8
CONSENSUS+FULLCONF (w/o curri.)	46.7	33.3	40.0
CONSENSUS+FULLCONF (w curri.)	47.1	33.6	40.4
GATED-CONSENSUS+FULLCONF (w curri.)	**53.9**	34.6	**44.3**

5.3 TRAINING DYNAMICS OF GATED CONSENSUS CORRECTION

Figure 1 compares optimization trajectories across raw reward, ungated consensus, and gated consensus. The key pattern is that gated consensus yields smoother entropy reduction, stronger re-

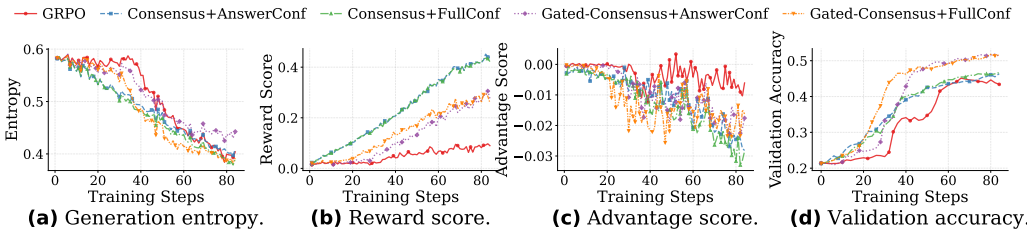


Figure 1: Training dynamics of reward designs. Panels trace generation entropy, reward, batch-averaged normalized advantage over the monitored subset, and clean validation accuracy, contrasting optimization stability across raw-reward and consensus variants.

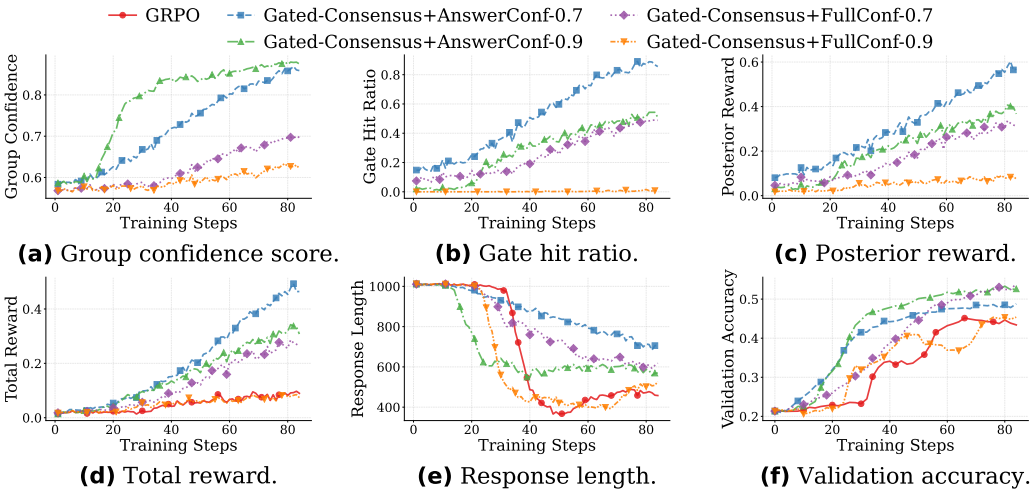


Figure 2: Dynamics of the confidence gate and reward correction. Panels track metrics for AnswerConf and FullConf ($\gamma = 0.7, 0.9$), illustrating how thresholds control consensus activation and performance.

ward/advantage signals, and a higher clean-validation plateau, while ungated consensus remains unstable because it occasionally over-trusts immature majorities.

5.4 Confidence Analysis

Table 23 in Appendix F and Figure 2 analyze how confidence source and threshold interact. AnswerConf naturally produces larger confidence scores because answer spans are shorter than full reasoning traces, so its best threshold is higher ($\gamma = 0.9$). FullConf is more conservative and works best at $\gamma = 0.7$. Both settings achieve a balanced gate-hit ratio of about 50%, which is high enough to repair noisy supervision but low enough to avoid premature posterior-reward injection.

6 Conclusion

We address the critical bottleneck of noisy supervision in reasoning-oriented RL, which limits the scalability of post-training on imperfect real-world data. We propose Denoising-GRPO, a critic-free reward extension for GRPO that transforms within-group answer agreement into a structured correction target, safeguarded by a generation confidence gate that prevents premature optimization collapse. Because it operates entirely on the reward side, this lightweight mechanism delivers substantial robustness gains over heavier external-model baselines across textual (MATH) and multimodal (Geo3K) domains. Confidence-gated consensus thus offers a practical design principle for letting reasoning models self-denoise corrupted supervision and generalize to clean evaluation without sacrificing efficiency.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *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, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Youngeun Kim. Mc-grpo: Median-centered group relative policy optimization for small-rollout reinforcement learning. *arXiv preprint arXiv:2601.22582*, 2026.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. *arXiv preprint arXiv:2504.13055*, 2025a.

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. In *2nd AI for Math Workshop@ ICML 2025*.

Zikang Liu, Tongtian Yue, Yepeng Tang, Longteng Guo, Junxian Cai, Qingbin Liu, Xi Chen, and Jing Liu. Prefix grouper: Efficient grpo training through shared-prefix forward. *arXiv preprint arXiv:2506.05433*, 2025b.

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

Omar El Mansouri, Mohamed El Amine Seddik, and Salem Lahlou. Noise-corrected grpo: From noisy rewards to unbiased gradients. *arXiv preprint arXiv:2510.18924*, 2025.

Longtian Qiu, Shan Ning, Jiaxuan Sun, and Xuming He. Noisygrpo: Incentivizing multimodal cot reasoning via noise injection and bayesian estimation. *arXiv preprint arXiv:2510.21122*, 2025.

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.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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.

Si Shen, Peijun Shen, Wenhua Zhao, and Danhao Zhu. Mitigating think-answer mismatch in llm reasoning through noise-aware advantage reweighting. *arXiv preprint arXiv:2508.05928*, 2025.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Jinpeng Wang, Chao Li, Ting Ye, Mengyuan Zhang, Wei Liu, and Jian Luan. Icpo: Intrinsic confidence-driven group relative preference optimization for efficient reinforcement learning. *arXiv preprint arXiv:2511.21005*, 2025a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024a.

Weiqin Wang, Yile Wang, Kehao Chen, and Hui Huang. Beyond majority voting: Towards fine-grained and more reliable reward signal for test-time reinforcement learning. *arXiv preprint arXiv:2512.15146*, 2025b.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024b.

Donglai Xu, Hongzheng Yang, Yuzhi Zhao, Pingping Zhang, Jinpeng Chen, Wenao Ma, Zhijian Hou, Mengyang Wu, Xiaolei Li, Senkang Hu, et al. From exploration to exploitation: A two-stage entropy rlvr approach for noise-tolerant mllm training. *arXiv preprint arXiv:2511.07738*, 2025a.

Fengli Xu, Qianyue Hao, Chenyang Shao, Zefang Zong, Yu Li, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, et al. Toward large reasoning models: A survey of reinforced reasoning with large language models. *Patterns*, 6(10):101370, 2025b.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*, 2025a.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2376–2385, 2025b.

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.

Wenzheng Zhang and Karl Stratos. Rence: Learning to reason by noise contrastive estimation. *arXiv preprint arXiv:2601.22432*, 2026.

Dingwei Zhu, Shihan Dou, Zhiheng Xi, Senjie Jin, Guoqiang Zhang, Jiazheng Zhang, Junjie Ye, Mingxu Chai, Enyu Zhou, Ming Zhang, et al. Vrpo: Rethinking value modeling for robust rl training under noisy supervision. *arXiv preprint arXiv:2508.03058*, 2025.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Kun Xu, and Yang You. Camel: Confidence-gated reflection for reward modeling. *arXiv preprint arXiv:2602.20670*, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A Detailed Method Discussion and Proofs

A.1 Notation and Mapping to the Implementation

The implementation used in this paper instantiates the reward module as `consensus_pred` with warmup interpolation and a configurable confidence gate. In practice, the gate can be driven by either full-response confidence or answer-level confidence. The supplementary material provides a clean mathematical description so that readers can reproduce the method without relying on code-specific naming conventions.

A.2 Reward Module Design Space

The compared reward-side methods span the key design axes of noisy GRPO training:

- **Raw reward:** no denoising and no relabeling.
- **Mean:** uniform scalar smoothing by the group-average reward.
- **Bayes:** posterior mixing from group successes.
- **ConfSoft + full confidence / answer confidence:** confidence-aware scalar soft labels without answer-conditioned relabeling.
- **ConfBayes + full confidence / answer confidence:** confidence-weighted Bayesian posteriors without answer-conditioned relabeling.
- **Consensus + full confidence / answer confidence:** answer-conditioned consensus relabeling without gate.
- **Gated consensus + full confidence / answer confidence:** answer-conditioned consensus relabeling that only activates above a confidence threshold.

This organization isolates three questions: whether scalar smoothing is enough, whether confidence should only modulate a scalar posterior, and whether answer-conditioned consensus should be applied always or only after a reliability gate.

A.3 Unified Formulation of Alternative Optimization Methods

To make the empirical comparisons in Section 5 mathematically explicit, we summarize the method families instantiated, or directly realizable, in the `qwen3_0b6_math_high` training recipe. All of them keep the same GRPO-style outer loop and differ primarily in how they construct the reward statistic. This is the design space represented by Table 2.

Raw reward and group-relative normalization. The simplest configuration uses the original reward without any case-level correction:

$$\tilde{r}_{g,i}^{\text{RAW-GRPO}} = r_{g,i}, \qquad z_{g,i}^{\text{RAW-GRPO}} = \frac{r_{g,i} - \mu_g}{\sigma_g + \epsilon_{norm}}, \tag{19}$$

with $\mu_g = \frac{1}{K}\sum_{i=1}^{K} r_{g,i}$ and $\sigma_g^2 = \frac{1}{K}\sum_{i=1}^{K}(r_{g,i} - \mu_g)^2$. This configuration is attractive because it is minimal and critic-free, but it inherits any corruption in the reward channel directly.

Bayesian posterior reward. The Bayesian variant replaces the raw group statistic with a posterior mean over group successes:

$$\pi_g^{\text{bayes}} = \frac{\alpha_0 + s_g}{\alpha_0 + \beta_0 + K}, \qquad \tilde{r}_{g,i}^{\text{BAYES}} = \lambda_t r_{g,i} + (1 - \lambda_t)\pi_g^{\text{bayes}}, \tag{20}$$

where $s_g = \sum_i \mathbb{I}[r_{g,i} \geq \tau_s]$ and $\tau_s = 1$ for the binary exact-match verifier used in our experiments. This scalar posterior baseline smooths the reward target, but it does not use answer identity and therefore cannot separate informative agreement from confident but systematically wrong majority behavior. We include it as a deliberately restricted reward-side alternative: it can change the reward scale entering normalization, but it lacks the within-group sample discrimination provided by answer-conditioned consensus.

Mean reward smoothing. The mean baseline replaces the per-sample reward target by the group-average reward:

$$\tilde{r}_{g,i}^{\text{MEAN}} = \lambda_t r_{g,i} + (1-\lambda_t)\mu_g, \qquad \mu_g = \frac{1}{K}\sum_{j=1}^{K} r_{g,j}. \tag{21}$$

This is the simplest denoising strategy in the design space. It reduces variance by smoothing within the group, but because it ignores both answer identity and confidence structure, it is also the least expressive alternative in Table 2.

Confidence-soft-label reward. The confidence-soft-label family uses group confidence itself as a soft posterior:

$$\lambda_{g,t}^{soft} = \max\Big(\lambda_{\min},\, \lambda_t(1-\bar{c}_g)^{\eta}\Big), \qquad \tilde{r}_{g,i}^{soft} = \lambda_{g,t}^{soft} r_{g,i} + (1-\lambda_{g,t}^{soft})\bar{c}_g, \tag{22}$$

where $\bar{c}_g$ is the average group confidence and η is the confidence exponent. In the experiments, this family is instantiated with either $\bar{c}_g^{full}$ or $\bar{c}_g^{ans}$, corresponding to CONFSOFT+FULLCONF and CONFSOFT+ANSWERCONF. The design is conservative and confidence-aware, but because it never commits to an answer-conditioned correction, it can remain too diffuse once a strong group structure has formed.

Confidence-weighted Bayesian reward. The confidence-Bayes family weights group evidence by confidence:

$$\pi_g^{conf\text{-}bayes} = \frac{\alpha_0 + \sum_i c_{g,i} s_{g,i}}{\alpha_0 + \beta_0 + \sum_i c_{g,i}}, \qquad \tilde{r}_{g,i}^{conf\text{-}bayes} = \lambda_t r_{g,i} + (1-\lambda_t)\pi_g^{conf\text{-}bayes}, \tag{23}$$

where $s_{g,i}$ is the binary success indicator and $c_{g,i}$ is the selected confidence source. In the experiments, this family is instantiated with either full-response confidence or answer-level confidence, corresponding to CONFBAYES+FULLCONF and CONFBAYES+ANSWERCONF. Compared with the plain Bayes variant, this design uses richer evidence, but it still collapses the whole group to a scalar posterior rather than to an answer-conditioned target.

Consensus relabeling. For consensus-based reward correction, let $\hat{a}_g = \text{mode}\{a_{g,1}, \ldots, a_{g,K}\}$ denote the unique most frequent valid predicted answer in group g, returning null on extraction failure or ties. The ungated consensus posterior is

$$p_{g,i}^{cons} = \begin{cases} \mathbb{I}[a_{g,i} = \hat{a}_g], & \hat{a}_g \text{ is not null}, \\ r_{g,i}, & \hat{a}_g \text{ is null}, \end{cases} \qquad \tilde{r}_{g,i}^{cons} = \lambda_t r_{g,i} + (1-\lambda_t) p_{g,i}^{cons}. \tag{24}$$

In the experiments, this family is instantiated with either full-response confidence or answer-level confidence to form CONSENSUS+FULLCONF and CONSENSUS+ANSWERCONF. It is stronger than scalar posterior smoothing because it leverages answer identity directly, but without a gate it can still over-trust an unstable early-training majority.

Confidence source inside consensus relabeling. The same consensus logic can be driven by two confidence summaries:

$$\bar{c}_g^{full} = \frac{1}{K}\sum_i c_{g,i}^{full}, \qquad \bar{c}_g^{ans} = \frac{1}{K}\sum_i c_{g,i}^{ans}. \tag{25}$$

The former aggregates full-response confidence, whereas the latter aggregates answer-level confidence. This distinction is central in our setting because full-response confidence is more easily confounded by verbosity, stylistic fluency, and multimodal uncertainty outside the answer span.

Gated consensus relabeling. The gated variant activates consensus override only when the group confidence is sufficiently reliable:

$$m_g = \mathbb{I}[\bar{c}_g \geq \gamma], \qquad p_{g,i}^{gate} = \begin{cases} r_{g,i}, & m_g = 0 \text{ or } \hat{a}_g \text{ is null}, \\ \mathbb{I}[a_{g,i} = \hat{a}_g], & m_g = 1 \text{ and } \hat{a}_g \text{ is not null}. \end{cases} \tag{26}$$

Instantiating $\bar{c}_g$ as $\bar{c}_g^{full}$ yields GATED-CONSENSUS+FULLCONF, while instantiating it as $\bar{c}_g^{ans}$ yields GATED-CONSENSUS+ANSWERCONF. This is the exact point at which the two confidence sources diverge in practice: the usefulness of the gate depends on whether the selected confidence statistic is aligned with answer correctness on the current dataset and modality.

Algorithm 1 Confidence-Gated Consensus Reward Correction within GRPO

Require: Policy π_θ, current training step t, group size K, gate threshold γ, warmup horizon T_w, minimum reward weight $\lambda_{\min}$, confidence source $s \in \{\text{full}, \text{answer}\}$
Ensure: Updated policy $\pi_{\theta'}$ after one GRPO step with corrected rewards

```
Sample rollout groups 𝒢 using current policy π_θ
for each group g ∈ 𝒢 do
    for each sample i ∈ {1, ..., K} do
        Decode response y_{g,i}, compute raw verifier reward r_{g,i}
        Extract predicted answer a_{g,i} from y_{g,i}
        Compute confidence c_{g,i}^{(s)} using source s
    end for
    Compute consensus â_g = mode{a_{g,1}, ..., a_{g,K}} (returns null on tie or extraction failure)
    Compute group confidence c̄_g = (1/K) Σ_{i=1}^{K} c_{g,i}^{(s)}
    Gate decision m_g = 𝕀[c̄_g ≥ γ]
    for each sample i ∈ {1, ..., K} do
        if m_g = 1 and â_g is not null then
            p_{g,i}^{gate} ← 𝕀[a_{g,i} = â_g]
        else
            p_{g,i}^{gate} ← r_{g,i}
        end if
        λ_t ← max(λ_min, 1 − t/T_w)
        Corrected reward r̃_{g,i} ← λ_t r_{g,i} + (1 − λ_t) p_{g,i}^{gate}
    end for
end for
Use {r̃_{g,i}} for group-relative normalization and apply the standard GRPO policy update
```

Why the proposed method is different. The alternatives above differ mostly in *how* they denoise or soften the reward target. By contrast, DENOISING-GRPO is built around two explicit reward-side questions: how to convert group agreement into a structured correction signal, and how to decide when that signal should be trusted:

$$\tilde{r}_{g,i} = \lambda_t r_{g,i} + (1 - \lambda_t) p_{g,i}^{gate}. \tag{27}$$

This formulation matters because noisy-reward RL fails first at the target-construction stage. The proposed method is therefore built around the underlying reward-side failure mode rather than a heavier optimizer-level intervention.

A.4 Algorithm Figure Walkthrough

Algorithm 1 provides a process-level view of the proposed reward-construction pipeline. It is placed before the proofs so that the symbolic statements in the next subsection can be read against a concrete training procedure.

Compared with vanilla GRPO, only the reward-construction block changes; rollout generation, group-relative normalization, and policy optimization remain unchanged. Algorithm 1 is therefore a drop-in replacement for GRPO-style reasoning RL pipelines.

A.5 Derivations

A.5.1 Derivation of Proposition 1

By definition, $\lambda_t = \max(\lambda_{\min}, 1 - t/T_w)$. Since $\lambda_{\min} \in [0, 1]$ and $1 - t/T_w \leq 1$ for all $t \geq 0$, it follows that $\lambda_t \in [\lambda_{\min}, 1] \subseteq [0, 1]$. Thus, Eq. equation 14 is a convex combination of $r_{g,i}$ and $p_{g,i}^{gate}$. If both terms lie in $[0, 1]$, then $\tilde{r}_{g,i} \in [0, 1]$ follows immediately from convexity.

A.5.2 Derivation of Proposition 2

When $\bar{c}_g < \gamma$, we have $m_g = 0$ by definition. From Eq. equation 13, this forces the gate to fall back, giving $p^{gate}_{g,i} = r_{g,i}$ for all i in group g. Substituting this into the final gated corrected reward Eq. equation 14 yields:

$$\tilde{r}_{g,i} = \lambda_t r_{g,i} + (1 - \lambda_t) r_{g,i} = r_{g,i}.$$

Hence low-confidence consensus never overrides the raw reward.

A.5.3 Derivation of Proposition 3

Consider the case where the gate is active, i.e., $m_g = 1$ and $\hat{a}_g$ is not null. From Eq. equation 13, we have $p^{gate}_{g,i} = p_{g,i}$. The error decomposition gives:

$$\tilde{r}_{g,i} - r^*_{g,i} = \lambda_t (r_{g,i} - r^*_{g,i}) + (1 - \lambda_t)(p^{gate}_{g,i} - r^*_{g,i}).$$

Taking the absolute value and applying the triangle inequality yields the bound:

$$|\tilde{r}_{g,i} - r^*_{g,i}| \le \lambda_t |r_{g,i} - r^*_{g,i}| + (1 - \lambda_t)|p^{gate}_{g,i} - r^*_{g,i}|.$$

Taking the conditional expectation given $m_g = 1$ on both sides:

$$\mathbb{E}[|\tilde{r}_{g,i} - r^*_{g,i}| \mid m_g = 1] \le \lambda_t \mathbb{E}[|r_{g,i} - r^*_{g,i}| \mid m_g = 1] + (1 - \lambda_t)\mathbb{E}[|p^{gate}_{g,i} - r^*_{g,i}| \mid m_g = 1].$$

By the assumption of the proposition, $\mathbb{E}[|p^{gate}_{g,i} - r^*_{g,i}| \mid m_g = 1] \le \mathbb{E}[|r_{g,i} - r^*_{g,i}| \mid m_g = 1]$. Substituting this upper bound into the second term, we obtain:

$$\mathbb{E}[|\tilde{r}_{g,i} - r^*_{g,i}| \mid m_g = 1] \le \lambda_t \mathbb{E}[|r_{g,i} - r^*_{g,i}| \mid m_g = 1] + (1 - \lambda_t)\mathbb{E}[|r_{g,i} - r^*_{g,i}| \mid m_g = 1].$$

Since $\lambda_t + (1 - \lambda_t) = 1$, the right-hand side simplifies exactly to $\mathbb{E}[|r_{g,i} - r^*_{g,i}| \mid m_g = 1]$, completing the proof.

B Alternative Robustification Strategies

This section expands on the integrated comparison presented in the main text. Table 3 compares the proposed reward correction against alternative robustification strategies within the exact same GRPO recipe. Because all rows share the same outer loop, rollout budget, verifier, and implementation context, the table isolates where the robustness gains actually come from.

Table 3: Extended integrated comparison across reward-design variants in the same GRPO training recipe. The subscripted number reports the difference from the GRPO result.

Family	Method	MATH high	Geo3K high	Avg.
Reference	GRPO	43.4	32.0	37.7
Alternative robustification	Mean	$40.6_{(-2.8)}$	$31.9_{(-0.1)}$	$36.3_{(-1.4)}$
	Bayes	$43.6_{(+0.2)}$	$33.6_{(+1.6)}$	$38.6_{(+0.9)}$
	ConfSoft+FullConf	$43.2_{(-0.2)}$	$34.3_{(+2.3)}$	$38.8_{(+1.1)}$
	ConfSoft+AnswerConf	$45.6_{(+2.2)}$	$34.3_{(+2.3)}$	$40.0_{(+2.3)}$
	ConfBayes+AnswerConf	$44.0_{(+0.6)}$	$33.7_{(+1.7)}$	$38.9_{(+1.2)}$
	ConfBayes+FullConf	$45.9_{(+2.5)}$	$33.5_{(+1.5)}$	$39.7_{(+2.0)}$
Consensus relabeling	Gated-Consensus+AnswerConf	$52.6_{(+9.2)}$	**35.0**$_{(+3.0)}$	$43.8_{(+6.1)}$
	Gated-Consensus+FullConf	**53.9**$_{(+10.5)}$	$34.6_{(+2.6)}$	**44.3**$_{(+6.6)}$

Comparison with alternative robustification methods. The alternative robustification block shows that the proposed gated consensus design dominates alternative strategies inside the same GRPO framework. Scalar denoising and posterior mixing help only modestly: the best competing average score is 40.0 (ConfSoft+AnswerConf), compared with 43.8 for Gated-Consensus+AnswerConf and 44.3 for Gated-Consensus+FullConf. The same gap is clear on MATH-high and remains visible on Geo3K-high. The strongest results therefore do not come from scalar smoothing alone, but from combining answer-conditioned consensus correction with selective confidence-based activation.

C CONSTRUCTION OF NOISY DATASETS AND DATA DETAILS

This section documents the two noisy training datasets used in the main paper and the clean evaluation suite used for reporting. It adds construction details and benchmark descriptions while keeping the same protocol, evaluation splits, and method definitions as the main text. Noise is injected only into training examples; all in-domain test splits and external benchmarks remain clean.

Table 4: Dataset summary table for the full paper. Noise levels and noise sources refer only to training data; every test or benchmark split is clean.

Dataset	Role	Modality	Train noise	Train source
MATH	noisy train / clean test	text	low, high	prompt + answer
Geo3K	noisy train / clean test	multimodal	low, high	text + image
GSM8K	clean cross-dataset eval.	text	none	benchmark only
MATH500	clean held-out same-corpus eval.	text	none	benchmark only
Minerva-Math	clean cross-dataset eval.	text	none	benchmark only
MathVista	clean cross-dataset eval.	multimodal	none	benchmark only
MathVerse	clean cross-dataset eval.	multimodal	none	benchmark only
MMK12	clean cross-dataset eval.	multimodal	none	benchmark only

Noisy-training construction. The noisy corpora are generated only for training, while all validation, test, and transfer benchmarks remain clean. The low-noise variants introduce weak-to-moderate perturbations that preserve most of the original task structure, such as redundant distractors, paraphrastic rewrites, local omissions, and mild visual degradation for Geo3K. The high-noise variants introduce more destructive prompt-side corruption and answer-side target corruption; for Geo3K, high noise additionally includes stronger visual perturbations such as image degradation or mismatched visual context. We report low/high as training-noise levels throughout the paper and never as properties of the test split.

MATH low-noise. This part documents the weak-to-moderate textual corruption pipeline used for the MATH training set. The implementation in `tests/math_low_noise.py` assigns each sampled training instance to exactly one low-noise transformation family, so the resulting corpus contains heterogeneous but controlled corruption. The three textual directions are:

- **Redundant distractor insertion:** the rewritten problem interleaves decoy numbers, irrelevant sub-conditions, and fake-but-plausible equations with the original question so that useful and useless evidence become harder to separate.
- **Synonymous rewriting with reordered logic:** the same mathematical content is retained, but the presentation order is aggressively shuffled and phrased more indirectly, increasing reading and condition-tracking difficulty without explicitly changing the target answer.
- **Small but essential omission:** one solvability-critical fact is removed or blurred, producing a low-noise form of ambiguity in which the question remains natural but key evidence is missing.

Across all three cases, the output prompt retains the original math-style reasoning instruction and boxed-answer format; the noise is injected into the question text rather than into the evaluation interface. This separation is important because it makes low-level corruption primarily a problem of identifying relevant information, not a problem of parsing a new output format.

MATH high-noise. The high-level MATH noise block is generated by `tests/math_high_noise.py` and intensifies the same general corruption directions. Concretely, the pipeline uses three stronger perturbations:

- **Massive information deletion:** large portions of the givens are removed so that the problem is no longer normally solvable.
- **Critical number or relation modification:** key numeric values or mathematical relations are changed to create contradictions, impossible constraints, or misleading pseudo-solutions.
- **Answer corruption:** instead of rewriting the question, the pipeline replaces the original final answer with a plausible but incorrect boxed answer and updates the stored reward target accordingly.

It explicitly distinguishes *prompt-side corruption* from *answer-side corruption*, because the latter changes the reward target itself and is therefore qualitatively different from paraphrastic or structural perturbation.

Geo3K low-noise. The Geo3K low-noise block, implemented in `tests/geo3k_low_noise.py`, follows the same *text-side directions* as low-noise MATH: redundant distractors are inserted into the question, conditions are synonymously rewritten and substantially reordered, or a small but necessary clue is omitted. This shared design is intentional: under the same noise regime, the text transformation direction is kept aligned across datasets so that differences in performance can be attributed to modality rather than to unrelated corruption semantics. Geo3K then adds a modality-specific branch in which some samples receive *image-only low noise*: the image is blurred, mixed with additive noise, contrast and brightness are perturbed, and a JPEG-style compression artifact is injected. The prompt still preserves the multimodal reasoning instruction and boxed-answer format, so the corruption again targets content quality rather than interface mismatch.

Geo3K high-noise. The Geo3K high-level block is implemented in `tests/geo3k_high_noise.py` and mirrors the same textual high-noise directions as MATH: severe information deletion, crucial number or relation modification, and answer corruption via a newly synthesized incorrect boxed answer. In addition, Geo3K introduces a multimodal failure mode that has no text-only counterpart: **cross-sample image swapping**. For these samples, the question text and answer target are kept, but the associated image is replaced by an image drawn from another training example, yielding image-question mismatch. These corruptions are critical for demonstrating that the proposed confidence-gated consensus reward design remains relevant under cross-modal failure modes rather than only text-side noise.

Clean test benchmark suite. The full evaluation suite is clean and is organized into textual and multimodal benchmarks so that transfer can be interpreted separately by modality:

- **Textual:** GSM8K, MATH500, and Minerva-Math.
- **Multimodal:** MathVista, MathVerse, and MMK12.

Benchmark descriptions. The textual suite mixes complementary difficulty profiles: **GSM8K** targets multi-step grade-school arithmetic word problems, **MATH500** provides a clean held-out benchmark from the same broad MATH corpus family rather than a fully out-of-distribution dataset, and **Minerva-Math** emphasizes olympiad-style and competition-style reasoning. The multimodal suite is likewise diverse: **MathVista** contains visual mathematical reasoning with mixed answer types, **MathVerse** broadens the visual reasoning space to more varied multimodal math questions, and **MMK12** captures school-level multimodal mathematics with diagram-heavy question formats. These benchmarks are all kept clean and are used only for reporting, not for noisy training or hyperparameter selection.

D Extended Training Configurations and Benchmark Details

This section collects the reproducibility details compressed in the main paper: optimization hyperparameters, prompt and verifier settings, the expanded model suite, and benchmark-level reporting details.

D.1 Hyperparameter Template

Table 5 summarizes the key optimization settings used in the main-text textual runs. These values are taken from the shared low-noise Qwen3-0.6B training recipes and remain fixed across the vanilla GRPO, full-confidence gated-consensus, and answer-confidence gated-consensus variants unless the reward construction block itself is changed.

D.2 Prompt and Verifier Details

For reproducibility, this section records the exact prompt formats, extraction rules, and verifier settings used during training.

Table 5: Key hyperparameters used in the main-text textual training runs. Consensus-specific entries are active only when case-reward correction is enabled.

Hyperparameter	Value
Group size K / rollout samples n	8
Training batch size	512
PPO mini-batch size	256
PPO micro-batch size per GPU	16
Log-prob micro-batch size per GPU	32
Learning rate	1×10^{-6}
Optimizer	AdamW-style default in `verl`
KL coefficient	0.001
KL type	low_var_kl
Reward warmup horizon T_w	84 steps
Minimum reward weight $\lambda_{\min}$	0.2
Confidence gate γ	0.7 for FullConf; 0.9 for AnswerConf
Confidence source	full-response or answer-level log-prob
Max prompt length	1024
Max response length	1024
Truncation / prompt filtering	error / enabled
Tensor parallel size	2
vLLM GPU memory utilization	0.6
Gradient checkpointing	enabled
Actor / ref param offload	False / True
Number of GPUs	8
Training epochs	6

System prompt and output format. For both textual MATH and multimodal Geo3K training, we append a fixed formatting instruction to every prompt to induce Chain-of-Thought (CoT) reasoning followed by a structured answer. The exact suffix appended to the user prompt is:

```
Please reason step by step, and put your final answer within \boxed{}.
```

For multimodal Geo3K prompts, the text is prepended with the necessary vision-language tags (e.g., `<image>`) before the question text, ensuring the backbone model processes the visual context correctly.

Answer extraction rule. The reward verifier extracts the predicted label by searching for the `\boxed{}` pattern in the generated response. To handle nested braces (e.g., `\boxed{\frac{1}{2}}`), the extraction script maintains a depth counter starting from the opening brace and extracts the substring up to the matching closing brace. If multiple `\boxed{}` strings exist, the last one is taken. If no `\boxed{}` is found, the extraction fails and the raw reward is automatically 0.

Confidence definitions. As formally defined in Section 3.5, the bounded confidence scores are deterministic functions of the raw generation log-probabilities. The full-response confidence $c_{g,i}^{full}$ aggregates over all generated tokens, whereas the answer-level confidence $c_{g,i}^{ans}$ aggregates only over the tokens strictly contained within the extracted `\boxed{}` span. If answer extraction fails, the answer-level confidence is set to 0 and the sample is treated as invalid for consensus voting.

Truncation and filtering heuristics. In the main-text textual runs, prompts are truncated only through the fixed maximum prompt/response lengths (1024 tokens each), with `filter_overlong_prompts=True` and `truncation=error`. All compared runs within a given setting keep the same rollout engine (`vllm`), grouped generation size ($n = 8$), KL regularization setup, and training budget. The only intentional changes across our GRPO-family scripts are in the reward-construction block: vanilla GRPO disables case-reward correction, while the two consensus variants enable `case_reward_mode=consensus_pred`, set `case_reward_group_key=uid`, reuse the same warmup and minimum-label-weight schedule, and use the source-specific gate thresholds selected in Table 23. Unless explicitly marked as a

retuned diagnostic, published-baseline comparisons use this shared protocol rather than per-baseline hyperparameter searches.

D.3 Training Recipe Notes

The three main scripts used for the textual low-noise experiments—`rl_math_10.sh`, `rl_math_15.sh`, and `rl_math_17.sh`—share the same backbone, data pipeline, optimization budget, and rollout configuration. As a result, the headline differences are not caused by changes in batch size, generation count, KL regularization, or hardware allocation. The comparison isolates the reward module: the baseline uses naive reward aggregation, whereas the two consensus variants add structured relabeling with the same warmup horizon and minimum label weight, differing in confidence source and the corresponding calibrated gate threshold.

D.4 Model Suite Summary

The supplementary model suite extends the two primary main-text backbones with a fixed, explicitly defined set of additional text-only and multimodal models:

- **Main-text models:** Qwen3-0.6B and Qwen3-VL-2B-Instruct.
- **Supplementary text extensions:** Qwen3-4B and Qwen2.5-Math-7B.
- **Supplementary multimodal extensions:** Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct.

Table 6 summarizes the parameter scale, modality, and role of each model used in the paper. Its purpose is to make clear that the supplement extends the main-text study rather than introducing a different experimental story.

Table 6: Summary of the extended model suite. Low/high denote noisy training variants; all reported evaluations use clean splits.

Model	Modality	Role in paper	Train noise	Notes
Qwen3-0.6B	text	primary main-text model	MATH low/high	main text
Qwen3-4B	text	extended validation	MATH low/high	supplement
Qwen2.5-Math-7B	text	extended validation	MATH low/high	supplement
Qwen3-VL-2B-Instruct	multimodal	primary main-text model	Geo3K low/high	main text
Qwen3-VL-4B-Instruct	multimodal	extended validation	Geo3K low/high	supplement
Qwen3-VL-8B-Instruct	multimodal	extended validation	Geo3K low/high	supplement

Table 7: Zero-shot clean MATH accuracy of textual backbones before noisy RL training. The non-monotonic baseline explains why model scale alone does not determine the post-training ranking.

Backbone	Zero-shot MATH
Qwen3-0.6B	21.4
Qwen3-4B	18.7
Qwen2.5-Math-7B	34.1

Table 7 contextualizes the apparent scaling anomaly in Table 11. The Qwen3-4B checkpoint starts from a weaker clean MATH zero-shot point than Qwen3-0.6B in our pipeline, so its earlier underperformance should not be interpreted as a failure mode introduced by DENOISING-GRPO. After retuning the shared training recipe, DENOISING-GRPO-Full still improves Qwen3-4B by +1.7 after low-noise training and +5.8 after high-noise training.

D.5 Benchmark Reporting Details

For each benchmark, we report the evaluation split, metric definition, decoding setting, and whether it is used only for clean benchmark reporting or also for clean in-domain testing. This keeps the textual suite (GSM8K, MATH500, Minerva-Math) and the multimodal suite (MathVista, MathVerse,

MMK12) directly comparable. Training-time validation is performed on clean in-domain validation splits every two epochs and checkpoints are saved every 70 steps; final tables are reported on clean held-out test or benchmark splits. Thus, the validation curve in Figure 1 is a clean training monitor, and the benchmark tables are clean-reporting results under a fixed evaluation template rather than separately tuned evaluation setups.

Table 8: Evaluation protocol details extracted from the benchmark scoring scripts. All rows report clean pass@1 accuracy under the same answer-extraction and normalization rules used by the verifier.

Benchmark group	Split	Metric	Decoding	Answer equivalence
MATH / MATH500 / Minerva-Math	clean held-out test	pass@1 accuracy	shared single-sample decoding	last boxed answer; fallback to final-answer or tail-number extraction; normalized exact match
GSM8K	clean test	pass@1 accuracy	shared single-sample decoding	strict `#### number` extraction with comma / dollar removal
Geo3K / MathVista / MathVerse / MMK12	clean benchmark split	pass@1 accuracy	shared single-sample decoding	boxed / final-answer extraction with mathruler grading and label normalization

All comparison methods use the same answer parser and verifier within each benchmark family in Table 8. For MATH-style tasks, the evaluator first extracts the last `\boxed{}` answer, then falls back to explicit "final answer" patterns or the last numeric candidate in the tail. For GSM8K, the strict parser follows the `####` numeric-answer convention. For Geo3K-style multimodal tasks, the parser uses boxed or explicit final-answer extraction followed by `mathruler` grading.

E Additional Experimental Results

E.1 Clean Benchmark Transfer

Table 9: Clean benchmark generalization after high-noise training. MATH500 is a held-out same-corpus benchmark; the remaining columns test broader cross-dataset or multimodal transfer. Subscripts denote improvements over GRPO.

Method	Textual benchmarks			Multimodal benchmarks		
	GSM8K	MATH500	Minerva-Math	MathVista	MathVerse	MMK12
PPO	57.6	19.2	4.8	–	–	–
CPPO	58.2	21.8	5.1	43.4	25.0	29.4
DCPO	63.5	26.2	4.8	43.4	27.3	33.0
GRPO	61.8	24.8	7.0	43.3	26.4	29.5
DrGRPO	61.8	23.2	5.1	43.4	27.7	35.7
VRPO	62.1	24.4	5.1	–	–	–
NC-GRPO	63.9	26.8	7.0	43.3	27.1	30.8
NoisyGRPO	59.7	21.2	3.3	43.9	25.7	31.5
MC-GRPO	62.5	27.2	5.9	43.2	26.4	33.5
Gated Consensus + AnswerConf	$65.2_{(+3.4)}$	$29.2_{(+4.4)}$	$7.7_{(+0.7)}$	$43.4_{(+0.1)}$	$27.0_{(+0.6)}$	$34.2_{(+4.7)}$
Gated Consensus + FullConf	$71.9_{(+10.1)}$	$37.0_{(+12.2)}$	$9.6_{(+2.6)}$	$43.8_{(+0.5)}$	$27.6_{(+1.2)}$	$34.8_{(+5.3)}$

E.2 Hard-Benchmark Results

Harder long-horizon mathematical benchmarks are a natural stress test because their low base accuracy can make confident-but-wrong consensus more likely. Table 10 reports clean AIME-style benchmark accuracy after high-noise MATH training.

Table 10: Clean AIME-style benchmark results after high-noise MATH training.

Method	Clean AIME accuracy
Zero-shot backbone	6.3
GRPO after high-noise training	6.9
Gated-Consensus+AnswerConf after high-noise training	9.6
Gated-Consensus+FullConf after high-noise training	9.4

Both gated variants improve over the high-noise GRPO baseline on this harder clean benchmark, with AnswerConf slightly ahead of FullConf. The absolute scores remain low, so we use this result as supporting evidence that gated consensus can help under higher difficulty, not as the main claim of the paper.

E.3 Textual Language Models across Noise Levels

This subsection focuses on clean in-domain MATH test performance after noisy MATH training. Table 11 compares vanilla GRPO and the two gated-consensus variants across textual backbones and across low/high training-noise settings, without mixing in out-of-domain benchmark columns.

Table 11: Extended clean in-domain textual test results after low/high-noise training. Rows are grouped by model for readability.

Model	Noise	Method	Clean test acc.	Δ vs GRPO
Qwen3-0.6B	low	GRPO	54.4	–
	low	DENOISING-GRPO-Answer	54.9	+0.5
	low	DENOISING-GRPO-Full	56.0	+1.6
	high	GRPO	43.4	–
	high	DENOISING-GRPO-Answer	52.6	+9.2
	high	DENOISING-GRPO-Full	53.9	+10.5
Qwen3-4B	low	GRPO (retuned)	45.7	–
	low	DENOISING-GRPO-Full (retuned)	47.4	+1.7
	high	GRPO (retuned)	39.7	–
	high	DENOISING-GRPO-Full (retuned)	45.5	+5.8
Qwen2.5-Math-7B	low	GRPO	66.3	–
	low	DENOISING-GRPO-Answer	66.0	-0.3
	low	DENOISING-GRPO-Full	66.0	-0.3
	high	GRPO	60.7	–
	high	DENOISING-GRPO-Answer	61.3	+0.6
	high	DENOISING-GRPO-Full	64.5	+3.8

Interpretation. This table is the textual in-domain complement to the main-paper results, with the backbone axis made explicit. The low-vs-high split reveals a clear noise-sensitivity pattern: gains over GRPO are moderate in low-noise training but become substantially larger in high-noise training, especially for DENOISING-GRPO-Full. The Qwen3-4B block reports the retuned diagnostic runs used to check the earlier scaling anomaly; after retuning, the 4B baseline is much stronger and DENOISING-GRPO-Full still provides clear gains, especially under high noise. The Qwen2.5-Math-7B low-noise setting is the main exception: both gated variants are slightly below GRPO (-0.3), suggesting that when noise is weak and the base model is already strong, consensus correction can have little or even slightly negative marginal value. Overall, the pattern supports the claim that the proposed reward correction primarily improves robustness to corrupted supervision rather than only fitting one specific architecture.

E.4 Multimodal Language Models across Noise Levels

Table 12 provides the clean in-domain multimodal counterpart after noisy Geo3K training. The comparison keeps only low/high training-noise results across backbones so that the section focuses on robustness under noisy supervision rather than benchmark transfer.

Interpretation. The multimodal clean in-domain results show the same mechanism-oriented pattern as the textual table. On the primary backbone (Qwen3-VL-2B-Instruct), both variants consistently outperform GRPO after low- and high-noise training, with larger margins after high-noise training. On larger backbones, improvements remain visible where measurements are available, especially for DENOISING-GRPO-Full in the high-noise setting. Overall, these results support that the proposed reward correction is not tied to one modality: it remains effective when training noise affects both textual reasoning and visual grounding.

Table 12: Extended clean in-domain multimodal test results after low/high-noise training. Rows are grouped by model for readability.

Model	Noise	Method	Clean test acc.	Δ vs GRPO
Qwen3-VL-2B-Instruct	low	GRPO	33.8	–
	low	DENOISING-GRPO-Answer	35.6	+1.8
	low	DENOISING-GRPO-Full	36.2	+2.4
	high	GRPO	32.0	–
	high	DENOISING-GRPO-Answer	35.0	+3.0
	high	DENOISING-GRPO-Full	34.6	+2.6
Qwen3-VL-4B-Instruct	low	GRPO	40.9	–
	low	DENOISING-GRPO-Answer	41.5	+0.6
	low	DENOISING-GRPO-Full	42.6	+1.7
	high	GRPO	36.8	–
	high	DENOISING-GRPO-Answer	38.6	+1.8
	high	DENOISING-GRPO-Full	38.3	+1.5
Qwen3-VL-8B-Instruct	low	GRPO	45.4	–
	low	DENOISING-GRPO-Answer	45.6	+0.2
	low	DENOISING-GRPO-Full	45.9	+0.5
	high	GRPO	41.9	–
	high	DENOISING-GRPO-Answer	42.7	+0.8
	high	DENOISING-GRPO-Full	43.6	+1.7

Table 13: Geo3K image-dependence diagnostic. The largest gain appears when the correct image is available, while text-only and shuffled-image settings remain lower.

Setting	GRPO	GATED-CONSENSUS+FULLCONF	Δ
Full multimodal input	32.0	34.6	+2.6
Text-only input	24.1	25.3	+1.2
Shuffled image input	22.8	23.4	+0.6

Table 13 checks whether the multimodal gain is a text-only shortcut. Removing or shuffling the image sharply lowers absolute accuracy and reduces the gain, while the full multimodal setting gives the strongest result. This indicates that the method benefits from image-grounded evidence rather than only exploiting textual artifacts in Geo3K.

E.5 SENSITIVITY, CALIBRATION, AND COST DIAGNOSTICS

This subsection collects additional diagnostics that are useful for reproducibility and reviewer interpretation but are too detailed for the main text. Unless otherwise stated, the numbers use the MATH-high setting with GATED-CONSENSUS+FULLCONF.

Table 14: Rollout group-size sensitivity on MATH-high. $K = 8$ is the main-paper setting and provides a strong accuracy–cost trade-off.

Group size K	MATH	Relative rollout cost
4	49.6	$0.5\times$
8	53.9	$1.0\times$
16	55.1	$2.0\times$

Table 14 shows that increasing K improves consensus reliability but with diminishing returns: moving from $K = 8$ to $K = 16$ adds only +1.2 points while doubling rollout cost. We therefore use $K = 8$ as the default budget-aligned setting rather than as a tuned upper bound.

Value of joint γ–T_w sensitivity. The gate threshold γ and warmup horizon T_w control two different failure modes of consensus correction. The threshold γ determines *which* rollout groups are trusted, so it directly trades off useful correction against harmful majority overrides. The warmup horizon T_w determines *when and how quickly* the posterior consensus reward replaces the raw verifier reward, so it controls optimization stability rather than group reliability itself. Studying them jointly is therefore more informative than varying either hyperparameter alone: a low threshold with a short warmup may inject unreliable consensus too early, while an overly high threshold or overly long warmup may suppress useful correction and reduce the method to nearly vanilla GRPO. The main configuration, $\gamma = 0.7, T_w = 84$, should be interpreted as a balanced setting that makes the gate active early enough to repair noisy supervision while still delaying strong posterior injection until the policy has accumulated usable agreement.

Table 15: Joint sensitivity of gate threshold γ and warmup horizon T_w on clean MATH test accuracy after MATH-high training.

Gate threshold γ	$T_w = 42$	$T_w = 84$
0.6	49.8	46.4
0.7	52.8	53.9
0.8	50.7	47.6

Table 16: Longer-training diagnostic on MATH-high. The gap between GRPO and GATED-CONSENSUS+FULLCONF persists under a $3\times$ longer training budget.

Training steps	GRPO	GATED-CONSENSUS+FULLCONF	Δ
84	43.4	53.9	+10.5
252	45.1	55.2	+10.1

Table 16 addresses whether the improvement is a transient warmup artifact. Extending training improves both methods slightly, but the performance gap remains essentially unchanged, indicating that confidence-gated correction continues to matter after the initial curriculum phase.

Table 17: Multi-seed clean in-domain high-noise results. Each row reports the mean and standard deviation across three independent runs.

Dataset	Method	Mean	Std.
MATH-H	GRPO	43.4	0.1
MATH-H	GATED-CONSENSUS+ANSWERCONF	52.6	0.5
MATH-H	GATED-CONSENSUS+FULLCONF	53.9	0.3
Geo3K-H	GRPO	32.0	0.2
Geo3K-H	GATED-CONSENSUS+ANSWERCONF	35.0	0.6
Geo3K-H	GATED-CONSENSUS+FULLCONF	34.4	0.2

Table 17 makes the high-noise comparison symmetric across the two clean in-domain test sets and shows that the gains are stable across runs. The textual setting favors FullConf, while the multimodal setting slightly favors AnswerConf on average, consistent with the idea that answer-span confidence is often more stable when visual grounding adds uncertainty to full-response likelihoods. The main comparison table reports the representative endpoint runs used for the broad baseline comparison, while this table summarizes independent-run variability.

Table 18 helps explain why gains can shrink on stronger and cleaner settings: better-calibrated and already capable models leave less room for reward-side denoising, especially under low noise. This is consistent with the small negative result for Qwen2.5-Math-7B under low-noise training in Table 11.

Table 20 clarifies the cost attribution. The expensive K rollouts are already required by the GRPO baseline for group-relative normalization. DENOISING-GRPO adds only lightweight post-generation processing before the standard normalization and policy update.

Table 18: Confidence calibration diagnostic. Lower ECE indicates better alignment between confidence and correctness.

Backbone	Expected calibration error
Qwen3-0.6B	0.14
Qwen2.5-Math-7B	0.06

Table 19: Calibration-bin diagnostic on clean MATH validation examples. The bin-level values are aligned with the aggregate ECE trend in Table 18.

Backbone	Confidence bin	Avg. confidence	Empirical accuracy	Count
Qwen3-0.6B	$[0.0, 0.2)$	0.12	0.18	90
Qwen3-0.6B	$[0.2, 0.4)$	0.31	0.39	180
Qwen3-0.6B	$[0.4, 0.6)$	0.51	0.63	250
Qwen3-0.6B	$[0.6, 0.8)$	0.70	0.54	290
Qwen3-0.6B	$[0.8, 1.0]$	0.88	0.66	190
Qwen2.5-Math-7B	$[0.0, 0.2)$	0.13	0.15	70
Qwen2.5-Math-7B	$[0.2, 0.4)$	0.32	0.35	120
Qwen2.5-Math-7B	$[0.4, 0.6)$	0.53	0.57	210
Qwen2.5-Math-7B	$[0.6, 0.8)$	0.72	0.78	300
Qwen2.5-Math-7B	$[0.8, 1.0]$	0.90	0.82	300

F Mechanism Analyses

This appendix section focuses only on the positive mechanism behind the proposed method: how confidence scale, gate activation, reward correction, and downstream optimization interact to produce the final robustness gains. The three subsections are organized as a causal chain: Figure 3 studies confidence formation, Figure 4 studies how confidence is converted into reward correction, and Figure 5 studies how the corrected reward alters optimization and final validation performance.

F.1 Main-Text Mechanism Diagnostics

Table 21 shows that the gain is not merely undoing synthetic answer flips. The largest improvement appears on answer-only training corruption (+16.6), where consensus directly repairs the corrupted target, but prompt-destructive training corruption also improves clean-test accuracy (+4.7). Table 22 places the result between practical and oracle controls: noisy-training GRPO loses 13.0 points relative to clean-supervision GRPO (56.4–43.4), while Gated-Consensus+FullConf recovers 10.5 points, or roughly 81% of that gap.

F.2 Gate Reliability Diagnostics

The central safety question is not whether the gate opens often, but whether it opens on groups where consensus is likely to be useful. Tables 24 and 25 report reliability diagnostics by corruption type and training phase.

These diagnostics should be interpreted together with the main-text harmful-override summary. Table 26 shows that confidence gating reduces early harmful overrides from 46% under ungated consensus to 24%, and further reduces the rate to 11% in the middle phase and 6% late in training. Table 25 explains why: early in training, consensus forms less often and the gate opens conservatively; as the policy stabilizes, both consensus quality and gate-hit rate increase, allowing more posterior correction without relying on all early majorities.

Table 20: Per-step runtime decomposition. Denoising-GRPO shares the same grouped rollout generation as vanilla GRPO; the added cost is the reward-construction block.

Component	Time per step	Relative change
Vanilla GRPO step	8400 ms	–
Added extraction + confidence + rewriting	260 ms	+3.1%

Table 21: Corruption-type decomposition after MATH-high training. Each row trains on the corresponding corrupted training variant and evaluates on the clean MATH test split. The method helps most on answer-side training corruption, but improves prompt-destructive corruption where the training input itself is degraded.

Corruption subset	GRPO	GATED-CONSENSUS+FULLCONF	Δ
Answer-only corruption	34.2	50.8	+16.6
Prompt-destructive corruption	48.5	53.2	+4.7
Mixed high-noise corruption	43.4	53.9	+10.5

F.3 GATE RELIABILITY BY CONSENSUS MARGIN

The probability model in Eq. equation 18 predicts that high-margin consensus should be substantially safer than weak plurality agreement. Table 27 verifies this empirically by grouping corrections according to their vote margin.

The sharp increase from 4% to 51% explains why the method combines answer consensus with a confidence gate rather than applying all plurality corrections. In practice, the gate suppresses many low-margin or low-confidence groups, reducing early harmful overrides relative to ungated consensus.

F.4 ADVANTAGE ALIGNMENT

Because GRPO optimizes normalized within-group advantages rather than raw scalar rewards, we also compare the corrected advantage $\tilde{z}$ against an oracle normalized advantage z^* constructed from clean rewards. Table 28 shows that reward correction improves the signal that actually drives the policy gradient.

This diagnostic addresses the gap between scalar reward correction and the actual GRPO update. The corrected reward not only reduces scalar target noise; after group normalization, it yields advantages that are closer to the oracle advantages in magnitude, correlation, and update sign.

F.5 CONFIDENCE-SCALE DYNAMICS

Interpretation. Figure 3 shows a few consistent regularities. The AnswerConf variant generally sits on a higher group-confidence trajectory than FullConf, simply because answer spans are much shorter than full reasoning traces, so the answer-side log-probability aggregation is more concentrated and larger in magnitude. Within the same confidence source, the better-performing configuration tends to end training with a higher and more stable confidence curve, so stronger optimization is reflected in the confidence signal and not only in final accuracy. The gate threshold in turn sets how quickly consensus becomes usable: with $\gamma = 0.7$ the confidence curve crosses the activation boundary earlier and consensus starts contributing sooner, whereas $\gamma = 0.9$ delays the same mechanism until the model has accumulated stronger evidence.

F.6 GATE-ACTIVATION AND REWARD-CORRECTION CHAIN

Interpretation. This figure provides the full logic chain from confidence to the actual training target. The gate is triggered by the group confidence through $m_g = \mathbb{I}[\bar{c}_g \geq \gamma]$, which determines whether the posterior reward is allowed to replace the raw verifier signal. The gated posterior is then combined with the raw reward through the warmup interpolation

$$\tilde{r}_{g,i} = \lambda_t r_{g,i} + (1 - \lambda_t) p_{g,i}^{gate}, \qquad \lambda_t = \max(\lambda_{\min}, 1 - t/T_w),$$

Table 22: Controls after MATH-high training, evaluated on the clean MATH test split. DENOISING-GRPO recovers most of the gap between noisy-training GRPO and clean-supervision GRPO without oracle labels.

Method / control	MATH
Pretrained, no RL	21.3
GRPO on noisy supervision	43.4
Confidence-only filtering	45.2
Oracle filtering	48.6
GATED-CONSENSUS+FULLCONF	53.9
Oracle correction	55.2
Clean-supervision GRPO	56.4

Table 23: Threshold robustness and sensitivity after high-noise text training, evaluated on the clean MATH validation/test protocol. Both variants outperform the GRPO baseline (43.4) across a wide range of thresholds.

Gate source (γ)	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
Full-response conf.	46.4	46.7	**53.9**	50.5	47.6	45.6	44.1	43.6
Answer-level conf.	46.3	46.7	46.7	45.8	47.6	47.4	**52.6**	47.3

which is exactly the reward-construction rule used in the full-confidence training recipe with $\gamma = 0.7$ in the main experiments. Reading the subplots in order therefore reveals the mechanism clearly: higher confidence yields a larger gate hit ratio; a smaller threshold activates the gate earlier and more often; a higher hit ratio lets the posterior reward affect training sooner; and the final reward moves away from the raw reward only to the extent permitted jointly by gate hits and the warmup weight λ_t. This is why the threshold does not merely change a binary switch: it changes the whole time profile of how quickly posterior reward enters the learning signal.

F.7 CONFIDENCE-TO-PERFORMANCE COUPLING

Interpretation. Figure 5 traces the next stage of the mechanism, from confidence through optimization quality to final performance. Higher and more stable confidence improves the posterior reward, and once that posterior is translated into a cleaner final reward, the advantage estimates also become stronger and less noisy—an effect that matters precisely because GRPO updates are driven by the normalized reward statistic, so an unstable reward target yields an unstable advantage. The entropy panels show the policy-level consequence: the better configurations reduce uncertainty in a smoother, more controlled way. The validation-accuracy curve then closes the causal chain, with confidence setting the gate hit ratio, the hit ratio governing how much posterior reward reaches the final reward, and that reward in turn shaping the advantage signal that drives improvement. The strongest runs are therefore those in which confidence is not merely high but converted into reward correction at the right speed and scale, which is also why lower-threshold settings tend to improve earlier: by raising the hit ratio sooner they let posterior and final reward separate from the raw-reward trajectory earlier, accelerating both the growth of the advantage signal and the rise in validation accuracy.

G OPTIONAL FURTHER MATERIAL

G.1 CASE STUDIES

This subsection summarizes qualitative evidence from rollout traces of the FullConf ($\gamma = 0.7$) and AnswerConf ($\gamma = 0.9$) runs. The schema itself follows the causal chain of the method: a problem induces group predictions, group predictions induce consensus and group confidence, confidence controls posterior reward injection, and the final reward determines whether the case becomes a useful learning signal.

Table 24: Gate reliability by training-corruption type on MATH-high. "Useful" denotes a correction that moves the reward target toward the clean label, while "harmful" denotes an override away from the clean label.

Training-corruption subset	Gate-open rate	Useful-correction rate	Harmful-override rate
Answer-only corruption	85%	96%	4%
Prompt-destructive corruption	20%	78%	22%
Mixed high-noise corruption	57%	91%	9%

Table 25: Gate and consensus evolution over MATH-high training. Correct/wrong consensus rates are measured over all groups; gated correct/wrong rates are measured over opened groups.

Step	Consensus	Correct cons.	Wrong cons.	Gate-hit	Correct gated	Wrong gated
0	32%	15%	17%	6%	70%	30%
20	58%	36%	22%	38%	82%	18%
40	78%	57%	21%	68%	89%	11%
60	90%	73%	17%	82%	93%	7%
80	96%	85%	11%	89%	96%	4%

Case analysis. The cases show that the method remains effective across several qualitatively different regimes. Most listed successes for both FullConf and AnswerConf start from raw reward 0.0 and reach posterior reward 1.0, turning the final reward positive; the raw reward is exactly 0.0 either because the noisy training target was synthetically corrupted (e.g., F8, A22) or because the noisy verifier rejected a correct semantic equivalent (e.g., A59). Because the table spans both early and late steps, the temporal progression is also visible: the repeated $\log_5 625$ case (F48→F69) and the repeated AnswerConf cases A51→A70 and A59→A80 show stable consensus with steadily stronger final reward as training matures. The group structures are deliberately varied as well—FullConf includes fully unanimous groups and a dominant but non-unanimous group in F77 ($7/8$), while AnswerConf includes full unanimity and a $6/8$ dominant majority in A66.

A19 is the instructive counter-case: the dominant prediction is mathematically incorrect due to deceptive problem features, but the AnswerConf group confidence (0.563) stays below its gate threshold ($\gamma = 0.9$), so the posterior correction is blocked and the final reward falls back to the raw reward. This zero-posterior outcome illustrates the intended safety property—when confidence is insufficient or agreement is fragile, the gate declines to force an unstable correction. Together the examples show that the method both exploits reliable agreement and conservatively rejects unreliable consensus.

G.2 Broader Societal Impacts

While our work focuses on the technical robustness of reasoning models under noisy supervision, improving the reliability of these models has broader societal implications. On the positive side, enhancing the mathematical and logical reasoning capabilities of language models could benefit educational applications, intelligent tutoring systems, and automated scientific discovery by reducing hallucination risks and providing more consistent reasoning traces.

However, we must also consider potential negative impacts. More capable logical models could be misused to generate deceptive yet highly structured and mathematically plausible arguments for disinformation or fraud. Furthermore, as reasoning models become more robust and automated, there are long-term considerations regarding the displacement of human labor in data analysis and educational roles. We encourage future research to explore robust alignment techniques alongside reasoning enhancements to ensure these capabilities are deployed responsibly.

G.3 Limitations and Future Work

While Denoising-GRPO provides a lightweight and effective solution for noisy reasoning RL, it remains intentionally simple, which creates clear boundaries. First, its effectiveness still depends

Table 26: Harmful-override rate with and without confidence gating over training.

Variant	Steps 1–20	Steps 21–50	Steps 51–80
Ungated consensus	46%	27%	15%
GATED-CONSENSUS+FULLCONF	24%	11%	6%

Table 27: Harmful-override rate by vote margin for $K = 8$. Bare majorities are much less reliable than unanimous or strong-majority groups.

Vote margin	Harmful override
$8/8$ unanimous	4%
6–$7/8$ strong majority	19%
$5/8$ bare majority	51%

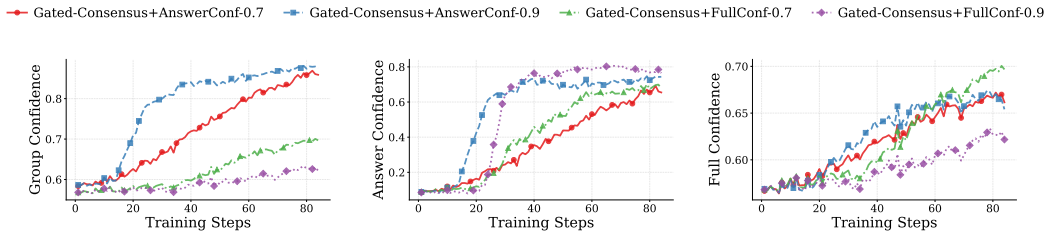
on answer extraction quality, verifier fidelity, and majority correctness. As discussed in the case studies, if a group consistently converges to a "confidently wrong consensus" due to deceptive prompt features, the gate will fail to block the incorrect target.

Second, the current linear time-based warmup schedule is intentionally simple. A variance-aware adaptive schedule improves MATH-high from 53.9 to 54.6 in a diagnostic run, suggesting that dynamic schedules can better balance raw and consensus rewards across training stages. We leave a systematic study of adaptive schedules to future work.

Third, our experiments show that there is no single universally optimal gate configuration: full-response confidence performs stronger on MATH-high, whereas answer-level confidence is slightly better on Geo3K-high. Finally, DENOISING-GRPO shares the same $K = 8$ rollout generation pattern as vanilla GRPO; the additional cost comes from answer extraction, confidence aggregation, and reward rewriting. In our implementation, this reward-construction block adds about 260 ms to an 8400 ms GRPO step, or roughly 3.1% overhead. These observations define the main boundary of the current design and point naturally toward future work on learned confidence modeling, adaptive scheduling, harder mathematical benchmarks, math-oriented multimodal datasets, and larger backbones.

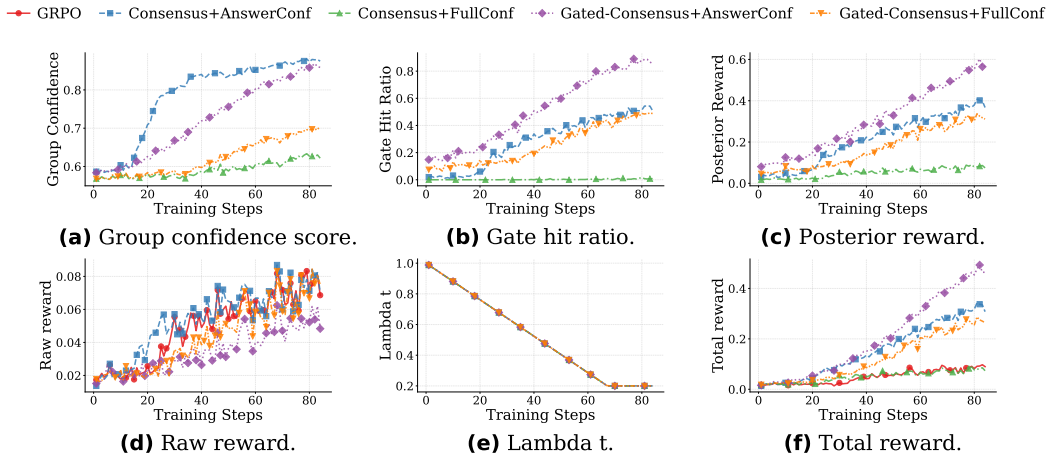
Table 28: Alignment between the training advantage and oracle clean-reward advantage on MATH-high. Lower MAE and higher correlation/sign agreement indicate better policy-gradient direction.

Advantage source	MAE to z^*	Pearson r	Sign agreement
Raw noisy z	0.42	0.61	0.74
Corrected $\tilde{z}$	0.19	0.87	0.91



(a) Group confidence score. **(b)** Answer confidence score. **(c)** Full confidence score.

Figure 3: Confidence-scale dynamics for the two confidence sources. The panels compare group confidence, answer confidence, and full-response confidence for the AnswerConf and FullConf variants under $\gamma = 0.7$ and $\gamma = 0.9$.



(a) Group confidence score. **(b)** Gate hit ratio. **(c)** Posterior reward.

(d) Raw reward. **(e)** Lambda t. **(f)** Total reward.

Figure 4: Gate-activation and reward-correction diagnostics. The panels track group confidence, gate hit ratio, posterior reward, raw reward, warmup weight λ_t, and final reward for the AnswerConf and FullConf variants under $\gamma = 0.7$ and $\gamma = 0.9$.

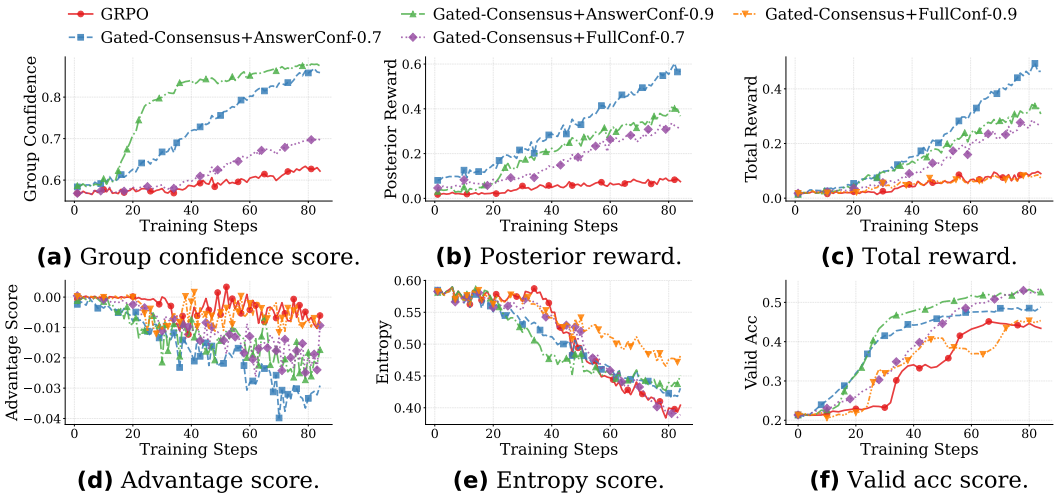


Figure 5: Confidence-to-performance coupling. The panels compare group confidence score, posterior reward, final reward, monitored-subset normalized advantage score, policy entropy, and validation accuracy for the AnswerConf and FullConf variants under $\gamma = 0.7$ and $\gamma = 0.9$.

Table 29: Representative case studies from rollout traces. Row prefixes F and A denote FullConf and AnswerConf respectively. Repeated questions across steps are intentionally retained to show how confidence and final reward evolve during training. The notation in the Predictions column (e.g., $\{289 \times 5\}$) denotes the predicted value and its frequency within the rollout group.

Step	Problem	Noisy target	Predictions	Consensus	Confidence	Raw reward	Posterior reward	Final reward
F8	Compute $17^9 \div 17^7$	17^{16}	$\{289 \times 5, 17^2 \times 3\}$	289	0.736	0.0	1.0	0.071
F14	Solve $\frac{1}{2} - \frac{1}{3} = \frac{3}{x}$	6	$\{18 \times 8\}$	18	0.758	0.0	1.0	0.167
F26	Solve $x - y = 6$, $x + y = 12$	3	$\{9 \times 8\}$	9	0.732	0.0	1.0	0.310
F38	Diagonals of a 23-gon	220	$\{230 \times 7, 220\}$	230	0.711	0.0	1.0	0.452
F48	Evaluate $\log_5 625$	5	$\{4 \times 8\}$	4	0.708	0.0	1.0	0.571
F59	Compute $17^9 \div 17^7$	17^{16}	$\{289 \times 8\}$	289	0.826	0.0	1.0	0.702
F64	Greatest prime factor of 221	13	$\{17 \times 8\}$	17	0.769	0.0	1.0	0.762
F69	Evaluate $\log_5 625$	5	$\{4 \times 8\}$	4	0.804	0.0	1.0	0.800
F77	Increase 50 by 120%	170	$\{110 \times 7, 56\}$	110	0.831	0.0	1.0	0.800
A19	Greatest integer of a floor expression	-1	$\{-2 \times 7, -1\}$	-2	0.563	0.0	0.0	0.000
A22	Simplify $16^{1/2} - 625^{1/2}$	21	$\{-21 \times 8\}$	-21	0.924	0.0	1.0	0.262
A26	Diagonals of a 23-gon	220	$\{230 \times 7, 220\}$	230	0.925	0.0	1.0	0.310
A33	Compute $\binom{5}{12}$	300	$\{0 \times 8\}$	0	0.986	0.0	1.0	0.393
A36	Compute $\binom{1293}{1}$	1292	$\{1293 \times 8\}$	1293	0.997	0.0	1.0	0.429
A41	y-intercept of $x - 2y = 5$	$\frac{5}{2}$	$\{-\frac{5}{2} \times 8\}$	$-\frac{5}{2}$	0.893	0.0	1.0	0.488
A51	Compute 9!/8!	72	$\{9 \times 8\}$	9	0.994	0.0	1.0	0.607
A59	Write $\frac{5}{8}$ as a decimal	0.652	$\{0.625 \times 8\}$	0.625	1.000	0.0	1.0	0.702
A66	$\gcd(39, 91)$	7	$\{13 \times 6, 1, 3\}$	13	0.999	0.0	1.0	0.786
A70	Compute 9!/8!	72	$\{9 \times 8\}$	9	0.936	0.0	1.0	0.800
A80	Write $\frac{5}{8}$ as a decimal	0.652	$\{0.625 \times 8\}$	0.625	1.000	0.0	1.0	0.800