CARM: CANCELLATION-AWARE RESPONSE MASKING FOR LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent years have witnessed the rapid adoption of reinforcement learning (RL) in large language model (LLM) post-training, with substantial gains in mathematical reasoning and code generation. In practical systems, however, policy updates and differences between rollout and training engines can make sampled responses off-policy. Sequence-level masking addresses this mismatch by deciding whether an entire response should contribute to optimization. A common masking rule uses the length-normalized geometric mean of sampled token probability ratios. Its signed log-ratios can cancel across positions, concealing substantial bidirectional policy drift. We propose *Cancellation-Aware Response Masking* (CARM), a sequence-level mask that takes the absolute value of each token log-ratio before averaging, preventing opposing probability changes from canceling. We prove that accepted responses satisfy a joint bound on the fraction of sampled-token ratios outside a prescribed band and their mean log-distance beyond its boundaries. Experiments on mathematical reasoning and code generation show that CARM improves mean@16 averaged over AIME 2024/2025/2026 and BeyondAIME by up to 3.13 percentage points over geometric-mean masking, and increases average pass@1 across four code benchmarks by 2.88 points over the strongest evaluated baseline. These findings support CARM as a theoretically grounded and effective method for response-level off-policy control in LLM reinforcement learning.

1 INTRODUCTION

Reinforcement learning (RL) is widely used to improve the reasoning ability of large language models (LLMs), building on advances in preference-based alignment and execution-feedback learning (Ouyang et al., 2022; Le et al., 2022). In an ideal on-policy setting, responses are generated by the same policy that is being optimized. In practical systems, however, several effects introduce policy mismatch. Mini-batch optimization makes stored responses stale as the policy changes, while actor–learner latency can delay their use. High-throughput rollout engines may also differ from the training backend in batching, numerical precision, and kernel implementations (Yao et al., 2025; Sheng et al., 2025). Consequently, the policy being updated can assign substantially different probabilities to sampled tokens than the rollout policy that generated them.

Proximal Policy Optimization (PPO) and Group Relative Policy Optimization (GRPO) regulate policy updates through clipping in their token-level surrogate losses (Schulman et al., 2017; Shao et al., 2024; Guo et al., 2025). Clipping changes how individual tokens contribute to the loss, but does not keep every current-to-rollout probability ratio within a prescribed interval. It also does not decide whether an entire response should participate in the update. Sequence-level masking complements clipping by making this decision from a score that summarizes mismatch across the response. Trust Region Masking (TRM), for example, motivates such filtering through an analysis of how policy mismatch affects optimization over long autoregressive sequences (Li et al., 2026).

A simple response-level criterion is geometric-mean masking: it takes the geometric mean of the token probability ratios and retains responses when this aggregate is close to one. We refer to this baseline as *GeoMean*. In log space, this aggregation corresponds to averaging signed token log-ratios, as used in DeepSeek-V3.2's off-policy sequence mask (Liu et al., 2025a). The problem is that positive and negative log-ratios can cancel. For example, token ratios of 10 and 0.1 have a geometric mean of one, although each sampled-token probability has changed tenfold. More generally, a

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

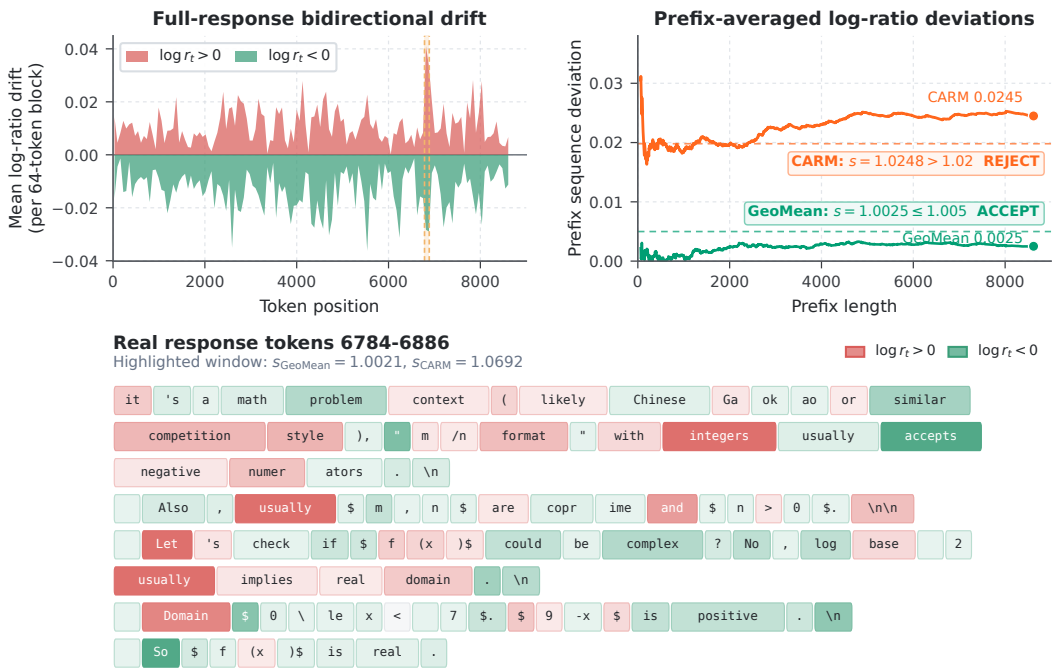


Figure 1: A real unfiltered rollout that GeoMean accepts and CARM rejects. This 8,619-token response has scores $s_{\text{GeoMean}} = 1.0025 \leq 1.005$ and $s_{\text{CARM}} = 1.0248 > 1.02$. *Top:* positive and negative log-ratio mass and prefix deviations. *Bottom:* a 103-token window; red tokens have $\log r_t > 0$, green tokens have $\log r_t < 0$, and color intensity encodes magnitude.

response can appear well aligned under this score because its probability changes balance out, rather than because those changes are small.

We propose *Cancellation-Aware Response Masking* (CARM), a sequence-level mask that takes the absolute value of each token log-ratio before averaging. This prevents opposing probability changes from canceling while treating reciprocal changes symmetrically. Figure 1 illustrates the distinction on a real 8,619-token response: substantial positive and negative log-ratio contributions largely cancel under GeoMean, whereas CARM retains their magnitudes. CARM changes which responses contribute to optimization while retaining the token-level PPO/GRPO surrogate.

Beyond preventing cancellation, CARM gives the response-level threshold a direct token-level interpretation. We prove that every accepted response satisfies a joint bound on the fraction of sampled-token ratios outside a prescribed, possibly asymmetric band and their mean log-distance beyond its boundaries. Experiments on mathematical reasoning and code generation show improvements over token- and sequence-level baselines. Threshold analyses further show that similar filtering rates can yield different accuracies, while filtering more responses is not consistently better.

Our contributions are:

- We formalize cancellation in signed-log sequence masking: the aggregate score can indicate perfect apparent alignment even when every sampled token ratio violates a prescribed band.
- We propose CARM, a sequence-level mask that prevents cancellation by taking absolute values before averaging token log-ratios. We derive a joint bound on the frequency and severity of ratio-band violations in accepted responses, including asymmetric bands, giving the filtering threshold a token-level interpretation.
- Experiments on mathematical reasoning and code generation show improvements over sequence- and token-level baselines. Paired-response diagnostics and threshold analyses

characterize the selection behavior: higher accuracy is observed at comparable average filtering rates, and accuracy is not monotone in the masked fraction.

2 Related Work

Policy optimization for LLM post-training. Trust Region Policy Optimization (TRPO) constrains Kullback–Leibler (KL) divergence, while PPO uses a clipped-ratio surrogate (Schulman et al., 2015; 2017). GRPO replaces a learned critic with within-group relative advantages (Shao et al., 2024). Answer correctness and program execution provide reward signals for improving mathematical reasoning and code generation (Le et al., 2022; Dou et al., 2024; Guo et al., 2025; Yang et al., 2025; Zeng et al., 2026), complementing inference-time reasoning and program-aided prompting (Wei et al., 2022; Chen et al., 2023; Gao et al., 2023). Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO) refines clipping and sampling (Yu et al., 2025); related work revisits advantage normalization, sequence-level objectives, and reward–policy learning (Hu et al., 2025; Xie et al., 2025; Lu et al., 2026a). These choices determine how sampled responses contribute to optimization. CARM addresses the complementary question of whether a sampled response should contribute to the current policy update.

Off-policy and token-level trust-region control. This selection problem arises even in nominally online training: actor–learner delay, repeated updates, and rollout–training engine differences separate the policy that generates responses from the policy being optimized (Espeholt et al., 2018; Yao et al., 2025; Sheng et al., 2025). Importance weighting and token-level filtering control individual update contributions (Espeholt et al., 2018; Shen et al., 2025). Adaptive clipping balances advantage contributions and mitigates entropy loss (Xi et al., 2026), while divergence-based controls constrain update size (Qi et al., 2026; Yao et al., 2026). These mechanisms regulate the update within a response. CARM instead decides whether to retain the complete response, separating response selection from the surrogate applied to retained tokens.

Sequence-level correction and masking. Response-level selection must summarize mismatch across an entire sampled path. Length-normalized geometric aggregation removes the direct length dependence of multiplying token ratios (Liu et al., 2025b); it also underlies sequence-level clipping in Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025a). DeepSeek-V3.2 explicitly uses off-policy sequence masking based on the mean signed log-ratio, equivalently the logarithm of a length-normalized geometric-mean ratio, to stabilize GRPO training (Liu et al., 2025a). Its rule applies a one-sided threshold to negative-advantage responses. Our GeoMean baseline uses the same signed-log aggregation with reciprocal-band acceptance (Eq. 5); we examine advantage conditioning separately in Section 5.2. TRM takes a worst-position view, rejecting a response when its maximum token-level divergence violates a long-horizon trust region (Li et al., 2026). CARM targets a different limitation: geometric aggregation averages signed log-ratios, so opposing shifts can cancel. Averaging their absolute values measures bidirectional drift rather than net displacement. The resulting mask changes which responses contribute to the update while retaining the token-level policy surrogate.

3 Method

3.1 Problem Setup

Let x be a prompt and let $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ be the i-th sampled response. We use *response* and *sequence* interchangeably for these generated tokens. We consider nonempty responses ($T_i \geq 1$) with finite token log-probabilities, so $0 < r_{i,t} < \infty$. The token-level ratio of the current policy to the rollout policy is

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{rollout}}(y_{i,t} \mid x, y_{i,<t})}. \tag{1}$$

For a sequence-level advantage $\hat{A}_i$, the conventional asymmetric PPO/GRPO token surrogate is

$$\ell_{i,t}(\theta) = \min\Big(r_{i,t}(\theta)\hat{A}_i, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{A}_i\Big). \tag{2}$$

For a binary response mask $M_i \in \{0, 1\}$, averaging the retained token losses gives

$$\mathcal{L}_{\text{masked}}(\theta) = -\frac{\sum_i M_i \sum_{t=1}^{T_i} \ell_{i,t}(\theta)}{\sum_i M_i T_i}. \tag{3}$$

The policy-loss contribution is defined as zero if no response is retained. CARM specifies the detached mask independently of loss normalization; Appendix B states the convention used in the experiments.

3.2 GeoMean Masking and Ratio Cancellation

Let $G(y_i) = \exp(\frac{1}{T_i} \sum_t \log r_{i,t})$ denote the geometric-mean token ratio. We define its log-space deviation and reciprocal-symmetric score as

$$d_{\text{GeoMean}}(y_i) = \left| \frac{1}{T_i} \sum_{t=1}^{T_i} \log r_{i,t} \right|, \qquad s_{\text{GeoMean}}(y_i) = \exp(d_{\text{GeoMean}}(y_i)). \tag{4}$$

The response is retained when

$$M_i^{\text{GeoMean}} = \mathbf{1}[s_{\text{GeoMean}}(y_i) \le \tau_{\text{GeoMean}}]. \tag{5}$$

For $\tau_{\text{GeoMean}} \ge 1$, this requires the geometric-mean ratio to lie in the reciprocal interval $[1/\tau_{\text{GeoMean}}, \tau_{\text{GeoMean}}]$. Equivalently, $d_{\text{GeoMean}} = |\log G|$ and $s_{\text{GeoMean}} = \max(G, G^{-1})$. We use GeoMean as a descriptive shorthand for this two-sided rule. It shares signed-log aggregation with DeepSeek-V3.2, whose mask instead uses a one-sided threshold on negative-advantage responses (Liu et al., 2025a).

Proposition 1 (Cancellation under signed aggregation). *For any $B > 1$ and any $1 < c < B$, there exists an even-length response for which every token ratio lies outside $[1/c, c]$, while $s_{\text{GeoMean}}(y) = 1$.*

Proof. Let the ratios alternate between B and $1/B$. Every token is outside $[1/c, c]$, but the log-ratios alternate between $\log B$ and $-\log B$. Their sum is zero, so $d_{\text{GeoMean}}(y) = 0$ and $s_{\text{GeoMean}}(y) = 1$. □

Proposition 1 does not claim that the exact trajectory likelihood ratio is incorrect. Indeed, $\pi_\theta(y \mid x)/\pi_{\text{rollout}}(y \mid x) = \prod_t r_t$. It shows that thresholding a signed aggregate does not bound the magnitudes of the sampled-token log-ratios when used as a response-level filter. Figure 1 shows this cancellation in an actual unfiltered rollout.

3.3 Cancellation-Aware Response Masking

CARM removes direction before aggregation:

$$d_{\text{CARM}}(y_i) = \frac{1}{T_i} \sum_{t=1}^{T_i} |\log r_{i,t}|, \qquad s_{\text{CARM}}(y_i) = \exp(d_{\text{CARM}}(y_i)). \tag{6}$$

Equivalently, each token contributes the reciprocal-invariant mismatch $\max(r_{i,t}, r_{i,t}^{-1})$ in ratio space, since $|\log(1/r_{i,t})| = |\log r_{i,t}|$. The CARM mask is

$$M_i^{\text{CARM}} = \mathbf{1}[s_{\text{CARM}}(y_i) \le \tau_{\text{CARM}}]. \tag{7}$$

During optimization, the response mask and token-level surrogate use the same current-policy forward pass and the stored rollout log-probabilities, before the optimizer update. The binary gate is treated as a constant during backpropagation; a rejected response contributes no policy-loss gradient in that update. Algorithm 1 and Appendix B give the computation schedule and numerical conventions.

The two scores measure different properties: GeoMean measures the magnitude of the average signed log-ratio, whereas CARM measures the average log-ratio magnitude. Mean-absolute aggregation combines contributions linearly, preventing cancellation without selecting only the most extreme token or squaring each deviation. Figure 2 contrasts this rule with GeoMean.

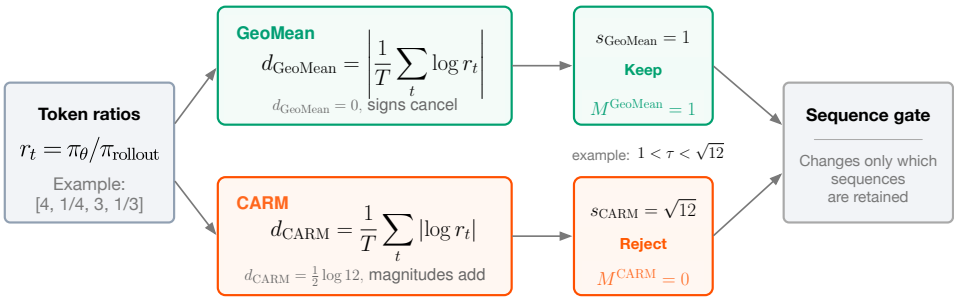


Figure 2: CARM detects bidirectional token shifts that geometric-mean masking can hide. GeoMean averages signed log-ratios before taking an absolute value; CARM averages their magnitudes. The response-level gate complements the token-level PPO/GRPO surrogate.

A real disagreement case from an unfiltered rollout. We replay both masks on an unfiltered batch of 1,024 responses generated by Qwen3.5-4B. With $\tau_{\text{GeoMean}} = 1.005$ and $\tau_{\text{CARM}} = 1.02$, 160 responses are accepted by both rules, 668 are rejected by both, 195 are accepted only by GeoMean, and one is accepted only by CARM. Figure 1 visualizes a representative GeoMean-only response whose opposing log-ratio masses cancel; Appendix C provides all 8,619 tokens. Its mean absolute log-ratio is 9.79 times the absolute signed mean, while the largest single-token contribution accounts for only 0.43% of the absolute log-ratio mass. The discrepancy is therefore not driven by one extreme token. These unequal thresholds illustrate different filtering decisions; the shared-threshold relation is examined separately below.

Proposition 2 (Acceptance-set inclusion). *For every response y, $s_{\text{CARM}}(y) \geq s_{\text{GeoMean}}(y)$. Thus, for any shared threshold $\tau \geq 1$,*

$$\{y : s_{\text{CARM}}(y) \leq \tau\} \subseteq \{y : s_{\text{GeoMean}}(y) \leq \tau\}. \tag{8}$$

Proof. The triangle inequality gives

$$\frac{1}{T}\sum_t |\log r_t| \geq \left|\frac{1}{T}\sum_t \log r_t\right|.$$

Exponentiation preserves the inequality, and the accepted-set inclusion follows immediately. □

Figure 3 replays both statistics on the same 1,024 responses from an unfiltered rollout batch, before any sequence-level gate can alter the training trajectory. Every response obeys $d_{\text{CARM}} \geq d_{\text{GeoMean}}$, with no numerical violations; the mean difference is 0.0319. At every shared threshold, CARM rejects at least as many responses as GeoMean, in agreement with Proposition 2.

3.4 Threshold Interpretation

Consider a token-ratio band $[\ell, u]$ with $0 < \ell < 1 < u$. Define the lower- and upper-side violation sets and their fractions as

$$\mathcal{V}_- = \{t : r_t < \ell\}, \qquad p_- = \frac{|\mathcal{V}_-|}{T}, \tag{9}$$

$$\mathcal{V}_+ = \{t : r_t > u\}, \qquad p_+ = \frac{|\mathcal{V}_+|}{T}. \tag{10}$$

Let $p_{\text{out}} = p_- + p_+$ and define the log-costs of reaching the boundaries by $b_- = \log(1/\ell)$ and $b_+ = \log u$. For $p_{\text{out}} > 0$, define

$$\bar{\delta}_{\text{out}}(y) = \frac{\sum_{t\in\mathcal{V}_-} \log\frac{\ell}{r_t} + \sum_{t\in\mathcal{V}_+} \log\frac{r_t}{u}}{|\mathcal{V}_-| + |\mathcal{V}_+|}, \tag{11}$$

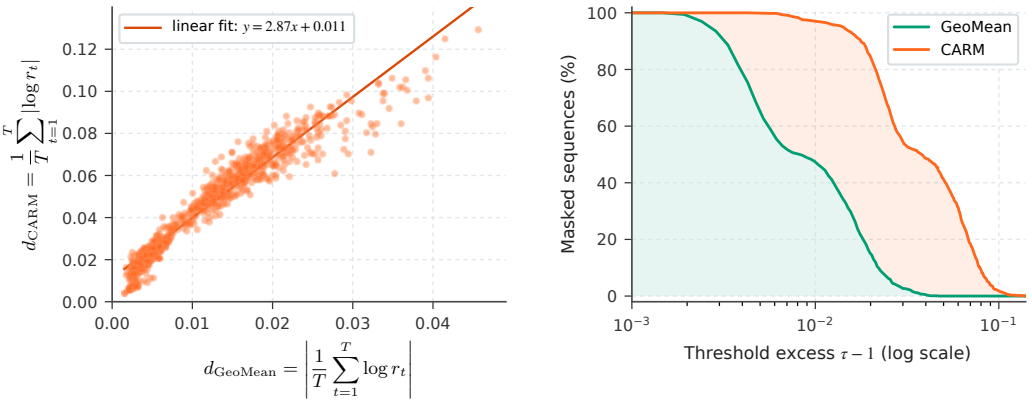


Figure 3: Paired deviations (left) and rejection rates at shared thresholds (right) on 1,024 unfiltered responses. Every response satisfies $d_{\mathrm{CARM}} \geq d_{\mathrm{GeoMean}}$. Green shading denotes rejection by both rules; orange shading denotes rejection only by CARM.

and set $\bar{\delta}_{\mathrm{out}}(y) = 0$ when $p_{\mathrm{out}} = 0$. We call $\bar{\delta}_{\mathrm{out}}$ the *mean violation severity*: it is the average log-distance by which an out-of-band token passes its nearest boundary.

Proposition 3 (Bounds on ratio-band violations). *If $s_{\mathrm{CARM}}(y) \leq \tau$ with $\tau \geq 1$, then*

$$p_- b_- + p_+ b_+ + p_{\mathrm{out}} \bar{\delta}_{\mathrm{out}}(y) \leq \log \tau. \tag{12}$$

Consequently, with $b_{\min} = \min(b_-, b_+)$,

$$p_{\mathrm{out}} \leq \min\left\{1, \frac{\log \tau}{b_{\min} + \bar{\delta}_{\mathrm{out}}(y)}\right\} \leq \min\left\{1, \frac{\log \tau}{b_{\min}}\right\}. \tag{13}$$

Thus each response accepted by CARM has a shared budget for the fraction of violating token ratios and their mean severity. At a fixed threshold, more severe violations imply a smaller admissible fraction; the bound does not require a reciprocal-symmetric band. This is a deterministic bound on the sampled-token ratios, not on the full conditional-policy KL divergence.

Proof. The acceptance condition is $\frac{1}{T}\sum_t |\log r_t| \leq \log \tau$. For a lower violation, $|\log r_t| = b_- + \log(\ell / r_t)$; for an upper violation, $|\log r_t| = b_+ + \log(r_t / u)$. Sum these identities over violating tokens, drop the non-negative in-band contribution, and divide by T to obtain Equation 12. Replacing both boundary costs by $b_{\min}$ gives $p_{\mathrm{out}}(b_{\min} + \bar{\delta}_{\mathrm{out}}) \leq \log \tau$. Division by the positive denominator and $p_{\mathrm{out}} \leq 1$ give the fraction bound. Appendix A provides the full derivation. □

Corollary 1 (Reciprocal-symmetric band). *For $[1/c, c]$ with $c > 1$, define $\mathcal{V}_c = \{t : r_t < 1/c \text{ or } r_t > c\}$, $p_c = |\mathcal{V}_c|/T$, and let $\bar{\delta}_c$ be the mean of $|\log r_t| - \log c$ over $t \in \mathcal{V}_c$ (zero when $p_c = 0$). If $s_{\mathrm{CARM}}(y) \leq \tau$, then*

$$p_c\big(\log c + \bar{\delta}_c\big) \leq \log \tau. \tag{14}$$

In particular, $p_c \leq \log \tau / \log c$.

Proof. Set $\ell = 1/c$ and $u = c$ in Proposition 3; then $b_- = b_+ = \log c$ and $\bar{\delta}_{\mathrm{out}} = \bar{\delta}_c$. □

The experimental PPO/GRPO clip parameters $\epsilon_{\mathrm{low}} = 0.2$ and $\epsilon_{\mathrm{high}} = 0.28$ correspond to the numerical band $[0.8, 1.28]$. For this band, the lower and upper boundary costs are $-\log 0.8 \approx 0.2231$ and $\log 1.28 \approx 0.2469$. Consequently, Equation 13 gives $p_{\mathrm{out}} \leq 8.88\%$ at $\tau = 1.02$ and $p_{\mathrm{out}} \leq 17.58\%$ at $\tau = 1.04$ for any accepted sequence. The PPO clipping interval serves only as a reference band for interpreting violations; CARM does not impose it as a hard per-token constraint.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Experiments use `verl` (Sheng et al., 2025) and GRPO. We train Qwen3.5-4B and Qwen3.5-9B (Qwen Team, 2026) on DAPO Math (Yu et al., 2025) and evaluate AIME 2024/2025/2026 (Mathematical Association of America, 2026) and BeyondAIME (Chen et al., 2025) with 16 responses per problem. For code generation, Qwen3.5-9B is trained on the TACO training split (Li et al., 2023) with executable rewards. Evaluation uses the held-out TACO test split, LiveCodeBench-v6 (Jain et al., 2025), and the EvalPlus suites HumanEval+ and MBPP+ (Liu et al., 2023). Metrics are obtained using three random seeds for both tasks. Math-Verify checks mathematical answers (Kydlíček, 2025); SandboxFusion executes and judges generated code (Cheng et al., 2025). Baselines are Null, IcePop, TRM, and GeoMean. Null retains the same GRPO procedure, including token-level clipping and advantage normalization, but applies no additional off-policy response mask. Appendix D and Table 3 give the datasets, prompts, and configuration.

4.2 MATHEMATICAL REASONING

Table 1 compares mathematical reasoning across AIME 2024/2025/2026 and BeyondAIME. CARM achieves the highest four-dataset average for both Qwen3.5-4B and Qwen3.5-9B. Its three-year AIME average also exceeds those of Null, IcePop, TRM, and GeoMean: relative to GeoMean, it rises from 69.17% to 70.42% on Qwen3.5-4B and from 74.03% to 77.50% on Qwen3.5-9B. The latter improvement holds separately on all three AIME years. For Qwen3.5-4B, CARM improves AIME 2024, ties on AIME 2025, and is slightly lower on AIME 2026, while improving the aggregate.

Table 1: Mathematical-reasoning mean@16 across methods (%, higher is better). *Average* is the mean of AIME 2024, AIME 2025, AIME 2026, and BeyondAIME. Bold and underline denote the best and second-best values within each model block.

Model	Method	AIME 2024	AIME 2025	AIME 2026	BeyondAIME	Average
Qwen3.5-4B	Null	49.38	48.33	54.79	22.25	43.69
	IcePop	47.08	40.83	40.42	19.31	36.91
	TRM	66.67	55.21	65.00	32.31	54.80
	GeoMean	<u>72.50</u>	**64.38**	**70.62**	<u>38.88</u>	<u>61.59</u>
	CARM (ours)	**77.08**	**64.38**	<u>69.79</u>	**42.56**	**63.45**
Qwen3.5-9B	Null	76.67	66.88	67.71	45.25	64.13
	IcePop	70.42	59.58	66.67	34.94	57.90
	TRM	78.33	<u>67.08</u>	<u>76.46</u>	45.81	66.92
	GeoMean	<u>80.42</u>	66.67	75.00	<u>47.06</u>	<u>67.29</u>
	CARM (ours)	**82.08**	**71.04**	**79.38**	**49.19**	**70.42**

On BeyondAIME, CARM raises mean@16 from 38.88% to 42.56% on Qwen3.5-4B and from 47.06% to 49.19% on Qwen3.5-9B relative to GeoMean. The gains therefore extend beyond the annual AIME benchmarks. Across all four datasets, the Average increases from 61.59% to 63.45% on Qwen3.5-4B and from 67.29% to 70.42% on Qwen3.5-9B. For the larger model, the aggregate lead accompanies an improvement on every benchmark; for the smaller model, it reflects an overall gain with a mixed year-by-year AIME pattern.

Relative to TRM, CARM improves the three-year AIME average by 8.13 points on Qwen3.5-4B and 3.54 points on Qwen3.5-9B despite TRM's higher rejection rates (Section 5.1). Relative to IcePop, CARM improves the four-dataset Average by 26.54 points on Qwen3.5-4B and 12.52 points on Qwen3.5-9B.

4.3 Code Generation

Table 2 reports pass@1 for Qwen3.5-9B trained on the TACO training split. Testing on TACO measures performance on held-out problems from the same dataset, while LiveCodeBench-v6, HumanEval+, and MBPP+ evaluate the resulting policy on three additional benchmarks.

Table 2: Evaluation pass@1 across code benchmarks (%, higher is better). TACO denotes the held-out test split, separate from the training split. *Average* is the mean of the four datasets. Bold and underline denote the best and second-best values.

Method	TACO	LiveCodeBench-v6	HumanEval+	MBPP+	Average
Null	34.50	38.46	85.57	75.22	58.44
IcePop	45.87	43.36	<u>92.68</u>	<u>79.81</u>	<u>65.43</u>
TRM	36.60	<u>44.97</u>	87.80	76.72	61.52
GeoMean	<u>46.03</u>	44.50	92.28	78.31	65.28
CARM (ours)	**46.70**	**51.20**	**94.11**	**81.22**	**68.31**

CARM improves average pass@1 by 3.03 points over GeoMean and 2.88 points over IcePop, the strongest baseline on average. The gains are not uniform: relative to GeoMean, the increase is modest on TACO, from 46.03% to 46.70%, and largest on LiveCodeBench-v6, from 44.50% to 51.20%. On HumanEval+ and MBPP+, CARM reaches 94.11% and 81.22%, also exceeding IcePop, the strongest baseline on these suites. Thus, the improvement is not confined to held-out TACO problems.

5 Analysis

We examine whether the gains are explained by filtering more responses, whether advantage sign should override the gate, and whether improvements persist across different sampling criteria.

5.1 Update Redundancy and Effective Filtering

We use *update redundancy* to describe improved validation accuracy despite omitting response-level updates, without implying that rejected responses lack useful information.

GeoMean and CARM mask an average of 55.65% and 54.52% of sequences during Qwen3.5-4B training, yet improve three-year AIME mean@16 over Null by 18.33 and 19.58 points. For Qwen3.5-9B, the corresponding fractions are 52.75% and 52.32%, with gains of 3.61 and 7.08 points. Filtering persists throughout training (Appendix Figure 9). Filtering already-generated responses complements selective rollout use and data-efficient reasoning (Yu et al., 2025; Sun et al., 2025; Lu et al., 2026b; Wang et al., 2025; Zheng et al., 2025b).

Figure 4 compares AIME accuracy with mean sequence-masked fraction. On Qwen3.5-9B, CARM outperforms GeoMean by 3.47 points while their masked fractions differ by only 0.43 points. Accuracy is not monotone in the masked fraction, and TRM rejects more responses without outperforming CARM.

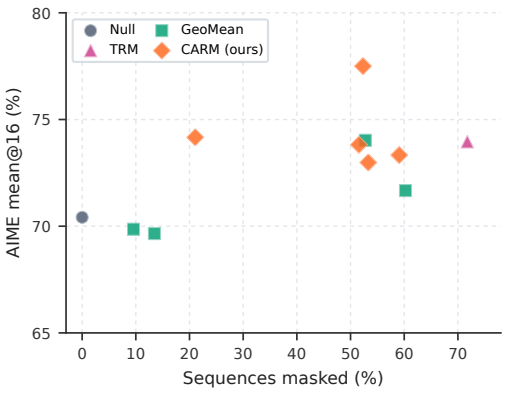


Figure 4: Sequence filtering versus AIME accuracy. Each point is a masking configuration. AIME mean@16 averages the 2024/2025/2026 benchmarks.

Paired-score differences (Figure 3) show that rejection rates alone do not characterize retained trajectories. Tightening CARM's frequency–severity budget (Proposition 3) limits sampled-token mismatch but removes update contributions, consistent with the observed non-monotonicity.

5.2 Advantage-Conditional Masking Ablation

DeepSeek-V3.2 masks only negative-advantage responses (Liu et al., 2025a). We test this conditioning on Qwen3.5-9B with CARM's score and threshold fixed within each task. Standard masking reaches 75.63% AIME mean@16 (2024/2025/2026) versus 60.28% for negative-only masking; LiveCodeBench-v6 scores are 50.58% versus 42.22% (Figure 5).

Even a correctly rewarded response can be stale if the current policy assigns substantially different probabilities to its tokens. PPO clips positive-advantage terms above the upper ratio bound, but other token updates can remain active (Schulman et al., 2017); it does not enforce a response-level trust region (Li et al., 2026). These results suggest that response-level mismatch checks can remain useful even for positive-advantage responses.

5.3 Sampling Consistency

We assess sampling consistency using bootstrapped best-of-k and worst-of-k: the former requires at least one correct response among k, while the latter requires all k to be correct. Worst-of-k measures sampling repeatability, not distribution-shift or adversarial robustness. Appendix G defines the estimators and reports results for $k \in \{1, 2, 4, 8, 16\}$.

For Qwen3.5-9B, Figure 6 averages results over AIME 2024/2025/2026. CARM improves mean@16, best@16, and worst@16 over GeoMean by 3.47, 2.74, and 2.78 points, respectively; the corresponding gains over TRM are 3.54, 0.30, and 4.61 points. CARM also achieves the highest best-of-k and worst-of-k values throughout the displayed range.

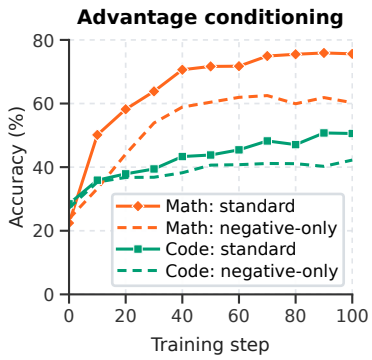


Figure 5: Standard (solid) versus negative-only (dashed) CARM masking.

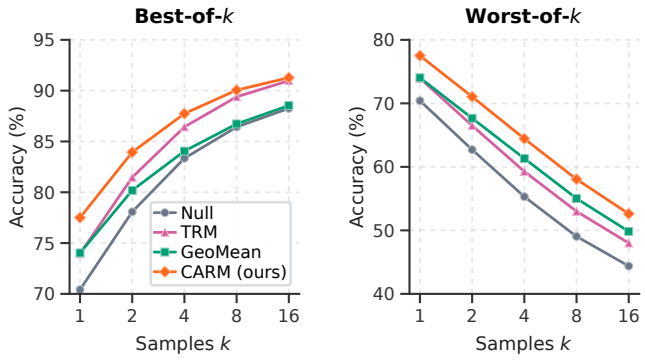


Figure 6: Bootstrapped best-of-k (left) and worst-of-k (right) accuracy on AIME.

6 Conclusion

We proposed CARM, a response-level off-policy mask that averages absolute token log-ratios to prevent opposing probability changes from canceling. Our analysis shows that signed aggregation can accept responses even when every sampled-token ratio lies outside a reciprocal-symmetric band. In contrast, CARMbounds the frequency and severity of ratio-band violations through a joint budget for each accepted response, giving the threshold a token-level interpretation. Experiments show gains in mathematical reasoning and code generation over the evaluated baselines, together with improved sampling consistency. These findings support CARM as a principled response-selection criterion that complements reward-directed PPO/GRPO optimization.

AI USE STATEMENT

Generative AI tools were used to assist with and polish the writing of this manuscript, primarily to improve clarity, readability, and language. The authors reviewed all AI-assisted content and take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work studies algorithmic stability in language-model reinforcement learning and does not involve human subjects or private user data. The broader risks are those associated with improving general-purpose language models; the proposed filtering mechanism does not introduce a new deployment capability by itself.

REPRODUCIBILITY STATEMENT

Section 3 gives the complete mask definitions and theoretical assumptions; Appendix A provides detailed proofs. Section 4.1 and Appendix D report the models, data, optimization settings, and evaluation protocol. Appendix B specifies the masking algorithm.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, et al. Seed1.5-Thinking: Advancing superb reasoning models with reinforcement learning, 2025. URL https://arxiv.org/abs/2504.13914.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

Yao Cheng, Jianfeng Chen, Jie Chen, Li Chen, Liyu Chen, Wentao Chen, Zhengyu Chen, et al. FullStack Bench: Evaluating LLMs as full stack coders, 2025. URL https://arxiv.org/abs/2412.00535.

Shihan Dou, Yan Liu, Haoxiang Jia, Enyu Zhou, Limao Xiong, Junjie Shan, Caishuang Huang, Xiao Wang, Xiaoran Fan, Zhiheng Xi, Yuhao Zhou, Tao Ji, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. StepCoder: Improving code generation with reinforcement learning from compiler feedback. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4571–4585. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.251. URL https://aclanthology.org/2024.acl-long.251/.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization, 2025. URL https://arxiv.org/abs/2501.03262.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hynek Kydlíček. Math-Verify: Math verification library. Software, 2025. URL https://github.com/huggingface/Math-Verify.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21314–21328, 2022. doi: 10.52202/068431-1549.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. TACO: Topics in algorithmic code generation dataset, 2023. URL https://arxiv.org/abs/2312.14852.

Yingru Li, Jiacai Liu, Jiawei Xu, Yuxuan Tong, Ziniu Li, and Baoxiang Wang. Trust region masking for long-horizon LLM reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dPaBOu0Xwk.

Aixin Liu, Aoxue Mei, Bangcai Lin, et al. DeepSeek-V3.2: Pushing the frontier of open large language models, 2025a. URL https://arxiv.org/abs/2512.02556.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Zhuo Jiang. When speed kills stability: Demystifying RL collapse from the training–inference mismatch, September 2025b. URL https://richardli.xyz/rl-collapse.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Songshuo Lu, Hua Wang, Zhi Chen, and Yaohua Tang. URPO: A unified reward & policy optimization framework for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):32329–32337, 2026a. doi: 10.1609/aaai.v40i38.40507. URL https://ojs.aaai.org/index.php/AAAI/article/view/40507.

Xiaodong Lu, Xiaohan Wang, Jiajun Chai, Guojun Yin, Wei Lin, Zhijun Chen, Yu Luo, Fuzhen Zhuang, Yikun Ban, and Deqing Wang. Contextual rollout bandits for reinforcement learning with verifiable rewards, 2026b. URL https://arxiv.org/abs/2602.08499.

Mathematical Association of America. MAA invitational competitions: American invitational mathematics examination, 2026. URL https://maa.org/maa-invitational-competitions/. Accessed September 2026.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. doi: 10.52202/068431-2011.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in LLM reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=f4g8zblbwp.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Anqi Shen, Baihui Li, Bin Hu, et al. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model, 2025. URL https://arxiv.org/abs/2510.18855.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Yan Sun, Jia Guo, Stanley Kok, Zihao Wang, Zujie Wen, and Zhiqiang Zhang. Efficient reinforcement learning for large language models with intrinsic exploration, 2025. URL https://arxiv.org/abs/2511.00794.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Advances in Neural Information Processing Systems*, volume 38, pp. 122721–122764, 2025. doi: 10.52202/085713-4093.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800.

Zhiheng Xi, Xin Guo, Yang Nan, et al. BAPO: Stabilizing off-policy reinforcement learning for LLMs via balanced policy optimization with adaptive clipping. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/pdf?id=jIeJJqG7dz.

Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. Simple policy optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68813–68824. PMLR, 2025. URL https://proceedings.mlr.press/v267/xie25m.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient RL framework secretly brings you off-policy RL training, August 2025. URL https://fengyao.notion.site/off-policy-rl.

Jiarui Yao, Xiangxin Zhou, Penghui Qi, Wee Sun Lee, Liefeng Bo, and Tianyu Pang. Rethinking the divergence regularization in LLM RL, 2026. URL https://arxiv.org/abs/2606.09821.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, et al. GLM-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025a. URL https://arxiv.org/abs/2507.18071.

Haizhong Zheng, Yang Zhou, Brian Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124321–124346, 2025b. doi: 10.52202/085713-4143.

A Detailed Proofs

A.1 Proof of Proposition 1

Step 1: construct an even-length response. Fix arbitrary constants $B > 1$ and $1 < c < B$. Choose an arbitrary positive integer m and let the response length be $T = 2m$. For every pair $j \in \{1, \dots, m\}$, define

$$r_{2j-1} = B, \qquad r_{2j} = \frac{1}{B}. \tag{15}$$

Thus the response contains exactly m ratios equal to B and exactly m ratios equal to $1/B$. These ratios can arise from valid conditional policies: at each odd position, assign the sampled token probabilities $1/(1+B)$ under the rollout policy and $B/(1+B)$ under the current policy, and swap them at each even position. Assign the remaining probability to a second token. Both policies are normalized, all probabilities are positive, and their sampled-token ratios are exactly those in Equation 15.

Step 2: verify that every ratio violates the band. For each odd position, $r_{2j-1} = B > c$, so it lies strictly above the upper endpoint of $[1/c, c]$. For each even position, $1 < c < B$ and all quantities are positive. Taking reciprocals reverses the inequalities and gives

$$1 > \frac{1}{c} > \frac{1}{B}. \tag{16}$$

Hence $r_{2j} = 1/B < 1/c$, so it lies strictly below the lower endpoint. Therefore every one of the T token ratios is outside $[1/c, c]$.

Step 3: evaluate the GeoMean statistic. The logarithm of a reciprocal satisfies $\log(1/B) = -\log B$. Grouping the signed log-ratios into the m adjacent pairs in Equation 15 yields

$$\sum_{t=1}^{T} \log r_t = \sum_{j=1}^{m} (\log B + \log(1/B)) \tag{17}$$

$$= \sum_{j=1}^{m} (\log B - \log B) \tag{18}$$

$$= \sum_{j=1}^{m} 0 \tag{19}$$

$$= 0. \tag{20}$$

Substitution into Equation 4 gives

$$d_{\mathrm{GeoMean}}(y) = \left| \frac{1}{T} \sum_{t=1}^{T} \log r_t \right| \tag{21}$$

$$= \left| \frac{0}{T} \right| \tag{22}$$

$$= 0, \tag{23}$$

and therefore

$$s_{\mathrm{GeoMean}}(y) = \exp(d_{\mathrm{GeoMean}}(y)) = \exp(0) = 1. \tag{24}$$

This proves the proposition: every token can violate the prescribed reciprocal band even though GeoMean reports its minimum possible score.

Step 4: evaluate the CARM statistic for comparison. Because $B > 1$, $\log B > 0$, and hence $|\log B| = |\log(1/B)| = \log B$. Consequently,

$$d_{\mathrm{CARM}}(y) = \frac{1}{T}\sum_{t=1}^{T} |\log r_t| \tag{25}$$

$$= \frac{1}{2m}\sum_{j=1}^{m} (|\log B| + |\log(1/B)|) \tag{26}$$

$$= \frac{1}{2m}\sum_{j=1}^{m} 2\log B \tag{27}$$

$$= \log B, \tag{28}$$

so $s_{\mathrm{CARM}}(y) = \exp(\log B) = B$. Thus any GeoMean threshold at least one accepts this construction, whereas CARM rejects it whenever $\tau_{\mathrm{CARM}} < B$.

A.2 Proof of Proposition 2

Step 1: move to log-ratio coordinates. For a fixed response of positive length T, set $z_t = \log r_t \in \mathbb{R}$. For $T = 1$, both scores are identical. For $T \geq 2$, the triangle inequality applied repeatedly to the finite sum gives

$$\left|\sum_{t=1}^{T} z_t\right| = |z_1 + (z_2 + \cdots + z_T)| \tag{29}$$

$$\leq |z_1| + |z_2 + \cdots + z_T| \tag{30}$$

$$\leq \cdots \leq \sum_{t=1}^{T} |z_t|. \tag{31}$$

Step 2: normalize by response length. Since $T > 0$, division by T preserves the direction of the inequality:

$$\left|\frac{1}{T}\sum_{t} z_t\right| \leq \frac{1}{T}\sum_{t} |z_t|. \tag{32}$$

Using $z_t = \log r_t$ and Equations 4 and 6, the left-hand side is exactly $d_{\mathrm{GeoMean}}(y)$ and the right-hand side is exactly $d_{\mathrm{CARM}}(y)$. Therefore

$$d_{\mathrm{GeoMean}}(y) \leq d_{\mathrm{CARM}}(y). \tag{33}$$

Step 3: return to ratio space. The exponential function is strictly increasing on $\mathbb{R}$. Applying it to both sides gives

$$\exp(d_{\mathrm{GeoMean}}(y)) \leq \exp(d_{\mathrm{CARM}}(y)), \tag{34}$$

$$s_{\mathrm{GeoMean}}(y) \leq s_{\mathrm{CARM}}(y). \tag{35}$$

This proves the pointwise score inequality.

Step 4: prove accepted-set inclusion. Take an arbitrary response $y \in \{y : s_{\mathrm{CARM}}(y) \leq \tau\}$. By membership in this set, $s_{\mathrm{CARM}}(y) \leq \tau$. Combining this fact with the pointwise result from Step 3 gives

$$s_{\mathrm{GeoMean}}(y) \leq s_{\mathrm{CARM}}(y) \leq \tau. \tag{36}$$

Thus $y \in \{y : s_{\mathrm{GeoMean}}(y) \leq \tau\}$. Since the argument holds for every element of the CARM accepted set, Equation 8 follows.

Equality condition. For real scalars, equality in the triangle inequality holds exactly when all nonzero z_t have the same sign. Accordingly, $s_{\mathrm{GeoMean}} = s_{\mathrm{CARM}}$ when all nonzero token log-ratios are nonnegative or all are nonpositive. If at least one $z_t > 0$ and at least one $z_{t'} < 0$, the inequality is strict.

A.3 Proof of Proposition 3

Step 1: convert acceptance into an additive budget. By definition,

$$s_{\mathrm{CARM}}(y) = \exp\left(\frac{1}{T}\sum_{t=1}^{T} |\log r_t|\right). \tag{37}$$

The acceptance assumption $s_{\mathrm{CARM}}(y) \le \tau$ and monotonicity of $\log(\cdot)$ imply

$$\log s_{\mathrm{CARM}}(y) \le \log \tau, \tag{38}$$

$$\frac{1}{T}\sum_{t=1}^{T} |\log r_t| \le \log \tau. \tag{39}$$

Multiplying by the positive length T gives the total absolute-log budget

$$\sum_{t=1}^{T} |\log r_t| \le T \log \tau. \tag{40}$$

Step 2: partition positions and define the boundary costs. Let $\mathcal{V}_0 = \{1, \dots, T\} \setminus (\mathcal{V}_- \cup \mathcal{V}_+)$. Because $\ell < 1 < u$, the sets $\mathcal{V}_-$, $\mathcal{V}_0$, and $\mathcal{V}_+$ are disjoint and cover all token positions. Define

$$b_- = \log \frac{1}{\ell} = -\log \ell, \qquad b_+ = \log u. \tag{41}$$

The assumptions $0 < \ell < 1 < u$ imply $b_- > 0$ and $b_+ > 0$.

Step 3: decompose a lower-side violation exactly. Take $t \in \mathcal{V}_-$. Then $0 < r_t < \ell < 1$, so $|\log r_t| = -\log r_t$. Inserting and subtracting $\log \ell$ gives

$$|\log r_t| = -\log r_t \tag{42}$$

$$= -\log \ell + \log \ell - \log r_t \tag{43}$$

$$= b_- + \log \frac{\ell}{r_t}. \tag{44}$$

Since $\ell / r_t > 1$, the final term is positive. It approaches zero at the lower boundary and grows with the token's log-distance below that boundary.

Step 4: decompose an upper-side violation exactly. Take $t \in \mathcal{V}_+$. Then $r_t > u > 1$, so $|\log r_t| = \log r_t$. Inserting and subtracting $\log u$ gives

$$|\log r_t| = \log r_t \tag{45}$$

$$= \log u + \log r_t - \log u \tag{46}$$

$$= b_+ + \log \frac{r_t}{u}. \tag{47}$$

Because $r_t / u > 1$, this final term is likewise a positive log-distance past the upper boundary.

Step 5: sum the exact decompositions. Partition the total absolute-log drift and substitute Equations 44 and 47 term by term:

$$\sum_{t=1}^{T} |\log r_t| = |\mathcal{V}_-| b_- + |\mathcal{V}_+| b_+ \tag{48}$$

$$+ \sum_{t \in \mathcal{V}_-} \log \frac{\ell}{r_t} + \sum_{t \in \mathcal{V}_+} \log \frac{r_t}{u} \tag{49}$$

$$+ \sum_{t \in \mathcal{V}_0} |\log r_t|. \tag{50}$$

Step 6: identify the conditional mean log-distance. If $p_{\text{out}} > 0$, Equation 11 implies

$$\sum_{t\in\mathcal{V}_-} \log\frac{\ell}{r_t} + \sum_{t\in\mathcal{V}_+} \log\frac{r_t}{u} \tag{51}$$

$$= \big(|\mathcal{V}_-| + |\mathcal{V}_+|\big)\bar{\delta}_{\text{out}}(y). \tag{52}$$

Divide by T and use $(|\mathcal{V}_-| + |\mathcal{V}_+|)/T = p_{\text{out}}$:

$$\frac{1}{T}\left(\sum_{t\in\mathcal{V}_-} \log\frac{\ell}{r_t} + \sum_{t\in\mathcal{V}_+} \log\frac{r_t}{u}\right) = p_{\text{out}}\bar{\delta}_{\text{out}}(y). \tag{53}$$

If $p_{\text{out}} = 0$, both sums are empty; our zero convention makes the same identity valid.

Step 7: obtain the master budget. The in-band sum in Equation 50 is nonnegative. Drop it, apply Equation 40, divide by T, and use Equation 53:

$$p_-b_- + p_+b_+ + p_{\text{out}}\bar{\delta}_{\text{out}}(y) \le \log\tau. \tag{54}$$

This is Equation 12.

Step 8: bound the total violation fraction. Let $b_{\min} = \min(b_-, b_+) > 0$. Keeping the severity contribution in the master budget gives

$$p_{\text{out}}(b_{\min} + \bar{\delta}_{\text{out}}) = (p_- + p_+)b_{\min} + p_{\text{out}}\bar{\delta}_{\text{out}} \tag{55}$$

$$\le p_-b_- + p_+b_+ + p_{\text{out}}\bar{\delta}_{\text{out}} \tag{56}$$

$$\le \log\tau. \tag{57}$$

Because $b_{\min} + \bar{\delta}_{\text{out}} > 0$, division yields

$$p_{\text{out}} \le \frac{\log\tau}{b_{\min} + \bar{\delta}_{\text{out}}} \le \frac{\log\tau}{b_{\min}}. \tag{58}$$

Combining each inequality with $p_{\text{out}} \le 1$ proves Equation 13, including the case $p_{\text{out}} = 0$ under the stated zero convention.

Step 9: recover the frequency-only budget. Every term defining $\bar{\delta}_{\text{out}}$ is nonnegative. Dropping the distance term from the master budget therefore gives $p_-b_- + p_+b_+ \le \log\tau$. Let $b_{\min} = \min(b_-, b_+) > 0$. Then

$$p_{\text{out}}b_{\min} = (p_- + p_+)b_{\min} \tag{59}$$

$$\le p_-b_- + p_+b_+ \tag{60}$$

$$\le \log\tau. \tag{61}$$

Division by $b_{\min}$ gives $p_{\text{out}} \le \log\tau/b_{\min}$. Independently, $p_{\text{out}} \le 1$ because it is a fraction. Taking the smaller upper bound recovers the severity-independent consequence of Equation 13.

A.4 Proof of Corollary 1

Step 1: specialize the two endpoints. Set $\ell = 1/c$ and $u = c$. Because $c > 1$, the assumptions $0 < \ell < 1 < u$ hold, and the two boundary costs are identical:

$$b_- = \log(1/\ell) = \log c, \tag{62}$$

$$b_+ = \log u = \log c. \tag{63}$$

Step 2: combine the violation sets. The lower and upper violation sets are $\mathcal{V}_- = \{t : r_t < 1/c\}$ and $\mathcal{V}_+ = \{t : r_t > c\}$. They are disjoint because $c > 1$, and their union is exactly $\mathcal{V}_c$. Hence

$$p_- + p_+ = \frac{|\mathcal{V}_c|}{T} = p_c. \tag{64}$$

Step 3: identify the reciprocal-band log-distance. For $t \in \mathcal{V}_-$,

$$\log \frac{\ell}{r_t} = \log \frac{1/c}{r_t} \tag{65}$$

$$= |\log r_t| - \log c, \tag{66}$$

where the second equality uses $r_t < 1$. For $t \in \mathcal{V}_+$,

$$\log \frac{r_t}{u} = \log r_t - \log c \tag{67}$$

$$= |\log r_t| - \log c, \tag{68}$$

where $r_t > 1$. Thus the conditional mean in Equation 11 is precisely $\bar{\delta}_c$.

Step 4: simplify the master budget. Substitute $p_- + p_+ = p_c$, $b_- = b_+ = \log c$, and $\bar{\delta}_{\text{out}} = \bar{\delta}_c$ into Equation 12:

$$p_- \log c + p_+ \log c + p_c \bar{\delta}_c \le \log \tau, \tag{69}$$

$$p_c \log c + p_c \bar{\delta}_c \le \log \tau, \tag{70}$$

$$p_c (\log c + \bar{\delta}_c) \le \log \tau. \tag{71}$$

This is Equation 14. Finally, $\bar{\delta}_c \ge 0$, so dropping $p_c \bar{\delta}_c$ gives $p_c \log c \le \log \tau$. Since $\log c > 0$, division by $\log c$ yields $p_c \le \log \tau / \log c$.

B Implementation of CARM

Algorithm 1 gives the response-level masking rule, treating the hard threshold decision as constant during backpropagation. The gate changes which response losses contribute to an update while retaining the token-level PPO/GRPO surrogate.

Algorithm 1 CARM response-level off-policy masking

Require: Finite current and rollout token log-probabilities $\log \pi_\theta(y_{i,t} \mid x_i, y_{i,<t})$ and $\log \pi_{\text{rollout}}(y_{i,t} \mid x_i, y_{i,<t})$; response-token mask $a_{i,t} \in \{0,1\}$ with $\sum_t a_{i,t} \ge 1$ for each response; threshold $\tau_{\text{CARM}} \ge 1$

Ensure: Response-acceptance indicators M_i^{CARM}

1: **for** each response y_i **do**
2: $\quad T_i \leftarrow \sum_t a_{i,t}$
3: $\quad \Delta_{i,t} \leftarrow \text{stopgrad}(\log \pi_\theta(y_{i,t} \mid \cdot) - \log \pi_{\text{rollout}}(y_{i,t} \mid \cdot))$
4: $\quad d_{\text{CARM},i} \leftarrow \frac{1}{T_i} \sum_t a_{i,t} |\Delta_{i,t}|$
5: $\quad s_{\text{CARM},i} \leftarrow \exp(d_{\text{CARM},i})$
6: $\quad M_i^{\text{CARM}} \leftarrow \mathbf{1}[s_{\text{CARM},i} \le \tau_{\text{CARM}}]$
7: **end for**
8: Gate each per-response PPO/GRPO loss by M_i^{CARM} and apply the training implementation's loss normalization.

Mask computation schedule. We store the sampled-token log-probabilities produced during rollout. During training, each optimizer mini-batch is divided into micro-batches; each micro-batch forward supplies the current log-probabilities used by both the response mask and the token surrogate. The optimizer updates the policy after gradient accumulation over the mini-batch. Threshold comparisons produce a binary gate with no gradient through the decision, as represented by stopgrad in Algorithm 1. Masking is local to loss evaluation rather than permanent removal from the rollout batch. Our experiments use one optimization epoch per rollout batch; with multiple epochs, the gate would be recomputed whenever a response is evaluated under the updated policy.

The PPO/GRPO loss implementation in `verl` also applies dual clipping to negative advantages. Writing $\ell_{i,t}$ for Equation 2, its surrogate is

$$\ell_{i,t}^{\text{dual}} = \begin{cases} \max\{\ell_{i,t}, c_{\text{dual}} \hat{A}_i\}, & \hat{A}_i < 0, \\ \ell_{i,t}, & \hat{A}_i \ge 0, \end{cases} \qquad c_{\text{dual}} = 3. \tag{72}$$

The sequence mask multiplies this surrogate without changing its clipping rule or normalization.

Numerical convention. The theoretical scores use exact log-ratios. The implementation uses `verl`'s numerical guard $\widetilde{z}_t = \mathrm{clip}(\log r_t, -20, 20)$ when computing the mask and token surrogate. If the guard is inactive, the implemented and theoretical scores coincide. Otherwise, the score inequalities and frequency–severity budget apply to the effective ratios $\widetilde{r}_t = \exp(\widetilde{z}_t)$. For a band strictly inside $[e^{-20}, e^{20}]$, including $[0.8, 1.28]$, clipping does not change which tokens violate the band. Thus the frequency-only guarantee also holds for the original ratios, while the severity term measures the guarded log-distances.

C Complete Response for the Disagreement Case

Figure 1 in the main paper visualizes the complete signed-drift trajectory and a tokenizer-aligned local window. For auditability, the five pages below show all 8,619 response tokens from the same rollout without elision. Each token is rendered in its tokenizer-aligned box: red denotes $\log r_t > 0$, green denotes $\log r_t < 0$, and color intensity encodes $|\log r_t|$. The first-page header repeats the response-level scores and masking decisions. For this response, $d_{\mathrm{GeoMean}} = 0.002502$ and $d_{\mathrm{CARM}} = 0.024486$: signed aggregation cancels 89.78% of the absolute log-ratio mass. The example demonstrates score cancellation and differing mask decisions.



Complete token-level view for the Figure 1 disagreement case | Part 2/5

Tokens 1,843-3,632 | Continuation of the same unabridged tokenizer-aligned response.

$\log r_t > 0$ $\log r_t < 0$



Complete token-level view for the Figure 1 disagreement case | Part 3/5

Tokens 3,633-5,377 | Continuation of the same unabridged tokenizer-aligned response.

$\log r_t > 0$ $\log r_t < 0$

tokens 3,633-4,521

$ f (5) = \ log _ 2 (4) = 2 $. \n

$ f (- 1 0 0) = - 1 / 2 $. \n\n

Let 's double check $ f (x + 7)f (x) = - 1 $. \n

Original : $ f (x + 3)f (x - 4) =- 1 $. \n

Shift $ x \ to x + 7 $: \n

$ f (x + 7 + 3)f (x + 7 - 4) = f (x + 1 0)f (x + 3) = - 1 $. \n

This doesn 't match $ f (x + 7)f (x) $. \n

Wait , derivation was : \n

Let $ y = x - 4 $. Then $ x = y + 4 $. \n

$ f (y + 4 + 3)f (y) = - 1 \ impl ies f (y + 7)f (y) = - 1 $. \n

This derivation is correct . \n

Wait , earlier I wrote " Replace $ x $ with $ x + 4 $ ". \n

Original equation : for any $ x $, $ f (x + 3)f (x - 4) =- 1 $. \n

Let 's substitute specific $ x $. \n

If we want to link $ f (x)$ and $ f (x + 7)$, we need the arguments of $ f $ in the equation to match $ x $ and $ x + 7 $. \n

In $ f (x + 3)f (x - 4)$, the difference between arguments is $(x + 3) - (x - 4) = 7 $. \n

So yes , $ f (u)f (v)$ with $ u -v = 7 $ equals $ - 1 $. \n

Let $ u = x + 7 $, $ v = x $. $ u -v = 7 $. \n

Then $ f (u)f (v) = - 1 $, so $ f (x + 7)f (x) = - 1 $. \n

Yes , derivation stands . \n\n

Now , check the mapping from $ - 1 0 0 $ to $ [0 , 7) $. \n

$ - 1 0 0 \ equiv 1 2 \ p mod 7 $. Wait , \n

$ f (x + 7) = - 1 /f (x) $. \n

$ f (x + 1 4) = f (x) $. \n

So we map $ - 1 0 0 $ by adding multiples of 1 4 \n

$ - 1 0 0 + 1 1 2 = 1 2 $. \n

We want an input to the function definition $ f (t) = \ log _ 2 (9 -t)$ for $ t \ in [0 , 7) $. \n

We have $ f (1 2) $. \n

$ f (1 2)f (5) = - 1 $ because $ 1 2 = 5 + 7 $. \n

Is $ 5 \ in [0 , 7)$? Yes . \n

So $ f (5) = \ log _ 2 (9 - 5) = 2 $. \n

$ f (1 2) = - 1 / 2 $. \n

$ f (- 1 0 0) = f (1 2) = - 1 / 2 $. \n

Seems solid . \n\n

Is there any ambiguity about $ m , n $ being integers ? \n

Usually , questions like " find m +n where m /n is the value " imply integer numerator and denominator in lowest terms . \n

$ m =- 1 , n = 2 $. \n

$ m +n = 1 $. \n

Sometimes they want positive $ n $. Yes , usually $ n > 0 $. \n

Sometimes they define $ m , n $ as positive integers and add a sign to the value ? No , $ m /n $ format usually allows negative $ m $. \n

If the answer was $ - 1 / 2 $, then $ m =- 1 , n = 2 $. \n

If the problem implied $ m , n \ in \ math bb { Z } ^ +$ and $ m $ is the numerator of the simplified fraction (including sign), it might be specified . But standard interpretation is $ m , n \ in \ math bb { Z }, n \ ne 0 , \ gcd (| m |, | n |)= 1 , n > 0 $. \n

So $ m =- 1 , n = 2 $. \n

tokens 4,522-5,377

Sum = 1 . \n\n

Wait , could the log be $\ ln $? The symbol is $\ log _ 2 $. \n

Could the base be something else ? $\ log _ 2 $. \n

$ f (5) = \ log _ 2 (4) = 2 $. \n

Calculation is correct . \n\n

Let me check if I missed any subtle constraints . \n

" defined on $\ math bb { R }$ ". \n

" for any real number $ x $ ". \n

Is the periodic ity unique ? \n

$ f (x + 1 4) = f (x) $. Could it have a smaller period ? \n

$ f (x + 7) = - 1 /f (x) $. \n

Since $ f (x)$ is not ident ically $ 1 $ or $ - 1 $ (value is 2), period cannot be 7 (unless $ f (x + 7)= f (x) \ impl ies f (x) =- 1 /f (x) \ impl ies f (x)^ 2 =- 1 $, impossible for real numbers). \n

So period is exactly 1 4 (or a multiple if $ f $ has other properties not given , but 1 4 is sufficient to solve). \n

Actually , $ f (x + 7) = - 1 /f (x)$ means $ f (x)$ is anti -period ic with period 7 . \n

So $ f (x + 1 4) = f (x) $. \n

$ - 1 0 0 = - 7 \ times 1 4 - 2 $. \n

So $ f (- 1 0 0) = f (- 2) $. \n

$ f (- 2) = - 1 /f (5) $. \n

Calculation : $ f (5) = 2 $. \n

Result $ - 0 . 5 $. \n\n

Let 's quickly check $ x $ mapping again . \n

$ f (x)$ is given for $ 0 \ le x < 7 $. \n

We need $ f (- 1 0 0) $. \n

$ - 1 0 0 \ p mod { 1 4 } $. \n

$ 1 0 0 = 7 0 + 3 0 = 7 0 + 2 8 + 2 $. $ 1 0 0 \ equiv 2 \ p mod { 1 4 } $. \n

$ - 1 0 0 \ equiv - 2 \ equiv 1 2 \ p mod { 1 4 } $. \n

So $ f (- 1 0 0) = f (1 2) $. \n

$ 1 2 = 7 + 5 $. \n

$ f (1 2) = f (5 + 7) $. \n

$ f (x + 7) = - 1 /f (x) $. \n

$ f (5 + 7) = - 1 /f (5) $. \n

$ 5 \ in [0 , 7) $. \n

$ f (5) = \ log _ 2 (9 - 5) = \ log _ 2 (4) = 2 $. \n

$ f (1 2) = - 1 / 2 $. \n

Everything is consistent . \n\n

What if I evaluate $ f (- 2)$ directly from the relation $ f (x + 3)f (x - 4) =- 1 $? \n

$ f (- 2 + 3)f (- 2 - 4) = - 1 \ impl ies f (1)f (- 6) = - 1 $. \n

$ f (- 6)$? $ - 6 \ equiv 8 \ p mod { 1 4 } $. $ f (- 6) = f (8) $. \n

$ 8 = 1 + 7 $. $ f (8) = - 1 /f (1) $. \n

So $ f (1)(- 1 /f (1)) = - 1 \ impl ies - 1 = - 1 $. Tautology . Doesn 't help find value of $ f (1) $. \n

We need the value in $ [0 , 7) $. \n

$ f (- 2) $. \n

$ f (- 2 + 7)f (- 2) = - 1 \ impl ies f (5)f (- 2) = - 1 $. \n

$ 5 \ in [0 , 7) $. $ f (5) = 2 $. \n

Complete token-level view for the Figure 1 disagreement case | Part 4/5

Tokens 5,378-6,954 | Continuation of the same unabridged tokenizer-aligned response.

$\log r_t > 0$ $\log r_t < 0$





D Experimental Settings

D.1 Data

Math training data. The mathematical-reasoning experiments use the competition-style problems in the DAPO Math training corpus (Yu et al., 2025). Math-Verify (Kydlíček, 2025) extracts the boxed final answer and checks its equivalence to the reference. Null, IcePop, TRM, GeoMean, and CARM use the same prompt order, rollout group size, rewards, optimizer settings, and token-level PPO/GRPO clipping; only the off-policy correction differs.

Math evaluation data. AIME 2024, AIME 2025, and AIME 2026 each contain 30 contest problems and are reported separately (Mathematical Association of America, 2026). BeyondAIME contributes 100 additional problems designed to extend evaluation beyond the three annual AIME sets

(Chen et al., 2025). We sample 16 responses per problem and compute `acc/mean@16` from exact final-answer correctness. The Average in Table 1 gives AIME 2024, AIME 2025, AIME 2026, and BeyondAIME equal weight. Evaluation is performed before training and periodically during optimization.

Code training data. The general-code experiments use problems from the TACO training split (Li et al., 2023), with natural-language task descriptions, input/output specifications, and executable tests. During RL, the reward checks up to the first ten available tests for each training problem. This fixed cap keeps execution cost and sandbox load comparable across methods; held-out benchmark tests are not used by the training reward. All five methods receive the same prompts, decoding budget, executable reward, and optimizer configuration.

Code evaluation data. We evaluate each frozen TACO-trained checkpoint on four complementary suites. TACO evaluation uses 1,000 problems from its held-out test split and up to the first ten tests per problem. LiveCodeBench-v6 contains 1,054 temporally collected coding problems and is judged with its complete released tests. HumanEval+ and MBPP+ contain 164 and 378 problems, respectively, and use the stronger EvalPlus test suites rather than only the original base tests. MBPP+ extends the MBPP benchmark (Austin et al., 2021). Every generated program is rejudged with SandboxFusion (Cheng et al., 2025), with functional correctness reported as pass@1.

D.2 Prompt Templates

Figures 7 and 8 reproduce the task-specific chat messages used throughout training and evaluation. Dataset problem statements replace the placeholders without changing the surrounding instructions.

System: A conversation between User and Assistant. The user asks a question, and the Assistant solves it.
User: Please solve the problem step by step, and put your final answer within `\boxed{}`.
`<problem>`

Figure 7: Prompt template for mathematical-reasoning experiments. The placeholder `<problem>` is replaced by the problem text.

System: You are an expert competitive programming assistant.
Solve the user's programming problem in Python 3. You may reason about the solution, algorithm, and edge cases. At the end, provide the final answer as exactly one fenced Python code block.
The final code must be complete and executable. It should read from standard input and write to standard output, unless the problem provides starter code requiring a specific function or class interface.
User: Solve the following programming problem.
Problem:
`<problem>`
`<starter code, when provided>`
Return your final solution in this format:
```python
# your complete Python solution
```

Figure 8: Prompt template for code-generation experiments. The placeholder `<problem>` is replaced by the problem text; the starter-code field is included only when provided by the dataset.

D.3 Training and Evaluation Configuration

Table 3 consolidates the shared settings, method-specific controls, and threshold grids used to produce the reported comparisons.

Table 3: Complete experimental configuration for mathematical reasoning and code generation.

Field	Value
Base models	Qwen3.5-4B and Qwen3.5-9B (math); Qwen3.5-9B (code)
Framework / algorithm	`verl`; GRPO-style optimization
Training data	DAPO Math (math); TACO training split with executable rewards (code)
Evaluation data	AIME 2024/2025/2026 and BeyondAIME; TACO test split, LiveCodeBench-v6, HumanEval+, and MBPP+
Evaluation sampling	16 responses/problem (math); one response/problem per evaluation (code)
Prompt batch / rollout group	128 prompts / 8 responses per prompt
PPO mini-batch size	8
Training budget	Up to 120 training iterations
Maximum response length	16,384 tokens
Learning rate	3×10^{-6}
PPO clipping band	$[1 - 0.2, 1 + 0.28] = [0.8, 1.28]$
Compared methods	Null; IcePop; TRM; GeoMean; CARM (ours)
Baseline settings	IcePop token-ratio interval $[0.5, 5.0]$; TRM cutoff 0.05
GeoMean thresholds	Qwen3.5-4B: $\{1.0025, 1.005, 1.01, 1.02, 1.03\}$ Qwen3.5-9B: $\{1.0025, 1.005, 1.01, 1.02, 1.03\}$
CARM (ours) thresholds	Qwen3.5-4B: $\{1.02, 1.025, 1.03, 1.04\}$ Qwen3.5-9B: $\{1.02, 1.025, 1.03, 1.04\}$

Evaluation protocol. Both tasks are evaluated with three random seeds. Mathematical-reasoning evaluation uses 16 responses per problem and reports mean@16; code-generation evaluation uses one response per problem and reports pass@1. Code decoding uses temperature 0.7, top-$p = 0.9$, top-$k = 50$, and a maximum response length of 16,384 tokens. Generated programs are judged with the same batched SandboxFusion evaluator across methods.

For each method, math results use the training iterate with the highest three-year AIME mean@16; code results use the iterate with the highest LiveCodeBench-v6 validation pass@1. Best- and worst-of-k use the same math iterate without separate optimization of either metric. Qwen3.5-4B BeyondAIME results for Null, GeoMean, and CARM use separate runs matched in configuration and training iteration.

TRM implementation. We follow the sampled-token max-filtering option described in Section 4.3 and Appendix E of Li et al. (2026). It rejects a response when $\max_t \frac{1}{2}(\log r_t)^2 > 0.05$. This is the paper's proposed approximation when full logits are unavailable, not a full-vocabulary KL implementation. All TRM results in this paper refer to this sampled-token variant.

E THRESHOLD RESULTS

Table 4 reports the GeoMean and CARM threshold sweeps, with AIME mean@16 averaged equally across the three years.

Table 4: Threshold sweep (%, higher is better). Each row reports the three-year AIME mean@16 for that threshold. Bold and underline denote the best and second-best values within each model block.

Qwen3.5-4B			Qwen3.5-9B		
Method	τ	Avg.	Method	τ	Avg.
Null	–	50.83	Null	–	70.42
GeoMean	1.0025	69.03	GeoMean	1.0025	71.67
GeoMean	1.005	69.17	GeoMean	1.005	74.03
GeoMean	1.01	66.74	GeoMean	1.01	69.65
GeoMean	1.02	52.57	GeoMean	1.02	71.67
GeoMean	1.03	52.43	GeoMean	1.03	69.86
CARM (ours)	1.02	**70.42**	CARM (ours)	1.02	72.99
CARM (ours)	1.025	<u>69.24</u>	CARM (ours)	1.025	**77.50**
CARM (ours)	1.03	67.99	CARM (ours)	1.03	73.82
CARM (ours)	1.04	66.81	CARM (ours)	1.04	<u>74.17</u>

F ADDITIONAL TRAINING DIAGNOSTICS

Figure 9 gives the time-resolved utilization view used in Section 5.1. It distinguishes sustained filtering from a mean dominated by a small number of unusually strict updates, and contrasts Qwen3.5-9B on mathematical reasoning and code generation.

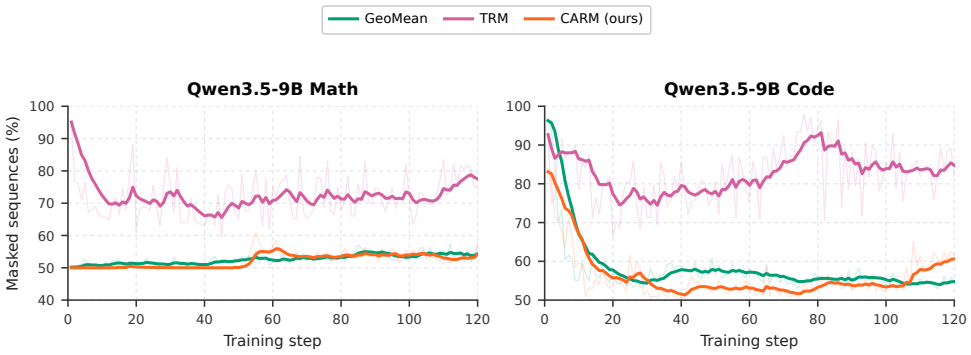


Figure 9: Sequence-masked fraction during training for the selected GeoMean, TRM, and CARM (ours) configurations. The left and right panels show Qwen3.5-9B mathematical-reasoning and code-generation training, respectively. Faint curves are raw measurements and opaque curves apply an exponential moving average with weight 0.8.

Figure 10 complements validation accuracy with generation- and optimization-level measurements. It compares the same five methods on Qwen3.5-9B mathematical-reasoning training. Faint solid curves show raw measurements, while opaque solid curves show exponential moving averages with weight 0.8. Gradient norm uses a logarithmic axis.

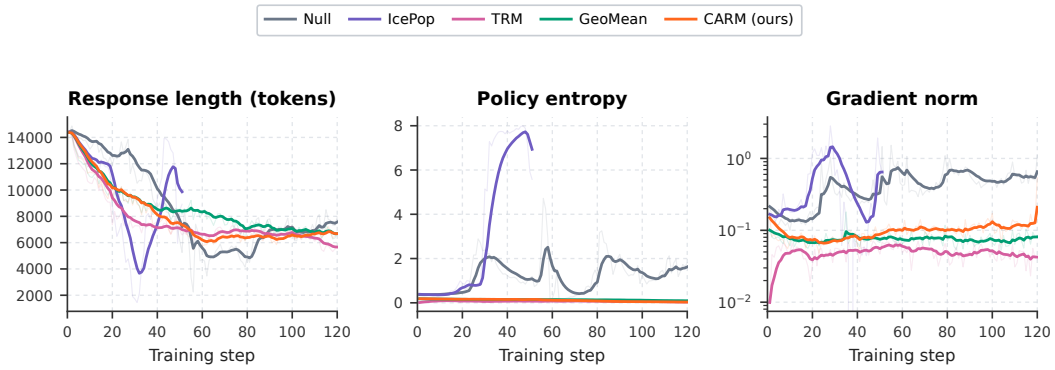


Figure 10: Training diagnostics for Qwen3.5-9B mathematical reasoning. Panels show response length, policy entropy, and gradient norm across the five methods. Faint curves show raw measurements; opaque curves show exponential moving averages with weight 0.8. Gradient norm is displayed on a logarithmic scale.

Figure 11 presents the matched diagnostics for code-generation training. The same metrics and smoothing convention make the domain-level differences directly comparable with Figure 10.

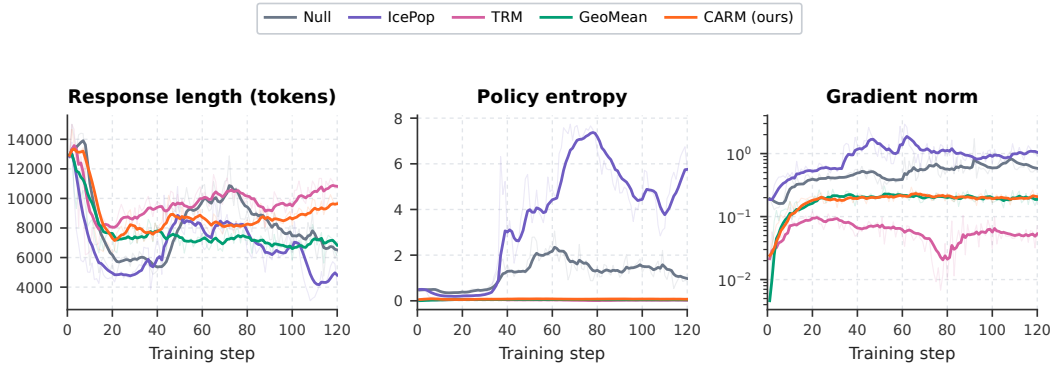


Figure 11: Code-generation training diagnostics: response length, policy entropy, and gradient norm for Qwen3.5-9B trained on TACO. Faint and opaque curves show raw and smoothed measurements, respectively, using the same smoothing and axis conventions as the mathematical-reasoning diagnostics.

G Best- and Worst-of-k Sampling Metrics

Figure 6 evaluates inference-time sample scaling at the same Qwen3.5-9B checkpoints reported in Table 5. For each problem, let $X_1, \ldots, X_{16} \in \{0, 1\}$ denote the correctness of the 16 independently sampled responses and let $c = \sum_{j=1}^{16} X_j$. The per-problem empirical accuracy is $c/16$, and `mean@16` averages this quantity over problems and then equally over AIME 2024, 2025, and 2026.

Best-of-k asks whether *at least one* of k sampled responses is correct. It captures the opportunity provided by additional inference-time sampling when an oracle verifier can select a correct candidate. Worst-of-k asks whether *all* k sampled responses are correct. It is therefore a strict measure of repeatability: a model scores highly only when its stochastic samples solve the same problem consistently. Worst-of-k should not be read as robustness to distribution shift or adversarial pertur-

bations. At $k = 1$, both statistics reduce exactly to `mean@16`, which anchors the two curves at the same observed single-sample accuracy.

For $k \in \{2, 4, 8, 16\}$, we resample k responses with replacement from the 16 available responses for each problem and apply the maximum (best-of-k) or minimum (worst-of-k) correctness operator. The plotted values average 1,000 bootstrap replicates, using the same models as Table 1.

For a problem with c correct responses among 16, the expected best-of-k value is

$$\widehat{B}_k(c) = 1 - \left(1 - \frac{c}{16}\right)^k. \tag{73}$$

Equation 73 is the probability that at least one of k independent draws from the empirical response distribution is correct. It is a with-replacement best-of-k statistic, not the standard without-replacement pass@k estimator. The corresponding expected worst-of-k value is

$$\widehat{W}_k(c) = \left(\frac{c}{16}\right)^k. \tag{74}$$

Equation 74 is the probability that all k draws are correct. It is non-increasing in k, and strictly decreasing for a problem when $0 < c < 16$. As with best-of-k, this is a with-replacement statistic rather than the literal maximum or minimum over a fixed subset of the original 16 responses. Together, the two curves characterize solution coverage and repeatability. Worst-of-k emphasizes problems with high per-problem correctness probability; an aggregate increase does not imply improvement on every problem.

Table 5: Exact Qwen3.5-9B values underlying Figure 6. All entries are equal-weighted three-year AIME averages in percent.

Metric	Method	$k = 1$	$k = 2$	$k = 4$	$k = 8$	$k = 16$
Best-of-k	Null	70.42	78.07	83.34	86.42	88.26
	TRM	73.96	<u>81.47</u>	<u>86.43</u>	<u>89.39</u>	<u>90.98</u>
	GeoMean	<u>74.03</u>	80.19	84.05	86.73	88.54
	CARM (ours)	**77.50**	**83.94**	**87.74**	**90.05**	**91.28**
Worst-of-k	Null	70.42	62.70	55.30	49.04	44.38
	TRM	73.96	66.49	59.27	52.99	48.00
	GeoMean	<u>74.03</u>	<u>67.66</u>	<u>61.32</u>	<u>55.02</u>	<u>49.83</u>
	CARM (ours)	**77.50**	**71.04**	**64.44**	**58.05**	**52.61**