DEPO: MITIGATING POLICY ENTROPY COLLAPSE VIA DENOISED POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has become a central tool for improving large language model reasoning, but rapid policy entropy decay can restrict exploration during training. We propose Denoised Policy Optimization (DePO), which filters importance sampling ratios along token positions and uses the filtered values as detached weights in a policy update with sign-dependent clipping gates. We study sliding-window averaging, exponentially weighted averaging, and Savitzky–Golay filtering as simple ways to control local ratio fluctuations. Building on prior analyses of entropy dynamics, we derive the detached-weight gradient and characterize noise reduction and smoothing bias under explicit assumptions. Experiments on mathematical reasoning and code generation cover three model backbones. On Qwen3-4B, DePO improves the reported average math accuracy from 65.5 to 68.4 and code accuracy from 39.0 to 42.3, while retaining higher final policy entropy than GRPO. Window studies, generation statistics, and a multi-turn search evaluation complement filter ablations and clipping-relaxation stress tests. These results support ratio filtering as a practical stabilization method; its effects on individual reasoning decisions remain to be established.

1 INTRODUCTION

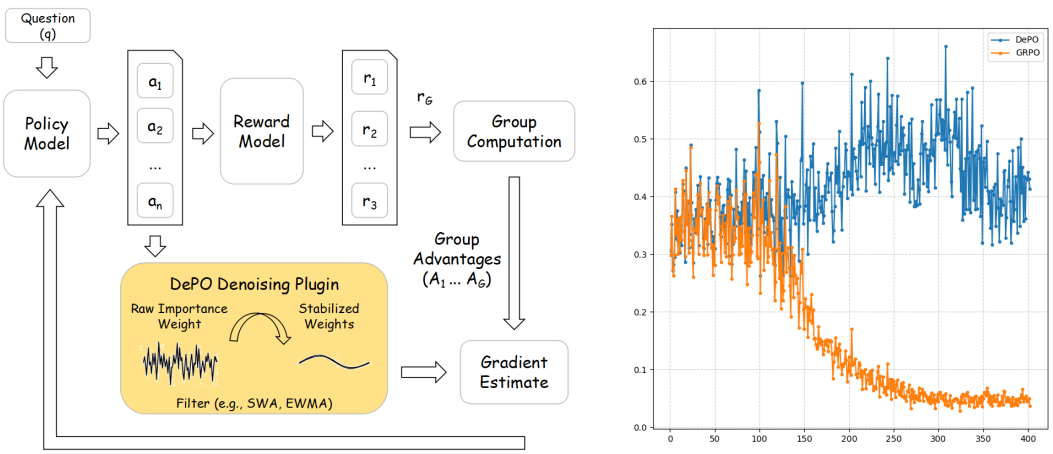
Large language models (LLMs) increasingly solve mathematical problems and generate code through multi-step reasoning (OpenAI, 2024; DeepSeek-AI, 2025). Reinforcement learning (RL) is a central component of this progress (Ouyang et al., 2022; Shao et al., 2024), but training can exhibit a rapid decline in policy entropy (Yu et al., 2025; He et al., 2025). Some concentration of probability is necessary for learning; premature concentration can limit the alternative responses explored by the policy. The practical goal is therefore to improve task performance while avoiding unstable or prematurely concentrated updates.

Existing approaches control different aspects of this process. Entropy regularization adds an explicit preference for less concentrated policies (Haarnoja et al., 2018); selective methods target particular tokens or decisions (Zheng et al., 2025b; Wang et al., 2025; Jiang et al., 2025). EPO studies entropy regularization in multi-turn agent training (Xu et al., 2025). Building on the entropy analysis of Cui et al. (2025), covariance-based methods instead constrain updates associated with large covariance contributions. These approaches motivate complementary ways of stabilizing training, each with its own tuning and approximation choices.

DePO takes a signal-processing perspective on the *importance-ratio sequence*. Small old-policy probabilities can cause large ratios: old/current probabilities $(0.45, 0.58)$ give a ratio of approximately 1.29, whereas $(0.001, 0.05)$ give 50. This numerical illustration is not a measured token-level case. We hypothesize that suppressing some local ratio fluctuations can improve optimization stability, without identifying all sharp changes as noise: a decisive mathematical sign or code operator can also carry a useful localized signal. Filtering therefore introduces a trade-off between smoothing and responsiveness.

We instantiate this idea using Sliding Window Average (SWA), Exponentially Weighted Moving Average (EWMA), and Savitzky–Golay filtering (SGF). The operation takes place after a response has been generated and does not add an inference-time module (Figure 1). Under an outcome reward, all tokens in a response share the same advantage. DePO changes the ratio signal and its role in the

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.



(a) DePO framework

(b) Policy entropy over training steps

Figure 1: **DePO framework and observed entropy dynamics.** Left: ratios along each generated response are filtered and detached before the policy-gradient update; the reward block denotes the rule-based task evaluator. Right: the Qwen3-4B trajectory shows higher entropy for DePO than GRPO (horizontal axis: training step; vertical axis: policy entropy). The curve illustrates the observed training dynamics.

objective; it does not temporally decorrelate a constant advantage or modify the reward assigned to a fixed response.

The reported experiments cover mathematical reasoning and code generation with Qwen3-1.7B, Qwen3-4B, and LLaMA-3.2-3B-Instruct. DePO has the highest average task scores among the compared methods in each model block. On Qwen3-4B, the math improvement over GRPO is 2.9 percentage points, or approximately 4.4% relative. Generation statistics and reasoning examples characterize the resulting behavior, while window studies and multi-turn search extend the evaluation. Higher entropy accompanies the gains, but entropy and truncation alone do not establish a distinct semantic form of exploration.

Our contributions are threefold: (i) a simple ratio-filtering framework with three kernels; (ii) an explicit account of the filtered objective, conditional noise reduction, and smoothing bias, including the limits of these statements; and (iii) an empirical comparison of task accuracy, training dynamics, and filter choices. We interpret clipping removal as a stress test under a specified setup, rather than as evidence that filtering generally replaces trust-region protection.

2 Methodology

2.1 Ratio Filtering with Detached Policy Weights

We build on group-relative policy optimization (Shao et al., 2024). For a prompt x, the old policy π_{old} samples G responses y_i of valid response lengths L_i. The outcome reward yields a group-normalized advantage A_i, shared by all response tokens. At the current parameter snapshot $\bar{\theta}$, define

$$\rho_{i,t} = \frac{\pi_{\bar{\theta}}(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})}, \qquad \widetilde{\boldsymbol{\rho}}_i = \mathcal{F}_\phi(\boldsymbol{\rho}_i). \tag{1}$$

Here filtering is along token positions within a response, not across successive optimization iterations. The filter does not change the outcome reward or recompute token-specific advantages.

Clipping gate. Following the sign-dependent active branches of PPO clipping (Schulman et al., 2017), let

$$m_{i,t} = \begin{cases} \mathbf{1}\{\widetilde{\rho}_{i,t} \le 1+\epsilon\}, & A_i \ge 0, \\ \mathbf{1}\{\widetilde{\rho}_{i,t} \ge 1-\epsilon\}, & A_i < 0. \end{cases} \tag{2}$$

Away from the clipping boundaries, this is the branch indicator of $\min(rA_i, \mathrm{clip}(r, 1-\epsilon, 1+\epsilon)A_i)$, evaluated at $r = \widetilde{\rho}_{i,t}$. Boundary subgradients do not affect the derivations below. When clipping is disabled, $m_{i,t} = 1$.

Detached-weight objective. Let $\mathrm{sg}[\cdot]$ denote stop-gradient. DePO maximizes the local weighted log-likelihood surrogate

$$J_{\mathrm{DePO}}(\theta; \bar{\theta}) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{t=1}^{L_i}\mathrm{sg}[m_{i,t}\widetilde{\rho}_{i,t}]\, A_i \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\right]. \tag{3}$$

The ratios, filter output, and gate are evaluated at the current snapshot and held fixed while differentiating the log-probability term. The weights are recomputed at the next optimizer step. Equivalently, a minimization implementation uses $\mathcal{L}_{\mathrm{DePO}} = -J_{\mathrm{DePO}}$. With $u_{i,t} = \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})|_{\theta=\bar{\theta}}$, the per-group gradient is

$$g_{\mathrm{DePO}} = \frac{1}{G}\sum_i \frac{1}{L_i}\sum_t m_{i,t}\widetilde{\rho}_{i,t} A_i u_{i,t}. \tag{4}$$

With the identity filter, Equation (4) recovers the clipped PPO gradient at the same parameter snapshot. For a nontrivial filter, no gradient is propagated through $\mathcal{F}_\phi$. The gate preserves the sign-dependent clipping decision, but Equation (3) is not the result of differentiating a clipped scalar objective through a linear filter. Appendix A.2 makes this distinction explicit. A constant advantage does not cancel the effect of filtering because the token score vectors $u_{i,t}$ differ.

2.2 Three Filtering Kernels

Sliding Window Average (SWA). Let h denote a radius and $w = 2h+1$ its full width. In the interior of a response,

$$\widetilde{\rho}_t = \frac{1}{2h+1}\sum_{j=-h}^{h}\rho_{t+j}. \tag{5}$$

SWA suppresses isolated ratio variations by averaging neighboring positions. This is a local smoothness prior, not a test of whether a token is semantically important. We distinguish h from w because the experimental records use "window size" without consistently specifying this convention.

Exponentially Weighted Moving Average (EWMA). A one-sided recurrence gives

$$\widetilde{\rho}_t = \alpha\rho_t + (1-\alpha)\widetilde{\rho}_{t-1}, \qquad 0 < \alpha \le 1. \tag{6}$$

This kernel is causal in token position. It should not be described as the same bidirectional operation as centered SWA. The ablation uses reported $\alpha = 0.3$; the historical initialization is not specified in the available configuration.

Savitzky–Golay Filter (SGF). SGF fits a polynomial of degree p over an odd-width window and evaluates the fit at its center (Savitzky & Golay, 1964):

$$\widetilde{\rho}_t = \sum_{j=-h}^{h} c_j \rho_{t+j}, \qquad \mathbf{c}^T = \mathbf{e}_0^T (V^T V)^{-1} V^T, \quad V_{j,q} = j^q \ (q = 0, \ldots, p). \tag{7}$$

For $p = 2, w = 5$, the interior coefficients are $(-3, 12, 17, 12, -3)/35$. Polynomial reproduction does not guarantee preservation of arbitrary extrema or reasoning decisions. Unlike SWA, SGF can have negative coefficients and can produce negative outputs even for positive input ratios. Any non-negativity correction changes the effective filter and must be reported; it is not specified in the current experimental record.

2.3 Scope of the Filtering Operation

The optimization stage has access to each completed response, so centered SWA and SGF can use both preceding and following tokens without changing autoregressive generation. The original configuration specifies reflect padding at boundaries. Exact endpoint conventions and the interaction with response masks remain necessary implementation details (Appendix B); filtering must be defined on the intended response rather than silently mixing prompts, padding, or separate samples. Because the weights are detached, smoothing affects the update coefficients, not the differentiation path through neighboring ratios. Algorithm 1 summarizes this computation.

3 Analysis and Its Scope

3.1 Entropy Dynamics as Motivation

For a softmax policy at a fixed state, the first-order entropy change under logit perturbation Δz is

$$\Delta H = -\operatorname{Cov}_{a\sim\pi}\big(\log \pi(a \mid s), \Delta z_a\big) + O(\|\Delta z\|^2). \tag{8}$$

This identity, used in prior entropy analyses (Cui et al., 2025), motivates examining unstable policy updates. Its covariance is over actions at a fixed state; it is not a temporal covariance with the response-constant advantage. Moreover, a parameter update couples many logits, so a change in ratio variance cannot be substituted directly for Δz in Equation (8). The derivation is included in Appendix A.1.

3.2 Conditional Noise Reduction and Signal Bias

Proposition 1 (Noise variance under a fixed filter)**.** *Suppose locally $\rho_t = s_t + \eta_t$, with deterministic signal s_t and zero-mean noise whose components are uncorrelated and have variance σ^2. For deterministic filter weights w_j, the filtered noise satisfies*

$$\operatorname{Var}\left(\sum_j w_j \eta_{t+j}\right) = \sigma^2 \sum_j w_j^2. \tag{9}$$

It is strictly reduced when $\sum_j w_j^2 < 1$.

For interior SWA of width $w > 1$, the variance factor is $1/w$. With correlated noise, the expression becomes $\mathbf{w}^T \Sigma \mathbf{w}$, so reduction is not automatic. Normalization $\sum_j w_j = 1$ alone is also insufficient. These are conditional properties of the ratio-noise component, not a claim that every filter reduces the full policy-gradient variance.

Filtering may alter the useful signal as well. If s admits a twice-differentiable local interpolation, $|s''| \le M$, and the kernel satisfies $\sum_j w_j = 1$ and $\sum_j j w_j = 0$, Taylor's theorem gives

$$\left|\sum_j w_j s(t+j) - s(t)\right| \le \frac{M}{2} \sum_j |w_j| j^2. \tag{10}$$

The cancellation of the first-order term holds under these moment conditions; it need not hold for one-sided EWMA or modified boundary weights. The bound grows with the window's second moment. Local curvature is an assumption to be examined, not a quantity established as small by linguistic coherence. Appendix A.3 provides the proofs and explains the additional assumptions required to transfer such bounds to gradients.

3.3 What Does and Does Not Follow for Policy Updates

For one response with clipping disabled, write the finite filter as W and form a matrix X with rows $A_t u_t^T$. Equation (4) then becomes

$$g_W = \frac{1}{L} X^T W \boldsymbol{\rho}, \qquad g_I = \frac{1}{L} X^T \boldsymbol{\rho}. \tag{11}$$

Even if W preserves the sum of ratios, it need not preserve this weighted gradient. Thus, detached ratio filtering can change the update with a constant outcome advantage and without clipping. Neither this rearrangement nor kernel normalization makes g_W unbiased for the original policy gradient. Its bias depends on the joint behavior of ratios, advantages, and token score vectors.

For fixed X and isotropic additive noise, the variance of a gradient projection in direction v is $\sigma^2 \|W^T X v\|^2 / L^2$. A contraction condition on W can provide a non-increase in this conditional model; strict reduction requires additional structure. Random, correlated score vectors and clipping gates require further analysis. Likewise, a smoothed ratio at one token is not generally the ratio of a normalized policy at that state. We therefore make no inference of a smaller policy KL or monotonic improvement from Jensen's inequality or Equation (9). The empirical studies evaluate whether the resulting bias and noise trade-off is useful in the reported settings.

4 EXPERIMENTS

We evaluate task accuracy, entropy dynamics, computational cost, and sensitivity to filtering and clipping. Beyond the three-model math/code comparison, we examine generation length and truncation, fixed and adaptive windows, and multi-turn search reasoning. Appendix C provides additional curves and protocol details.

Table 1: Main results on Mathematical Reasoning (Mean@32) and Code Generation (Mean@16). `Ent.` denotes the final policy entropy, and `Time` indicates the average training time per step (s). Best task scores within each model block are bolded. Entropy is not a task score.

Mathematical Reasoning (Mean@32)

Model	Method	AIME24	AIME25	MATH500	AMC24	Avg.
Qwen3-1.7B	GRPO	60.1	41.5	78.9	53.5	58.5
	Ent-Reg	55.7	40.9	77.6	52.6	56.7
	Clip-Cov	58.3	42.1	81.2	50.3	58.0
	DAPO	61.2	44.7	80.4	66.7	63.3
	SIREN	60.4	45.4	79.6	66.1	62.9
	DePO (Ours)	**62.5**	**46.5**	**83.2**	**68.4**	**65.2**
Qwen3-4B	GRPO	63.5	48.1	82.0	68.2	65.5
	Ent-Reg	62.8	47.5	81.4	67.5	64.8
	Clip-Cov	64.0	48.9	82.8	69.0	66.2
	DAPO	65.2	50.1	83.5	70.1	67.2
	SIREN	65.8	50.5	84.1	70.5	67.7
	DePO (Ours)	**66.4**	**51.2**	**85.0**	**71.1**	**68.4**
LLaMA-3.2-3B	GRPO	56.2	44.1	80.5	67.9	62.2
	Ent-Reg	55.4	43.5	79.8	67.1	61.5
	Clip-Cov	57.0	45.0	81.3	68.5	63.0
	DAPO	57.8	46.2	81.9	69.2	63.8
	SIREN	58.1	46.8	82.5	69.8	64.3
	DePO (Ours)	**58.6**	**47.3**	**83.1**	**70.2**	**64.8**

Code Generation (Mean@16) and Statistics

Model	Method	LCB	BigCode	MBPP	Avg.	Ent.	Time
Qwen3-1.7B	GRPO	24.1	30.8	55.4	36.8	0.08	60
	Ent-Reg	23.5	26.9	52.8	34.4	0.51	61
	Clip-Cov	23.0	29.5	53.1	35.2	0.22	63
	DAPO	24.8	27.2	54.0	35.3	0.13	76
	SIREN	24.5	30.1	53.5	36.0	0.36	60
	DePO (Ours)	**25.5**	**32.2**	**57.1**	**38.3**	0.44	65
Qwen3-4B	GRPO	25.4	33.8	57.8	39.0	0.06	86
	Ent-Reg	25.1	34.7	55.9	38.6	0.33	88
	Clip-Cov	26.2	35.2	57.2	39.5	0.16	90
	DAPO	27.4	36.1	58.3	40.6	0.09	121
	SIREN	26.6	35.5	60.1	40.7	0.31	95
	DePO (Ours)	**28.3**	**37.4**	**61.3**	**42.3**	0.30	97
LLaMA-3.2-3B	GRPO	23.3	31.8	54.3	36.5	0.28	75
	Ent-Reg	22.7	28.6	52.1	34.5	0.55	77
	Clip-Cov	25.8	31.4	55.3	37.5	0.47	90
	DAPO	26.3	33.5	57.4	39.1	0.36	92
	SIREN	25.2	32.1	58.2	38.5	0.50	74
	DePO (Ours)	**27.8**	**35.7**	**59.5**	**41.0**	0.51	78

4.1 Experimental Setup

Tasks and metrics. Mathematical reasoning uses the DAPO-Math-17K training corpus (Yu et al., 2025), with evaluation on AIME24, AIME25, MATH500, and AMC24. Code generation uses AceCode-87K from AceCoder (Zeng et al., 2025), with evaluation on LiveCodeBench (LCB), BigCodeBench, and MBPP. Mean@32 for math and Mean@16 for code denote the mean success rate over the indicated number of sampled responses, averaged over evaluation problems. They are not pass@k (at least one successful response), and multiple responses are not independent training seeds. Within each domain, `Avg.` is the unweighted mean of its benchmark scores. The additional search experiment evaluates Qwen3-1.7B on 2wiki and Bamboogle.

Models and baselines. We use Qwen3-1.7B, Qwen3-4B, and LLaMA-3.2-3B-Instruct. Comparisons include GRPO (Shao et al., 2024), entropy regularization (Ent-Reg), Clip-Cov (Cui et al., 2025), DAPO (Yu et al., 2025), and SIREN (Jiang et al., 2025). Ent-Reg adds $\beta_H H(\pi)$ to the maximization objective, with reported coefficient $\beta_H = 0.01$, distinct from the KL coefficient. SWA is the default DePO kernel, with reported window parameter 5. Its radius/full-width convention remains to be verified (Appendix B).

Reporting scope. Appendix B records the task-specific configurations, including zero KL coefficient for math. Independent training seeds, confidence intervals, and some evaluation and decoding settings are not specified in the available records. We interpret all tables as descriptive point estimates, without claims of statistical significance; this particularly limits conclusions about small differences between filters. The additional studies retain their own model and metric settings.

4.2 Task Performance and Computational Cost

Table 1 reports all model/method comparisons. DePO achieves the highest reported average scores in each model block. On Qwen3-4B, math accuracy rises from 65.5 to 68.4 over GRPO, a 2.9 percentage-point gain (approximately 4.4% relative); code accuracy rises from 39.0 to 42.3. Relative to DAPO, the math gain is 1.2 points; relative to SIREN, the code gain is 1.6 points.

Across Qwen3-1.7B, Qwen3-4B, and LLaMA-3.2-3B, the math gains over GRPO are 6.7, 2.9, and 2.6 points; code gains are 1.5, 3.3, and 4.5 points. Ent-Reg retains more entropy than DePO in these model blocks but has lower task accuracy. Higher entropy alone therefore does not account for the performance differences, nor do aggregate scores identify which individual tokens benefited from filtering.

Efficiency. GRPO versus DePO takes 60 versus 65, 86 versus 97, and 75 versus 78 seconds per step across the three models. The overheads are 8.3%, 12.8%, and 4.0%, with an unweighted mean of approximately 8.4%. These latency measurements do not imply a uniform overhead below 10%. Filtering acts on scalar ratio sequences: direct window evaluation costs $O(Lw)$, whereas cumulative-sum SWA and recurrent EWMA can use $O(L)$ arithmetic. Actual cost depends on implementation and hardware.

4.3 Entropy, Generation Length, and Reasoning Examples

Figures 1 and 2 show that DePO retains higher entropy than GRPO alongside improving validation accuracy. Ent-Reg also maintains entropy, but its accuracy falls later in the displayed run. We examine generation behavior to distinguish these training trajectories beyond entropy alone.

Table 2: Qwen3-4B generation statistics. Length is in tokens; truncation is the fraction reaching the configured limit without EOS. The aggregation window is not specified.

Method	Math avg. (%)	Final entropy	Mean length	Truncation (%)
GRPO	65.5	0.06	3014	7.5
Ent-Reg	64.8	0.33	3486	15.6
DePO	68.4	0.30	3031	6.2

Table 2 reports mean lengths of 3014, 3486, and 3031 tokens for GRPO, Ent-Reg, and DePO, with truncation rates of 7.5%, 15.6%, and 6.2%. Truncation means reaching the configured generation limit



Figure 2: Qwen3-4B math training traces. Top: validation accuracy. Bottom: policy entropy (axis labeled "Value"). Horizontal axes denote training step. These trajectories are distinct from the final statistics: their corresponding checkpoint and aggregation window are not specified in the experiment records.

without EOS. DePO's length is close to GRPO's while its truncation rate is lower by 1.3 points; Ent-Reg is longer and more frequently truncated. This supports a difference in termination behavior, not a general preference for shorter responses. Long or truncated reasoning is not necessarily incoherent, and these statistics do not isolate a causal effect of the entropy coefficient.

Illustrative reasoning case. Table 3 compares two recorded responses to the same digit problem. Ent-Reg enumerates possible tens digits after reaching $9x = xy$, whereas DePO uses the nonzero tens digit to conclude $b = 9$ directly. Displaying both excerpts makes the difference in solution style explicit while retaining their common correct answer.

Table 3: Shortened reasoning excerpts for $N = P(N) + S(N)$, where N is a two-digit number and P, S are the product and sum of its digits. The question asks for the units digit. Both methods answer correctly; ellipses mark omissions in the source record.

Ent-Reg ($H \approx 0.48$)	**DePO** ($H \approx 0.30$)
"… $10x + y = xy + x + y \Rightarrow 9x = xy$. What if $x = 1$? $N = 19$, $P(19) + S(19) = 19$. Matches. What if $x = 2$? $N = 29$, $P(29) + S(29) = 29$. Matches. … [enumerates $x = 3$ to 5] … The units digit is always 9. $\boxed{9}$"	"… $10a + b = ab + a + b \Rightarrow 9a = ab$. Since N is a two-digit number, $a \neq 0$. Dividing both sides by a gives $b = 9$. The units digit is 9. $\boxed{9}$"

The example entropies, approximately 0.48 and 0.30, characterize these excerpts rather than the aggregate statistics in Table 2. The pair does not establish an infinite loop, EOS avoidance, or a population-level improvement in reasoning quality: complete rollouts and selection criteria are unavailable. Together, the generation statistics and case support examining termination behavior and reasoning efficiency separately from final correctness; a token-level diagnosis of which useful updates survive smoothing remains open.

4.4 Clipping-Relaxation Stress Test

Table 4 reports Qwen3-4B AIME25 accuracy as clipping is relaxed. The math configuration has KL coefficient zero; No Clip additionally removes clipping. This is a stress test under reduced controls on policy change.

Table 4: Robustness to clipping relaxation (AIME25 Acc.).

Method	$\epsilon = 0.2$	$\epsilon = 0.5$	$\epsilon = 1.0$	**No Clip** (∞)
GRPO	48.1	45.5	12.4	Diverged
DAPO	50.1	48.3	30.1	25.1
DePO	**51.2**	**50.0**	**50.8**	**50.2**

DePO ranges from 50.0 to 51.2, including 50.2 without clipping. GRPO is reported as diverging without clipping, while DAPO scores 25.1. These observations indicate reduced sensitivity in the tested setup, not a KL trust region, monotonic improvement, or superiority to baselines with tuned KL protection. Detachment remains active without clipping, so the filtered weights still modify the update. Appendix C.5 provides the corresponding curves.

4.5 Kernels, Window Sensitivity, and Adaptive Filtering

Table 5: Ablation of filtering kernels on Qwen3-4B.

Kernel	**Math Avg.**	**Code Avg.**	**Final Ent.**
SWA (Default)	68.4	42.3	0.30
EWMA ($\alpha = 0.3$)	68.2	41.5	0.32
SGF (p=2, w=5)	67.5	42.9	0.37

Table 5 compares the three kernels on Qwen3-4B. SWA has the highest reported math average (68.4), while SGF has the highest code average (42.9); all three exceed the GRPO averages. Similar aggregate scores do not establish statistical equivalence or prove that rare, useful token-level signals are preserved.

Table 6: Fixed and adaptive SWA windows in two distinct model settings. Accuracy is in percent. Window values retain the reported convention; the two panels are not pooled.

(a) Qwen3-1.7B: fixed windows

Window	Math avg.	Ent.
10	64.7	0.43
100	65.0	0.39
500	65.3	0.40

(b) Qwen3-4B: adaptive versus fixed

Method	Math avg.	Code avg.	Ent.
GRPO	65.5	39.0	0.06
Fixed SWA (5)	68.4	42.3	0.30
Adaptive SWA (1%)	68.2	42.7	0.32

Fixed windows. Table 6a varies the reported SWA window parameter from 10 to 500 on Qwen3-1.7B. Math accuracy spans 64.7–65.3, a 0.6-point range, while final entropy spans 0.39–0.43. This provides evidence of limited task-level sensitivity over the tested range, without a significance test or token-level diagnosis. The separate Qwen2.5-1.5B/GSM8K study (windows 5, 50, and 500) remains in Appendix C.2; its results are not pooled with this experiment.

Adaptive windows. On Qwen3-4B, the adaptive SWA variant scales the window parameter with approximately 1% of response length (Table 6b). It yields 68.2 math accuracy, 42.7 code accuracy, and 0.32 entropy, versus 68.4, 42.3, and 0.30 for fixed SWA. Thus, the adaptive setting trades a 0.2-point math decrease for a 0.4-point code increase. These results show that a length-dependent variant was evaluated, without establishing consistent superiority. Exact rounding and width conventions are among the configuration details in Appendix C.

4.6 Multi-Turn Search Reasoning

The additional experiment follows a Search-R1-style interaction pattern (Jin et al., 2025), interleaving reasoning, search queries, and retrieved context. This tests applicability beyond standalone math and code responses.

Table 7: Multi-turn search reasoning with Qwen3-1.7B.

Method	2wiki accuracy (%)	Bamboogle accuracy (%)	Final entropy
GRPO	26.5	22.7	0.05
DePO	29.2	24.4	0.37

On Qwen3-1.7B, Table 7 reports gains of 2.7 points on 2wiki and 1.7 points on Bamboogle, with final entropy 0.37 rather than 0.05. The results extend the empirical evidence to multi-turn search, but do not establish which retrieval decisions improved. Search/retriever settings, splits, rollout budget, and retrieval-token masks are not fully specified in the available records (Appendix C); in particular, no smoothing across retrieved and generated tokens is assumed here.

5 Discussion and Conclusion

DePO filters token-level importance ratios and uses the detached outputs to weight policy updates. The math/code comparisons, generation statistics, window studies, and multi-turn search results support its practical use for stabilizing training in the evaluated settings. Our analysis characterizes conditional noise reduction and smoothing bias; it does not establish a smaller policy KL or monotonic improvement.

The main limitations are possible attenuation of useful sharp updates, sensitivity to filter and boundary choices, and incomplete run-level evidence for uncertainty and evaluation settings. Entropy, truncation, and individual examples do not certify semantic exploration quality. Resolving implementation details and measuring effects on important token-level decisions are priorities for further evaluation.

Reproducibility Statement

The objective and filter definitions are given in Section 2; conditional results and their derivations appear in Section 3 and Appendix A. Section 4 presents the main comparison and additional generation, window, and search studies. Appendices B and C record available settings, missing reproduction details, and example-selection limitations. The tables report existing experimental records; this revision does not constitute an independent rerun.

Impact Statement

This work studies optimization for language-model reasoning. Improved benchmark performance does not establish reliability or safety outside the evaluated tasks. The method does not itself address misuse or deployment safeguards.

AI Use Disclosure

Generative AI was used for language polishing. All content was reviewed and verified by the authors, who take full responsibility for the paper.

REFERENCES

Ganqu Cui et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pp. 1861–1870. PMLR, 2018.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 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. *arXiv preprint arXiv:2103.03874*, 2021.

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. *arXiv preprint arXiv:2403.07974*, 2024.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

OpenAI. Openai o1 system card. *OpenAI Technical Report*, 2024. URL https://openai.com/index/openai-o1-system-card/.

Long Ouyang et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D. Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=7pufO0SJAC.

Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 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, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

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

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

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning, 2025. URL https://arxiv.org/abs/2509.22576.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. Acecoder: Acing coder rl via automated test-case synthesis. *arXiv preprint arXiv:2502.01718*, 2025.

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

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return, entropy-eliciting explore, 2025b. URL https://arxiv.org/abs/2507.07017.

Brian D Ziebart et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438, 2008.

A Derivations and Implementation-Dependent Bounds

A.1 Entropy Identity

Following the softmax entropy analysis underlying Cui et al. (2025), fix a state and write $\pi_a = \exp z_a / \sum_b \exp z_b$ and $H = -\sum_a \pi_a \log \pi_a$. The softmax derivative gives

$$\frac{\partial H}{\partial z_a} = -\sum_b \pi_b(\mathbf{1}\{a=b\} - \pi_a)(1 + \log \pi_b) \tag{12}$$

$$= -\pi_a(\log \pi_a + H). \tag{13}$$

Consequently,

$$\Delta H = -\sum_a \pi_a(\log \pi_a + H)\Delta z_a + O(\|\Delta z\|^2) \tag{14}$$

$$= -\mathbb{E}_\pi[\log \pi_a \, \Delta z_a] - H\mathbb{E}_\pi[\Delta z_a] + O(\|\Delta z\|^2) \tag{15}$$

$$= -\operatorname{Cov}_\pi(\log \pi_a, \Delta z_a) + O(\|\Delta z\|^2). \tag{16}$$

The final equality uses $\mathbb{E}_\pi[\log \pi_a] = -H$. This is a local identity about action probabilities and logit changes. It does not assert that a sequence filter necessarily reduces the covariance, and the entropy mechanism is prior work rather than a new contribution of DePO.

A.2 Detached-Weight Gradient and Clipping Branches

For a sampled response, write $r_t = (W\rho)_t$ and m_t as in Equation (2). At snapshot $\bar{\theta}$, the local surrogate is $J(\theta;\bar{\theta}) = L^{-1}\sum_t \operatorname{sg}[m_t r_t] A_t \log \pi_\theta(y_t \mid x, y_{<t})$. Since neither m_t nor r_t is differentiated,

$$\nabla_\theta J(\theta;\bar{\theta})\big|_{\bar{\theta}} = \frac{1}{L}\sum_t m_t r_t A_t u_t. \tag{17}$$

For $A_t > 0$, the upper saturated branch $r_t > 1+\epsilon$ contributes zero; for $A_t < 0$, the lower saturated branch $r_t < 1-\epsilon$ contributes zero. The other branches contribute the detached weight times the score vector. When $A_t = 0$, the contribution vanishes regardless of the gate.

For comparison, directly differentiating $L^{-1}\sum_t \min(r_t A_t, \operatorname{clip}(r_t)A_t)$ through a fixed linear W would instead give

$$\frac{1}{L}\sum_j \rho_j u_j \left(\sum_t m_t A_t W_{tj}\right). \tag{18}$$

This is a different update and is not the detached-weight implementation used here. A sum-preserving filter and constant advantage would leave the corresponding *unclipped scalar sum of ratios* unchanged; that observation does not imply invariance of the detached weighted log-likelihood update.

With clipping disabled, the actual update can be rearranged exactly as

$$g_W = \frac{1}{L}\sum_t\sum_j W_{tj}\rho_j A_t u_t = \frac{1}{L}\sum_j \rho_j \left(\sum_t W_{tj} A_t u_t\right). \tag{19}$$

This identity holds for the finite matrix including boundary effects. It describes a weighted combination of token score vectors, not an unbiasedness theorem. For example, with constant A, $L = 2$, $\rho = (1,3)^T$, scalar scores $u = (1,-1)^T$, and $W = \frac{1}{2}\mathbf{1}\mathbf{1}^T$, the unfiltered gradient is $-A$, whereas the filtered gradient is zero. The example is algebraic, not a benchmark measurement.

A.3 Noise Variance, Signal Bias, and Gradient Scope

Proof of Proposition 1. By bilinearity of covariance,

$$\operatorname{Var}\left(\sum_j w_j \eta_{t+j}\right) = \sum_{j,k} w_j w_k \operatorname{Cov}(\eta_{t+j}, \eta_{t+k}) = \sigma^2 \sum_j w_j^2. \tag{20}$$

The last equality requires uncorrelated, equal-variance noise. For a general covariance matrix it becomes $\mathbf{w}^T\Sigma\mathbf{w}$. In particular, perfectly correlated noise is unchanged by weights summing to one. Row normalization alone does not ensure $\|\mathbf{w}\|_2 < 1$, especially with signed coefficients.

Signal-bias bound. Taylor's theorem gives $s(t+j) = s(t) + js'(t) + \frac{1}{2}j^2 s''(\xi_j)$ for an intermediate point ξ_j. With normalized weights and zero first moment,

$$b_t = \sum_j w_j s(t+j) - s(t) = \frac{1}{2}\sum_j w_j j^2 s''(\xi_j), \qquad |b_t| \le \frac{M}{2}\sum_j |w_j| j^2. \tag{21}$$

This is a bound on the smoothed signal, not a proof that gradient bias is negligible. If token scores and advantages are held fixed in the conditional model, the difference in expected unclipped updates is

$$\mathbb{E}[g_W - g_I \mid X] = \frac{1}{L} X^T (W - I)s. \tag{22}$$

It can be bounded by $L^{-1}\sum_t |A_t|\|u_t\||b_t|$ when the local conditions apply. Large curvature, large score norms, signed filters, or boundaries can make this bound uninformative. No local smoothness assumption is inferred solely from the fact that the tokens form a reasoning chain.

Conditional gradient variance. With X held fixed and $\mathrm{Cov}(\eta) = \sigma^2 I$, for any parameter-space vector v,

$$\mathrm{Var}(v^T g_W \mid X) = \frac{\sigma^2}{L^2}\|W^T X v\|_2^2. \tag{23}$$

If $\|W\|_2 \le 1$, this is no greater than the unfiltered value. Equality can hold, for example when Xv lies entirely in modes unchanged by the filter. This argument cannot be applied without qualification to random score vectors correlated with ratios, or to clipping gates that depend on the noisy ratio. It therefore does not establish a general strict reduction in policy-gradient variance.

Finite boundaries. For example, width-three averaging of a length-three response with reflection that excludes the endpoint has matrix

$$W = \frac{1}{3}\begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 1 \\ 0 & 2 & 1 \end{bmatrix}. \tag{24}$$

Its rows sum to one but its columns do not. Repeated reflected indices also change the effective weights multiplying the same noise variable. Thus, interior convolution arguments do not automatically describe the whole response. This example illustrates why endpoint and mask conventions must be included in a reproducible implementation; it does not assert that a particular library convention was used in the original runs.

SGF and policy constraints. For $p = 2, w = 5$, the coefficients sum to one and reproduce interior polynomials through degree two. Their squared sum is $17/35 < 1$, but their signs are mixed. Applying them to $(100, 1, 1, 1, 1)$ yields $-262/35 < 0$. Clamping such outputs is a nonlinear modification and requires a separate analysis. Even for nonnegative averaging weights, a bound on $\sum_t(\widetilde{\rho}_t - 1)^2$ is not a bound on the KL divergence of the updated policy: the smoothed quantity mixes different token states, while a policy likelihood ratio is defined over actions at one state. Trust-region performance results (Schulman et al., 2015) require assumptions and constraints not supplied by this smoothing inequality.

B Algorithm and Experimental Configuration

B.1 Algorithm

Algorithm 1 makes the detached gradient path and the sign-dependent clipping gate explicit. The reward, old-policy likelihoods, and group advantages are treated as fixed during optimization of a sampled batch. The precise number of optimization steps per rollout batch is a separate configuration item.

Algorithm 1 DePO with detached ratio weights

Require: Policy parameters θ, prompt distribution, filter $\mathcal{F}_\phi$, clipping threshold ϵ

for each rollout iteration **do**
  Snapshot the behavior policy $\pi_{\text{old}} \leftarrow \pi_\theta$
  Sample prompts and G responses per prompt with π_{old}
  Compute task rewards and group-normalized response advantages A_i
  Retain old log-probabilities and valid response lengths/masks
  for each optimization step on the sampled batch **do**
    Evaluate current response log-probabilities $\ell_{i,t} = \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})$
    Compute $\rho_{i,t} = \exp(\ell_{i,t} - \ell_{i,t}^{\text{old}})$
    for each response i **do**
      Compute $\widetilde{\boldsymbol{\rho}}_i = \mathcal{F}_\phi(\boldsymbol{\rho}_i)$ with specified boundaries
      Set $m_{i,t}$ using Equation (2); use $m_{i,t} = 1$ for No Clip
      Set detached weights $a_{i,t} = \text{sg}[m_{i,t}\widetilde{\rho}_{i,t}]A_i$
    end for
    Compute $\mathcal{L} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\frac{1}{L_i}\sum_{t=1}^{L_i} a_{i,t}\ell_{i,t}$
    Apply an optimizer step minimizing $\mathcal{L}$
  end for
end for

This pseudocode specifies the update rather than a library-specific implementation. In particular, it does not choose an undocumented window-width interpretation, EWMA initialization, SGF output correction, or padding/mask order. Such choices must match a runnable release before an exact reproduction can be claimed.

B.2 Task Configurations

The reported setup uses the `verl` framework, DAPO-Math-17K for math, and AceCode-87K for code. Math uses a rule-based binary correctness reward; code uses a rule-based test pass rate. The configuration records below explicitly cover Qwen3-1.7B/4B for math and Qwen3-4B for code. They do not establish complete per-model settings for every row of Table 1.

Table 8: Available task-level training settings. "Not recorded" identifies an absent value, not an assumed default.

Configuration	Math	Code
Models in configuration record	Qwen3-1.7B / 4B	Qwen3-4B
Optimizer	AdamW	AdamW
Learning rate	1×10^{-6}	5×10^{-7}
LR schedule	Linear decay; 20 warmup steps	Cosine decay
Training steps	Approximately 500	Not recorded
Rollout batch (prompts)	512	256
Responses per prompt	16	8
Training mini-batch size	512	128
KL coefficient	0.0	Not recorded
Reward	Binary $\{0, 1\}$	Test pass rate in $[0, 1]$
Maximum sequence length	8192	4096
Default filter	SWA	SWA
Reported window parameter	5	5

The method description specifies reflect padding for centered filtering. Equation (5) uses radius h and full width $w = 2h + 1$, while the experimental records label a parameter "window size $k = 5$." They do not resolve whether that 5 was a radius or a full width. The same ambiguity applies to the recorded window sweeps. We retain their reported labels and values instead of silently recoding them. For SGF, the stated $p = 2, w = 5$ is the polynomial-degree/full-width pair used to describe the standard interior kernel; the historical handling of boundaries and negative filtered values still needs verification.

Prompts. The math prompt is: "The following is a math problem. Please reason step by step and put your final answer within `\boxed{}`." The code prompt contains "You are an intelligent programming assistant," the question description, a request to complete the supplied function signature in a Python code block, and an assistant response marker, following AceCoder. Tokenization and the exact chat-template versions are needed for reproduction.

Hardware and timing. The recorded hardware is NVIDIA H800 with 80GB memory. Per-model GPU counts and the precise timing scope for Table 1 are not available in the accompanying configuration. A separate 7B timing example in the earlier description does not correspond to the main models and is not used to characterize their cost. The table supports the per-model time ratios reported in Section 4.2, not an unconditional overhead bound.

B.3 Remaining Reproduction Details

The detached weighting and sign-dependent saturation gate are explicit in Equations (2)–(4). Exact reproduction additionally requires the following items, which are not fully specified by the available experiment records: response-mask and padding order; endpoint conventions; the full-width interpretation of reported window parameters; EWMA initial conditions; SGF non-negativity handling; optimizer steps and mini-batch scheduling per rollout; per-model settings and seeds; evaluation set versions, decoding temperatures, and sample aggregation; and the checkpoint/averaging window for entropy and truncation statistics.

In particular, the math and code maximum sequence lengths in Table 8 should not be confused with an unspecified truncation threshold in a separate diagnostic run. Likewise, entropy values read from a training trace are not substituted for the final statistics in the main table. Resolving these details requires the original configuration and logs rather than inference from a plotted curve.

C Additional Experimental Details and Examples

The generation statistics, Qwen3-1.7B window sweep, adaptive-window results, and multi-turn search experiment appear in Section 4. This appendix records protocol limitations, the separate GSM8K window study, qualitative-example limitations, and additional training curves. These are existing experiment records, not independent reruns.

C.1 Protocol and Aggregation Details

Window studies. Table 6a reports average math accuracy for Qwen3-1.7B, without specifying the exact benchmark composition or per-seed uncertainty in its summary. Window values retain their reported parameter convention, which does not resolve radius versus full width. For the adaptive variant in Table 6b, the parameter scales with approximately 1% of response length; rounding, minimum width, and the radius/full-width interpretation of this percentage are not specified. No exact implementation of these rules is inferred from the aggregate results.

Multi-turn search. Table 7 follows a Search-R1-style pattern, but the record does not fully specify the search corpus/retriever, benchmark splits, rollout budget, retrieval-token masks, or filter boundaries across tool turns. The treatment of retrieved context must be established before reproducing the policy-generated token updates. The reported accuracy and entropy support potential applicability without identifying which part of the search interaction benefited.

Generation statistics. Table 2 uses truncation to denote reaching a generation-length limit without an EOS token. Its precise aggregation window and checkpoint are not specified. Entropy and lengths in the table are aggregate statistics; the example-level entropies below are separate measurements. The truncation trace in Figure 3 is not substituted for the table's reported rates.

C.2 Separate Window Study on GSM8K

The original window experiment uses Qwen2.5-1.5B on GSM8K (Table 9), whereas Table 6a uses Qwen3-1.7B and average math accuracy. Their model, metric, and tested parameter values differ. The GSM8K accuracy spans 1.3 percentage points and the Qwen3-1.7B math accuracy spans 0.6

points. These numerical ranges are not confidence intervals, and neither study determines whether some useful individual token updates were attenuated.

Table 9: Ablation of SWA window size on Qwen2.5-1.5B over the GSM8K benchmark. Window values use the original reporting convention.

Window Size	GSM8K Acc.	Final Ent.
5	80.4%	0.28
50	79.8%	0.23
500	79.1%	0.26

C.3 Truncation Trace

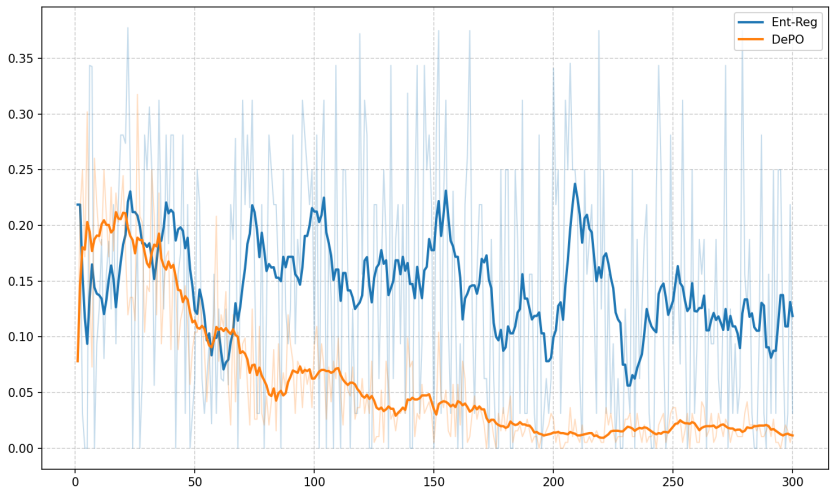


Figure 3: Truncation trace: horizontal axis is training step, vertical axis is truncation fraction. Its late-run values and the aggregation underlying Table 2 cannot be equated from the available logs.

The figure shows differing termination trajectories for Ent-Reg and DePO. Higher truncation can reflect long or unfinished reasoning as well as repetition; it does not establish that every long response is incoherent or that an entropy coefficient caused the entire difference.

C.4 Qualitative-Example Scope

Table 3 presents both author-reported excerpts in the main text. They were already shortened in the source: ellipses and the bracketed omission are retained. Complete token-level generations, checkpoints, selection criteria, and token-aligned importance ratios are not available. Consequently, the pair can illustrate enumeration versus direct algebraic deduction but cannot estimate how frequently either style occurs or whether the filter selectively preserves decisive reasoning tokens. Both excerpts reach the correct answer; their example-level entropy values are distinct from the aggregates in Table 2.

C.5 Clipping-Relaxation Curves

These curves supplement the tabulated stress test without adding an independent training seed. Their overlapping trajectories do not imply identical gradients or a general convergence guarantee.

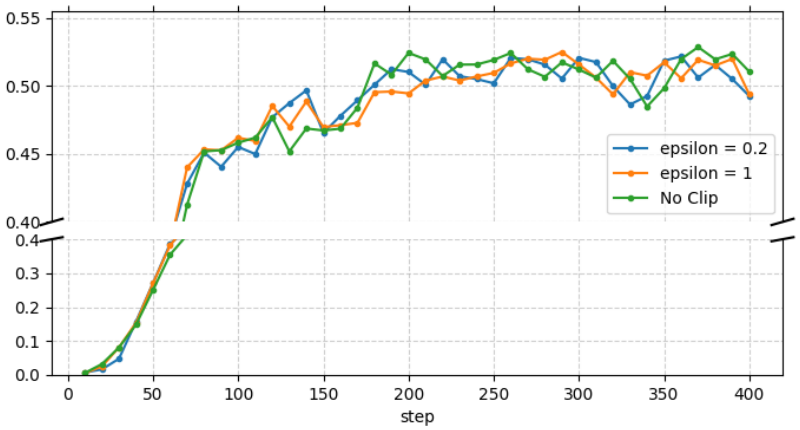


Figure 4: DePO accuracy traces for $\epsilon = 0.2$, $\epsilon = 1$, and No Clip. Horizontal axis: training step; vertical axis: AIME25 accuracy as a fraction, with a broken axis. The plot does not display $\epsilon = 0.5$; its reported result is in Table 4.

D Related Work and Mechanism Comparison

Reasoning and policy optimization. Chain-of-thought prompting and self-consistency improve reasoning at inference time (Wei et al., 2022; Wang et al., 2022). RL methods such as GRPO instead change the model through outcome-based optimization (Shao et al., 2024; DeepSeek-AI, 2025). DePO operates within this training setting. It neither replaces outcome rewards with a new reward model nor changes the reasoning benchmarks into an alternative task. Related sequence-level objectives such as GSPO (Zheng et al., 2025a) use a different aggregation of policy ratios; their success does not prove that arbitrary local smoothing preserves all useful token-level updates.

Entropy and selective updates. Classical maximum-entropy methods motivate explicit entropy objectives (Ziebart et al., 2008; Haarnoja et al., 2018). More recent work studies entropy dynamics and selective optimization for reasoning models (Cui et al., 2025; Jiang et al., 2025; Zheng et al., 2025b; Wang et al., 2025). Table 10 summarizes the distinctions relevant to DePO. The comparison concerns mechanisms and possible trade-offs; it does not rank methods independently of their tuning and empirical setting.

Table 10: Mechanism comparison. Each method has a tuning or approximation trade-off.

Method	Operation	Relevant trade-off
Ent-Reg	Adds an entropy term to the objective	Coefficient balances reward optimization and entropy preference
Clip-Cov	Selectively suppresses updates according to covariance statistics	Thresholds and selection can alter useful as well as harmful updates
DePO	Filters ratios used as detached score-function weights	Smoothing can attenuate useful sharp signals; kernel and boundary choices matter

Filtering and stochastic optimization. SWA, EWMA, and SGF are standard filtering operations; SGF's local polynomial construction is due to Savitzky & Golay (1964). Averaging has a broader role in stochastic approximation (Polyak & Juditsky, 1992), but averaging optimization iterates is not equivalent to filtering ratios across token positions. Here the relevant contribution is the choice of ratio signal and its integration into a detached policy update, together with the evaluation of that choice. Claims about polynomial reproduction or conditional noise variance should not be promoted into semantic guarantees.

Reward variance and sources of fluctuation. Razin et al. (2025) show that low reward variance can make reward optimization slow. This concerns the reward-induced optimization landscape. DePO leaves the rewards and advantages of a fixed sampled batch unchanged and modifies ratio weights, so it does not simply minimize reward variance. Nevertheless, ratios and rewards need not

be statistically independent, and filtering can still attenuate updates associated with useful sparse rewards. Small old-policy probabilities, rapid policy change, and imperfect trajectory-level credit assignment are possible contributors to ratio fluctuations; the present experiments do not isolate their causal contributions.

Tasks and complementary approaches. DAPO and AceCoder motivate the math and code training setups used here (Yu et al., 2025; Zeng et al., 2025). MATH and LiveCodeBench provide established reasoning evaluations (Hendrycks et al., 2021; Jain et al., 2024). The additional multi-turn study is inspired by Search-R1 (Jin et al., 2025). These connections motivate further evaluations, but the existing comparisons and incomplete supplementary configurations do not establish generality over all reward, retrieval, or task settings.