GRPODROPOUT: LESS IS MORE FOR ONLINE REINFORCEMENT LEARNING ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) methods such as GRPO substantially improve large language model reasoning but often suffer from policy entropy collapse: the loss of sampling diversity weakens exploration and limits further improvement. Existing methods address this issue either through algorithm-level interventions, such as reward modification and entropy/KL regularization, or through token-level reweighting. We investigate a complementary perspective: entropy collapse can also be mitigated by changing which generated rollouts contribute to policy updates. Under the same sampling budget, not all rollouts contribute positively to an update, and selectively excluding some can improve learning. To address this, we propose GRPODROPOUT: before the standard update, we use a simple strategy that selectively removes a small number of high-probability positive-advantage rollouts and recenters the retained advantages. To motivate this design, we develop a rollout-level theoretical analysis that guides method design and threshold selection. The method changes only rollout usage, and adds negligible computational overhead. Experiments show higher accuracy than original GRPO and higher actor entropy while using fewer rollout samples for updates, illustrating "less is more." This work provides insight into RL rollout usage: removing some rollouts can improve performance. Code is available at `https://anonymous.4open.science/r/GRPODropout/`.

1 INTRODUCTION

Online RL substantially improves the reasoning ability of large language models (Shao et al., 2024; Yu et al., 2025). However, these methods often suffer from policy entropy collapse during training, reducing sampling diversity and weakening exploration of alternative reasoning paths (Cui et al., 2025; Jin et al., 2026). Because online RL relies on newly sampled rollouts for subsequent updates, this reduced diversity can limit opportunities to discover new solutions and improve further (Yue et al., 2025). This highlights a fundamental challenge in RL: the exploration–exploitation trade-off (Auer et al., 2002).

To address this problem, existing approaches follow two routes. The first modifies the RL algorithm through rewards or advantages (Hao et al., 2025; Jin et al., 2026), clipping bounds (Yu et al., 2025; Park et al., 2025), or temperature and sampling control (Wang et al., 2025a). The second focuses on data usage at the token level, targeting high-entropy tokens and critical branching points for masking or reweighting (Du et al., 2025; Cui et al., 2025; Wang et al., 2025b), branching (Li et al., 2025; Zhao et al., 2026), or calibration (Yang et al., 2026). These methods exploit differences in token importance to mitigate entropy collapse. Here, we argue that changing rollout usage alone can mitigate entropy collapse. However, existing work still offers limited theoretical and methodological guidance for selecting complete rollouts to balance exploration and exploitation.

To bridge this gap, we propose GRPODROPOUT, which selectively removes rollouts to improve performance, mitigate entropy collapse, and preserve exploration in the trained model. Specifically, under our rollout-level theoretical analysis, we find that high-probability positive rollouts and low-probability negative rollouts can both reduce policy entropy, with the former having a more pronounced effect. Intuitively, high-probability positive rollouts are already well represented under the current policy; further reinforcement may provide limited additional information while accelerating entropy collapse. We therefore selectively remove these rollouts and recenter the retained advan-

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.

tages using old-policy probability weights to correct the imbalance introduced by deletion. Across three models and ten benchmarks, the average score improves over original GRPO by 1.55 percentage points. Despite using 12.1% fewer retained rollouts in policy updates on average in groups with nonzero advantages, training and held-out actor entropy are 446.1% and 256.0% higher than original GRPO, respectively, at the end of training, demonstrating better results with fewer update rollouts. Our contributions are:

- **A rollout-level analysis of data importance.** We extend the theoretical analysis to the rollout level to motivate our approach, and use experiments to further characterize which rollouts can reduce policy entropy, providing insight into how rollouts are used during RL training.
- **Rollout filtering and recentering.** We design a data-selection algorithm to filter potentially harmful rollouts and recenter retained advantages to correct the imbalance introduced by deletion.
- **Empirical evidence across model sizes.** We evaluate accuracy and entropy across three model sizes under reduced update-level rollout usage.

2 Related Work

Token-level entropy interventions. One line of work stabilizes policy entropy by identifying less informative tokens and attenuating their updates. UloRL (Du et al., 2025) identifies high-probability tokens in successful rollouts as already well learned and masks them when policy entropy falls below a target. RSI-S (Lv et al., 2026) extends this idea by normalizing sampled-token surprisal by local entropy. Cui et al. (2025) analyze first-order entropy changes and use this analysis to motivate Clip-Cov and KL-Cov, which clip or penalize updates on high-covariance tokens. Wang et al. (2025b) show that updating only the 20% highest-entropy tokens can match or outperform full-token training. Later methods identify critical branching points to construct more diverse training data. CURE (Li et al., 2025) and ROSE (Zhao et al., 2026) branch at positions where likely next tokens are semantically distinct and train on the resulting trees using segment-level advantages. ERPO (Yu et al., 2026) detects Critical Decision Pivots using group-normalized entropy and combines entropy gating, reasoning progress, and final outcomes into token-level advantages. PAEC (Yang et al., 2026) combines normalized entropy with top-two competition to construct a detached soft position mask. STARE (Luo et al., 2026) derives an advantage–surprisal four-quadrant structure from first-order entropy variation and uses target-entropy feedback to regulate high-surprisal tokens. ICT (Feng et al., 2026) selects positions whose full next-token distributions exhibit large Jensen–Shannon divergences, capturing both high- and low-entropy forks.

These token-level interventions leave open how entropy contributions can guide the selective removal of complete rollouts sampled for each prompt. We address this gap through rollout-level entropy analysis and principled rollout selection, as unmasked tokens may still reinforce existing paths and limit exploration (Du et al., 2025; Zhu et al., 2025).

Entropy-aware reinforcement learning algorithms. Another line addresses entropy collapse by modifying the training signal or optimization rule. One group modifies rewards or advantages. STEER (Hao et al., 2025) estimates the first-order entropy change induced by each token update and downweights updates that sharply decrease or increase entropy. Jin et al. (2026) identify positive-advantage tokens as the main source of entropy reduction and reweight them according to the training stage or current entropy. HAPO (He et al., 2026) reallocates advantages toward high-entropy positions through capacity-guided credit assignment. Papo (Zhang et al., 2026) predicts the update direction from the advantage sign, sampled-token probability, and local distribution, and regulates opposing entropy polarities by training phase. A second group modifies the optimization rule directly. DAPO (Yu et al., 2025) introduces Clip-Higher to relax the upper importance-ratio bound. Clip-Low/High (Park et al., 2025) show the underlying optimizer bias: lower clipping tends to increase entropy, whereas upper clipping decreases it. SCOPE-RL (Wang et al., 2025a) uses temperature-adaptive positive samples and an auxiliary regularizer to control policy entropy.

These methods mitigate entropy collapse through algorithmic changes to rewards, advantages, clipping, or regularization. We find that data-level interventions alone can also mitigate entropy collapse, through rollout selection and advantage recentering while preserving the RL objective form.

3 METHOD

We find that not all rollouts benefit model training and demonstrate that removing a subset of the data can improve performance. We therefore propose GRPODROPOUT, as illustrated in Figure 1. Each subsection presents a local theoretical analysis to motivate the corresponding design choice, while experiments evaluate the complete method. Section 3.2 develops rollout-level first-order entropy analysis and candidate deletion; Section 3.3 motivates positive-only deletion through a conditional comparison; Section 3.4 analyzes deletion-induced drift of zero-advantage sequences, derives re-centering, and presents the full procedure.

3.1 PRELIMINARIES

Problem Formulation. A rollout is a complete output sequence generated by the policy for a given prompt. For a fixed prompt x, let π_{old} denote the policy before the policy update, and let $\mathcal{Y}_x = \{\mathbf{y}_i\}_i$ be the set of all possible complete output sequences generated from x under π_{old}. For this fixed prompt, its Shannon entropy is $H(\pi_{\text{old}}) = -\sum_{\mathbf{y}_i \in \mathcal{Y}_x} \pi_{\text{old}}(\mathbf{y}_i \mid x) \log \pi_{\text{old}}(\mathbf{y}_i \mid x)$. We seek to optimize task return while mitigating policy entropy collapse and preserving exploration. For policy updates, we independently sample a training group of G complete rollouts from this space according to the old policy: $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(G)} \overset{\text{i.i.d.}}{\sim} \pi_{\text{old}}(\cdot \mid x)$. Here i indexes the space of complete output sequences, whereas $g \in \{1, \ldots, G\}$ indexes the sampled group. Rollout g is $\mathbf{y}^{(g)} = (y_1^{(g)}, \ldots, y_{T_g}^{(g)})$, with length T_g, giving sequence probability $\pi_{\text{old}}(\mathbf{y}^{(g)} \mid x) = \prod_{t=1}^{T_g} \pi_{\text{old}}(y_t^{(g)} \mid x, \mathbf{y}_{<t}^{(g)})$.

Policy optimization. Let π_θ be the policy to optimize, with parameters θ. Given a rollout reward $R(x, \mathbf{y})$, let $A(x, \mathbf{y})$ denote advantage relative to a baseline $b(x) = \bar{R}(x, *)$, e.g., $R(x, \mathbf{y}) - b(x)$. Using the old policy for sampling, we write the advantage objective with reference-policy regularization (Stiennon et al., 2020; Ouyang et al., 2022) and its sampled surrogate as

$$\begin{aligned} &\max_\theta \Big\{ \mathbb{E}_{\mathbf{y} \sim \pi_{\text{old}}(\cdot|x)} \Big[\frac{\pi_\theta(\mathbf{y} \mid x)}{\pi_{\text{old}}(\mathbf{y} \mid x)} A(x, \mathbf{y}) \Big] - \beta \, \mathrm{KL}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big) \Big\} \\ \rightsquigarrow \quad &\max_\theta \Big\{ \frac{1}{G} \sum_{g=1}^{G} A^{(g)} \frac{\pi_\theta(\mathbf{y}^{(g)} \mid x)}{\pi_{\text{old}}(\mathbf{y}^{(g)} \mid x)} - \beta \, \mathrm{KL}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big) \Big\}. \end{aligned} \tag{1}$$

Here π_{ref} is a fixed reference policy, $\beta \geq 0$ is its regularization coefficient, and KL denotes Kullback–Leibler divergence. The probability-ratio identity assumes that the candidate policy is supported within the old policy and the advantage expectation exists. The arrow denotes Monte Carlo approximation using sampled group of rollouts and advantages $A^{(g)}$.

3.2 ROLLOUT-LEVEL ENTROPY EFFECT AND CANDIDATE DELETION

In this subsection, we analyze which rollouts decrease policy entropy under a local update model. Motivated by this analysis and intuition, we design a filtering rule to mitigate entropy collapse.

Theoretical motivation: rollout updates and entropy collapse. Reward optimization can improve return while increasing the risk of entropy collapse and loss of exploration. To identify which rollouts decrease entropy, we first characterize each rollout's first-order entropy contribution and then give the criterion for the overall entropy change.

To extend token-level entropy analysis to complete rollouts, we analyze a local KL-proximal policy update following Abdolmaleki et al. (2018). With the sampled group and its advantages fixed, let $\pi^{(\eta)}$ be the updated policy at step size η, with $\pi^{(0)} = \pi_{\text{old}}$. Under this model, we obtain the following results:

Proposition 1 (First-order entropy contribution of a single rollout)**.** *Fix a prompt x and adopt the local update model and assumptions in Appendix A.1. Let $s_g = -\log \pi_{\text{old}}(\mathbf{y}^{(g)} \mid x)$ be the surprisal of sampled rollout g under the old policy. Its additive first-order entropy contribution to the whole-group update is*

$$h_g = \frac{A^{(g)}}{G} \big(s_g - H(\pi_{\text{old}}) \big). \tag{2}$$

Proposition 2 (Entropy derivative from sampled rollouts)**.** *Under the setting of Proposition 1, assume $\sum_{g=1}^{G} A^{(g)} = 0$. Using the surprisals s_g defined above, let $\bar{s} = G^{-1} \sum_{g=1}^{G} s_g$. For the fixed*

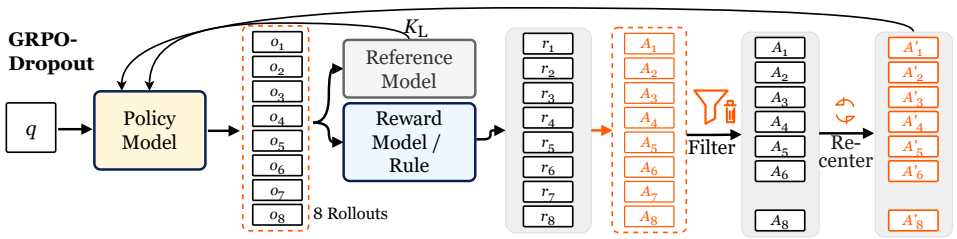


Figure 1: GRPODropout procedure. Filter removes only selected positive-advantage rollouts; Re-center applies probability-weighted advantage recentering. Other GRPO updates are unchanged.

sampled group, the proximal update has entropy derivative

$$\left.\frac{dH(\pi^{(\eta)})}{d\eta}\right|_{\eta=0} = \mathrm{Cov}_{g=1}^{G}\big(A^{(g)}, s_g\big) = \frac{1}{G}\sum_{g=1}^{G} A^{(g)}\big(s_g - \bar{s}\big). \tag{3}$$

The proofs are in Appendix A.1. Proposition 1 shows that a positive-advantage rollout with $s_g < H(\pi_{\text{old}})$ (relatively high probability) and a negative-advantage rollout with $s_g > H(\pi_{\text{old}})$ (relatively low probability) both have negative first-order entropy contributions. Conversely, low-probability positives and high-probability negatives contribute to entropy increase. Intuitively, further reinforcing already likely positive-advantage rollouts or suppressing already unlikely negative-advantage rollouts may offer limited gains in reasoning ability. Repeatedly prioritizing such updates may instead accelerate entropy collapse and weaken exploration.

Proposition 2 shows that the proximal update decreases entropy to first order when rollout advantages and log-probabilities are positively correlated, which is often indeed tend to be positively correlated in original models (Kadavath et al., 2022; Zenn & Geiping, 2026), providing an intuitive explanation for entropy collapse during reward optimization. Further, during RL training, by further increasing the likelihood of positive-advantage rollouts and decreasing that of negative-advantage ones, reward optimization can further reinforce this positive association and intensify entropy reduction over training.

Method: candidate filtering. Based on the preceding analysis, we design a rollout-selection method that uses Proposition 1 to identify candidates with negative estimated entropy contributions and Proposition 2 to set the deletion threshold. Specifically, we reuse h_g from Eq. (2), estimating $H(\pi_{\text{old}})$ by the Monte Carlo mean $\bar{s}$ of the original sampled group. In the filtering rule below, h_g denotes the contribution computed with this substitution. Using Proposition 2, we employ a greedy algorithm to select the largest retained set that ensures the estimated entropy does not decrease:

$$\mathcal{I}_{\text{keep}} \in \arg\max_{\mathcal{I}\subseteq\{1,\dots,G\}} |\mathcal{I}| \quad \text{s.t.} \quad \mathrm{Cov}_{g=1}^{G}\big(A^{(g)}\mathbf{1}\{g\in\mathcal{I}\}, s_g\big) \geq 0. \tag{4}$$

Here $\mathcal{I}_{\text{keep}}$ is the set of rollout indices selected for retention. The covariance is evaluated over the original G rollouts, with the indicator setting advantages of the deleted rollouts to zero. To solve this argmax, we start with all rollouts and greedily remove those with the most negative estimated contributions, until the constraint in Eq. (4) is satisfied. In practice, we length-normalize log-probabilities for both filtering and recentering to avoid direct length bias.

3.3 Deleting Only Positive-Advantage Rollouts

Section 3.2 shows that both high-probability positive-advantage and low-probability negative-advantage rollouts can decrease entropy. That result, however, uses a KL-proximal update whose logit coordinates contain old-policy-dependent scaling. Under ordinary gradient updates with a shared learning rate, their entropy contributions generally differ even when their h_g values are equal. To compare these two classes under the latter update, we additionally analyze one gradient-ascent step on independent sequence logits at the starting point.

Theoretical motivation: conditional entropy effects of positive and negative rollouts. For comparison, we consider gradient updates of logits z_i and explicitly require the same learning rate for every logit coordinate. With $L(z) = G^{-1}\sum_g A^{(g)} \log[\mathrm{softmax}(z)]_{i(g)}$, where $i(g)$ is the

sequence-space index of sampled rollout g, the update is:

$$z_i(\alpha) = z_i + \alpha \frac{\partial L(z)}{\partial z_i}, \qquad \alpha > 0. \tag{5}$$

Here α is the learning rate shared by all logit coordinates: it is the multiplier of each coordinate's gradient that is shared, not the logit or probability change of every sequence. Under this relatively more realistic assumption of a shared learning rate, we obtain the following result.

Proposition 3 (Greater Entropy Impact of Positive-Advantage Rollouts)**.** *In the independent-logit model above, assume $\sum_g A^{(g)} = 0$. Here h_g denotes rollout g's additive first-order entropy contribution under the KL-proximal update in Proposition 1, and h_g^{SGD} denotes its corresponding contribution under the logit gradient update above. Consider sampled rollouts $\mathbf{y}_+$ and $\mathbf{y}_-$ with advantages $A_\pm$ and surprisals $s_\pm = -\log \pi_{\mathrm{old}}(\mathbf{y}_\pm \mid x)$. Suppose $A_+ > 0 > A_-$ and $s_+ < H(\pi_{\mathrm{old}}) < s_-$. If their entropy contributions in Proposition 1 are equal, $h_+ = h_- < 0$, then the high-probability rollout contributes more to entropy decrease to first order in a single logit gradient update:*

$$\frac{|h_+^{\mathrm{SGD}}|}{|h_-^{\mathrm{SGD}}|} = \frac{\pi_{\mathrm{old}}(\mathbf{y}_+ \mid x)}{\pi_{\mathrm{old}}(\mathbf{y}_- \mid x)} = \exp(s_- - s_+) > 1. \tag{6}$$

Thus, even when the two rollouts have equal entropy contributions in Proposition 1, the high-probability positive-advantage rollout makes a larger first-order contribution to entropy reduction in a single logit update, strengthening the tendency toward entropy collapse (see Appendix A.2 for the proof). This conditional comparison provides additional motivation for preferentially filtering high-probability positive-advantage candidates.

Method: final deletion rule. Based on the above analysis, considering that deleting data itself can negatively reduce data informativeness and potentially degrade accuracy to some extent, we aim to minimize perturbations to the original training process and preserve the original algorithm and data distribution as much as possible. Guided by the preceding theory, we therefore delete only positive-advantage rollouts while retaining negative-advantage ones. Starting from the retained set $\mathcal{I}_{\mathrm{keep}}$ in Section 3.2, we restore any excluded rollouts with nonpositive advantages:

$$\mathcal{I}_{\mathrm{keep}} \leftarrow \mathcal{I}_{\mathrm{keep}} \cup \{g \in \{1, \ldots, G\} : A^{(g)} \le 0\}. \tag{7}$$

3.4 Recentering After Deletion and the Full Procedure

Selective deletion changes the composition of retained samples, may make their distribution depart from the old policy, and disrupts the original centering of group advantages. This motivates recentering the retained advantages after deletion.

Theoretical motivation: normalization drift after deletion. Generalization can legitimately increase the probability of unsampled solutions similar to rewarded rollouts. Beyond generalization, a normalization offset that does not depend on similarity can also change the probabilities of unsampled paths, which is what we aim to avoid. However, unrecentered advantages and the resulting perturbed distribution tend to exacerbate this issue. To isolate it, we assume i.i.d. sampling of the original rollout group for a fixed prompt and, conditional on the realized group and retained set, analyze an auxiliary distribution-space update without transfer through shared parameters. Using the retained set $\mathcal{I}_{\mathrm{keep}}$, define the accumulated post-deletion advantage as $\tilde{A}_i^{\mathrm{keep}} = \sum_{g \in \mathcal{I}_{\mathrm{keep}}} A^{(g)} \mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\}$. We then obtain the following result:

Proposition 4 (Invariance condition for zero-advantage sequences)**.** *Fix a prompt x and a realized group of rollouts sampled i.i.d. from $\pi_{\mathrm{old}}(\cdot \mid x)$. Let $U = \{i : \pi_{\mathrm{old}}(\mathbf{y}_i \mid x) > 0,\ \tilde{A}_i^{\mathrm{keep}} = 0\}$ be the set of unsampled and other zero-advantage paths in the old-policy support. Evaluate π_θ along the auxiliary retained-sample proximal update specified in Appendix A.3, with θ varying with η and $\pi_\theta = \pi_{\mathrm{old}}$ at $\eta = 0$. For every $i \in U$,*

$$\frac{d}{d\eta} \log \pi_\theta(\mathbf{y}_i \mid x)\Big|_{\eta=0} = -\mathbb{E}_{\pi_{\mathrm{old}}}[\tilde{A}^{\mathrm{keep}}]. \tag{8}$$

This proposition shows that all zero-advantage paths share the same first-order log-probability shift: regardless of their quality, their probabilities decrease when the old-policy-weighted mean advantage is positive and increase when it is negative. This shift expresses no preference for more promising

paths, and its first-order strength is determined by $|\mathbb{E}_{\pi_{\text{old}}}[\tilde{A}^{\text{keep}}]|$. Deletion changes the retained score and its old-policy-weighted mean, making such unwarranted probability increases and decreases more pronounced, thereby leading to training instability. The proof is in Appendix A.3.

Method: probability-weighted recentering. Based on the above analysis, we recenter a nonempty retained set as follows. Recomputing $\tilde{A}^{\text{keep}}$ with the recentered advantages yields a zero old-policy-weighted mean and cancels the first-order normalization shift in the proximal model of Proposition 4:

$$b_{\text{keep}} = \frac{\sum_{g\in\mathcal{I}_{\text{keep}}} e^{-s_g} A^{(g)}}{\sum_{g\in\mathcal{I}_{\text{keep}}} e^{-s_g}}, \qquad \bar{A}^{(g)}_{\text{keep}} = A^{(g)} - b_{\text{keep}}, \quad g \in \mathcal{I}_{\text{keep}}. \tag{9}$$

This local analysis motivates the recentering rule; its effectiveness in training is evaluated empirically in Section 4.2. Note that this probability weighting is introduced precisely because our data deletion is biased, which violates the i.i.d. assumption.

GRPODropout procedure. Based on the above theory and analysis, we present the full GRPO-DROPOUT procedure. To minimize changes to the original training process, we use the selected rollouts and recentered advantages in the original loss, preserving its formulation. The procedure reuses existing rewards and log-probabilities and requires no value network, adding negligible computational overhead for filtering and recentering.

GRPODropout

Input: prompt x, old policy π_{old}, sample count G, and the original RL loss.
1. Sample G rollouts and compute rewards, native advantages $A^{(g)}$, and old-policy rollout surprisals s_g; set advantages to zero if the reward standard deviation is zero.
2. Estimate $H(\pi_{\text{old}})$ by $\bar{s}$, compute and fix h_g using Eq. (2), then solve Eq. (4) by removing the most negative estimated contributions first to obtain $\mathcal{I}_{\text{keep}}$.
3. Apply Eq. (7) to restore excluded rollouts with nonpositive native advantages to $\mathcal{I}_{\text{keep}}$.
4. For the retained set, compute and subtract b_{keep} using Eq. (9) to recentered advantages.
5. Update the policy using the original RL loss on retained rollouts.

4 EXPERIMENTS

We compare GRPODROPOUT, original GRPO, and other entropy-aware baselines across three model sizes (1.7B, 4B, and 8B), trained on mathematics with matched prompts. Evaluation spans ten benchmarks covering mathematical reasoning and general QA, with the latter assessing generalization beyond mathematics. Given their small test sets, AMC23/AIME24/AIME25/HMMT25 use Mean@32; the others use Mean@1. Full settings, baselines, and metric conventions appear in Appendix B.1.

4.1 MAIN RESULTS

GRPODropout targets rollouts that may reduce policy entropy. Under the same rollout sampling budget, we assess whether using fewer retained rollouts in policy updates can preserve or improve performance over original GRPO. Table 1 reports results on three model sizes: selective deletion with recentering improves average accuracy over GRPO by 1.55 percentage points on average. These results support our data-usage insight that using all sampled rollouts is not always the most effective training strategy, and that selectively removing a subset can instead yield better performance. GRPODropout also exceeds KL-Cov, the strongest entropy-aware baseline by average score, by 0.11 points on average. Finally, our method is orthogonal in design to these optimization methods, offering both a data-selection insight and a complementary strategy for mitigating entropy collapse.

Removing few rollouts substantially mitigates entropy collapse. We seek to preserve performance and exploration with minimal changes to training. Because sequence-level entropy is strongly affected by response length, we follow prior work (Cui et al., 2025) and report token-level conditional entropy averaged over valid response tokens. We measure this entropy as $-\sum_v \pi_\theta(v \mid$

Table 1: Main results on three backbones. Avg. averages ten benchmarks; Base denotes the untrained backbone. Bold marks the best result within each backbone block.

	Method	Mathematical reasoning							General QA			Avg.
		GSM8k	Math500	AMC23	Olympiad	AIME24	AIME25	HMMT25	ARC-C	MMLUP	SGPQA	
Qwen3-1.7B	Base	89.92	80.60	59.92	42.96	22.08	21.04	11.46	87.37	57.01	29.28	50.16
	GRPO	90.60	84.00	72.27	51.56	28.54	23.44	14.06	87.80	**57.91**	**31.07**	54.13
	DAPO	88.86	85.20	72.27	52.00	**33.75**	**26.15**	15.21	87.63	55.62	29.70	54.64
	Clip-Cov	82.34	77.20	53.91	36.15	11.56	12.71	5.21	87.37	55.19	28.26	44.99
	KL-Cov	90.07	**86.40**	70.39	50.07	31.67	25.73	14.79	**88.31**	56.60	29.68	54.37
	Beyond8020	89.01	84.00	72.19	50.22	29.48	25.00	**18.02**	86.69	56.03	29.78	54.04
	GRPODropout	**90.83**	85.60	**75.55**	**52.15**	31.46	25.62	15.83	86.35	56.64	30.02	**55.01**
Qwen3-4B	Base	94.77	85.40	72.50	47.41	37.92	26.25	14.27	94.20	69.19	39.85	58.18
	GRPO	94.77	91.80	89.45	58.37	42.60	33.23	18.96	94.11	68.35	38.64	63.03
	DAPO	94.47	89.80	86.72	57.48	42.40	36.88	19.06	94.11	67.48	38.52	62.69
	Clip-Cov	94.54	**92.00**	86.33	59.41	54.06	41.56	27.08	94.45	69.32	40.79	65.95
	KL-Cov	**94.92**	**92.00**	87.89	59.11	**54.79**	40.83	**27.60**	94.03	69.27	**40.83**	**66.13**
	Beyond8020	94.39	91.60	**89.92**	**60.00**	51.88	41.25	23.65	**94.54**	68.21	39.39	65.48
	GRPODropout	94.77	91.80	88.28	58.96	51.98	**43.02**	25.94	94.11	**69.35**	40.25	65.85
Qwen3-8B	Base	95.45	82.80	67.50	46.07	37.40	24.69	11.98	95.90	73.14	43.61	57.85
	GRPO	95.30	91.20	85.86	59.11	53.54	38.44	24.38	95.73	**74.22**	**45.54**	66.33
	DAPO	95.53	92.20	**90.31**	**61.33**	52.40	38.85	21.25	95.65	72.39	43.75	66.37
	Clip-Cov	95.38	91.20	84.77	60.44	54.90	42.08	**27.60**	95.56	73.76	45.05	67.07
	KL-Cov	**95.75**	**93.20**	86.72	**61.33**	54.17	40.31	27.50	95.48	73.81	44.88	**67.32**
	Beyond8020	95.45	92.20	87.58	60.59	**57.29**	39.06	24.69	**95.99**	73.23	44.57	67.07
	GRPODropout	95.53	92.20	87.66	60.74	55.31	**42.81**	25.21	94.97	73.50	44.99	67.29

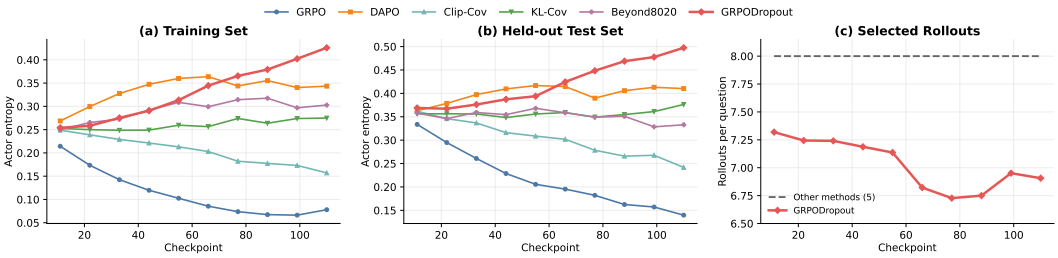


Figure 2: Qwen3-4B actor entropy and rollout usage across checkpoints. (a–b) Token-mean entropy on training DAPO-17K and held-out MMLU-Pro (MMLUP); (c) mean update rollouts per question in training batches, excluding all-zero-advantage groups. The other five methods use 8 rollouts at every recorded checkpoint (gray dashed line). Curves connect raw observations without smoothing.

$x, \mathbf{y}_{<t}) \log \pi_\theta(v \mid x, \mathbf{y}_{<t})$, where v ranges over the vocabulary. Figure 2 tracks this metric for Qwen3-4B on DAPO-17K and held-out MMLU-Pro. While GRPO exhibits entropy collapse, GRPODropout achieves 446.1% and 256.0% higher entropy, respectively, at the end of training. Across the ten recorded checkpoints, excluding all-zero-advantage groups, it retains 7.03 rollouts per question for the update—12.1% fewer than GRPO's baseline of eight. Notably, compared to Beyond8020, which masks out 80% of tokens, our method removes a mere 12.1% of the data while achieving comparable or even superior policy entropy and performance, demonstrating the efficacy and value of rollout-level deletion.

4.2 Ablation Studies

Our method has two main components: rollout deletion and advantage recentering. We ablate the effects of deleting positive-advantage (Pos.) and negative-advantage (Neg.) rollouts, enabling recentering (Center), and using probability weighting during recentering (P-wt.). Specifically, with Center/P-wt. set to ✓/✓, we subtract the probability-weighted mean advantage as in Eq. (9); with ✓/×, we subtract the equal-weight mean of retained advantages; with ×/×, no recentering is performed. The first row of each model block reports original GRPO without rollout deletion. Table 2 reports results on Qwen3-1.7B and Qwen3-4B.

Table 2: Qwen3-1.7B and Qwen3-4B ablations of positive- and negative-advantage rollouts, advantage recentering, and old-policy probability weighting of the mean. ✓/×: enabled/disabled; –: not applicable. Avg. Ent reports benchmark-average actor entropy.

Deletion		Advantages		Mathematical reasoning							General QA			Avg.	Avg. Ent
Pos.	Neg.	Center	P-wt.	GSM8k	Math500	AMC23	Olympiad	AIME24	AIME25	HMMT25	ARC-C	MMLUP	SGPQA		
Qwen3-1.7B															
×	×	–	–	90.60	84.00	72.27	51.56	28.54	23.44	14.06	87.80	**57.91**	**31.07**	54.13	0.0503
✓	✓	✓	✓	90.30	**86.80**	70.47	49.04	30.94	24.90	13.75	87.71	56.53	29.77	54.02	0.2025
✓	×	×	×	77.33	39.00	10.70	8.74	0.00	0.10	0.00	87.12	44.04	21.09	28.81	0.3282
✓	×	✓	×	89.76	85.00	65.31	46.67	27.29	22.40	11.46	**87.88**	56.64	29.44	52.19	0.2239
✓	×	✓	✓	**90.83**	85.60	**75.55**	**52.15**	**31.46**	**25.62**	**15.83**	86.35	56.64	30.02	**55.01**	0.1739
Qwen3-4B															
×	×	–	–	**94.77**	91.80	**89.45**	58.37	42.60	33.23	18.96	94.11	68.35	38.64	63.03	0.0528
✓	✓	✓	✓	**94.77**	91.00	87.03	58.22	51.46	41.25	25.00	93.34	69.09	40.10	65.13	0.2425
×	✓	✓	✓	94.54	91.40	87.97	**59.26**	51.04	41.88	23.44	**94.28**	68.80	40.43	65.30	0.2043
✓	×	×	×	0.38	0.20	2.19	0.00	0.00	0.00	0.00	88.57	42.35	26.34	16.00	0.0200
✓	×	✓	×	94.62	**92.20**	89.30	58.22	**52.81**	42.08	25.10	93.94	**69.48**	**40.44**	65.82	0.2174
✓	×	✓	✓	**94.77**	91.80	88.28	58.96	51.98	**43.02**	**25.94**	94.11	69.35	40.25	**65.85**	0.2140

Deleting only positive-advantage rollouts performs best. With probability-weighted recentering fixed, we compare deleting positive- and/or negative-advantage candidates. Deletion generally increases entropy and improves final accuracy in most settings, while the magnitude of the accuracy gain depends on the deletion pattern and recentering scheme. Deleting both positive and negative candidates yields the largest entropy gain but incurs a slight accuracy loss, showing that data removal can sacrifice performance and supporting our motivation to delete as little data as possible. Deleting only positive candidates achieves higher entropy and accuracy than deleting only negative candidates, consistent with our theoretical result that positive-advantage rollouts have a greater effect on entropy reduction.

Recentering is necessary. With deletion restricted to positive-advantage candidates, removing recentering causes severe performance degradation and training collapse, demonstrating the importance of recentering in the tested deletion configurations. We further introduce probability weighting to calibrate the retained advantages under the old-policy measure. On Qwen3-4B, probability weighting has little effect, improving the average score by only 0.03 points over equal-weight recentering. Its benefit is more pronounced on Qwen3-1.7B, where removing the weighting lowers the average score by 2.82 points, suggesting that probability weighting can sometimes improve training stability.

4.3 More Precise Exploration and More Diverse Reasoning Paths

The full GRPODropout method increases entropy while largely preserving or improving average accuracy across the three model sizes. We further examine the value of this retained entropy from two perspectives: exploration precision, to assess accuracy preservation; and reasoning-path diversity, to evaluate the preservation of exploration capacity.

Metrics. We first assess how accurately model sampling produces correct solutions while limiting repeated incorrect final answers. We compare all methods on Qwen3-4B across AMC23, AIME24, AIME25, and HMMT25, sampling 32 rollouts per question. Pass@32 is computed per question within each benchmark and then averaged equally across AMC23, AIME24, AIME25, and HMMT25. For Table 3, Entropy is token-mean actor entropy over valid response tokens at the final checkpoint, averaged equally across the four benchmarks. Inspired by the answer-consistency perspective of Wang et al. (2022), we define answer-string dispersion W as the number of distinct nonempty final-answer strings among a question's incorrect responses. W measures answer-string dispersion among incorrect final answers and can reveal recurring answer patterns. To distinguish relatively easier and harder questions, we use a fixed cross-method partition: a question enters the ≥ 8 subset only if every one of the six systems has at least eight correct responses, and enters the < 8 subset only if every system has fewer than eight. We average W within each subset.

We next assess whether models approach the same problem through diverse reasoning paths. To this end, we use GLM-5.2 to summarize each response's reasoning path in three to five sentences,

Table 3: Qwen3-4B solution coverage, wrong-answer dispersion, and reasoning diversity. Definitions are in Section 4.3. Bold marks the best and underline marks the second-best in each column.

Method	Correctness			Path diversity				Entropy ↑	Overall mean rank ↓
	pass@32 ↑	$W_{\geq 8}$ ↓	$W_{<8}$ ↓	Dist.$_{\geq 8}$ ↑	Dist.$_{<8}$ ↑	Vendi$_{\geq 8}$ ↑	Vendi$_{<8}$ ↑		
GRPO	63.54	1.1846	7.6216	0.1063	0.1641	1.8311	2.4194	0.0448	5.63
DAPO	69.38	1.6154	8.0811	<u>0.1150</u>	0.1798	<u>1.9080</u>	2.6013	<u>0.1796</u>	3.88
Clip-Cov	**76.88**	0.7692	5.2162	0.1139	**0.2141**	1.9041	**2.9851**	0.0900	<u>2.63</u>
KL-Cov	72.71	**0.4615**	**3.0000**	0.1134	0.2014	1.8930	2.8510	0.1509	3.25
Beyond8020	73.54	1.1231	7.8378	0.1142	0.1726	1.9037	2.5252	0.1671	4.00
GRPODropout	<u>75.21</u>	<u>0.4923</u>	<u>4.2432</u>	**0.1151**	<u>0.2090</u>	**1.9114**	<u>2.8992</u>	**0.2291**	**1.63**

excluding abandoned reasoning, and then encode the summaries with bge-m3 (Chen et al., 2024). Following Kirk et al. (2023); Chen et al. (2026), Summary distance measures the mean pairwise cosine distance between summary embeddings for the same question, reflecting semantic separation. Following Friedman & Dieng (2023); Encheng et al. (2026), Vendi measures effective diversity as the exponential of the entropy of the eigenvalues of the Gram matrix normalized by the number of summaries, accounting for similarity and repetition. Higher values indicate greater diversity. Here, W measures diversity among incorrect final answers, whereas Summary distance and Vendi measure diversity in the summarized reasoning paths, including paths that may lead to the same final answer. Full prompt and run settings are given in Appendix B.2.

A favorable trade-off between correctness and diversity. The results show that our method achieves the best or second-best reported mean on all eight metrics and ranks first among the six systems, with an overall mean rank of 1.63 under equal weighting. In particular, our method attains the highest reasoning-path diversity on the ≥ 8 subset and the highest actor entropy while achieving lower W than all baselines except KL-Cov. Compared with GRPO, higher solution coverage and actor entropy accompany fewer distinct incorrect answers and higher values on all four path-diversity measures, demonstrating that our improvement in Pass@1 accuracy does not come at the expense of Pass@32 or the model's exploratory capability, but rather enhances them simultaneously. Notably, although DAPO exhibits relatively high entropy and reasoning-path diversity on the ≥ 8 subset, this does not translate into the highest Pass@32, indicating that it sacrifices accuracy for exploratory diversity. Correspondingly, our method maximally preserves exploration accuracy—namely, higher Pass@1 and Pass@32, and fewer distinct incorrect answers—while fully maintaining the path diversity and entropy of correct answers, ensuring no loss of the model's exploratory capability and achieving a simultaneous preservation of accuracy and entropy.

5 Conclusion

We find that some high-probability positive rollouts can hinder training in reinforcement learning algorithms such as GRPO. Removing these rollouts improves performance and mitigates entropy collapse. Based on this finding, we propose GRPODropout, which selectively removes high-probability positive rollouts and recenters the retained advantages to stabilize training, while leaving the original RL loss unchanged. We further extend the theoretical analysis of entropy dynamics from tokens to complete rollouts, providing theoretical motivation for our rollout-deletion strategy. Extensive experiments across three model scales and ten benchmarks demonstrate that GRPODropout outperforms the original GRPO while using fewer retained rollouts in policy updates.

Limitations. First, GRPODropout does not outperform the strongest baseline at every model scale. It is worth pointing out that we do not merely provide an entropy collapse mitigation method; rather, our work includes the insight that reinforcing some rollouts—even correct ones—can be counterproductive, which also possesses clear value. Second, given the complexity of RL training, our theory itself cannot fully explain our method. In fact, the primary role of the theory is to serve as motivation for method design, acting as the initial starting point. The final method is validated through its advantages over the original GRPO and the ablation study results. Finally, GRPODropout relies on within-prompt differences in advantages and probabilities across multiple rollouts. It therefore does not directly apply to single-rollout training, while its effectiveness with very small groups remains to be evaluated.

AI USE STATEMENT

Generative AI was not used to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose hypotheses, design or implement methods or experiments, translate text, clean or reformat datasets, or interpret results. It was used only as an off-the-shelf measurement tool: GLM-5.2 summarizes each sampled response into a three-to-five-sentence description of its reasoning path, which is then embedded with bge-m3 for the diversity metrics in Section 4.3. These summaries do not enter the training data, reward signal, or model updates. Generative AI tools were used to assist in drafting and editing parts of the manuscript to improve readability, as well as in searching for and retrieving related work. We reviewed all AI-assisted material and take responsibility for the final content, including the text, claims, and artifacts.

ETHICS STATEMENT

This work modifies the rollout-selection step of online reinforcement learning for language models and involves no human subjects or newly collected personal data. We use public mathematics and question-answering benchmarks, model-generated responses, and automated verifiers. The method inherits the behaviors, limitations, and potential biases of the base models and training data; these dimensions are not measured here, and we make no claim that GRPODropout reduces them. Our theoretical analysis is local and first-order and does not characterize parameter sharing, clipping, or repeated optimization. The reported gains are limited to the models, benchmarks, and sampling budget studied here, and the path-level measurements rely on an automatic summarizer. They should not be interpreted as evidence of broader safety or reliability improvements.

REPRODUCIBILITY STATEMENT

Section 3.1 defines the notation, sampling setup, and policy-update objective. Sections 3.2–3.4 state the propositions and derive the filtering and recentering rules, with complete proofs in Appendices A.1, A.2, and A.3. Section 3.4 gives the full five-step procedure, which reuses existing rewards and old-policy log-probabilities, preserves the original loss formulation, and requires no value network. Sections 4.1, 4.2, and 4.3 report the model scales, ten benchmarks, baselines, evaluation protocols, and diversity metrics. Appendix B.1 provides the training data, batch composition, sampling settings, optimizer and clipping hyperparameters, sequence lengths, decoding configuration, and scoring procedures. Code is available at `https://anonymous.4open.science/r/GRPODropout/`. The implementation uses length-normalized surprisals for filtering and recentering, whereas the theoretical results use full-sequence probabilities; Appendix A.1 documents this correspondence.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. *CoRR*, abs/1806.06920, 2018. URL `http://arxiv.org/abs/1806.06920`.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002. doi: 10.1023/A:1013689704352. URL `https://doi.org/10.1023/A:1013689704352`.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 2318–2335. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.137. URL `https://doi.org/10.18653/v1/2024.findings-acl.137`.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aris Sotiras, Yalin Wang, and Abolfazl Razi. DRA-GRPO: your GRPO needs to know diverse

reasoning paths for mathematical reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 13995–14019. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.685. URL https://doi.org/10.18653/v1/2026.findings-acl.685.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025. doi: 10.48550/ARXIV.2505.22617. URL https://doi.org/10.48550/arXiv.2505.22617.

Dong Du, Shulin Liu, Tao Yang, Shaohua Chen, and Yang Li. Ulorl:an ultra-long output reinforcement learning approach for advancing large language models' reasoning abilities. *CoRR*, abs/2507.19766, 2025. doi: 10.48550/ARXIV.2507.19766. URL https://doi.org/10.48550/arXiv.2507.19766.

Cui Encheng, Shaowen Peng, Kazuhiro Ito, Jinsha Xu, Shohei Hisada, Shoko Wakamiya, and Eiji Aramaki. Single-agent generation surpasses multi-agent systems in semantic diversity. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 37993–38008. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.1894. URL https://doi.org/10.18653/v1/2026.findings-acl.1894.

Xuanzhi Feng, Zhengyang Li, Zeyu Liu, Haoxi Li, Yuming Jiang, Bing Guo, Jingcai Guo, Jie Zhang, and Song Guo. Beyond entropy: Learning from token-level distributional deviations for LLM reasoning. *CoRR*, abs/2606.19771, 2026. doi: 10.48550/ARXIV.2606.19771. URL https://doi.org/10.48550/arXiv.2606.19771.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=g97OHbQyk1.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in RLVR: an entropy change perspective. *CoRR*, abs/2510.10150, 2025. doi: 10.48550/ARXIV.2510.10150. URL https://doi.org/10.48550/arXiv.2510.10150.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *CoRR*, abs/2402.14008, 2024. doi: 10.48550/ARXIV.2402.14008. URL https://doi.org/10.48550/arXiv.2402.14008.

Yuhang He, Haodong Wu, Siyi Liu, Hongyu Ge, Hange Zhou, Keyi Wu, Zhuo Zheng, Qihong Lin, Zixin Zhong, and Yongqi Zhang. Rethinking token-level credit assignment in RLVR: A polarity-entropy analysis. *CoRR*, abs/2604.11056, 2026. doi: 10.48550/ARXIV.2604.11056. URL https://doi.org/10.48550/arXiv.2604.11056.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the*

Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026, pp. 25300–25322. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.1266. URL https://doi.org/10.18653/v1/2026.findings-acl.1266.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. doi: 10.48550/ARXIV.2207.05221. URL https://doi.org/10.48550/arXiv.2207.05221.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. *CoRR*, abs/2310.06452, 2023. doi: 10.48550/ARXIV.2310.06452. URL https://doi.org/10.48550/arXiv.2310.06452.

Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, and Jing Yang. CURE: critical-token-guided re-concatenation for entropy-collapse prevention. *CoRR*, abs/2508.11016, 2025. doi: 10.48550/ARXIV.2508.11016. URL https://doi.org/10.48550/arXiv.2508.11016.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *CoRR*, abs/2305.20050, 2023. doi: 10.48550/ARXIV.2305.20050. URL https://doi.org/10.48550/arXiv.2305.20050.

Haipeng Luo, Qingfeng Sun, Songli Wu, Can Xu, Wenfeng Deng, Han Hu, and Yansong Tang. STARE: surprisal-guided token-level advantage reweighting for policy entropy stability. *CoRR*, abs/2606.19236, 2026. doi: 10.48550/ARXIV.2606.19236. URL https://doi.org/10.48550/arXiv.2606.19236.

Outongyi Lv, Yanzhao Zheng, Yuanwei Zhang, Zhenghao Huang, Xingjun Wang, Baohua Dong, Hangcheng Zhu, and Yingda Chen. Which tokens matter? adaptive token selection for RLVR with the relative surprisal index. *CoRR*, abs/2606.31575, 2026. doi: 10.48550/ARXIV.2606.31575. URL https://doi.org/10.48550/arXiv.2606.31575.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022. doi: 10.48550/ARXIV.2203.02155. URL https://doi.org/10.48550/arXiv.2203.02155.

Jaesung R. Park, Junsu Kim, Gyeongman Kim, Jinyoung Jo, Sean Choi, Jaewoong Cho, and Ernest K. Ryu. Clip-low increases entropy and clip-high decreases entropy in reinforcement learning of large language models. *CoRR*, abs/2509.26114, 2025. doi: 10.48550/ARXIV.2509.26114. URL https://doi.org/10.48550/arXiv.2509.26114.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020. URL https://arxiv.org/abs/2009.01325.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller (eds.), *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pp. 1057–1063. The MIT Press, 1999. URL http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixing Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jingyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Chen Wang, Zhaochun Li, Jionghao Bai, Yuzhi Zhang, Shisheng Cui, Zhou Zhao, and Yue Wang. Arbitrary entropy policy optimization: Entropy is controllable in reinforcement finetuning. *CoRR*, abs/2510.08141, 2025a. doi: 10.48550/ARXIV.2510.08141. URL https://doi.org/10.48550/arXiv.2510.08141.

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. *CoRR*, abs/2506.01939, 2025b. doi: 10.48550/ARXIV.2506.01939. URL https://doi.org/10.48550/arXiv.2506.01939.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171, 2022. doi: 10.48550/ARXIV.2203.11171. URL https://doi.org/10.48550/arXiv.2203.11171.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574, 2024. doi: 10.48550/ARXIV.2406.01574. URL https://doi.org/10.48550/arXiv.2406.01574.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Shumeng Yang, Yisu Liu, Jiayi Zheng, Zhaohui Yang, and Linjing Li. PAEC: position-aware entropy calibration for LLM reasoning in RLVR. *CoRR*, abs/2606.08543, 2026. doi: 10.48550/ARXIV.2606.08543. URL https://doi.org/10.48550/arXiv.2606.08543.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, 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, Weinan Dai, 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. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/ARXIV.2503.14476. URL https://doi.org/10.48550/arXiv.2503.14476.

Song Yu, Li Li, Wenwen Zhao, and Zhisheng Yang. ERPO: token-level entropy-regulated policy optimization for large reasoning models. *CoRR*, abs/2603.28204, 2026. doi: 10.48550/ARXIV.2603.28204. URL https://doi.org/10.48550/arXiv.2603.28204.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837, 2025. doi: 10.48550/ARXIV.2504.13837. URL https://doi.org/10.48550/arXiv.2504.13837.

Johannes Zenn and Jonas Geiping. When are likely answers right? on sequence probability and correctness in llms. *CoRR*, abs/2606.27359, 2026. doi: 10.48550/ARXIV.2606.27359. URL https://doi.org/10.48550/arXiv.2606.27359.

Jiazheng Zhang, Ziche Fu, Junrui Shen, Yunbin Zhao, Yunke Zhang, Zhiheng Xi, Long Ma, Chenxin An, Zhihao Zhang, Shichun Liu, Dingwei Zhu, Shihan Dou, Shaofan Liu, Han Li, Wiggin Zhou, Aiden Adams, Tao Gui, Fei Huang, Qi Zhang, and Xuanjing Huang. Entropy polarity in reinforcement fine-tuning: Direction, asymmetry, and control. *CoRR*, abs/2605.11775, 2026. doi: 10.48550/ARXIV.2605.11775. URL https://doi.org/10.48550/arXiv.2605.11775.

Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, Dawei Yin, and Xin Xin. Reinforced efficient reasoning via semantically diverse exploration. *CoRR*, abs/2601.05053, 2026. doi: 10.48550/ARXIV.2601.05053. URL https://doi.org/10.48550/arXiv.2601.05053.

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. *CoRR*, abs/2507.18071, 2025. doi: 10.48550/ARXIV.2507.18071. URL https://doi.org/10.48550/arXiv.2507.18071.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. *CoRR*, abs/2506.01347, 2025. doi: 10.48550/ARXIV.2506.01347. URL https://doi.org/10.48550/arXiv.2506.01347.

A SUPPLEMENTARY MATERIAL

A.1 PROOFS OF PROPOSITION 1 AND PROPOSITION 2

Local update approximation. To extend token-level entropy analysis to complete rollouts, we adopt a distribution-space KL-proximal policy-improvement model following Abdolmaleki et al. (2018). For a candidate policy $\pi(\cdot \mid x)$, we define its proximal update as

$$L_A(\pi) = \frac{1}{G}\sum_{g=1}^{G} A^{(g)} \frac{\pi(\mathbf{y}^{(g)} \mid x)}{\pi_{\text{old}}(\mathbf{y}^{(g)} \mid x)}, \quad \pi^{(\eta)} = \arg\max_{\pi \in \Delta_{\text{old}}} \Big\{ L_A(\pi) - \frac{1}{\eta}\mathrm{KL}\big(\pi \,\|\, \pi_{\text{old}}\big) \Big\}. \tag{A.1}$$

Here $\eta > 0$ controls the size of the proximal update, and Δ_{old} contains normalized distributions over complete sequences supported on $\{\mathbf{y}_i : \pi_{\text{old}}(\mathbf{y}_i \mid x) > 0\}$. Eq. (A.1) specifies an idealized update directly over sequence distributions. Its old-policy KL penalty specifies the update geometry, while the KL regularizer relative to the reference policy is omitted. We analyze the update around $\eta = 0$, taking $\pi^{(0)} = \pi_{\text{old}}$, with the sampled group and its finite advantages held fixed. We assume $H(\pi_{\text{old}}) < \infty$ and use rollout advantages without length normalization. We derive the sequence-score representation and closed-form update below.

Notation for the proofs. For the fixed prompt x, we use the shorthand $p(\mathbf{y}) = \pi_{\text{old}}(\mathbf{y} \mid x)$ and $q(\mathbf{y}) = \pi(\mathbf{y} \mid x)$ for the old and candidate policies, with $p_i = p(\mathbf{y}_i)$ and $q_i = q(\mathbf{y}_i)$. The proximal policies are written $q_i^{(\eta)} = \pi^{(\eta)}(\mathbf{y}_i \mid x)$ and $q_{\text{keep},i}^{(\eta)} = \pi_\theta(\mathbf{y}_i \mid x)$, where in the latter case θ varies with η along the retained-sample proximal update. These abbreviations are used only in this appendix.

Sampling-aligned sequence score. With the sampled group and its advantages fixed, express the sampled surrogate in sequence space using

$$\tilde{A}_i = \frac{1}{G p_i}\sum_{g=1}^{G} A^{(g)} \mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\}, \qquad p_i > 0. \tag{A.2}$$

Here $\mathbf{1}\{\cdot\}$ is the indicator function, and sums over sequence indices are restricted to the old-policy support. Repeated samples of the same sequence contribute additively, while a sequence not sampled in the current group has $\tilde{A}_i = 0$ and receives no direct update signal. Indeed,

$$\sum_i q_i \tilde{A}_i = \sum_i q_i \frac{1}{G p_i}\sum_{g=1}^{G} A^{(g)} \mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\} = \frac{1}{G}\sum_{g=1}^{G} A^{(g)} \frac{q(\mathbf{y}^{(g)})}{p(\mathbf{y}^{(g)})} = L_A(q).$$

Intuitively, $\tilde{A}_i$ measures the marginal value of assigning probability to sequence $\mathbf{y}_i$: transferring a small probability mass δ from $\mathbf{y}_j$ to $\mathbf{y}_i$ changes L_A by $\delta(\tilde{A}_i - \tilde{A}_j)$. Consequently, at the starting point, $\nabla_\theta L_A = G^{-1}\sum_g A^{(g)} \nabla_\theta \log \pi_\theta(\mathbf{y}^{(g)} \mid x)|_{\theta=\theta_{\text{old}}}$; the old policy and denominator are held fixed. Omitting $1/p_i$ from Eq. (A.2) would change the coefficient of sample g to $e^{-s_g} A^{(g)}/G$, which cannot generally be corrected by one global learning rate. The inverse-probability factor is needed only for the mathematical correspondence and is not explicitly computed by the algorithm; its difference from a direct logit update is analyzed in Appendix A.2.

Setup and regularity conditions. Fix a prompt x and consider only the positive-probability support of the old policy, $\{\mathbf{y}_i : p_i > 0\}$. Assume that this set is finite or countable, $H(p) < \infty$, the sample count G is finite, and every advantage $A^{(g)}$ is a finite real number; samples and advantages are fixed throughout the update. Since the score in Eq. (A.2) can be nonzero only on finitely many sampled rollouts and each has $p_i > 0$, the empirical score is bounded. Therefore $Z(\eta) = \sum_i p_i e^{\eta \tilde{A}_i}$ exists and is positive for every finite η. In a neighborhood of zero, $q_i^{(\eta)}/p_i$ and its derivative with respect to η are uniformly bounded, and each summand of the entropy derivative is dominated by a constant multiple of $p_i(1 + |\log p_i|)$. Interchanging differentiation and summation below is consequently valid. This argument is conditional on a fixed finite sampled group and does not mistake a finite first moment for exponential integrability of an arbitrary unbounded score. Taking a further expectation over prompts requires the corresponding additional integrability conditions.

Closed-form KL-proximal update. The stationarity condition for Eq. (A.1) under normalization gives $q_i \propto p_i \exp(\eta \tilde{A}_i)$. Normalizing over the old-policy support yields

$$\pi^{(\eta)}(\mathbf{y}_i \mid x) = \frac{\pi_{\text{old}}(\mathbf{y}_i \mid x) \exp(\eta \tilde{A}_i)}{\sum_j \pi_{\text{old}}(\mathbf{y}_j \mid x) \exp(\eta \tilde{A}_j)}. \tag{A.3a}$$

The updated policy converges to π_{old} as $\eta \to 0^+$, so we extend the update continuously to $\pi^{(0)} = \pi_{\text{old}}$. Equivalently, using the shorthand defined above,

$$q_i^{(\eta)} = \frac{p_i \exp(\eta \tilde{A}_i)}{Z(\eta)}, \qquad Z(\eta) = \sum_j p_j \exp(\eta \tilde{A}_j). \tag{A.3b}$$

To verify that this is the unique global optimizer rather than merely a stationary point, observe that for every $q \in \Delta_{\text{old}}$,

$$\sum_i q_i \tilde{A}_i - \frac{1}{\eta}\text{KL}(q\|p) = \frac{1}{\eta}\log Z(\eta) - \frac{1}{\eta}\text{KL}(q\|q^{(\eta)}).$$

By nonnegativity of KL, the right-hand side is maximized exactly at $q = q^{(\eta)}$; a distribution with $\text{KL}(q\|p) = \infty$ cannot improve the objective. This argument applies to finite and countable supports.

Taking logs of Eq. (A.3b) and differentiating yields

$$\frac{1}{q_i^{(\eta)}} \frac{dq_i^{(\eta)}}{d\eta} = \tilde{A}_i - \frac{Z'(\eta)}{Z(\eta)} = \tilde{A}_i - \mathbb{E}_{q^{(\eta)}}[\tilde{A}].$$

Hence

$$\frac{dq_i^{(\eta)}}{d\eta} = q_i^{(\eta)} \left(\tilde{A}_i - \mathbb{E}_{q^{(\eta)}}[\tilde{A}] \right). \tag{A.4}$$

At $\eta = 0$, $q_i^{(0)} = p_i$, and therefore

$$\left.\frac{dq_i^{(\eta)}}{d\eta}\right|_{\eta=0} = p_i \left(\tilde{A}_i - \mathbb{E}_p[\tilde{A}] \right). \tag{A.5}$$

Differentiating $H(q^{(\eta)}) = -\sum_i q_i^{(\eta)} \log q_i^{(\eta)}$ gives

$$\frac{dH(q^{(\eta)})}{d\eta} = -\sum_i \frac{dq_i^{(\eta)}}{d\eta} \left(1 + \log q_i^{(\eta)} \right).$$

Because $\sum_i q_i^{(\eta)} = 1$, we have $\sum_i dq_i^{(\eta)}/d\eta = 0$, so the constant term vanishes. Therefore,

$$\frac{dH(q^{(\eta)})}{d\eta} = -\sum_i \frac{dq_i^{(\eta)}}{d\eta} \log q_i^{(\eta)}.$$

Substituting $\eta = 0$ and Eq. (A.5) gives

$$\left.\frac{dH(q^{(\eta)})}{d\eta}\right|_{\eta=0} = -\sum_i p_i \left(\tilde{A}_i - \mathbb{E}_p[\tilde{A}] \right) \log p_i = -\text{Cov}_p(\tilde{A}, \log p). \tag{A.6}$$

This is the entropy derivative of the reward-driven KL-proximal update specified by Eq. (A.1); it does not apply to an arbitrary online-RL parameter update.

Proposition 1: per-sample contribution. For each sampled index g, define the score component $a_i^{[g]} = A^{(g)} \mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\}/(G p_i)$, so that $\tilde{A}_i = \sum_g a_i^{[g]}$. Since

$$\mathbb{E}_p[a^{[g]}] = \frac{A^{(g)}}{G}, \qquad \mathbb{E}_p[a^{[g]} \log p] = -\frac{A^{(g)}}{G} s_g, \qquad \mathbb{E}_p[\log p] = -H(p),$$

and Eq. (A.6) is linear in its score argument,

$$h_g = -\text{Cov}_p(a^{[g]}, \log p) = \frac{A^{(g)}}{G}\big(s_g - H(p)\big). \tag{A.7}$$

This proves Eq. (2). Throughout, g denotes a sampled index and i an index in the complete-sequence space. This additive decomposition includes the normalization-induced changes of other sequences, not only the entropy term of rollout g itself. Its sign is determined by $A^{(g)}(s_g - H(p))$: a relatively

high-probability positive-advantage rollout and a relatively low-probability negative-advantage rollout decrease entropy; the reverse combinations increase it. The contribution is zero if $A^{(g)} = 0$ or $s_g = H(p)$.

Proposition 2: group entropy derivative. Summing Eq. (A.7) over sampled indices gives the entropy derivative of $q^{(\eta)}$ at $\eta = 0$. Under the zero-sum condition of Proposition 2, $\mathbb{E}_p[\tilde{A}] = G^{-1}\sum_g A^{(g)} = 0$, so

$$\begin{aligned}\left.\frac{dH(q^{(\eta)})}{d\eta}\right|_{\eta=0} &= \sum_g h_g = \frac{1}{G}\sum_g A^{(g)} s_g \\ &= \frac{1}{G}\sum_g A^{(g)}(s_g - \bar{s}) = \mathrm{Cov}_{g=1}^{G}\big(A^{(g)}, s_g\big).\end{aligned} \tag{A.8}$$

This proves Eq. (3), using the same s_g and $\bar{s}$ as in the main text. Thus a negative empirical covariance means that the KL-proximal update driven by this fixed sampled group decreases entropy at the starting point, while a positive value means that it increases entropy. When the covariance is zero, only the first-order term vanishes; finite-step entropy change is undetermined. Eq. (A.8) is exact conditional on the current samples and weights, not an entropy guarantee for a neural-network update.

Monte Carlo evaluation and greedy selection. For the sequence-level criterion, we estimate $H(p)$ by the sample mean $\bar{s}$ of the existing G rollouts and substitute it into Eq. (2). Each $s_g = -\sum_{t=1}^{T_g} \log \pi_{\mathrm{old}}(y_t^{(g)} \mid x, \mathbf{y}_{<t}^{(g)})$ sums negative old-policy log-probabilities over valid response tokens; the length-normalized implementation is described below.

The covariance in Eq. (4) is computed over the original G rollouts. Since $\sum_{g=1}^{G}(s_g - \bar{s}) = 0$, centering the masked advantages leaves

$$\begin{aligned}\mathrm{Cov}_{g=1}^{G}\big(A^{(g)}\mathbf{1}\{g \in \mathcal{I}\}, s_g\big) &= \frac{1}{G}\sum_{g\in\mathcal{I}} A^{(g)}(s_g - \bar{s}) \\ &= \sum_{g\in\mathcal{I}} h_g|_{H(p)\to\bar{s}}\,.\end{aligned}$$

Thus the covariance constraint is exactly equivalent to a nonnegative sum of the remaining MC contributions. Recomputing the covariance on the retained subset would generally change the criterion because its mean surprisal need not equal $\bar{s}$.

For the greedy-selection argument, use h_g with the MC substitution specified in Section 3.2. Start with all G rollout indices. If their total contribution is negative, remove negative contributions in ascending order until the retained sum is nonnegative. Among sets with the same number of removals, removing the most negative contributions produces the largest retained sum, while removing nonnegative contributions cannot help. Thus the first feasible set uses the fewest removals and retains the most rollouts, solving Eq. (4). Removing all negative contributions leaves a nonnegative sum, so a feasible retained set exists. If $\sum_{g=1}^{G} h_g \geq 0$, retaining all G rollouts is optimal. This establishes maximum-cardinality optimality for the fixed MC contributions.

Length normalization in implementation. To avoid direct length bias from summing token log-probabilities, the implementation uses the length-normalized surprisal $s_g^{\mathrm{len}} = s_g/T_g$ for both filtering and recentering, with T_g counting valid response tokens. Filtering uses the original-group mean of s_g^{len}, and recentering uses weights proportional to $\exp(-s_g^{\mathrm{len}}) = \pi_{\mathrm{old}}(\mathbf{y}^{(g)} \mid x)^{1/T_g}$, computed by a softmax over retained rollouts. Thus responses with equal mean token log-probability receive equal recentering weights regardless of length. These normalized quantities are implementation proxies: their group mean is a filtering baseline rather than a complete-sequence entropy estimate, and the implemented zero-mean condition is defined by these weights. The theoretical results retain full sequence probabilities as their analysis model.

General objective and probability-ratio surrogate. Eq. (A.1) specifies a distribution-space KL-proximal geometry for analyzing the sampled reward update. Its penalty $\eta^{-1}\mathrm{KL}(q\|p)$ limits change relative to the old policy, whereas the reference penalty $\beta\,\mathrm{KL}(q\|\pi_{\mathrm{ref}})$ in the original RL objective penalizes deviation from a fixed reference model. The generalized step size η and reference-KL co-

efficient β serve different purposes; the proximal model analyzes the reward contribution separately from clipping, length weights, and reference regularization. For the reward term, subtracting a fixed baseline that depends only on x does not change the expected policy gradient (Williams, 1992; Sutton et al., 1999). For any fixed advantage function $A(x, \mathbf{y})$, if the relevant expectation exists and the support of q is contained in that of p, the probability-ratio identity gives

$$\sum_i q_i A(x, \mathbf{y}_i) = \sum_i p_i \frac{q_i}{p_i} A(x, \mathbf{y}_i) = \mathbb{E}_{i\sim p}\left[\frac{q_i}{p_i} A(x, \mathbf{y}_i)\right].$$

Taking $A^{(g)} = A(x, \mathbf{y}^{(g)})$ gives a sampled estimator of this expectation. More generally, an algorithm can construct advantage estimates from the full sampled group, in which case we directly define the empirical surrogate conditional on the current samples:

$$L_A(q) = \frac{1}{G}\sum_{g=1}^{G} A^{(g)} \frac{q(\mathbf{y}^{(g)})}{p(\mathbf{y}^{(g)})},$$

$$\nabla_\theta L_A(\pi_\theta(\cdot \mid x))|_{\theta=\theta_{\text{old}}} = \frac{1}{G}\sum_{g=1}^{G} A^{(g)}\ \nabla_\theta \log \pi_\theta(\mathbf{y}^{(g)} \mid x)\Big|_{\theta=\theta_{\text{old}}}.$$

The key condition for extending the local surrogate is therefore that the algorithm's reward gradient at the starting point can be expressed in this rollout-level weighted form. Group-dependent advantage estimates are generally correlated with the current rollout, so the probability-ratio identity alone does not make this an unbiased estimator of the original expected-return gradient.

Sequence-level policy gradients. When a complete generated sequence is treated as one action, the REINFORCE sampled gradient corresponds to $A^{(g)} = R(x, \mathbf{y}^{(g)}) - b(x)$ for a fixed prompt-level baseline $b(x)$. An actor–critic update that treats the sequence as one action and holds its advantage fixed while differentiating has the same form. This correspondence concerns the reward gradient and excludes gradients from additional regularizers.

Group-relative policy optimization. For outcome-supervised GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025), define the native group-relative advantage by

$$A^{(g)} = \frac{R(x, \mathbf{y}^{(g)}) - \bar{R}}{\sigma_R}, \qquad \bar{R} = \frac{1}{G}\sum_{g=1}^{G} R(x, \mathbf{y}^{(g)}), \qquad \sigma_R = \left[\frac{1}{G}\sum_{g=1}^{G}(R(x, \mathbf{y}^{(g)}) - \bar{R})^2\right]^{1/2}.$$

If $\sigma_R = 0$, all advantages are defined as zero, so $\sum_g A^{(g)} = 0$ always holds. Under outcome supervision, every token within a rollout shares $A^{(g)}$. Original GRPO averages the token loss within each rollout, whereas GSPO uses the length-normalized sequence probability ratio $[q(\mathbf{y}^{(g)})/p(\mathbf{y}^{(g)})]^{1/T_g}$. Ignoring the reference KL and differentiating at $\theta = \theta_{\text{old}}$ with clipping inactive, both correspond to replacing $A^{(g)}$ in L_A by $A^{(g)}/T_g$. If GRPO sums token losses rather than averaging within each rollout, it corresponds to L_A without length normalization. These correspondences match parameter gradients only at the starting point and do not make the clipped objectives or their off-starting-point gradients identical.

A.2 Proof of Proposition 3

Independent-logit update and entropy derivative. Restrict the complete-sequence space to be finite, associate each sequence with an independent parameter z_i, and let $p = \mathrm{softmax}(z)$. Fix a sampled group with zero-sum advantages $\sum_g A^{(g)} = 0$ and perform ordinary gradient ascent on $L(z) = G^{-1}\sum_g A^{(g)} \log p(\mathbf{y}^{(g)})$ by $z(\alpha) = z + \alpha \nabla_z L(z)$, using the same learning rate α for every coordinate. The gradient with respect to each logit and the corresponding entropy derivative are

$$\begin{aligned} \frac{\partial L}{\partial z_i} &= \frac{1}{G}\sum_g A^{(g)}\big(\mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\} - p_i\big) = \frac{1}{G}\sum_g A^{(g)}\mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\}, \\ \frac{\partial H(p)}{\partial z_i} &= p_i(-\log p_i - H(p)). \end{aligned} \tag{A.9}$$

Therefore

$$\left.\frac{dH(\mathrm{softmax}(z(\alpha)))}{d\alpha}\right|_{\alpha=0} = \sum_i \frac{\partial H(p)}{\partial z_i}\frac{\partial L}{\partial z_i} = \sum_g \underbrace{\frac{e^{-s_g}A^{(g)}}{G}(s_g - H(p))}_{h_g^{\mathrm{SGD}}}. \tag{A.10}$$

Finite-dimensional smoothness also gives $H(\mathrm{softmax}(z(\alpha))) - H(p) = \alpha \sum_g h_g^{\mathrm{SGD}} + O(\alpha^2)$. The probability factor e^{-s_g} arises from the entropy gradient with respect to logits, not from weighting a sampled rollout a second time. If $\sum_g A^{(g)} \neq 0$, Eq. (A.10) acquires the additional term $-\big(G^{-1}\sum_g A^{(g)}\big)\sum_i p_i^2(-\log p_i - H(p))$. The contribution h_g^{SGD} is therefore an additive attribution within a zero-sum whole-group update, not the complete entropy derivative obtained by training one rollout in isolation; it also cannot be reused automatically after deletion or length weighting.

Conditional dominance of a high-probability positive rollout. Assume $A_+ > 0 > A_-$, $s_+ < H(p) < s_-$, and equal entropy contributions $h_+ = h_- < 0$ in Proposition 1. By Eq. (2), this equality is equivalent to $A_+(H(p) - s_+) = |A_-|(s_- - H(p)) > 0$. By Eq. (A.10), both rollouts decrease entropy, and their contribution magnitudes satisfy

$$\frac{|h_+^{\mathrm{SGD}}|}{|h_-^{\mathrm{SGD}}|} = \frac{e^{-s_+}A_+(H(p) - s_+)}{e^{-s_-}|A_-|(s_- - H(p))} = e^{s_- - s_+} > 1. \tag{A.11}$$

This proves Eq. (6). The comparison concerns two rollouts whose entropy-decrease magnitudes before probability weighting are equal; it neither requires equal absolute advantages nor says that every positive-advantage rollout is more influential than every negative one.

Scale difference from the KL-proximal update. For the same sampled group with $\sum_g A^{(g)} = 0$, Eq. (A.2) satisfies $\tilde{A}_i = p_i^{-1}\partial L/\partial z_i$. Eq. (A.3a) can therefore be written $q^{(\eta)} = \mathrm{softmax}(z + \eta\tilde{A})$. The two logit-coordinate changes are

$$\Delta z_i^{\mathrm{KL}} = \frac{\eta}{p_i}\frac{\partial L}{\partial z_i}, \qquad \Delta z_i^{\mathrm{SGD}} = \alpha\frac{\partial L}{\partial z_i}. \tag{A.12}$$

These are logit representatives of the corresponding distributions; adding the same constant to every coordinate does not change softmax. Section 3.2 also uses one global parameter η, but relative to the ordinary logit gradient its effective scale is η/p_i, whereas Section 3.3 uses the shared gradient multiplier α. The inverse-probability factor in the former cancels the p_i in the entropy gradient; the latter retains that probability factor. This is a difference between update rules, not a difference between having and not having a step size, and it does not make the independent-logit model more accurate than the KL-proximal model.

Scope of the conclusion. Proposition 3 is a first-order comparison under independent logits, zero-sum whole-group advantages, and the stated equal-magnitude condition. It is not a direct consequence of the KL-proximal update in Section 3.2 and does not establish positive-sample dominance for arbitrary finite steps or actual neural-network updates. It supplies additional motivation for the final positive-advantage filter. The maximum-cardinality property of Eq. (4) applies only to the initial retained set with fixed original-group scores; the filtering in Eq. (7) and recentering in Eq. (9) do not inherit its threshold constraint or any global optimality claim. These results are rigorous within their stated local models and guide method design, while the experiments in Sections 4.1 and 4.2 evaluate the complete method. They do not guarantee entropy preservation under shared-parameter AdamW updates.

A.3 Proof of Proposition 4

First-order change of zero-advantage sequences. With the native advantages fixed and satisfying $\sum_{g=1}^{G} A^{(g)} = 0$, deleting a nonempty set $\mathcal{D}$ of positive-advantage rollouts leaves $\sum_{g\in\mathcal{I}_{\mathrm{keep}}} A^{(g)} = -\sum_{g\in\mathcal{D}} A^{(g)} < 0$. Retention depends on rollout content and within-group comparisons, so the retained samples cannot generally be treated as fresh i.i.d. draws from p. These changes motivate reconsidering the advantage baseline after selection, but do not by themselves determine the probability weights.

Fix a prompt x, a realized group of rollouts sampled i.i.d. from p, a nonempty retained set $\mathcal{I}_{\mathrm{keep}}$, and a bounded post-deletion advantage $\tilde{A}_i^{\mathrm{keep}}$. To isolate normalization effects, assume the local

update $q^{(\eta)}_{\mathrm{keep},i} = p_i \exp(\eta \tilde{A}^{\mathrm{keep}}_i)/Z_{\mathrm{keep}}(\eta)$, where $Z_{\mathrm{keep}}(\eta) = \sum_j p_j \exp(\eta \tilde{A}^{\mathrm{keep}}_j)$. This has the same KL-proximal form as Appendix A.1, but uses the accumulated retained advantages as its score rather than the sampling-aligned score in Eq. (A.2). The distribution-space update couples paths only through its normalizing denominator and contains no shared-parameter effects. Let $U = \{i : p_i > 0,\ \tilde{A}^{\mathrm{keep}}_i = 0\}$. The set U contains sequences that receive no retained signal and may also include sampled sequences whose recentered advantage is zero or whose signals cancel; it must not be identified with the set of unsampled sequences. Under this update, $q^{(\eta)}_{\mathrm{keep},i} = p_i/Z_{\mathrm{keep}}(\eta)$ for $i \in U$. For $i \in U$, the unnormalized weight remains p_i, while normalization multiplies all such probabilities by the same factor $1/Z_{\mathrm{keep}}(\eta)$. Equivalently, every zero-advantage path receives the same additive log-probability offset $-\log Z_{\mathrm{keep}}(\eta)$, regardless of its similarity to the trained rollouts, and probability ratios between zero-advantage paths remain unchanged. In particular, any two paths $i, j \in U$ satisfy $q^{(\eta)}_{\mathrm{keep},i}/q^{(\eta)}_{\mathrm{keep},j} = p_i/p_j$: normalization cannot preferentially favor the one more similar to a rewarded rollout. The i.i.d. assumption concerns the original sampling group. The derivation is conditional on the realized group and retained set and does not require the retained samples to remain i.i.d.; excluding transfer through shared parameters is a separate assumption of the local update model. Using $Z_{\mathrm{keep}}(0) = 1$ and $Z'_{\mathrm{keep}}(0) = \mathbb{E}_p[\tilde{A}^{\mathrm{keep}}]$ gives

$$\left.\frac{dq^{(\eta)}_{\mathrm{keep},i}}{d\eta}\right|_0 = -p_i \mathbb{E}_p[\tilde{A}^{\mathrm{keep}}], \qquad \left.\frac{d \log q^{(\eta)}_{\mathrm{keep},i}}{d\eta}\right|_0 = -\mathbb{E}_p[\tilde{A}^{\mathrm{keep}}], \quad i \in U. \tag{A.13}$$

This proves Eq. (8). At $\eta = 0$, the identical log-probability derivatives imply equal relative changes across zero-advantage paths, while their absolute changes are scaled by p_i. When U is nonempty, every sequence in it has zero first-order probability change if and only if $\mathbb{E}_p[\tilde{A}^{\mathrm{keep}}] = 0$. If U is empty, there are no such sequences and one cannot infer a zero mean from a condition on an empty set. The result does not imply finite-step invariance: for a nontrivial post-deletion advantage with zero mean, typically $Z_{\mathrm{keep}}(\eta) > 1$ for $\eta > 0$, so zero-advantage sequences can still change at second and higher orders. Normalization here occurs only over the response space of a fixed prompt and does not prove cross-prompt or OOD invariance after a shared-network update.

Subset entropy and global entropy attribution. Let $H_U(q) = -\sum_{i \in U} q_i \log q_i$ be the unnormalized entropy of the subset U. Eq. (A.13) yields

$$\left.\frac{dH_U(q^{(\eta)}_{\mathrm{keep}})}{d\eta}\right|_0 = \mathbb{E}_p[\tilde{A}^{\mathrm{keep}}]\big(\sum_{i \in U} p_i - H_U(p)\big), \qquad -\sum_{i \in U} q'_{\mathrm{keep},i}(0) \log p_i = -\mathbb{E}_p[\tilde{A}^{\mathrm{keep}}] H_U(p). \tag{A.14}$$

The left expression is the derivative of the unnormalized subset entropy itself. The right expression is the contribution assigned to U after the constant term has been canceled globally using $\sum_i q'_{\mathrm{keep},i}(0) = 0$. They must not be conflated, and the sign of $\mathbb{E}_p[\tilde{A}^{\mathrm{keep}}]$ alone does not determine the direction of global entropy. The general global expression remains $-\mathrm{Cov}_p(\tilde{A}^{\mathrm{keep}}, \log p)$ and reduces to the empirical covariance in Eq. (3) only for the original zero-sum score specified in Appendix A.1.

Which zero-mean condition probability-weighted recentering satisfies. For the nonempty retained set in Eq. (9), replacing the native advantages in the accumulated post-deletion advantage by the recentered advantages gives

$$\tilde{A}^{\mathrm{keep}}_i = \sum_{g \in \mathcal{I}_{\mathrm{keep}}} \bar{A}^{(g)}_{\mathrm{keep}} \mathbf{1}\{\mathbf{y}_i = \mathbf{y}^{(g)}\}, \qquad \mathbb{E}_p[\tilde{A}^{\mathrm{keep}}] = \sum_{g \in \mathcal{I}_{\mathrm{keep}}} e^{-s_g} \bar{A}^{(g)}_{\mathrm{keep}} = 0. \tag{A.15}$$

The last equality follows directly from $\bar{A}^{(g)}_{\mathrm{keep}} = A^{(g)} - b_{\mathrm{keep}}$ and $b_{\mathrm{keep}} = \sum_{g \in \mathcal{I}_{\mathrm{keep}}} e^{-s_g} A^{(g)} / \sum_{g \in \mathcal{I}_{\mathrm{keep}}} e^{-s_g}$. Eq. (9) therefore eliminates the first-order normalization drift of zero-advantage sequences in the KL-proximal model driven by this particular post-deletion advantage. The probability weights arise from the old-policy expectation in Eq. (A.13): Eq. (9) imposes this weighted zero-mean condition on the retained score without assuming that it held before deletion. Subtracting a common baseline preserves pairwise differences between retained advantages; the weights calibrate the retained score rather than reconstruct the original sampling distribution through importance weighting. Advantage contributions are accumulated by sampled index; when sequences repeat, the denominator $\sum_{g \in \mathcal{I}_{\mathrm{keep}}} e^{-s_g}$ counts each sampled occurrence and need not equal the probability mass of the distinct retained sequences.

B EXPERIMENTAL DETAILS

B.1 EXPERIMENTAL SETUP

Training settings. We conduct experiments on Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Team, 2025), using DAPO-17K mathematics data (Yu et al., 2025) for reinforcement learning. All model sizes use the same training configuration and are trained for one epoch. Each training batch contains 128 prompts with eight rollouts per prompt, totaling 1,024 candidate rollouts before selection. The maximum prompt and response lengths are 2,048 and 8,192 tokens, respectively, and overlong prompts are filtered. Rollouts are sampled with vLLM at temperature 1.0 and top-p 1.0. Original GRPO uses all eight rollouts; GRPODropout removes selected rollouts and recenters the retained advantages before updating the policy. We use AdamW with learning rate 10^{-6} and weight decay 0.01, a clipping width of 0.2, and a policy loss averaged over valid response tokens. The original GRPO baseline uses no additional entropy bonus or reference-policy KL regularization.

Baselines. We use original GRPO (Shao et al., 2024) as the base control and compare representative methods from the two categories discussed in Related Work. The first category selects or masks token-level training signals: Beyond8020 (Wang et al., 2025b) optimizes only the approximately 20% highest-entropy tokens, while Clip-Cov (Cui et al., 2025) masks updates for a subset of high-covariance tokens. The second category modifies the RL objective or update rule: KL-Cov (Cui et al., 2025) applies KL constraints to selected high-covariance tokens, while DAPO (Yu et al., 2025) uses Clip-Higher to relax the upper clipping bound and facilitate exploration of low-probability actions. These baselines cover token-level data usage, regularization, and clipping, allowing us to compare complete-rollout selection with different strategies for mitigating entropy collapse. GRPO-Dropout operates on data usage, applying the original RL loss to retained rollouts with recentered advantages while preserving its objective form.

Evaluation settings. We evaluate on ten benchmarks: GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2023), OlympiadBench (He et al., 2024), AMC23, AIME24, AIME25, and HMMT25 for mathematical reasoning, and ARC-C (Clark et al., 2018), MMLU-Pro (Wang et al., 2024), and SuperGPQA (Team et al., 2025) for general question answering. The latter evaluates generalization beyond the mathematics training domain. All evaluations use a maximum generation length of 8,192 tokens, temperature 0.6, top-p 0.95, top-k 20, and a fixed random seed. GSM8K, MATH-500, OlympiadBench, and the three QA benchmarks use one sampled response per question and report Mean@1. For the smaller AMC23, AIME24, AIME25, and HMMT25 test sets, we sample 32 responses per question through repeated single-response requests and report Mean@32, averaging correctness over the 32 responses and then over questions. Mathematical prompts request step-by-step reasoning and a final answer in `\boxed{}`. QA prompts request a final option letter, which is extracted, case-normalized, and compared with the reference label. Avg. is the unweighted mean accuracy across all ten benchmarks.

B.2 DIVERSITY-ANALYSIS PROTOCOL

This subsection records the protocol used to compute the path-diversity columns in Table 3. The analysis covers 130 unique questions in AMC23 (40), AIME24 (30), AIME25 (30), and HMMT25 (30), with 32 sampled responses per question and system. Pass@32 is computed per question within each benchmark and then averaged equally across AMC23, AIME24, AIME25, and HMMT25. The Entropy column uses token-mean actor entropy over valid response tokens at the final checkpoint, averaged equally across the four benchmarks. All path-diversity aggregates are unweighted means over questions.

Question partition and answer dispersion. Let $c_{m,q}$ be the number of correct responses among the 32 rollouts for method m on question q. We define the common high-correctness set as

$$\mathcal{Q}_{\geq 8} = \bigcap_m \{q : c_{m,q} \geq 8\}, \qquad |\mathcal{Q}_{\geq 8}| = 65,$$

and the common low-correctness set as

$$\mathcal{Q}_{<8} = \bigcap_m \{q : c_{m,q} < 8\}, \qquad |\mathcal{Q}_{<8}| = 37.$$

Thus the threshold is based on the original rollout correctness counts, and the same question sets are used for every method. The remaining 28 questions have mixed membership across systems and are not used in the two split columns. For each method and question, we compute

$$W_{m,q} = \left|\{\mathrm{strip}(a_i) : r_i = 0,\ \mathrm{strip}(a_i) \neq \emptyset\}\right|,$$

where a_i is the extracted final answer and r_i is its correctness indicator. W measures answer-string dispersion among incorrect final answers and can reveal recurring answer patterns.

Summary generation and embedding. For each rollout, the summarizer receives the original question and the full generated response after removing explicit `<think>` tags. We use the model `GLM-5.2`, temperature 0, with thinking disabled. The summarizer is instructed to retain only the reasoning path that actually leads to the final boxed answer, place abandoned ideas in a separate list, and return strict JSON. Its exact prompt is:

GLM-5.2 SUMMARIZER PROMPT

```
You compress ONE chain-of-thought solution to a math problem.

The trace is a stream-of-consciousness: the solver often floats several ideas,
tries some, abandons most, and finally carries ONE line of reasoning through to
the boxed answer. Your job is to separate the two:

- FINAL PATH: the reasoning that was ACTUALLY carried through to produce the
  final (boxed) answer. This is the solution's real solving path. Identify it by
  looking at how the last, answer-producing derivation actually went -- NOT by
  what was merely proposed earlier.
- ATTEMPTED: ideas that were floated, tried, then dropped / not used for the
  final answer. Record each briefly with why it was abandoned.

Be faithful. Do NOT invent steps. If the solver kept switching, the FINAL PATH
is the one whose result equals the boxed answer at the very end.

Output STRICT JSON only:
{
  "final_path": {
    "summary": "<3-5 sentences: the actual path that yielded the answer>",
    "method": "<short tag: main machinery, e.g. cartesian-algebra / polar-form / calculus
    ↪  / geometric-series-identity>",
    "angle": "<short tag: the entry point / first decisive framing that led to the
    ↪  answer>",
    "key_trick": "<short tag: the crucial simplification/insight used, or 'none'>"
  },
  "attempted": [
    {"idea": "<short: an approach that was tried>", "why_abandoned": "<short>"}
  ]
}
```

The returned summary is encoded with the local `bge-m3` checkpoint with L2 normalizations, and float32 arithmetic. For each question, Summary distance is the mean pairwise cosine distance over all 32 summary embeddings. Vendi is computed from the eigenvalues of the symmetrized Gram matrix divided by 32 as $\exp(-\sum_j \lambda_j \log \lambda_j)$. Both measures therefore use all 32 summaries, including correct and incorrect responses.