THE CANCELLATION HYPOTHESIS IN CRITIC-FREE RL: FROM OUTCOME REWARDS TO TOKEN CREDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

A commonly accepted explanation of critic-free RL (e.g., GRPO) for LLMs, based on sequence-level rewards, is that it reinforces successful rollouts with a positive advantage while penalizing failed ones. In contrast, we study critic-free RL from a token-level perspective, revealing the *token-flipping* phenomenon: positive and negative rollouts exhibit remarkably similar proportions of tokens whose probabilities are boosted or suppressed during RL training. To explain this phenomenon, we further show that a token's change in probability is not fully determined by its own advantage; coupled gradient interactions with other tokens also play a non-negligible role. Specifically, these *token coupling effects* occur primarily between identical tokens that are both predicted with low confidence. Building upon this analysis, we propose the *cancellation hypothesis*: as a result of coupling, opposing signals cancel out for tokens shared by positive and negative rollouts, while tokens more specific to successful rollouts receive stronger reinforcement, thereby inducing hidden token-level credit assignment from rollout-level rewards. We support this hypothesis with complementary empirical evidence. (1) Compared with training on only positive rollouts, critic-free RL shifts updates from template and formatting tokens toward reasoning tokens; (2) Tokens boosted by critic-free RL consistently demonstrate higher value than suppressed tokens, regardless of whether they originate from positive or negative rollouts. Guided by this view, we implement two batching interventions to encourage or preserve cancellation in critic-free RL training: query-preserved mini-batching, which keeps rollouts from the same prompt together to encourage token coupling, and reward-balanced batching, which ensures that positive and negative rollouts are balanced in each update. Despite their simplicity, these interventions improve RLVR training across multiple model scales, supporting cancellation as both an explanatory principle and a practical design criterion for critic-free RL training.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become one of the standard recipes for post-training large language models (LLMs) (Guo et al., 2025; Jaech et al., 2024; Team et al., 2025). Among RLVR methods, critic-free approaches such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and its variants (Yu et al., 2025; Liu et al., 2025) are widely adopted for their simplicity. They remove the value network (Schulman et al., 2017) and instead assign a single, shared advantage to every token within a rollout. Despite such coarse, rollout-level supervision, these methods achieve strong results across reasoning and broader agentic domains (Guo et al., 2025). This success motivated growing interest in understanding how critic-free RL optimizes and reshapes model behavior. A common view is that RL reinforces successful rollouts and suppresses failed ones, thereby shifting the model distribution toward higher-reward responses (Mroueh, 2025; Wu et al., 2025b). Recent work complicates this view by analyzing the different contributions of positive and negative rollouts to optimization dynamics (Zhu et al., 2025a; Tang et al., 2025; Deng et al., 2025a;b;c; Simoni et al., 2025; Xue et al., 2025).

However, a fundamental disconnect remains in the current discourse: most existing analyses focus on rollout-level likelihoods or outcomes, while treating the token-level dynamics as a black box. Though the model improves, it still lacks a mechanistic to understand how a shared rollout-level

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

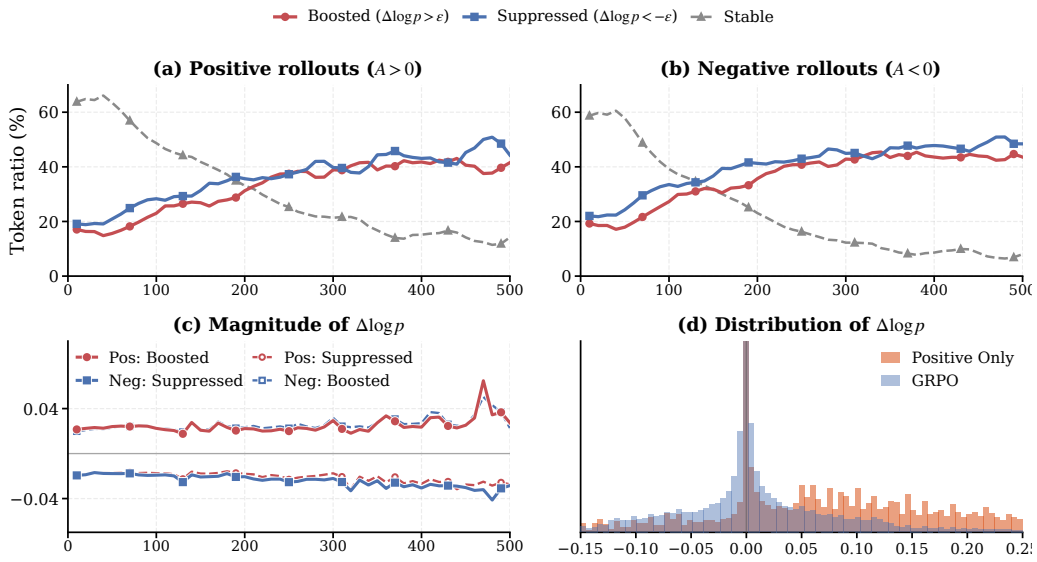


Figure 1: Token-level updates do not simply follow rollout-level advantage signs. Tokens could be categorized into three groups according to their change in log probability: boosted, suppressed, and stable. Figures (a) and (b) show that positive and negative rollouts exhibit remarkably similar boosted/suppressed/stable token ratios during training. Figure (c) shows that the update magnitudes are also similar. Figure (d) further confirms that the flipping pattern greatly decreases when training with only positive rollouts, highlighting the coupling effect between positive and negative rollouts.

advantage distributes credit or blame across tokens. Under the *de-facto* sequence-level intuition, tokens in successful rollouts should be mostly reinforced, and vice versa.

In this work, we bridge this gap by shifting the analytical lens from rollouts to tokens. We measure the shift in log-probability for every token before and after individual GRPO training step, as shown in Fig. 1(a,b). Our findings reveal a clear departure from conventional intuition: tokens in successful rollouts are not mostly reinforced, nor are tokens in failed rollouts mostly suppressed. Instead, nearly half of the tokens move in the direction opposite to what their rollout-level advantage sign suggests. More surprisingly, positive and negative rollouts exhibit remarkably similar ratios of boosted and suppressed tokens during training, even the magnitudes of these token-level changes are also similar (Fig. 1(c)). We term this mismatch *Token Flipping*, which suggests that the rollout-level advantage sign is a poor predictor for token-level updates, implying that the learning mechanism in critic-free RL is more granular than the standard sequence-level view suggests. Furthermore, this pattern largely disappears under positive-rollout-only training, where negative rollouts are masked (Fig. 1(d)), indicating that token flipping arises from interactions between positive and negative rollouts.

The *Token Flipping* phenomenon suggests that a token's update cannot be explained by its advantage alone. To understand where the additional force arises, we revisit GRPO's training objective at the token level. A first-order expansion shows that the log-probability change of token j can be decomposed into a self-term and a within-batch coupling term. The former depends on its own advantage, while the latter aggregates interactions with all other tokens within the batch. We then ask when such coupling is large enough to affect the update. By approximating the full gradient with an output-layer proxy, we obtain an interpretable factorization of the coupling kernel into a representation term and a distributional term. This factorization reveals a sparse structure: non-negligible coupling occurs primarily between identical tokens that are both predicted with low confidence. This helps explain why token-level updates could deviate from their own advantage signs, and suggests that the joint composition of an RL batch plays a central role in shaping the effective learning signal.

Building on this sparse coupling structure, we propose the *cancellation hypothesis*: tokens that appear in both positive and negative rollouts receive opposing coupled signals that cancel in the net update, while tokens more specific to successful rollouts retain stronger reinforcement. In this way, batch-level token coupling can induce hidden token-level credit assignment from rollout-level rewards,

even without an explicit critic: (1) compared with training on only positive rollouts, GRPO greatly weakens updates on template and formatting tokens and shifts emphasis toward reasoning tokens (Fig. 4). (2) we estimate token value by Monte Carlo rollouts and find that GRPO-boosted tokens consistently have higher estimated value than suppressed tokens, regardless of whether they originate from positive or negative rollouts (Fig. 5(a)). This value gap further peaks in an intermediate-entropy region, connecting cancellation to entropy-based token selection (Wang et al., 2025).

Finally, the cancellation view leads us to implement two simple yet effective interventions for RL training. (1) *query-preserved mini-batching*, which keeps rollouts from the same query together to ensure related positive and negative samples are within the same optimizer step. (2) *reward-balanced batching*, which avoids batches that are overwhelmed by positive or negative rollouts to preserve the cancellation structure. Across multiple model scales and math benchmarks, these interventions improve RLVR training, demonstrating the practical value of preserving the cancellation hypothesis.

2 Token Coupling Effect

The *Token Flipping* suggests that a token's update direction may be shaped by other tokens in the same batch, rather than by its own advantage alone. In this section, we formalize this batch-level token coupling for a single RL update. Unless stated otherwise, we consider GRPO with binary rewards: for each prompt q, a group of G responses $\{o_i\}_{i=1}^{G}$ is sampled from the old policy $\pi_{\theta_{\text{old}}}$, and each response is assigned a rollout-level advantage $\hat{A}_i$ after verification. We study the change in the token's log-probability before and after an update step. For the t-th token in response i, denoted $o_{i,t}$,

$$\Delta \log p_{i,t} \triangleq \log \pi_{\theta'}(o_{i,t} \mid q, o_{i,<t}) - \log \pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t}) \,. \tag{1}$$

For this analysis, we rewrite the GRPO update as a weighted token log-likelihood:

$$\mathcal{J}(\theta) \;\triangleq\; \frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} A_i \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}), \qquad N \triangleq \sum_{i=1}^{G} |o_i|. \tag{2}$$

A_i denotes the update weight obtained from the original rollout advantage $\hat{A}_i$. As the importance-sampling ratio and clipping term only rescale or mask the gradient, we absorb them into A_i. Under a first-order approximation, a single gradient step on $\mathcal{J}(\theta)$ with step size η yields

$$\theta' \;=\; \theta + \eta \nabla_\theta \mathcal{J}(\theta) \;=\; \theta + \frac{\eta}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} A_i \, \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}). \tag{3}$$

We now analyze the $\Delta \log p_{i,t}$ under the one-step update: let $g_{i,t} \triangleq \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$ denote the score function with respect to full parameters θ, a first-order Taylor expansion gives

$$\Delta \log p_{i,t} \;\approx\; \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})^\top (\theta' - \theta) \;=\; g_{i,t}^\top (\theta' - \theta). \tag{4}$$

Substituting Eq. (3) into Eq. (4) and flattening tokens in the batch into indices $j, k \in \{1, \ldots, N\}$:

$$\Delta \log p_j \approx \frac{\eta}{N} \sum_{k=1}^{N} A_k \langle g_j, g_k \rangle = \frac{\eta}{N} \left(A_j \|g_j\|_2^2 + \sum_{k \neq j} A_k \langle g_j, g_k \rangle \right) \tag{5}$$

where A_k denotes the effective weight inherited from the rollout of token k. Eq. 5 clearly reveals that the update of token j is not only driven by the advantage A_j, but also the coupling term $\sum_{k \neq j} A_k \langle g_j, g_k \rangle$. We define the coupling kernel $K_{j,k} \triangleq \langle g_j, g_k \rangle$, which measures the coupling strength between tokens j and k. Computing this kernel in the full parameter space is expensive, since it requires inner products between full-model gradients. We therefore follow Deng et al. (2025b) and adopt the *output-layer proxy*, which approximates g_j using only the gradient with respect to the final unembedding matrix, denoted $g_j^{(W)}$. Appendix D.1 shows that this proxy closely tracks the full-parameter update in practice. Let $z_\theta(s) = W h_\theta(s)$ denote the logits, where $h_\theta(s)$ is the final hidden state, the gradient with respect to W becomes

$$g_j^{(W)} \;\triangleq\; \nabla_W \log \pi_\theta(o_j \mid s_j) \;=\; \underbrace{(\nabla_z \log \pi_\theta(o_j \mid s_j))}_{\text{error term } r_j} \underbrace{h_j^\top}_{\text{input}} \;=\; r_j h_j^\top, \tag{6}$$

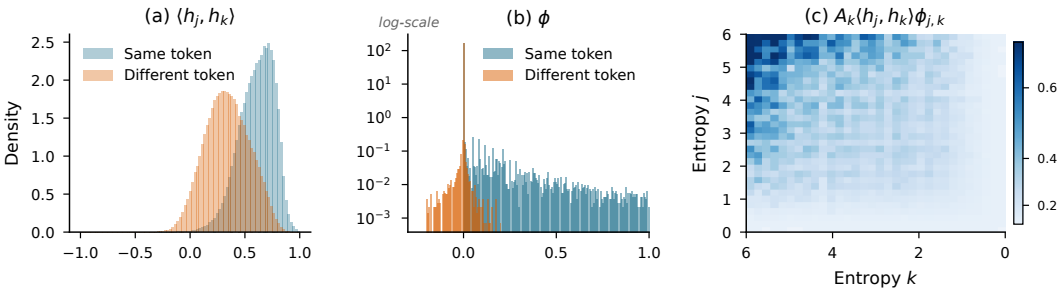


Figure 2: Coupling concentrates on same-token, low-confidence pairs. **(a)** Hidden-representation similarities $\langle h_j, h_k \rangle$ are generally positive across token pairs, with same-token pairs showing stronger alignment than different-token pairs. **(b)** The output-distribution factor $\phi_{j,k}$ is non-negligible primarily for same-token pairs. **(c)** The full coupling contribution $A_k \langle h_j, h_k \rangle \phi_{j,k}$ is strongest when both tokens have high entropy, corresponding to low-confidence predictions.

where s_j is the prefix state of token j, $h_j = h_\theta(s_j)$, and $r_j \triangleq e_{o_j} - \pi_\theta(\cdot \mid s_j) \in \mathbb{R}^{|\mathcal{V}|}$. Substituting Eq. (6) into the kernel definition and using the Frobenius inner product property $\langle uv^\top, xy^\top \rangle_F = \langle u, x \rangle \langle v, y \rangle$, we obtain the output-layer proxy factorization:

$$K_{j,k} \approx \langle g_j^{(W)}, g_k^{(W)} \rangle_F = \langle r_j h_j^\top, r_k h_k^\top \rangle_F = \underbrace{\langle h_j, h_k \rangle}_{\text{Representation}} \cdot \underbrace{\langle r_j, r_k \rangle}_{\text{Distribution}}. \tag{7}$$

Thus, non-negligible coupling requires both aligned hidden representations and high similarity between the output distributions. We denote the second factor by $\phi_{j,k} \triangleq \langle r_j, r_k \rangle$ and expand it:

$$\phi_{j,k} = (e_{o_j} - \pi_j)^\top (e_{o_k} - \pi_k) = \mathbb{I}[o_j = o_k] - \pi_j(o_k) - \pi_k(o_j) + \langle \pi_j, \pi_k \rangle. \tag{8}$$

Substituting this factorized kernel back into Eq. 5, we obtain the proxy delta:

$$\delta_j \approx \frac{\eta}{N} \sum_{k=1}^{N} A_k \langle h_j, h_k \rangle \phi_{j,k} = \frac{\eta}{N} \left(A_j \langle h_j, h_j \rangle \phi_{j,j} + \sum_{k \neq j} A_k \langle h_j, h_k \rangle \phi_{j,k} \right). \tag{9}$$

We now inspect $\phi_{j,k}$ in two cases:

$$\phi_{j,k} = \begin{cases} -\pi_j(o_k) - \pi_k(o_j) + \langle \pi_j, \pi_k \rangle, & o_j \neq o_k, \\ (1 - \pi_j(o))(1 - \pi_k(o)) + \sum_{v \neq o} \pi_j(v) \pi_k(v), & o_j = o_k = o. \end{cases} \tag{10}$$

For distinct output tokens $o_j \neq o_k$, the cross-token probabilities $\pi_j(o_k)$ and $\pi_k(o_j)$ are usually small, and the overlap term $\langle \pi_j, \pi_k \rangle$ is also small when the two output distributions have weak overlap. Thus, different-token pairs typically contribute little through $\phi_{j,k}$. For same-token pairs $o_j = o_k = o$, the non-target overlap is bounded by the uncertainty product, yielding $(1 - \pi_j(o))(1 - \pi_k(o)) \leq \phi_{j,k} \leq 2(1 - \pi_j(o))(1 - \pi_k(o))$. Thus, $\phi_{j,k}$ is controlled within a factor of two by the two prediction uncertainties. Consequently, under the small-overlap conditions above and for comparable hidden-state inner products, coupling is preferentially concentrated among same-token pairs for which both predictions have low confidence.

Empirical Verification We first look into the similarity of hidden representation and output distribution between sampled tokens in Eq. 7 and the sparsity pattern predicted by Eq. 10. As shown in Fig. 2(a), hidden-representation similarities are generally positive across token pairs, and same tokens tend to be aligned more strongly than different-token pairs. Fig. 2(b) further shows that non-negligible $\phi_{j,k}$ values are primarily on the same-token pairs, though most of them are zero. Finally, Fig. 2(c) shows that the full coupling contribution is strongest when both tokens have high entropy, corresponding to low-confidence predictions. Together, these observations support our factorization: *non-negligible coupling is sparse and occurs primarily between identical, low-confidence tokens.*

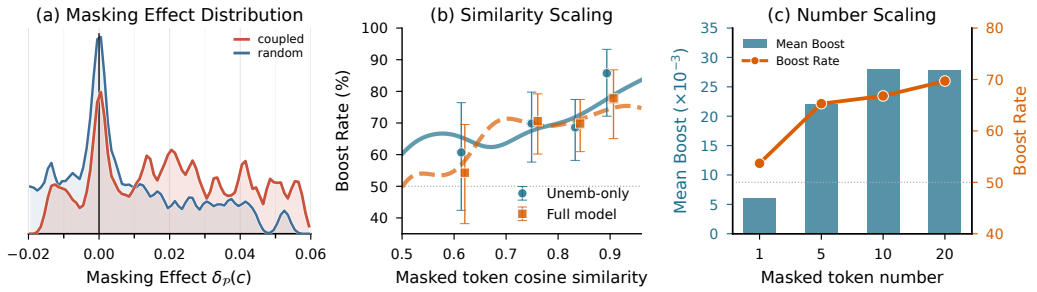


Figure 3: Empirical verification via masked updates. **(a)** Coupled tokens produce a right-shifted Masking Effect distribution compared with random tokens. **(b)** Boost Rate increases with coupling strength and is consistent between the output-layer proxy and the full-parameter update. **(c)** Masking more coupled tokens increases both Mean Boost and Boost Rate.

For further verification, we test whether tokens selected by this coupling structure measurably affect actual updates. Specifically, we compare the normal update with a masked update, where a selected token set $\mathcal{M}(c)$ is removed from the loss. We define the *Masking Effect* as

$$\delta_{\mathcal{M}}(c) = \log \pi_\theta(c \mid s_c) - \log \pi_\theta^{-\mathcal{M}}(c \mid s_c),$$

where $\log \pi_\theta^{-\mathcal{M}}(c \mid s_c)$ is the log probability with the masked update. A positive value of δ indicates that the removed tokens would have increased the candidate token's log-probability under the unmasked update. We summarize this effect with two statistics: *Boost Rate*, the fraction of candidates for which the selected tokens boost the candidate token's log-probability, and *Mean Boost*, the average Masking Effect:

$$\text{Boost Rate} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathbf{1}\{\delta_{\mathcal{M}(c)}(c) > 0\}, \qquad \text{Mean Boost} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \delta_{\mathcal{M}(c)}(c).$$

As shown in Table 1, masking random tokens yields a random Boost Rate (≈50%) and a near-zero Mean Boost. In contrast, masking tokens that satisfy both conditions, same token identity and low confidence, raises Boost Rate to approximately 75% and produces clearly positive Mean Boost under both the output-layer proxy and the full-parameter update. This verifies that the sparse coupling structure identified by Eq. 10 selects tokens with a consistent and measurable influence on the candidate update. The close agreement between output-layer and full-parameter update further suggests that the proxy approximates the full-parameter training dynamics well. Fig. 3 provides a complementary view, showing that Masking Effect scales with predicted coupling strength and is not driven by a few outlier pairs.

Table 1: Masked updates verify the effect of coupled tokens. Tokens satisfying both same-token and low-confidence conditions yield higher Boost Rate and larger Mean Boost than random tokens under both the output-layer proxy and full-parameter update.

Masked Tokens		LM-head		Full	
Same	LowConf.	Boost Rate	Mean Boost	Boost Rate	Mean Boost
		50.60	−.0039	50.21	+.0048
✓		66.20	+.0224	66.41	+.0241
	✓	44.20	−.0004	43.04	−.0012
✓	✓	74.12	+.0301	75.00	+.0306

3 Coupling Effect Induces Cancellation

The previous section indicates that token updates are not isolated: same-token, low-confidence, and representation-aligned tokens can exert coupled influence on one another within the same batch. In a group-relative batch, such coupled directions often appear in both successful and failed rollouts from the same query. Because these rollouts carry opposite-signed group-normalized advantages, shared directions are suppressed in the aggregate update, while directions that are more correlated with successful rollouts are reinforced. We term this mechanism the *cancellation hypothesis*. It has two consequences: it could reduce variance along shared directions, and it turns the group-relative update into an advantage-weighted filter that induces hidden token-level credit assignment.

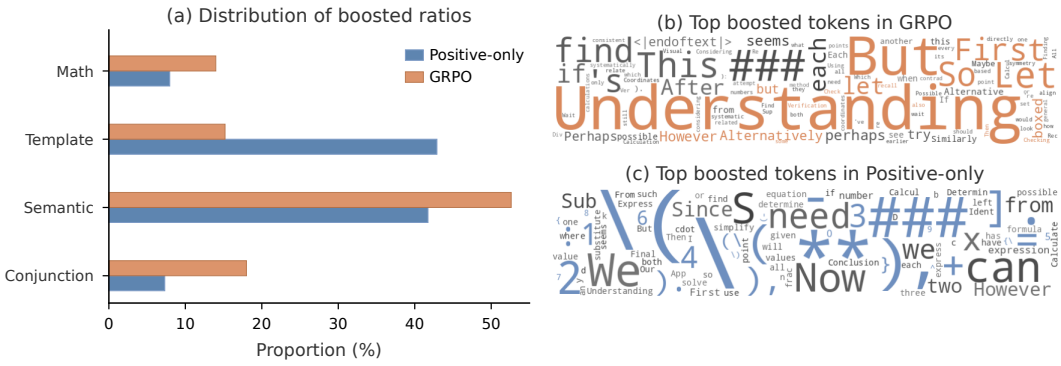


Figure 4: (a) Tokens are grouped by NLTK POS tagging (Bird et al., 2009) into Math (Number, Latex), Template (Symbol, Determiner, Conj), Semantic (Noun, Verb), and Conjunction (Conjunction); Each bar shows the fraction of total log-prob boost contributed by that group. Positive-only updates concentrate on Template, whereas GRPO update toward Semantic and Conjunction. (b/c) At the token level, Positive-only boosts formatting tokens, GRPO boosts reasoning conjunctions.

3.1 Cancellation as Advantage-Weighted Filter and Variance Reducer

Consider a query q with G sampled rollouts and group-normalized rollout advantages $\{\hat{A}_i\}_{i=1}^G$ satisfying $\sum_{i=1}^G \hat{A}_i = 0$. Let $d_i = \sum_t g_{i,t}$ denote the response-level score direction for rollout i, where $g_{i,t} = \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$. The group-relative part of the query-level update is $\Delta\theta_q \propto \sum_{i=1}^G \hat{A}_i d_i$. The squared norm of the query-level update could be expanded:

$$\left\| \sum_{i=1}^G \hat{A}_i d_i \right\|^2 = \sum_{i=1}^G \hat{A}_i^2 \|d_i\|^2 + \sum_{i \neq j} \hat{A}_i \hat{A}_j \langle d_i, d_j \rangle. \tag{11}$$

As rollouts from the same query contain identical tokens and similar hidden states, their response-level gradients have more positive overlap. In the idealized case where the common within-query overlap is approximately a positive constant c_q, the cross term becomes

$$\sum_{i \neq j} \hat{A}_i \hat{A}_j \langle d_i, d_j \rangle \approx c_q \sum_{i \neq j} \hat{A}_i \hat{A}_j = -c_q \sum_{i=1}^G \hat{A}_i^2 < 0, \tag{12}$$

where the last equality uses $\sum_i \hat{A}_i = 0$. Thus, positive gradient overlap is multiplied by negative advantage cross-products, reducing the update norm relative to the self-term $\sum_i \hat{A}_i^2 \|d_i\|^2$. Since gradient variance is controlled by the second moment of the update, this norm reduction gives the variance-reduction role of cancellation. The same algebra also gives a filtering interpretation. For any token or feature direction u, let $m_{i,u} = \langle d_i, u \rangle$ measure how strongly rollout i loads on that direction. The surviving signal along u is

$$S_u = \sum_{i=1}^G \hat{A}_i m_{i,u} = \sum_{i=1}^G \hat{A}_i \left(m_{i,u} - \bar{m}_u\right), \qquad \bar{m}_u = \frac{1}{G} \sum_{i=1}^G m_{i,u}. \tag{13}$$

Since $\sum_i \hat{A}_i = 0$, S_u/G is the empirical covariance between advantage and the loading on direction u. Directions that appear equally in positive and negative rollouts are filtered out, while ones enriched in higher-advantage rollouts survive. This advantage-weighted filtering provides a mechanism for hidden token-level credit assignment in the cancellation view.

3.2 Cancellation Acts as Hidden Credit Assignment

Variance reduction and filtering alone does not imply better learning: a smaller update could simply be a weaker update. The key question is whether cancellation is objective-driven, i.e., whether the residual left after cancellation secretly retains tokens that contribute to the training task. We compare GRPO with two complementary probes: *positive-only* keeps only positive-reward rollouts, *negative-only* keeps only negative-reward rollouts. Full experimental details are provided in Appendix E.1.

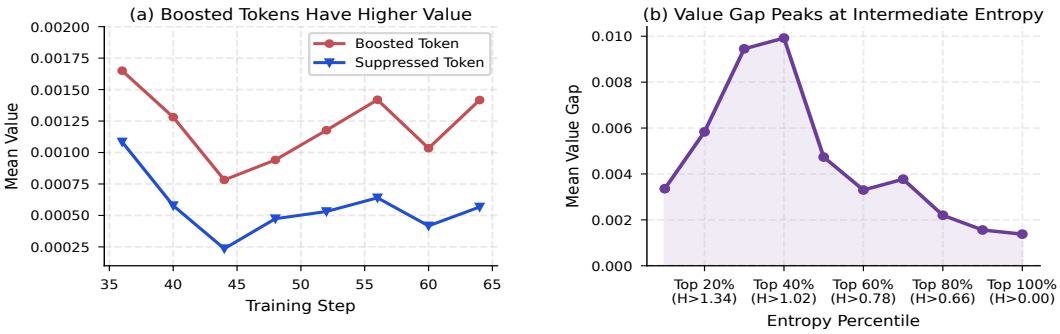


Figure 5: Cancellation acts as hidden token-level credit assignment. (a) Boosted tokens have higher Monte Carlo estimated value than suppressed tokens. (b) The value gap first increases and then decreases with entropy, suggesting that low-entropy tokens dilute the learning signal.

Cancellation decreases learning on template and formatting tokens. Under the positive-only update, formatting and template tokens dominate the boosted directions, indicating that positive rollouts alone strongly reinforce surface patterns. By contrast, the joint GRPO update suppresses many of these shared directions and shifts boosted mass toward more reasoning-bearing tokens. This is the qualitative signature of the filter in Eq. 13: directions common across rollout polarities are removed, while directions more correlated with success remain.

Cancellation favors tokens with higher value. We further ask whether the tokens that survive cancellation are more critical. We thus estimate token value with Monte Carlo rollouts. We randomly sample equal numbers of boosted and suppressed tokens from positive or negative rollouts, and for each token with state $s_t = (q, o_{<t})$ and action $a_t = o_t$, we estimate the marginal contribution of a_t by rolling out M continuations from both $s_t \oplus a_t$ and s_t:

$$\hat{\Delta}_M(s_t, a_t) \;=\; \frac{\frac{1}{M}\sum_{m=1}^{M} r\big(\tau^{(m)} \mid s_t \oplus a_t\big) - \frac{1}{M}\sum_{m=1}^{M} r\big(\tilde{\tau}^{(m)} \mid s_t\big)}{1 - \pi(a_t \mid s_t)},$$

where $r \in \{0, 1\}$ is the binary reward. As shown in Fig. 5(a), boosted tokens consistently have higher Monte Carlo estimated value than suppressed tokens. Full details are provided in Appendix E.2.

The value gap peaks at intermediate entropy. To examine where cancellation carries the most informative signal, we sort all tokens by entropy and compute the mean value gap on the top-$k\%$ highest-entropy subset for varying k. As shown in Fig. 5(b), the gap peaks at about top-40%. The trend reflects that high entropy could amplify the value gap. And low-entropy tokens, anchored by pretraining priors, resist displacement and contribute little net signal, diluting the average gap. The observation connects to recent high-entropy token update strategies (Wang et al., 2025; Zheng et al., 2025; Wu et al., 2025a). At the same time, the rise-then-fall pattern suggests that entropy is not a monotonic proxy for token value; the strongest hidden credit signal appears before low-entropy tokens are included, rather than simply at the maximum-entropy extreme.

Larger sampling budget strengthens cancellation. Cancellation relies on the co-occurrence of positive and negative rollouts within each update. With a small sampling budget, many query groups lack sufficient rollout diversity, and the opposing coupled signals that drive cancellation cannot fully form. We verify this by independently varying batch size and rollout-n, while keeping other hyperparameters fixed. As shown in Fig. 6, value gap increases along both axes: the hidden value discovery grows with larger sampling budget, particularly for the top-25% highest-value subset. This confirms that larger batches yield more complete coupling groups, strengthening cancellation and amplifying the value signal they expose.

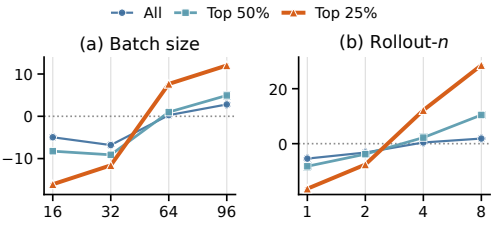


Figure 6: Cancellation strengthens with the sampling budget. The value gap (10^{-3}) grows as batch size increases from 16 to 96 (a) and rollout-n from 1 to 8 (b).

4 REPAIRING BROKEN CANCELLATION IN RLVR

The previous section shows that cancellation filters shared update directions, exposing selective token-level learning signals. We next examine whether common practices in RL frameworks disrupt this mechanism and whether repairing it yields consistent performance improvements.

Query-preserved mini-batching To improve throughput, RL frameworks commonly partition rollout batches into smaller mini-batches based on token count (Sheng et al., 2024; Zhu et al., 2025b). This implementation detail could scatter rollouts from the same query across different optimizer steps. From the cancellation view, this could be harmful, as the positive and negative rollouts of a query are better updated together to reflect their shared directions for cancellation.

(1) *Local imbalance.* For a complete query group, group-normalized advantages satisfy $\sum_i \hat{A}_i = 0$. Splitting the group across mini-batches generally yields $S_{\mathcal{B}} = \sum_{i\in\mathcal{B}} \hat{A}_i \neq 0$, and hence $\sum_{i\neq j\in\mathcal{B}} \hat{A}_i\hat{A}_j = S_{\mathcal{B}}^2 - \sum_{i\in\mathcal{B}} \hat{A}_i^2$. The positive residual $S_{\mathcal{B}}^2$ weakens the negative cross-advantage structure underlying cancellation, increasing update variance.

(2) *Optimization drift.* One might expect later mini-batches to provide the missing counterparts, but GRPO updates the policy after every mini-batch. Later counterparts therefore contribute under different model parameters and cannot reproduce the cancellation of a joint update. We therefore keep all rollouts from the same query within one mini-batch, preserving their joint cancellation.

Reward-balanced batching Cancellation mechanism relies on the co-occurrence of positive and negative advantage within the same update. When rewards are sparse or saturated, one sign may dominate the batch, weakening cross-sign coupling. We examine this failure mode through two complementary experiments (Fig. 7). (1) We separate positive and negative-advantage rollouts into different mini-batches, removing cross-sign coupling; training performance degrades accordingly. (2) Using the same token-value analysis, we vary the reward-sign composition within each query group and measure the boosted–suppressed value gap.

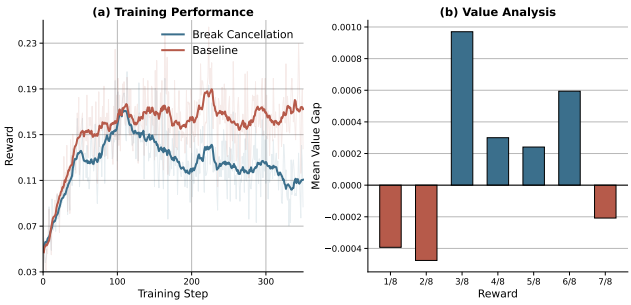


Figure 7: (a) Partitioning mini-batches by reward sign maximally breaks cancellation, causing performance drops. (b) The boosted–suppressed value gap narrows or reverses in reward-sparse or reward-saturated groups.

These observations motivate a simple remedy: preserve reward-sign balance during batch construction. We accumulate rollouts and defer updates until both reward signs are sufficiently represented, then select an update batch $\mathcal{B}$ satisfying $\min\{N_+(\mathcal{B}), N_-(\mathcal{B})\} \geq \tau|\mathcal{B}|$, where N_+ and N_- count positive- and negative-advantage rollouts in $\mathcal{B}$, and τ sets the minimum fraction of each sign.

Experiment results Full experiment details can be found in Appendix B. Table 2 presents the evaluation results across all model scales and benchmarks. Our proposed interventions achieve consistent improvements, with particularly notable results on the AMC and Olympiad Bench. This phenomenon is most pronounced for Qwen2.5-Math-7B, which exhibits the most unstable training when RL is from base models. As shown in Fig. 8, the baseline configuration suffers from severe collapse, whereas batching strategies stabilize the training and improve final performance.

5 RELATED WORK

A common sequence-level intuition for LLM RL training is that it reinforces successful rollouts and suppresses failed ones, shifting the model distribution toward higher-reward responses (Mroueh, 2025; Wu et al., 2025b). However, recent studies suggest that this explanation could be incomplete.

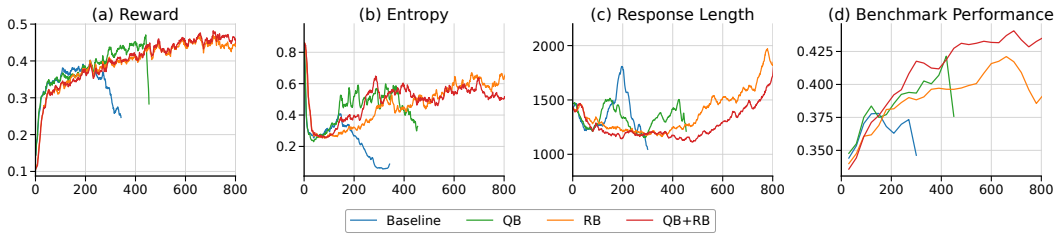


Figure 8: Query-preserved mini-batching (QB) and reward-balanced batching (RB) stabilize GRPO training on Qwen2.5-Math-7B. The baseline suffers severe training collapse, while QB and RB individually mitigate instability. Their combination (QB+RB) achieves the most stable training dynamics and the strongest final performance across all evaluation benchmarks.

Table 2: Evaluation results across math reasoning benchmarks under different method combinations (%). QB: query-preserved mini batching RB: reward-balanced batching

Model	Methods		Benchmarks						
	QB	RB	AIME24	AIME25	AMC	Olympiad	Minerva	Math	Avg
Qwen2.5-Math-7B			21.3	11.3	54.9	37.8	29.4	72.6	37.8
	✓		20.7	10.8	61.2	42.6	36.7	81.6	42.3
		✓	25.4	13.3	63.0	39.0	**37.1**	76.6	42.4
	✓	✓	**27.8**	**15.9**	**63.9**	**44.5**	36.0	**82.2**	**45.0**
Qwen3-1.7B-Base			12.4	9.4	37.0	34.5	28.3	69.8	31.9
	✓		11.6	7.9	40.2	37.6	33.8	71.0	33.7
		✓	16.2	**11.8**	42.6	37.3	30.9	**74.2**	35.5
	✓	✓	**16.5**	9.0	**43.5**	**38.2**	**35.7**	74.0	**36.1**
Qwen3-4B-Base			28.3	**24.2**	58.0	53.2	44.5	86.2	49.1
	✓		29.2	22.8	**64.4**	53.6	44.9	86.4	50.2
		✓	27.9	22.4	61.5	54.8	**46.0**	88.0	50.1
	✓	✓	**30.7**	23.0	64.0	**55.4**	43.4	**88.4**	**50.8**

The *Lazy Likelihood* phenomenon shows that positively supervised responses are not necessarily upweighted during training (Razin et al., 2024; Pal et al., 2024; Ren & Sutherland, 2024; Deng et al., 2025b;c). Related positive-suppressed phenomena have also been reported across diverse settings of LLM post-training, including reasoning with format-reward (Simoni et al., 2025), search-integrated RL (Deng et al., 2025a), and multi-turn tool-integrated reasoning (Xue et al., 2025). Existing studies primarily focused on domain-specific remedies for particular instability modes the *Lazy Likelihood* may cause, or on rollout-level analyses of likelihood dynamics, while leaving the in-depth analysis of how individual tokens are updated in critic-free RL remains underexplored.

6 Conclusion

In this work, we revisit critic-free RL for LLMs from the perspective of token-level probability shifts. We identified *token flipping*, showing that positive and negative rollouts exhibit remarkably similar proportions of boosted and suppressed tokens, despite receiving opposite signs of advantage. To explain this mismatch, we derived a first-order decomposition of GRPO updates, showing that a token's log-probability change is shaped by a coupling kernel that concentrates on same-token and low-confidence token pairs. Building on this view, we propose the *cancellation hypothesis*: opposing coupled signals cancel out for tokens shared across rollouts, while tokens more specific to successful ones receive stronger reinforcement, thereby inducing hidden token-level credit assignment from outcome-level supervision. Guided by this view, we further introduce two batching interventions for RL training and validate their performance. We hope this work offers a new perspective on the learning dynamics of critic-free RL for LLMs and will pave the way for more principled approaches to batch design and credit assignment in LLM post-training.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and presentation. All AI-assisted suggestions were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009. ISBN 9780596516499.

Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. On grpo collapse in search-r1: The lazy likelihood-displacement death spiral. *arXiv preprint arXiv:2512.04220*, 2025a.

Wenlong Deng, Yi Ren, Muchen Li, Danica J Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. *arXiv preprint arXiv:2505.18830*, 2025b.

Wenlong Deng, Yi Ren, Yushu Li, Boying Gong, Danica J Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. Token hidden reward: Steering exploration-exploitation in group relative deep reinforcement learning. *arXiv preprint arXiv:2510.03669*, 2025c.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need adam? surprisingly strong and sparse reinforcement learning with sgd in llms. *arXiv preprint arXiv:2602.07729*, 2026.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. *arXiv preprint arXiv:2410.08847*, 2024.

Yi Ren and Danica J Sutherland. Learning dynamics of llm finetuning. *arXiv preprint arXiv:2407.10490*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, Andrea Saracino, and Paolo Mori. Gtpo: Stabilizing group relative policy optimization via gradient and entropy control. *arXiv preprint arXiv:2508.03772*, 2025.

Xinyu Tang, Yuliang Zhan, Zhixun Li, Wayne Xin Zhao, Zhenduo Zhang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Rethinking sample polarity in reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2512.21625*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Jianghao Wu, Yasmeen George, Jin Ye, Yicheng Wu, Daniel F Schmidt, and Jianfei Cai. Spine: Token-selective test-time reinforcement learning with entropy-band regularization. *arXiv preprint arXiv:2511.17938*, 2025a.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, et al. It takes two: Your grpo is secretly dpo. *arXiv preprint arXiv:2510.00977*, 2025b.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025a.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. `https://github.com/THUDM/slime`, 2025b. GitHub repository. Corresponding author: Xin Lv.

A LIMITATIONS

Our coupling analysis relies on an output-layer proxy that approximates the full-model gradient with its last-layer counterpart, so we validate the reasonableness of this proxy and provide corresponding empirical justifications: Appendix D.1 shows that the proxy and full-parameter updates exhibit stable token-flip sign consistency throughout training, while Table 1 shows that the same-token, low-confidence coupling pattern identified by the proxy produces comparable masking effects under full-parameter updates. Our two interventions are evaluated on mathematical reasoning with binary rewards. Extending them and validating them in more diverse settings and broader domains remains future work. Reward-balanced batching may reduce sample utilization because some rollouts are discarded to meet the target reward composition. However, our design naturally fits modern asynchronous RL frameworks: it can be implemented simply as an additional update condition on the rollout buffer.

B EXPERIMENT DETAILS

Experiment Settings We employ math reasoning problems as a playground to test the proposed method, for math problems usually have reliable and cheap verifiers. Our training set is the publicly released DAPO-17K-dataset. Each training batch consists of 128 prompts with 8 responses sampled per prompt, yielding a total of 1024 samples. In our configuration, updates are performed with a mini-batch of 256 samples, resulting in 4 gradient updates per sampling iteration. We evaluate our analysis on three backbone models spanning both math-specialized and general-purpose LLMs: QWEN2.5-MATH-7B, QWEN3-1.7B-BASE, and QWEN3-4B-BASE. This model suite allows us to examine whether the observed token-level dynamics remain consistent across different model families and scales. We use a maximum response length of 8192 for all models. We use the DAPO dataset throughout our experiments and adopt MATH-VERIFY as the verifier.

Implementation We implement all RL experiments in the VERL framework and use GRPO as our critic-free RL algorithm. Following DAPO (Yu et al., 2025), we adopt the token-level policy gradient loss and enable `clip-higher`. However, we do not use Dynamic Sampling due to limited computational budget. In all experiments, we set `clip_low` $= 0.2$ and `clip_high` $= 0.28$, remove the KL regularization term by setting $\beta = 0$, and set the entropy loss coefficient to 0. We use Math-Verify as our reward function and include no format or length reward. Our training algorithm maximizes the following clipped surrogate objective:

$$\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\left(\rho_{i,t}(\theta)A_i,\text{clip}(\rho_{i,t}(\theta),1-\epsilon_{\text{low}},1+\epsilon_{\text{high}})A_i\right)\right] \tag{14}$$

where $\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$, and $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance ratio. Table 3 summarizes the default training configurations.

Table 3: Training hyperparameters across different models.

Hyperparameters	Qwen2.5-Math-7B	Qwen3-1.7B-Base	Qwen3-4B-Base
`max_prompt_length`	1024	1024	1024
`max_response_length`	8192	8192	8192
`train_batch_size`	128	128	128
`mini_batch_size`	32	32	32
`optim.lr`	1e−6	1e−6	1e−6
`rollout.temperature`	1.0	1.0	1.0
`rollout.n`	8	8	8

Implementation Details for proposed batching tricks We implement all experiments within the VERL framework, a widely adopted open-source RL training framework. By default, VERL partitions mini-batches by token count to balance computational load across data-parallel nodes. For *query-preserved mini-batching*, we ensure all rollouts from the same query are assigned to the same mini-batch. For *reward-balanced batching*, we accumulate rollouts in a buffer and trigger gradient updates only when the collected batch meets the target reward composition. We set $\tau = 0.5$ for QWEN2.5-MATH-7B and QWEN3-4B-BASE, and $\tau = 0.25$ for QWEN3-1.7B-BASE.

Evaluation For evaluation, we mainly focus on six widely used math reasoning benchmarks, including AIME2024, AIME2025, AMC, Minerva, OlympiadBench and Math-500. For AIME2024, AIME2025, and AMC, we report avg@32 because the test set is relatively small; for the other three benchmarks, we report pass@1. To determine the evaluation hyperparameters, we test four common sampling configurations on both the base and aligned models, as shown in Table 4. We adopt temperature $= 0.6$ and top-$p = 0.95$ for all evaluations, as this setting yields the highest average performance on the aligned model while maintaining competitive results on base model.

Table 4: Evaluation results under different temperature and top-p settings.

Model	Temp.	Top-p	AIME24	AIME25	AMC	Math	Olymp.	Minerva	Avg.
qwen3-4b-base	1.0	0.7	**10.6**	6.7	36.4	62.8	33.4	24.2	29.0
qwen3-4b-base	0.6	0.7	10.0	**8.5**	**41.7**	**73.6**	**36.7**	26.1	**32.8**
qwen3-4b-base	1.0	0.95	6.6	3.6	27.1	49.0	23.8	17.6	21.3
qwen3-4b-base	0.6	0.95	10.2	6.6	39.9	64.8	34.5	**27.9**	30.7
hf-4b-step100	1.0	0.7	16.0	14.1	49.5	**82.6**	46.2	**37.5**	41.0
hf-4b-step100	0.6	0.7	14.2	12.3	49.9	81.0	**46.9**	37.1	40.2
hf-4b-step100	1.0	0.95	**16.8**	13.9	49.4	79.8	44.4	**37.5**	40.3
hf-4b-step100	0.6	0.95	15.8	**14.8**	**50.6**	82.4	**46.9**	36.7	**41.2**

C Token Flipping

Measuring token-level log-probability shift. We measure how a single GRPO update changes token probabilities on a fixed rollout batch. In typical RL training pipelines, responses are sampled by a fast inference engine as VLLM, which also return rollout-time log-probabilities. However, these values can differ slightly from those produced by the FSDP training forward pass, even for the same model state, due to numerical differences between the inference and training stacks. To ensure an exact comparison, we compute both the pre-update and post-update log-probabilities via dedicated FSDP forward passes on the training model over the same rollout batch. Concretely, before the GRPO update we obtain $\log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$, and after the update we obtain $\log \pi_{\theta'}(o_{i,t} \mid q, o_{i,<t})$ through a second FSDP forward pass. The per-token shift and its classification are:

$$\Delta \log p_{i,t} = \log \pi_{\theta'}(o_{i,t} \mid q, o_{i,<t}) - \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}), \tag{15}$$

$$\textit{boosted}: \ \Delta \log p_{i,t} > \epsilon, \qquad \textit{suppressed}: \ \Delta \log p_{i,t} < -\epsilon, \qquad \textit{stable}: \ |\Delta \log p_{i,t}| \le \epsilon,$$

with $\epsilon = 10^{-6}$. We report the ratio of boosted and suppressed tokens separately for positive-advantage and negative-advantage rollouts. The full procedure is summarized in Algorithm 1.

Algorithm 1 Computing token-level $\Delta \log p$ after one GRPO step.

Require: Batch of prompts $\{q\}$, current policy π_θ, number of rollouts G

Rollout & Scoring

1: For each prompt q, sample G responses $\{o_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid q)$ — via vLLM

2: Compute $\ell_{i,t}^{\text{old}} \leftarrow \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$ for all response tokens — via FSDP forward

3: Verify each response; compute group-normalized advantage A_i

Policy Update

4: $\theta' \leftarrow \text{GRPO-UPDATE}(\theta, \{o_i, A_i\})$

Measure Token Displacement

5: Compute $\ell_{i,t}^{\text{new}} \leftarrow \log \pi_{\theta'}(o_{i,t} \mid q, o_{i,<t})$ for all response tokens — via FSDP forward

6: $\Delta \log p_{i,t} \leftarrow \ell_{i,t}^{\text{new}} - \ell_{i,t}^{\text{old}}$

Additional empirical evidence for token flipping. Fig. 9 shows that *Token Flipping* is a stable phenomenon across different models. We report the proportions of boosted, suppressed, and stable tokens, defined by $\Delta \log p$, for positive rollouts ($A > 0$), negative rollouts ($A < 0$), and the full rollout batch. Across different models, Positive and negative rollouts display highly similar token-displacement profiles throughout generation, even though their advantages have opposite signs.

D Coupling Analysis

D.1 Empirical Justification for the Output-Layer Proxy

The coupling kernel in Eq. (5) is defined in the full parameter space, making it expensive to compute and difficult to interpret. This motivates a simpler proxy based on the output layer. We provide empirical evidence that this proxy remains faithful to the full-parameter update in practice.

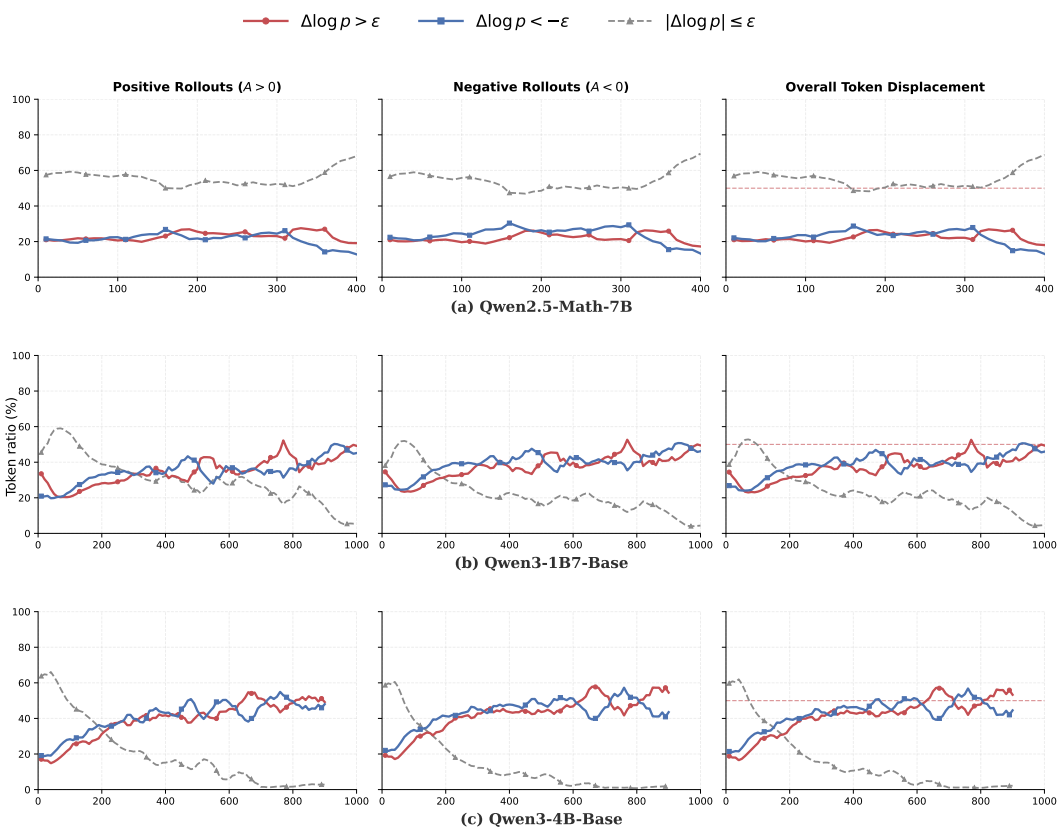


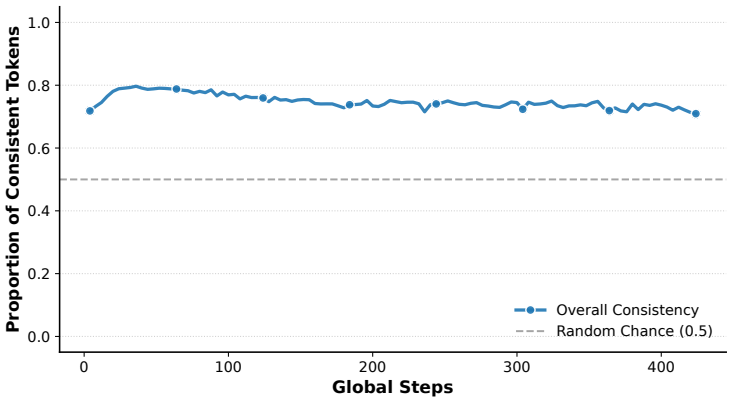
Figure 9: **Reversed update persists across backbone models.** Each row corresponds to a backbone model, and the three columns report the proportions of boosted ($\Delta \log p > \epsilon$), suppressed ($\Delta \log p < -\epsilon$), and stable ($|\Delta \log p| \leq \epsilon$) tokens in positive rollouts ($A > 0$), negative rollouts ($A < 0$), and the full rollout batch. Despite opposite advantage signs, positive and negative rollouts exhibit remarkably similar token-displacement profiles across models.

To validate the proxy, we compare the per-token displacement $\Delta \log p$ produced by a standard full-parameter GRPO step with the displacement produced by updating only the unembedding matrix W, using the same rollout batch. For each token we record whether its $\Delta \log p$ under the two update paradigms shares the same sign. As shown in Fig. 10, the sign-consistent proportion remains stable throughout training, plateauing near 0.7.

Beyond sign consistency, the proxy also faithfully recovers the *structural* predictions of the coupling analysis. Specifically, the factorisation in Eq. 10 predicts that non-negligible coupling concentrates on same-token, low-confidence pairs—a conclusion derived entirely under the output-layer approximation. Table 1 provides a direct causal test of this prediction: masking tokens that satisfy both conditions yields comparable Boost Rate and Mean Boost under the LM-head-only and full-parameter updates ($\sim$74–75% and $\sim$0.030, respectively), whereas relaxing either condition substantially weakens the effect. The close agreement between the two update paradigms confirms that the sparse coupling structure identified by the proxy is not an artifact of the approximation but reflects a property of the full-parameter training dynamics.

D.2 Empirical Verification Details for Token Coupling and Additional Results

For the masked-update experiments reported in Table 1 and Fig. 3, we deliberately use SGD with a learning rate of $1e{-}1$ and no momentum rather than Adam. The reason is that Adam's first parameter update reduces to an approximate sign-descent step (see Appendix D.3 for a formal derivation), which normalizes away the relative magnitude differences across tokens and would obscure the per-token causal signal our analysis aims to detect. Following Mukherjee et al. (2026), we adopt SGD for RL



(a) Sign Consistency

Figure 10: **Agreement between full-parameter and unembedding-layer updates.** The proportion of tokens where the token displacement ($\Delta \log p$) shares the same sign under both update paradigms. The consistency remains highly stable across training steps, plateauing near 0.7, providing empirical support for the output-layer proxy adopted in Section 3.

updates with a learning rate of $1\mathrm{e}{-1}$, which preserves the true gradient magnitudes and therefore faithfully reflects the contribution of each masked token set to the candidate update.

As shown in Fig. 3, the masked-update results provide three complementary pieces of evidence. (1) Masking coupled tokens (same-token, low-confidence) produces a clearly right-shifted Masking Effect distribution relative to random tokens (panel a), indicating that these tokens exert a consistently positive influence on the candidate's log-probability under the unmasked update. (2) Boost Rate increases monotonically with coupling strength as measured by the proxy cosine similarity, and the trend is closely matched between the LM-head-only and full-parameter updates (panel b), further validating the output-layer proxy. (3) Progressively masking more top-coupled tokens amplifies both Mean Boost and Boost Rate (panel c). Together, these results corroborate the prediction of Eq. 10: non-negligible coupling is sparse and concentrates on identical, low-confidence token pairs.

D.3 Why SGD Instead of Adam for Causal Analysis

In our causal experiments we deliberately use SGD rather than Adam. The reason is that Adam's first update is governed almost entirely by the *sign* of the gradient, not its magnitude, which would obscure the token-level causal signal we aim to measure.

Concretely, Adam maintains a first-moment estimate $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ and a second-moment estimate $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$. At the very first step ($t = 1$), with $m_0 = v_0 = 0$, these reduce to

$$m_1 = (1-\beta_1)\, g_1, \qquad v_1 = (1-\beta_2)\, g_1^2. \tag{16}$$

After bias correction ($\hat{m}_1 = m_1/(1-\beta_1)$, $\hat{v}_1 = v_1/(1-\beta_2)$), the parameter update becomes

$$\theta_1 = \theta_0 - \alpha \cdot \frac{\hat{m}_1}{\sqrt{\hat{v}_1} + \epsilon} = \theta_0 - \alpha \cdot \frac{g_1}{\sqrt{g_1^2} + \epsilon} = \theta_0 - \alpha \cdot \frac{g_1}{|g_1| + \epsilon}. \tag{17}$$

When $|g_1| \gg \epsilon$ (which holds for the vast majority of parameters), this simplifies to

$$\theta_1 \approx \theta_0 - \alpha \cdot \mathrm{sign}(g_1). \tag{18}$$

This "sign-descent" behavior means that the relative contribution of individual tokens to the update is washed out at the first step. SGD preserves the true gradient magnitudes and therefore faithfully reflects the per-token causal effect, making it the appropriate choice for our analysis.

E CANCELLATION HYPOTHESIS

E.1 DECOMPOSING UPDATES BY REWARD SIGN: POSITIVE-ONLY AND GRPO

To isolate how positive rollouts contribute to token-level displacement, we construct a controlled comparison in which the rollout batch is fixed and only the polarity composition of the update is changed. Given a rollout batch $\mathcal{B}$ with binary rewards $r_i \in \{0, 1\}$ and group-normalized advantages A_i, we instantiate two update variants by reweighting the rollout-level advantage:

- **Positive-only:** $A_i^+ = A_i \cdot \mathbb{I}[r_i = 1]$
- **GRPO (Joint):** use the original A_i.

Both variants start from the same saved checkpoint θ_0 and operate on the same rollout batch $\mathcal{B}$. This design removes variation from sampling and initialization, so any difference in the resulting token displacement can be attributed to the polarity composition of the update itself. Importantly, after each update, we measure token displacement on the *full original batch* $\mathcal{B}$ rather than on the masked subset used to form the loss. This shared evaluation protocol makes the displacement maps from Positive-only and GRPO directly comparable. The full procedure is summarized in Algorithm 2.

Algorithm 2 Controlled measurement of token displacement under single-polarity and joint updates.

Require: Dataset $\mathcal{D}$, current policy π_θ, number of rollouts G

Rollout & Scoring

Sample a batch of prompts $\{q\}$ from $\mathcal{D}$
For each prompt q, sample G responses $\{o_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid q)$
Verify each rollout to obtain reward $r_i \in \{0, 1\}$; compute group-normalized advantage A_i
Collect rollout batch $\mathcal{B} = \{(q, o_i, r_i, A_i)\}$

Shared Preparation

Compute $\ell_{i,t}^{\text{old}} \leftarrow \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$ for all tokens in $\mathcal{B}$
Save checkpoint $\theta_0 \leftarrow \theta$

Positive-Only Update

    Define $A_i^+ = A_i \cdot \mathbb{I}[r_i = 1]$ for all rollouts
    $\theta^+ \leftarrow \text{GRPO-UPDATE}(\theta_0,\ \mathcal{B},\ \{A_i^+\})$
    $\Delta \log p_{i,t}^{\text{POS}} \leftarrow \log \pi_{\theta^+}(o_{i,t} \mid q, o_{i,<t}) - \ell_{i,t}^{\text{old}}$ measured on full $\mathcal{B}$

GRPO: Joint Update

    Load checkpoint $\theta \leftarrow \theta_0$
    $\theta' \leftarrow \text{GRPO-UPDATE}(\theta_0,\ \mathcal{B},\ \{A_i\})$
    $\Delta \log p_{i,t}^{\text{GRPO}} \leftarrow \log \pi_{\theta'}(o_{i,t} \mid q, o_{i,<t}) - \ell_{i,t}^{\text{old}}$ measured on full $\mathcal{B}$

E.2 EXPERIMENT DETAILS FOR VALUE ESTIMATION

In standard RL, the value function is defined for a state rather than for an action:

$$V^\pi(s_t) = \mathbb{E}\big[r \mid s_t, \pi\big].$$

A token o_t is an action taken at the language-model state $s_t = (q, o_{<t})$, so it does not have a context-free value by itself. The standard action-related quantity is instead the state-action Q fuction:

$$Q^\pi(s_t, o_t) = \mathbb{E}\big[r \mid s_t \oplus o_t, \pi\big],$$

which measures the expected reward after forcing o_t and then following policy π.

Our analysis focuses on a different quantity: at the same state s_t, how much better is taking token o_t than not taking it and instead sampling an alternative token from the policy? We define this *counterfactual token value* as

$$\Delta^\pi(o_t \mid s_t) = Q^\pi(s_t, o_t) - Q^\pi_{\neg o_t}(s_t),$$

where

$$Q^\pi_{\neg o_t}(s_t) = \mathbb{E}_{a \sim \pi(\cdot|s_t),\, a \neq o_t} Q^\pi(s_t, a)$$

is the expected value of the counterfactual alternatives.

Let $p_t = \pi(o_t \mid s_t)$. The state value mixes the forced-token branch and the alternatives:

$$V^\pi(s_t) = p_t Q^\pi(s_t, o_t) + (1 - p_t) Q^\pi_{\neg o_t}(s_t),$$

Therefore,

$$Q^\pi(s_t, o_t) - V^\pi(s_t) = (1 - p_t) \left(Q^\pi(s_t, o_t) - Q^\pi_{\neg o_t}(s_t) \right),$$

and hence

$$\Delta^\pi(o_t \mid s_t) = \frac{Q^\pi(s_t, o_t) - V^\pi(s_t)}{1 - p_t}.$$

Monte Carlo estimation. For each candidate token o_t at state $s_t = (q, o_{<t})$, we estimate the two terms above with Monte Carlo rollouts. We sample $M{=}256$ complete continuations from $s_t \oplus o_t$ and denote the mean reward by $\mathrm{avg}[0]$. We also sample $M{=}256$ complete continuations directly from s_t and denote the mean reward by $\mathrm{avg}[-1]$. Since rollouts from s_t already choose o_t with probability $p(o_t)$, the naive difference $\mathrm{avg}[0] - \mathrm{avg}[-1]$ is attenuated by this self-inclusion. We therefore use the estimator

$$\hat{\Delta}(o_t \mid s_t) = \frac{\mathrm{avg}[0] - \mathrm{avg}[-1]}{1 - p(o_t)}. \tag{19}$$

This estimator measures the reward gain of taking token o_t over sampling a counterfactual alternative from the policy. We refer to this quantity as the counterfactual token value, or token value for short. For each result reported in the main text, we conduct three independent sampling runs with different random seeds. Each run samples 1,500 tokens, balanced across boosted and suppressed tokens from positive and negative rollouts, and we report the average over the three runs.

Repeated Updates widen value gap We apply repeated GRPO updates on the same fixed batch and track the mean value gap between boosted and suppressed tokens across update steps. As shown in Fig. 11, the value gap grows monotonically with the number of updates, indicating that each gradient step further separates high-value boosted tokens from low-value suppressed ones. This accumulation suggests that the hidden credit assignment embedded in cancellation is not a one-step coincidence but a stable structural property of critic-free RL: successive updates consistently surface and amplify the implicit value signal latent in the rollout batch.

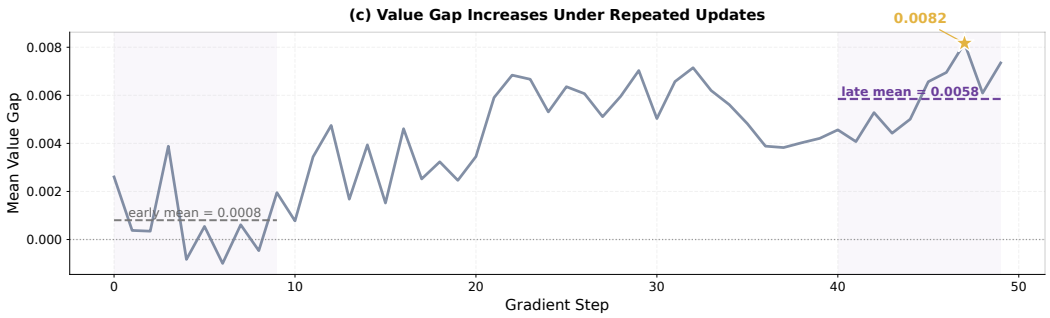


Figure 11: Repeated updates on a fixed batch progressively widen the value gap between boosted and suppressed tokens.