DELTA: DISCRIMINATIVE TOKEN CREDIT ASSIGNMENT FOR REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has emerged as a central technique for improving the reasoning capabilities of large language models. Despite its effectiveness, how response-level rewards translate into token-level probability changes remains poorly understood. We introduce a discriminator view of RLVR updates, showing that the policy-gradient update direction implicitly acts as a linear discriminator over token-gradient vectors and thereby determines which token probabilities are increased or decreased during learning. Under standard sequence-level RLVR, this discriminator is constructed from positive- and negative-side centroids formed by advantage-weighted averaging of token-gradient vectors. However, such centroid construction can be dominated by shared high-frequency patterns, such as formatting tokens, diluting sparse yet discriminative directions that better distinguish high-reward responses from low-reward ones. To address this limitation, we propose **DelTA**, a discriminative token credit assignment method that estimates token coefficients to amplify side-specific token-gradient directions and downweight shared or weakly discriminative ones. These coefficients reweight a self-normalized RLVR surrogate, making the effective side-wise centroids more contrastive and thereby reshaping the RLVR update direction. On seven mathematical benchmarks, DelTA outperforms the strongest same-scale baselines by 3.26 and 2.62 average points on Qwen3-8B-Base and Qwen3-14B-Base, respectively. Additional results on code generation, a different backbone, and out-of-domain evaluations further demonstrate the generalization ability of DelTA.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a key paradigm for improving the reasoning ability of large language models (LLMs), with strong gains in mathematics (Shao et al., 2024; Yang et al., 2024), code generation (Hui et al., 2024; Shojaee et al., 2023; Le et al., 2022), and formal problem solving (Guo et al., 2025; Team et al., 2025). RLVR optimizes response-level verifiable rewards, such as answer correctness, without requiring dense process-level annotations. This response-level supervision creates a granularity mismatch: each response provides a single scalar advantage, while the policy update is accumulated through token-level terms. Recent studies show that RLVR induces sparse token-level distributional shifts, where substantial probability changes concentrate on a small subset of tokens while most token distributions change little (Meng et al., 2026; Ma et al., 2026). This contrast suggests that sequence-level RLVR contains an implicit token-level selection mechanism that is not directly specified by the reward signal. Hence, an essential question arises: *which token probabilities are increased or decreased by an RLVR update, and what determines these changes?*

We introduce a discriminator view of RLVR to explain this implicit token selection. Although an RLVR update is usually viewed as a parameter-space movement, the same update also defines a token-level decision rule: it determines whether a candidate-token probability is increased or decreased by the update. The rule works by comparing token-gradient directions. For a sequence-level RLVR objective, the update direction contrasts token-gradient aggregates from positive-advantage responses and negative-advantage responses. After normalization, these aggregates define positive- and negative-side reference directions. A candidate-token probability is increased when its token-gradient vector

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.

aligns more with the positive-side reference direction than with the negative-side reference direction, and is decreased otherwise. In this sense, the RLVR update acts as an implicit linear discriminator over candidate token-gradient vectors. This view suggests that RLVR update directions can be understood and improved by analyzing and shaping the discriminator induced by the update.

Following the insights, further investigation indicates that standard sequence-level RLVR updates form the two-side directions by averaging token-gradient vectors from positive- and negative-advantage responses, yielding two centroids. Such centroids are natural summaries of each side, but a good within-side summary is not necessarily a good between-side discriminator. In reasoning tasks, higher- and lower-reward responses often share substantial common structure, such as formatting tokens or problem-specific entities. Because these shared patterns appear on both sides and occur frequently, their token-gradient directions can pull both centroids toward common background structure. Consequently, the induced discriminator may overemphasize task-agnostic commonalities and undermine sparse directions that better distinguish higher- from lower-reward responses.

To address this limitation, we propose **D**iscriminativ**e** signa**l**-guided **T**oken Credit **A**ssignment (**DelTA**). DelTA reshapes the induced RLVR discriminator by reweighting token-gradient terms in the RLVR surrogate. It estimates token coefficients that assign larger weights to token-gradient vectors more characteristic of their own advantage side than of the opposite side, while assigning smaller weights to shared or weakly discriminative directions. These coefficients change the effective aggregates that define the discriminator, making its positive and negative reference directions more contrastive and thereby reshaping the RLVR update direction. Empirically, DelTA consistently improves strong RLVR baselines. On seven mathematical benchmarks, it surpasses the strongest same-scale baseline by 3.26 average points on Qwen3-8B-Base and 2.62 points on Qwen3-14B-Base. It also improves code generation and generalizes to another backbone and out-of-domain evaluations.

In summary, our contributions are threefold. First, we introduce a local discriminator view of sequence-level RLVR, showing that policy-gradient updates induce an implicit linear discriminator over token-gradient vectors and thereby determine local token-probability changes. Second, using this view, we trace a limitation of standard sequence-level RLVR to the construction of the update direction: the side-wise centroids that form the induced discriminator can be pulled toward shared, high-frequency token-gradient directions, weakening its ability to separate token-gradient directions from higher- and lower-reward responses. Third, we propose DelTA, which reweights token-gradient terms by their positive-negative discriminative signal in a self-normalized RLVR surrogate, making the effective side-wise centroids more contrastive and consistently improving strong RLVR baselines across mathematical reasoning, code generation, different backbones, and out-of-domain evaluations.

2 PRELIMINARIES

We review the critic-free group-relative RLVR framework, taking DAPO as the main concrete example. For a prompt q, let $\{o_i\}_{i=1}^{G}$ be a group of sampled responses, where $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$. Each response receives a sequence-level reward R_i. Let μ_R and σ_R be the mean and standard deviation of rewards within the sampled group, and let $\epsilon_A > 0$ be a small numerical constant. The group-normalized advantage and token-level importance ratio are given by

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon_A}, \qquad r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}.$$

DAPO-style Surrogate. DAPO (Yu et al., 2025) is a state-of-the-art critic-free group-relative RLVR method that optimizes a clipped surrogate objective with two key designs relevant to this work: asymmetric clipping and token-level normalization over all response tokens. The expected objective is defined as

$$J_{\text{DAPO}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \text{clip}\big(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_i\Big)\right].$$

Here $\hat{A}_i$ is defined at the response level and is therefore shared by all tokens in the same response. The per-token contribution to the surrogate is nevertheless accumulated through the token-level ratio $r_{i,t}(\theta)$, which provides the basic object for the token-gradient analysis in the next section.

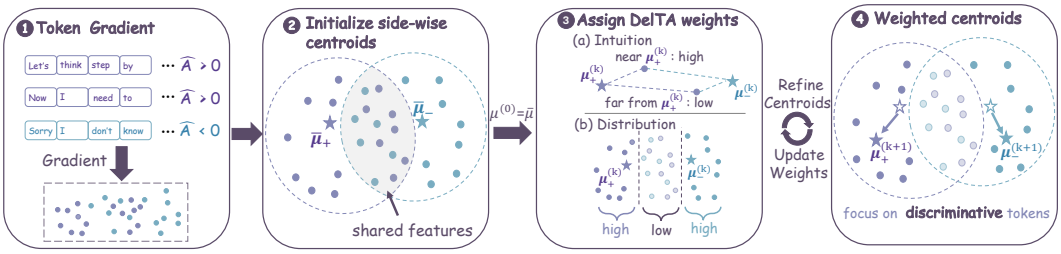


Figure 1: Overview of DelTA. DelTA estimates token coefficients from the contrast between positive- and negative-advantage token-gradient aggregates, and uses the coefficients to reweight the sequence-level RLVR objective.

The original DAPO algorithm also includes dynamic sampling to encourage each sampled group to contain both correct and incorrect responses. This component affects rollout filtering rather than the form of the per-token surrogate above. We disable dynamic sampling for all methods in our experiments, and focus our analysis on the update rule induced by the surrogate objective.

3 METHOD

Recent studies suggest that RLVR induces sparse and targeted changes at the token-distribution level: only a small fraction of token distributions undergo substantial shifts, while most remain nearly unchanged (Meng et al., 2026; Ma et al., 2026). For sequence-level RLVR, this sparsity is not directly explained by the reward signal itself, since all tokens in the same response share the same scalar advantage. This suggests that the token-level selection effect is induced not by explicit token rewards, but by how token-gradient vectors are aggregated in the policy-gradient update.

We therefore analyze this induced selection effect by examining how sequence-level RLVR updates determine which candidate-token probabilities are increased or suppressed. Section 3.1 investigates this question using DAPO as a concrete instance and derives the discriminator view induced by RLVR updates. Although DAPO serves as the primary showcase, the resulting conclusions rely only on the update being expressible as an advantage-weighted aggregation of token-gradient vectors, and therefore naturally extend to a broader class of sequence-level RLVR objectives, as discussed in Appendix C. Building on this analysis, Section 3.2 introduces DelTA, a new critic-free group-relative RLVR method that reweights token-gradient terms to reshape the induced discriminator and the corresponding RLVR update direction.

3.1 A LOCAL DISCRIMINATOR VIEW OF RLVR UPDATES

To understand how sequence-level RLVR implicitly selects tokens, we view the local policy update not only as a parameter update, but also as an implicit discriminator in token-gradient space. For sequence-level RLVR objectives, the update direction contrasts token-gradient aggregates from positive- and negative-advantage responses. After normalization, these aggregates define two side-wise reference directions. A candidate token log-probability is locally increased when its token-gradient vector aligns more with the positive reference direction than with the negative one, and is decreased otherwise.

Formally, let c denote an arbitrary generation context, x a candidate next token under this context, and $\pi_\theta(x \mid c)$ the policy model. For a local parameter update $\Delta\theta$ around θ_{old}, a first-order Taylor expansion gives

$$\Delta \log \pi(x \mid c) := \log \pi_{\theta_{\text{old}}+\Delta\theta}(x \mid c) - \log \pi_{\theta_{\text{old}}}(x \mid c) \approx \left(\nabla_\theta \log \pi_\theta(x \mid c) \Big|_{\theta=\theta_{\text{old}}} \right)^\top \Delta\theta. \quad (1)$$

Thus, once x, c, and θ_{old} are fixed, the local increase or decrease of the candidate token probability is determined by the inner product between its token-gradient vector and the update direction $\Delta\theta$. In the following analysis, we focus on the local update direction (i.e., $\Delta\theta$).

For concreteness, consider the DAPO-style sequence-level surrogate used in our analysis. Let $\{o_i\}_{i=1}^G$ be a rollout group, and let $\hat{A}_i$ be the group-normalized advantage of response o_i. Around θ_{old}, clipping

is locally inactive because $r_{i,t}(\theta_{\text{old}}) = 1$. The local policy-gradient update can therefore be written as an advantage-weighted aggregation of sampled-token gradients. Separating this aggregation by the sign of the response-level advantage gives

$$\Delta\theta_{\text{RLVR}} \propto \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i \, v_{i,t} - \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| \, v_{i,t}, \qquad v_{i,t} := \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Big|_{\theta=\theta_{\text{old}}}. \tag{2}$$

We refer to token-gradient vectors from responses with $\hat{A}_i > 0$ as the *positive side*, and those from responses with $\hat{A}_i < 0$ as the *negative side*. Throughout this analysis, $v_{i,t}$ denotes the exact full-parameter token-gradient vector. A detailed derivation of Eq. (2) from the DAPO surrogate is provided in Appendix A. We use this local characterization as an analysis and design principle for shaping the policy-update direction, rather than as an exact description of the full nonlinear clipped RLVR training trajectory.

Eq. (2) contains two components: the total mass of each side and the reference direction of each side. Let $M_+ = \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i$ and $M_- = \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i|$ denote the total positive and negative advantage masses. Then Eq. (2) can be rewritten as

$$\Delta\theta_{\text{RLVR}} \propto M_+ \bar{\mu}_+ - M_- \bar{\mu}_-, \quad \bar{\mu}_+ = \frac{\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i \, v_{i,t}}{M_+}, \quad \bar{\mu}_- = \frac{\sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| \, v_{i,t}}{M_-}. \tag{3}$$

Here, M_+ and M_- determine the total strength of the two advantage sides, while $\bar{\mu}_+$ and $\bar{\mu}_-$ are their normalized aggregate directions. Substituting Eq. (3) into Eq. (1) yields

$$\Delta \log \pi(x \mid c) \propto M_+ \left(\nabla_\theta \log \pi_\theta(x \mid c)\Big|_{\theta=\theta_{\text{old}}} \right)^\top \bar{\mu}_+ - M_- \left(\nabla_\theta \log \pi_\theta(x \mid c)\Big|_{\theta=\theta_{\text{old}}} \right)^\top \bar{\mu}_-. \tag{4}$$

The two terms in Eq. (4) define the positive-side and negative-side scores assigned to the candidate token-gradient vector. The candidate token probability is locally increased when its positive-side score exceeds its negative-side score, and decreased otherwise. In this sense, the update direction has a dual role: in parameter space, it is a policy-update direction; in token-gradient space, it acts as an implicit linear discriminator over candidate token-gradient vectors. This discriminator is not explicitly parameterized or separately trained; it is induced by the policy-gradient update itself. This duality suggests a reverse design perspective: since the update direction induces a discriminator in token-gradient space, we can instead ask how to shape this induced discriminator and adjust the update direction accordingly. Thus, **RLVR update directions can be understood and improved by studying and shaping the local discriminator induced by the update**.

For the induced discriminator, the central objects are the side-wise reference directions $\bar{\mu}_+$ and $\bar{\mu}_-$. Under the standard sequence-level RLVR update, these directions are simply the advantage-weighted centroids of the token-gradient vectors on the positive and negative sides. Equivalently, they are weighted least-squares summaries that minimize within-side squared distances, as shown in Appendix B. Such centroids are natural if the goal is to summarize each side independently. However, the induced discriminator uses them for a different purpose: distinguishing positive-advantage token gradients from negative-advantage token gradients.

This creates a mismatch between within-side summarization and between-side discrimination. In RLVR, positive- and negative-advantage responses often share frequent token patterns, such as common formatting tokens or problem-specific entities. The corresponding token-gradient directions can dominate both side-wise centroids, making the positive and negative reference directions less discriminative and diluting rarer directions that better separate higher-reward responses from lower-reward responses. From a classical discriminative perspective, good within-side summaries are not necessarily good between-side discriminators (Cohen et al., 2013; Zhao et al., 2024; Khosla et al., 2020).

This motivates a centroid-level design principle: to obtain a better local update direction, we can reshape the side-wise centroids that define the induced discriminator. Changing these centroids changes the scores assigned to candidate token-gradient vectors, and therefore changes which token probabilities are locally increased or decreased. This principle motivates DelTA: we reshape the effective side-wise centroids by assigning larger weights to token-gradient directions that better distinguish the two advantage sides.

3.2 DELTA: DISCRIMINATIVE SIGNAL-GUIDED TOKEN CREDIT ASSIGNMENT

DelTA implements the centroid-level principle above by reweighting token terms in the RLVR surrogate. Since the side-wise centroids are induced by weighted token-gradient aggregation rather than separately parameterized, changing token weights directly reshapes these centroids, and hence the induced discriminator and local update direction. At a high level, DelTA has three steps. First, it initializes the positive and negative reference directions from the original advantage-weighted centroids. Second, it refines these directions through a small number of alternating steps: with the current centroids fixed, DelTA estimates discriminative token scores; with the scores fixed, it recomputes each side-wise centroid as a score-weighted average of token-gradient vectors from that side. Third, it maps the final scores to bounded coefficients and uses them to reweight the sequence-level RLVR surrogate.

The formulation below is written in terms of token-gradient vectors $v_{i,t}$. In the exact version, these vectors are the full-parameter gradients defined in Section 3.1. In practice, explicitly materializing full-parameter gradients for all rollout tokens is computationally prohibitive at LLM scale, so we use a layer-restricted LM-head gradient representation to compute the stop-gradient token coefficients. This proxy affects only the coefficient computation; the analysis remains formulated with full-parameter token gradients, and the resulting weighted RLVR objective is still optimized over the full policy parameters. Further details and proxy ablations are provided in Appendix D.

We initialize the refinement from the original advantage-weighted centroids, $\mu_+^{(0)} = \bar{\mu}_+$ and $\mu_-^{(0)} = \bar{\mu}_-$. DelTA then runs K stop-gradient refinement iterations. At iteration $k = 0, \dots, K-1$, DelTA first estimates a soft discriminative score $\alpha_{i,t}^{(k)}$ for each token-gradient vector. We describe the positive side; the negative side is obtained symmetrically by swapping $\mu_+^{(k)}$ and $\mu_-^{(k)}$, and by replacing $\gamma_+^{(k)}$ with $\gamma_-^{(k)}$. For a positive-advantage token, DelTA assigns a larger score when $v_{i,t}$ is closer to the positive centroid than to the negative centroid. Specifically, $\alpha_{i,t}^{(k)}$ is defined by the entropy-regularized assignment problem

$$\alpha_{i,t}^{(k)} = \arg\max_{\alpha \in [0,1]} \alpha\Big(\|v_{i,t} - \mu_-^{(k)}\|_2^2 - \|v_{i,t} - \mu_+^{(k)}\|_2^2\Big) + \gamma_+^{(k)} h(\alpha), \qquad \hat{A}_i > 0, \tag{5}$$

where $h(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ is the binary entropy regularizer, and $\gamma_+^{(k)} > 0$ is a side-specific temperature for the positive-side assignment. The distance-margin term is positive when $v_{i,t}$ is closer to the positive centroid than to the negative centroid, so maximizing Eq. (5) assigns a larger score to tokens that are more characteristic of their own side. The entropy regularizer and the temperature jointly control the softness of this assignment: smaller temperatures make the score closer to a hard decision, while larger temperatures produce smoother scores. We use squared Euclidean distances to stay consistent with the centroid construction, as detailed in Appendix B.

For fixed centroids and temperatures, the closed-form solution is

$$\alpha_{i,t}^{(k)} = \begin{cases} \sigma\left(\dfrac{\|v_{i,t} - \mu_-^{(k)}\|_2^2 - \|v_{i,t} - \mu_+^{(k)}\|_2^2}{\gamma_+^{(k)}}\right), & \hat{A}_i > 0, \\ \sigma\left(\dfrac{\|v_{i,t} - \mu_+^{(k)}\|_2^2 - \|v_{i,t} - \mu_-^{(k)}\|_2^2}{\gamma_-^{(k)}}\right), & \hat{A}_i < 0, \end{cases} \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function. The side-specific temperatures $\gamma_+^{(k)}$ and $\gamma_-^{(k)}$ adapt the assignment scale for the two advantage sides; their computation is detailed in Appendix F. A derivation of Eq. (6) is provided in Appendix E.

Thus, $\alpha_{i,t}^{(k)}$ is large when the token-gradient vector is more representative of its own advantage side than of the opposite side, and small for shared or weakly discriminative directions.

Given these scores, DelTA updates the centroids as score-weighted within-side averages:

$$\mu_+^{(k+1)} = \frac{\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i\, \alpha_{i,t}^{(k)}\, v_{i,t}}{\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i\, \alpha_{i,t}^{(k)}}, \qquad \mu_-^{(k+1)} = \frac{\sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i|\, \alpha_{i,t}^{(k)}\, v_{i,t}}{\sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i|\, \alpha_{i,t}^{(k)}}. \tag{7}$$

This refinement gives larger influence to token-gradient vectors that are more characteristic of their own side, while downweighting shared or weakly discriminative directions. It is used only to compute

stop-gradient token coefficients; no gradients are propagated through the refinement, and no additional loss term is added.

After the final refinement step, DelTA recomputes raw scores $\alpha^{\star}_{i,t}$ with the refined centroids and maps them to bounded coefficients $\lambda_{i,t} = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\alpha^{\star}_{i,t}$. The bounded range prevents extreme reweighting while preserving the ranking of the discriminative scores. DelTA then replaces the uniform token average in DAPO with the following self-normalized weighted surrogate:

$$J_{\mathrm{DelTA}}(\theta) = \mathbb{E}\left[\frac{1}{Z_\lambda}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\lambda_{i,t}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}\big(r_{i,t}(\theta), 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}}\big)\hat{A}_i\Big)\right]. \quad (8)$$

Here, $Z_\lambda = \sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\lambda_{i,t}$ is the total token-coefficient mass. Around θ_{old}, Eq. (8) changes each sampled-token contribution from $\hat{A}_i v_{i,t}$ to $\lambda_{i,t}\hat{A}_i v_{i,t}$. This reweighting reshapes the effective side-wise centroids, and hence the induced discriminator and local RLVR update direction, by amplifying side-specific token-gradient directions and downweighting shared or weakly discriminative ones. The coefficients are stop-gradient quantities computed once per rollout batch, fixed across optimization epochs, and recomputed after new trajectories are sampled. Full details are provided in Appendix F.

4 Experiments

4.1 Experimental Setup

We train on two backbones, Qwen3-8B-Base (Yang et al., 2025) and Qwen3-14B-Base, using DeepMath-103K (He et al., 2025) with VeRL (Sheng et al., 2024). For DelTA, we set $[\lambda_{\min}, \lambda_{\max}] = [0.8, 1.2]$ and use one refinement iteration ($K = 1$). We compare against DAPO (Yu et al., 2025), DAPO w/ Forking Tokens (DAPO w/ FT) (Wang et al., 2025), SAPO (Gao et al., 2025), and FIPO (Ma et al., 2026) using the same shared training settings, with method-specific hyperparameters detailed in Appendix H. We disable dynamic sampling for all methods to isolate the effect of the policy-update objective. We evaluate our models on seven mathematical benchmarks: AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), AIME26 (Zhang & Math-AI, 2026), HMMT25 (February) (Balunović et al., 2025), HMMT25 (November), HMMT26 (February) and Brumo25. To better reveal each model's long-reasoning capability, we set the maximum generation length during evaluation to 30,000 tokens. For each problem, we independently sample 16 responses and report their average performance. Detailed settings and baseline descriptions are provided in Appendix G and Appendix H, respectively.

4.2 Main Results

Table 1 reports the main results on mathematical reasoning benchmarks. DelTA consistently outperforms all same-scale RL baselines on both Qwen3-8B-Base and Qwen3-14B-Base, achieving the best result on every benchmark and the highest average score at both scales. Compared with the strongest same-scale baseline, DelTA improves the average score from 25.14 to 28.40 on the 8B backbone and from 37.29 to 39.91 on the 14B backbone. Appendix J.1 details the computational overhead of DelTA, while Appendix J.12 shows that DelTA continues to outperform SAPO under two approximately matched end-to-end wall-clock budgets. We further evaluate DelTA beyond the main math setting: Appendix J.3 shows that DelTA also improves DAPO on code generation benchmarks, and Appendix J.2 shows that DelTA remains effective on Olmo3-7B-Base (Olmo et al., 2025).

4.3 Training Dynamics

Figure 2 compares the training dynamics of DelTA and DAPO on Qwen3-8B-Base. The two methods show similar reward trajectories in the early stage, but diverge afterwards: DAPO plateaus and slightly degrades, whereas DelTA continues to improve and reaches a higher final reward. The response-length and entropy curves suggest that this divergence is not merely due to shorter answers. DAPO shifts toward shorter responses with rising entropy, whereas DelTA maintains longer responses with lower entropy and higher reward, indicating more stable and confident long-reasoning behavior.

Table 1: Main results on seven mathematical reasoning benchmarks for Qwen3-8B-Base and Qwen3-14B-Base. DelTA consistently outperforms all compared same-scale RL baselines. The best results are in **bold**, and the second-best results are underlined.

Method	AIME24	AIME25	AIME26	HMMT25 (Feb.)	HMMT25 (Nov.)	HMMT26 (Feb.)	Brumo25	Avg.
Qwen3-8B-Base								
DAPO	34.79	23.33	24.17	13.54	12.08	16.86	36.46	22.95
DAPO w/ FT	36.67	23.96	26.46	15.62	15.42	17.05	39.17	24.80
SAPO	38.75	24.37	26.25	14.58	16.04	17.42	39.37	25.14
FIPO	37.50	23.13	23.96	14.58	12.92	17.99	37.71	23.89
DelTA	**43.13**	**26.46**	**28.12**	**18.33**	**18.54**	**20.27**	**44.79**	**28.40**
Qwen3-14B-Base								
DAPO	51.25	32.29	39.79	19.79	30.00	25.38	48.13	35.09
DAPO w/ FT	54.37	33.75	41.46	20.42	31.67	24.81	52.08	36.77
SAPO	53.96	34.17	41.46	20.62	28.33	24.05	50.21	35.94
FIPO	54.58	35.00	42.50	21.46	32.29	24.43	52.08	37.29
DelTA	**56.87**	**37.92**	**45.21**	**26.04**	**32.92**	**26.89**	**54.79**	**39.91**

This behavior is consistent with our discriminator view. In standard sequence-level aggregation, shared background directions can dominate the side-wise centroids and weaken the contrast of the induced update direction. DelTA counteracts this by upweighting side-specific token-gradient directions and downweighting shared or weakly discriminative ones. As a result, the effective reference directions become more contrastive, helping the update sustain useful long-reasoning trajectories without an explicit length incentive.

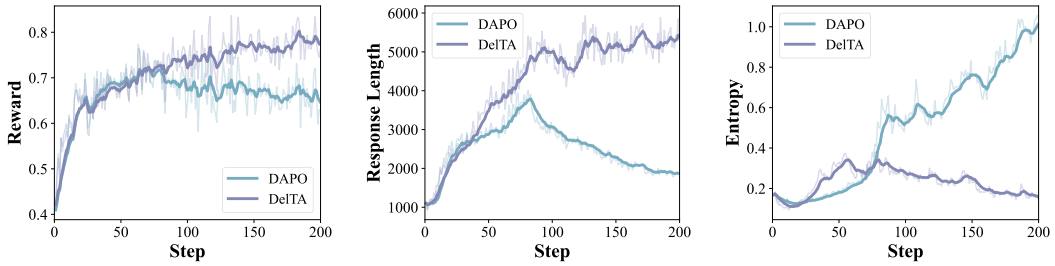


Figure 2: Training dynamics of DelTA compared with DAPO. **Left:** Reward; **Middle:** Response Length; **Right:** Entropy.

5 Analysis

Unless stated otherwise, we analyze DelTA on Qwen3-8B-Base using four representative math benchmarks: AIME25, AIME26, HMMT25-Nov, and HMMT26-Feb, abbreviated as HMMT25 and HMMT26. We organize the analysis around five questions: **Q1:** Is the opposite-side comparison necessary? **Q2:** Does $\lambda_{i,t}$ capture useful token-level learning signals? **Q3:** Are DelTA's design components necessary? **Q4:** Is DelTA sensitive to its hyperparameters? **Q5:** Does DelTA generalize to out-of-domain evaluation? We address Q1 to Q3 below, and Q4 and Q5 in Appendices J.4 and J.5, respectively.

5.1 Q1: Is the opposite-side comparison necessary?

To test whether DelTA's gain can be explained by own-side centrality alone, we construct a within-side-only variant. This variant keeps the same coefficient normalization and weighted DAPO objective as DelTA, but removes the opposite-side distance from the assignment score. For to-

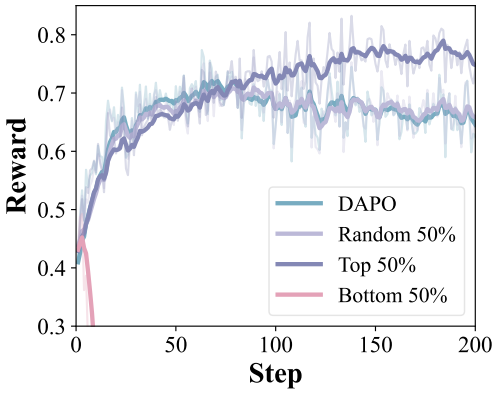


Figure 3: Training reward under different token-selection strategies.

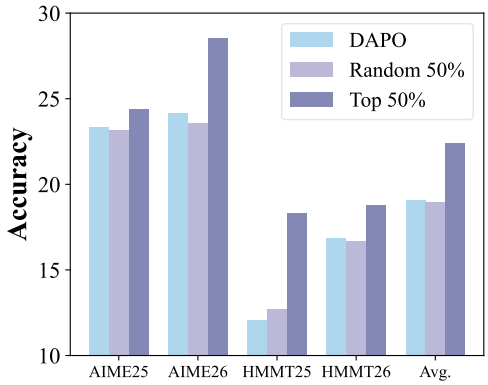


Figure 4: Evaluation accuracy under different token-selection strategies.

kens from positive-advantage responses, we use $\alpha_{i,t} = \sigma\left(-\frac{\|v_{i,t}-\mu_+\|_2^2}{\gamma_+}\right)$, and define the negative side symmetrically with μ_- and γ_-. Thus, this variant assigns larger coefficients to tokens closer to their own-side centroid, without considering their distance to the opposite-side centroid.

Table 2 shows that the within-side-only variant performs worse than both DelTA and the DAPO baseline. This result indicates that DelTA's gains cannot be explained by simply assigning larger weights to tokens close to their own-side centroid. In fact, own-side centrality alone can be misleading: tokens near a side-wise centroid may correspond to shared patterns rather than directions that distinguish positive- from negative-advantage responses. The opposite-side comparison is therefore essential, because it assigns high coefficients only to directions that are relatively more representative of their own side than of the opposite side.

Table 2: Effect of using only within-side information. The best results are in **bold**.

Method	AIME25	AIME26	HMMT25	HMMT26	Avg.
DelTA	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**
DAPO	23.33	24.17	12.08	16.86	19.05
Within-side only	21.67	22.08	11.04	17.05	17.94

5.2 Q2: Does $\lambda_{i,t}$ capture useful token-level learning signals?

We next test whether the DelTA coefficient $\lambda_{i,t}$ identifies token-gradient directions with useful learning value. To isolate the ranking effect from the continuous reweighting objective, we use $\lambda_{i,t}$ only for hard token selection. Specifically, we compute DelTA coefficients for all valid response tokens and train DAPO using only the top 50% tokens ranked by $\lambda_{i,t}$. As controls, we train on a random 50% subset and on the bottom 50% tokens, while keeping the standard DAPO loss unchanged.

Figure 3 shows a sharp separation between these selections. Training on the top-λ tokens consistently outperforms both full-token DAPO and random 50% selection, even though it uses only half of the token-gradient terms. In contrast, training on the bottom-λ tokens quickly collapses. The evaluation results in Figure 4 show the same trend: top-λ training improves accuracy across benchmarks, whereas random selection remains close to DAPO. These results indicate that DelTA's coefficients capture more than a benign sparsification signal. If the gain came merely from reducing the number of optimized tokens, random 50% selection should provide a similar benefit, which it does not.

Together, these results indicate that DelTA's coefficients identify token subsets with markedly different learning utility: high-weight tokens support more effective policy optimization, whereas training on low-weight tokens alone fails to sustain improvement.

5.3 Q3: Are the design components of DelTA necessary?

We ablate five design choices in DelTA while keeping all other training settings unchanged: **w/o adaptive** γ fixes both assignment temperatures to their mean value computed from the first rollout batch and keeps this value unchanged throughout training; **w/o** $h(\alpha)$ removes the entropy regularizer and turns the soft assignment into a hard $0/1$ decision; **w/o** λ**-norm** keeps the token coefficients in the numerator, but replaces DelTA's coefficient-mass normalizer $1/\sum_{i,t}\lambda_{i,t}$ with the standard DAPO token-count normalizer $1/\sum_i |o_i|$; **w/o range map** removes the linear mapping from assignment scores to $[\lambda_{\min}, \lambda_{\max}]$, and directly uses the raw scores $\alpha_{i,t} \in (0, 1)$ as token weights; and **w/o refinement** estimates coefficients only from the initial side-wise centroids.

Table 3 shows that each component contributes to DelTA. Removing any design choice reduces the average score. The largest drop comes from **w/o refinement**, indicating that one-shot coefficients from the initial centroids are insufficient. The drops from **w/o range map** and **w/o** $h(\alpha)$ suggest that bounded soft coefficients are more stable than raw scores or hard assignments. Finally, the degradation from **w/o adaptive** γ and **w/o** λ**-norm** shows that scale adaptation and coefficient-mass normalization are both useful.

Table 3: Ablation study of DelTA. The best results are in **bold**, and the second-best results are underlined.

Method	AIME25	AIME26	HMMT25	HMMT26	Avg.
Full DelTA	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**
w/o adaptive γ	<u>25.00</u>	26.04	<u>16.04</u>	17.99	21.19
w/o $h(\alpha)$	24.37	<u>26.87</u>	15.42	17.42	20.93
w/o λ-norm	24.37	26.25	15.83	<u>19.32</u>	<u>21.39</u>
w/o range map	24.79	25.83	15.83	17.05	20.78
w/o refinement	23.13	25.42	15.42	16.29	19.97

6 Related Work

Reinforcement learning has become an effective paradigm for improving LLM reasoning, especially in domains with verifiable feedback such as mathematics. Representative methods include PPO-style optimization and recent critic-free group-relative objectives such as GRPO and DAPO (Schulman et al., 2017; Shao et al., 2024; Yu et al., 2025). Recent work further studies the mechanisms of RLVR (Yue et al., 2025; Huan et al., 2025; Meng et al., 2026), improves training stability and efficiency (Zheng et al., 2025; Gao et al., 2025; Liu et al., 2025), and explores off-policy or semi-off-policy training (Yan et al., 2025; Zhang et al., 2025a).

A central challenge in RLVR is that rewards are usually provided at the response level, while policy updates are applied at the token level. Prior work addresses this mismatch through token-level or step-level reweighting (Kazemnejad et al., 2025; Xie et al., 2025), process reward models or learned value functions (Cui et al., 2025; Zhang et al., 2025b), and token-selection signals such as entropy or future influence (Wang et al., 2025; Ma et al., 2026). Instead of relying on external dense rewards, value estimates, or auxiliary token-selection rules, DelTA reweights the RLVR surrogate using token coefficients derived from the positive-negative discriminator induced by the update itself.

7 Conclusion

We studied token-level learning in RLVR from a local discriminative perspective. Using DAPO as a representative sequence-level objective, we showed that its policy-gradient update induces an implicit linear discriminator over candidate token-gradient vectors. This discriminator is defined by positive and negative side-wise centroids, which can be dominated by shared, weakly discriminative directions rather than directions that distinguish higher- from lower-reward responses. Motivated by the mismatch between within-side summarization and between-side discrimination, we proposed **DelTA**, which estimates token coefficients from refined centroid contrasts and uses them to reweight a self-normalized clipped RLVR surrogate. Experiments on mathematical reasoning, code generation, another model family, and out-of-domain benchmarks show consistent improvements over strong RLVR baselines. Overall, our results suggest that shaping the discriminator induced by RLVR updates is a useful route for improving token-level credit assignment under sequence-level supervision.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and language polishing, including improving grammar, clarity, and presentation, as well as for consistency and error checking. We also used generative AI tools to assist with the implementation, debugging, and verification of a limited portion of the code. We did not use generative AI tools to generate the core research ideas, formulate the main scientific claims, design the experiments, or generate experimental results.

All AI-assisted text was carefully reviewed and revised by the authors. AI-assisted code was manually inspected, tested, and verified to ensure that it correctly implemented the intended algorithms and experimental procedures. All analyses, interpretations, and conclusions were independently verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies token-level credit assignment for reinforcement learning with verifiable rewards. Its potential positive impact is to improve the effectiveness and efficiency of training reasoning-capable language models, especially in domains where correctness can be automatically verified, such as mathematics and code generation. By providing a more interpretable view of how sequence-level RLVR updates allocate token-level credit, the work may also help researchers better diagnose and improve RL training dynamics.

At the same time, stronger reasoning models may also be misused in settings such as automated generation of misleading content, scalable code generation for harmful purposes, or other dual-use applications. DelTA does not introduce new user-facing deployment mechanisms, new datasets containing sensitive information, or new privacy-invasive capabilities, but it may improve the capabilities of models trained with RLVR. Responsible deployment of models trained with such methods should therefore follow standard safety practices, including appropriate evaluation, monitoring, access control when needed, and task-specific safeguards for high-risk applications.

REPRODUCIBILITY STATEMENT

Our code is available at `https://anonymous.4open.science/r/DelTA_code`. The assumptions and derivations underlying our local analysis are provided in Appendices A to C. Gradient-proxy definitions, token-score derivations, and the coefficient-computation procedure are detailed in Appendices D to F. Training and evaluation settings, baseline descriptions, and the significance-testing protocol are documented in Appendices G, H, and I, respectively.

REFERENCES

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL `https://matharena.ai/`.

Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

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

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Zhewei Kang, Aosong Feng, Sergey Levine, Dawn Song, and Xuandong Zhao. Vimpo: Value-implicit policy optimization for llms. *arXiv preprint arXiv:2606.20008*, 2026.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms, 2025. URL https://arxiv.org/abs/2410.01679.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

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.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

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.

Parshin Shojaee, Aneesh Jain, Sindhu Tipirneni, and Chandan K Reddy. Execution-based code generation using deep reinforcement learning. *arXiv preprint arXiv:2301.13816*, 2023.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 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.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. Capo: Towards enhancing llm reasoning through generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025.

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

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chih-Kuan Yeh, Ankur Taly, Mukund Sundararajan, Frederick Liu, and Pradeep Ravikumar. First is better than last for language data influence. *Advances in Neural Information Processing Systems*, 35:32285–32298, 2022.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*, 2025a.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025b.

Shuping Zhao, Bob Zhang, Jian Yang, Jianhang Zhou, and Yong Xu. Linear discriminant analysis. *Nature Reviews Methods Primers*, 4(1):70, 2024.

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

A Derivation of the Local DAPO Update

In this appendix, we derive the local first-order update form used in Section 3.1. We focus on the update rule induced by the DAPO surrogate. Since dynamic sampling is disabled in our training recipe, it is not included in the derivation.

For a fixed prompt q, let $\{o_i\}_{i=1}^{G}$ denote a sampled group of responses drawn from the old policy $\pi_{\theta_{\text{old}}}$, where $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$. Let

$$N := \sum_{i=1}^{G} |o_i|$$

be the total number of valid response tokens in the group. Conditioning on this rollout batch, the DAPO surrogate is

$$J_{\text{DAPO}}(\theta) = \frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta)\hat{A}_i,\ \text{clip}\big(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_i \Big),$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}.$$

We analyze a local update around θ_{old}. At this point, $r_{i,t}(\theta_{\text{old}}) = 1$ for every sampled token. Since 1 lies inside the clipping interval, clipping is locally inactive, and the local gradient of J_{DAPO} agrees with that of the unclipped surrogate:

$$\frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} r_{i,t}(\theta)\hat{A}_i.$$

Therefore,

$$\nabla_\theta J_{\text{DAPO}}(\theta)\Big|_{\theta=\theta_{\text{old}}} = \frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \hat{A}_i \nabla_\theta r_{i,t}(\theta)\Big|_{\theta=\theta_{\text{old}}}.$$

Because the denominator of $r_{i,t}(\theta)$ is fixed with respect to θ,

$$\nabla_\theta r_{i,t}(\theta)\Big|_{\theta=\theta_{\text{old}}} = \frac{\nabla_\theta \pi_\theta(o_{i,t} \mid q, o_{i,<t})\big|_{\theta=\theta_{\text{old}}}}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} = \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Big|_{\theta=\theta_{\text{old}}}.$$

Define the token-gradient vector

$$v_{i,t} := \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Big|_{\theta=\theta_{\text{old}}}.$$

Then

$$\nabla_\theta J_{\text{DAPO}}(\theta)\Big|_{\theta=\theta_{\text{old}}} = \frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \hat{A}_i\, v_{i,t}.$$

For a first-order gradient ascent step with learning rate $\eta > 0$,

$$\Delta\theta = \eta \nabla_\theta J_{\text{DAPO}}(\theta)\Big|_{\theta=\theta_{\text{old}}} = \frac{\eta}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \hat{A}_i\, v_{i,t}.$$

Thus, up to a positive proportionality constant,

$$\Delta\theta \propto \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \hat{A}_i\, v_{i,t}.$$

Separating this sum by the sign of the sequence-level advantage gives the positive-negative decomposition used in Section 3.1:

$$\Delta\theta \propto \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i \, v_{i,t} - \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| \, v_{i,t}.$$

Finally, we derive the induced local change in next-token log-probabilities. Let c denote an arbitrary generation context, and let x be a candidate next token under this context. Define

$$\Delta \log \pi(x \mid c) := \log \pi_{\theta_{\text{old}}+\Delta\theta}(x \mid c) - \log \pi_{\theta_{\text{old}}}(x \mid c).$$

A first-order Taylor expansion around θ_{old} gives

$$\Delta \log \pi(x \mid c) \approx \left(\nabla_\theta \log \pi_\theta(x \mid c) \Big|_{\theta=\theta_{\text{old}}} \right)^\top \Delta\theta.$$

Substituting the local DAPO update yields

$$\Delta \log \pi(x \mid c) \propto \left(\nabla_\theta \log \pi_\theta(x \mid c) \Big|_{\theta=\theta_{\text{old}}} \right)^\top \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \hat{A}_i \, v_{i,t}.$$

Equivalently, using the positive-negative decomposition,

$$\Delta \log \pi(x \mid c) \propto \left(\nabla_\theta \log \pi_\theta(x \mid c) \Big|_{\theta=\theta_{\text{old}}} \right)^\top \left(\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i \, v_{i,t} - \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| \, v_{i,t} \right).$$

This is the first-order next-token log-probability change analyzed in Section 3.1.

This expression describes the raw surrogate-gradient direction before optimizer preconditioning, weight decay, or gradient clipping. In practical training with Adam-type optimizers, the actual parameter step may apply an additional preconditioner to this gradient. Our analysis focuses on the update direction induced by the RLVR surrogate itself.

Appendix J.7 empirically examines whether this local discriminator retains predictive information about token log-probability changes after a practical DAPO optimization cycle.

A degenerate case occurs when one advantage side is empty. Under group-normalized advantages, if there is no positive-advantage response in a group, then there is also no negative-advantage response: all advantages in the group are zero. Such a group contributes zero to the local policy-gradient update and is therefore skipped in the positive-negative decomposition. Equivalently, the sums over empty sides are treated as zero, and centroid quantities that require nonzero side mass are computed only for rollout groups with both positive and negative advantage sides.

B Mean Directions as Weighted Least-Squares Centroids

In this appendix, we show that the side-wise mean directions used in the local discriminator view are exactly the weighted least-squares centroids of the positive and negative sides.

Recall the local token-gradient vectors

$$v_{i,t} := \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \Big|_{\theta=\theta_{\text{old}}},$$

where i indexes responses in the rollout group and t indexes valid response tokens. We split tokens according to the sign of the response-level advantage $\hat{A}_i$. Define the total positive and negative advantage masses as

$$M_+ := \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i, \qquad M_- := \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i|.$$

When $M_+ > 0$ and $M_- > 0$, the corresponding normalized side-wise mean directions are

$$\bar{\mu}_+ = \frac{\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i v_{i,t}}{M_+}, \qquad \bar{\mu}_- = \frac{\sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| v_{i,t}}{M_-}.$$

These are the positive- and negative-side reference directions used by the local discriminator.

We now show that $\bar{\mu}_+$ and $\bar{\mu}_-$ are weighted least-squares centroids. Consider first the positive side and define

$$F_+(\mu) := \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i \, \|v_{i,t} - \mu\|_2^2.$$

Differentiating with respect to μ gives

$$\nabla_\mu F_+(\mu) = 2 \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i(\mu - v_{i,t}) = 2M_+\mu - 2 \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i v_{i,t}.$$

Setting the gradient to zero yields

$$\mu = \frac{\sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i v_{i,t}}{M_+} = \bar{\mu}_+.$$

Moreover,

$$\nabla_\mu^2 F_+(\mu) = 2M_+ I,$$

which is positive definite whenever $M_+ > 0$. Thus, $\bar{\mu}_+$ is the unique minimizer of $F_+(\mu)$.

The negative side follows analogously. Define

$$F_-(\mu) := \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| \, \|v_{i,t} - \mu\|_2^2.$$

Then

$$\nabla_\mu F_-(\mu) = 2M_-\mu - 2 \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| v_{i,t}.$$

The first-order optimality condition gives

$$\mu = \frac{\sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i| v_{i,t}}{M_-} = \bar{\mu}_-.$$

Since

$$\nabla_\mu^2 F_-(\mu) = 2M_- I \succ 0$$

whenever $M_- > 0$, $\bar{\mu}_-$ is also the unique minimizer.

Therefore, the side-wise mean directions $\bar{\mu}_+$ and $\bar{\mu}_-$ are exactly the weighted least-squares centroids of the positive and negative token-gradient vectors, respectively. If one side has zero total mass, the corresponding centroid is undefined; such degenerate groups contribute no positive-negative centroid contrast and are skipped in the centroid construction.

C Representative RLVR Variants as Token-Weighted Centroid Estimators

In this appendix, we show that several representative RLVR variants can be interpreted through the same local centroid view. The goal is not to claim that these methods are identical, but to clarify how their local update rules modify the effective token weights used to estimate the positive and negative directions.

Consider a generic RLVR surrogate whose unclipped local update around the old policy can be written as

$$\Delta\theta \propto \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \rho_{i,t}\hat{A}_i v_{i,t}, \tag{9}$$

where

$$v_{i,t} = \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Big|_{\theta=\theta_{\text{old}}},$$

and $\rho_{i,t} \geq 0$ denotes the effective token weight induced by the objective. Global normalizers shared by all tokens are omitted when they do not affect the update direction.

Throughout this appendix, the effective token weights $\rho_{i,t}$ are treated as fixed stop-gradient quantities. Thus, the local update keeps only the gradient through the policy ratio $r_{i,t}(\theta)$.

Separating positive- and negative-advantage responses gives

$$\Delta\theta \propto M_+(\rho)\mu_+(\rho) - M_-(\rho)\mu_-(\rho), \tag{10}$$

where

$$\mu_+(\rho) = \frac{\sum_{i:\hat{A}_i>0}\sum_{t=1}^{|o_i|}\rho_{i,t}\hat{A}_i v_{i,t}}{\sum_{i:\hat{A}_i>0}\sum_{t=1}^{|o_i|}\rho_{i,t}\hat{A}_i}, \qquad \mu_-(\rho) = \frac{\sum_{i:\hat{A}_i<0}\sum_{t=1}^{|o_i|}\rho_{i,t}|\hat{A}_i| v_{i,t}}{\sum_{i:\hat{A}_i<0}\sum_{t=1}^{|o_i|}\rho_{i,t}|\hat{A}_i|}. \tag{11}$$

Here $M_+(\rho)$ and $M_-(\rho)$ are the corresponding total positive and negative masses, i.e., the denominators of the two centroids above. Therefore, under the local view, changing $\rho_{i,t}$ changes both the side-wise centroids and their relative masses.

GRPO. GRPO averages token losses within each response and then averages over responses. Around the old policy, where clipping is locally inactive, its update direction is

$$\Delta\theta_{\text{GRPO}} \propto \frac{1}{G}\sum_{i=1}^{G}\frac{\hat{A}_i}{|o_i|}\sum_{t=1}^{|o_i|} v_{i,t}. \tag{12}$$

Thus, up to a global constant,

$$\rho_{i,t}^{\text{GRPO}} = \frac{1}{|o_i|}.$$

Equivalently, GRPO first forms a response-level average token-gradient vector,

$$\bar{v}_i = \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} v_{i,t},$$

and then aggregates these response-level vectors with group-relative advantages:

$$\Delta\theta_{\text{GRPO}} \propto \sum_{i=1}^{G}\hat{A}_i\bar{v}_i.$$

Therefore, each response contributes one averaged direction. Longer responses have smaller per-token weights, while shorter responses have larger per-token weights. In the centroid view, GRPO estimates side-wise directions from response-balanced token-gradient averages.

DAPO. DAPO replaces response-level averaging with token-level normalization over all valid response tokens. Locally, its update direction is

$$\Delta\theta_{\text{DAPO}} \propto \sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\hat{A}_i v_{i,t}, \tag{13}$$

where the global token-count normalizer is omitted. Hence

$$\rho_{i,t}^{\text{DAPO}} = 1.$$

Compared with GRPO, DAPO gives equal weight to every valid token. Consequently, the total contribution of a response is proportional to its length. In the centroid view, DAPO changes the side-wise estimator from a response-balanced average to a token-balanced average.

DAPO with forking tokens. DAPO with forking tokens keeps only high-entropy tokens in the policy-gradient loss. Let

$$m_{i,t}^{\mathrm{FT}} = \mathbb{I}\big[H_{i,t} \geq \tau_{\rho}^{\mathcal{B}}\big]$$

denote the high-entropy token mask in the current batch. The local update becomes

$$\Delta\theta_{\mathrm{FT}} \propto \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} m_{i,t}^{\mathrm{FT}} \hat{A}_i v_{i,t}. \tag{14}$$

Thus,

$$\rho_{i,t}^{\mathrm{FT}} = m_{i,t}^{\mathrm{FT}}.$$

This objective changes the support of the centroid estimator: low-entropy tokens are removed, and the positive and negative centroids are estimated only from high-entropy tokens. In this view, forking-token filtering emphasizes high-entropy tokens as the main contributors to the side-wise directions.

FIPO. FIPO introduces a future-influence weight for each token. In its objective, the standard advantage is multiplied by a Future-KL importance weight $f_{i,t}$, and this weighted advantage is used inside a DAPO-style token-level surrogate. FIPO defines this weight from a discounted future probability-shift signal and clips it to control variance.

Ignoring clipping locally, the update direction is

$$\Delta\theta_{\mathrm{FIPO}} \propto \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} f_{i,t}\hat{A}_i v_{i,t}. \tag{15}$$

Therefore,

$$\rho_{i,t}^{\mathrm{FIPO}} = f_{i,t}.$$

In the centroid view, FIPO estimates the positive and negative centroids from future-influence-weighted token-gradient vectors. Tokens with larger future-influence weights receive larger centroid mass. This differs from entropy filtering: FIPO does not simply select uncertain tokens, but weights tokens according to a forward-looking estimate of their influence on later trajectory behavior.

Summary. Under a local first-order view, several representative RLVR variants can be interpreted as modifying the effective token weights $\rho_{i,t}$ in an advantage-weighted token-gradient aggregation. These weights determine the side-wise centroids $\mu_{+}(\rho)$ and $\mu_{-}(\rho)$, as well as their total masses. Existing methods adjust this estimator through length normalization, entropy filtering, or future-influence weighting. DelTA differs by assigning token weights according to whether a token gradient is more representative of its own side than of the opposite side, thereby aligning the centroid estimator with the discriminative structure of the local DAPO update.

D Last-layer Token-Gradient Proxy

For scalable coefficient estimation, we use a layer-restricted token-gradient proxy based on the standard fixed-representation view of the output layer. Let the LM head produce logits $z_t = Wh_t$, where $h_t \in \mathbb{R}^d$ is the final hidden state at step t, $W \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the LM-head matrix, and $p_t = \mathrm{softmax}(z_t)$. For the realized token y_t, the token log-probability is

$$\log p_t(y_t) = W_{y_t}^{\top} h_t - \log \sum_j \exp(W_j^{\top} h_t).$$

Differentiating with respect to the corresponding LM-head row gives

$$\nabla_{W_{y_t}} \log p_t(y_t) = \big(1 - p_t(y_t)\big) h_t.$$

Thus, under a frozen-representation approximation, we use $\big(1 - p_t(y_t)\big)h_t$ as a layer-restricted proxy for the token-gradient vector.

We do not use this proxy as an exact reconstruction of the full-parameter gradient. Instead, we use it as a scalable layer-restricted representation for estimating relative token coefficients. It is a pragmatic

Table 4: Proxy ablation study of DelTA. The best results are in **bold**.

Method	AIME25	AIME26	HMMT25	HMMT26	Avg.
Base DelTA	26.46	**28.12**	18.54	20.27	23.27
Top-K hidden-gradient proxy	**27.08**	27.71	**20.83**	**21.78**	**24.29**
Random λ	22.50	22.50	11.87	16.67	18.34

reduction that makes token-level analysis and reweighting feasible in large autoregressive language models. Prior work on gradient-based influence has similarly relied on first-order approximations and layer-restricted computations for scalability (Pruthi et al., 2020), and in NLP it is common to restrict such analysis to the last layer when full-parameter computations are impractical at scale (Yeh et al., 2022). This restriction should therefore be understood as an approximation, not an exact reconstruction of the full gradient signal (Yeh et al., 2022).

Proxy ablations. In addition to the default proxy above, we ablate the choice of layer-restricted proxy used for DelTA coefficient estimation. These ablations do not change the DelTA objective or the policy-gradient update; they only replace the stop-gradient proxy vectors used to compute centroids, distances, and token coefficients.

Let $\mathcal{K}_t$ denote the top-K vocabulary indices under the current logits, and let $\widetilde{p}_t$ be the softmax distribution renormalized within $\mathcal{K}_t$. We consider a top-K hidden-gradient proxy

$$\widehat{v}_t^{\mathrm{hid}} = W_{y_t} - \sum_{j \in \mathcal{K}_t} \widetilde{p}_t(j) W_j,$$

which is a top-K approximation to $\nabla_{h_t} \log p_t(y_t)$. Compared with the default output-row proxy $\big(1 - p_t(y_t)\big)h_t$, this proxy explicitly incorporates the competing high-probability tokens under the current policy.

We also include a random-coefficient baseline to test whether DelTA benefits merely from perturbing token weights. In this baseline, $\lambda_{i,t}$ is sampled randomly from the same bounded range $[\lambda_{\min}, \lambda_{\max}]$, and then used with the same coefficient normalization and weighted DAPO objective as DelTA. All other training settings are kept unchanged.

Table 4 shows that DelTA is robust to the choice of last-layer proxy. The top-K hidden-gradient proxy achieves the best average performance, improving over the default proxy from 23.27 to 24.29. This suggests that the contrast-sensitive reweighting mechanism does not rely on a single proxy choice, and that incorporating local competition among top vocabulary candidates can provide a stronger signal for coefficient estimation.

In contrast, the random-λ baseline substantially underperforms both DelTA variants, dropping to an average score of 18.34. This confirms that the benefit of DelTA is not due to arbitrary token reweighting or stochastic perturbation of the loss. Rather, the coefficients need to preserve meaningful discriminative structure: tokens should receive larger weights when their gradients are more representative of their own advantage side than of the opposite side.

Therefore, the gains reported in our main experiments are obtained with a conservative and computationally minimal proxy. The stronger performance of the top-K hidden-gradient variant further suggests that DelTA is not tied to this specific proxy choice and may benefit from more informative gradient proxies. We keep the output-row proxy as the default in the main experiments because it is the minimal and most direct layer-restricted instantiation of the token-gradient view; the stronger top-K proxy shows that DelTA can further benefit from richer proxy choices.

Appendix J.8 further evaluates the agreement between proxy-based token coefficients and those computed from full-parameter gradients, and examines how proxy choice interacts with refinement depth.

E Derivation of the DelTA Soft Assignment Score

In this appendix, we derive the closed-form solution of the soft assignment score used in DelTA. We follow the notation in Section 3.2. The centroids μ_+ and μ_- are fixed in this derivation, and $\alpha_{i,t} \in [0, 1]$ denotes the raw assignment score before the final remapping and normalization step.

Side-specific squared-distance margin. For a token from a positive-advantage response, DelTA assigns a larger score when its token-gradient vector is closer to the positive centroid than to the negative centroid. We define the positive-side squared-distance margin as

$$\Delta^+_{i,t} := \|v_{i,t} - \mu_-\|_2^2 - \|v_{i,t} - \mu_+\|_2^2, \qquad \hat{A}_i > 0.$$

Thus, $\Delta^+_{i,t} > 0$ means that $v_{i,t}$ is closer to μ_+ than to μ_-. For a token from a negative-advantage response, the two centroids are swapped, and we define

$$\Delta^-_{i,t} := \|v_{i,t} - \mu_+\|_2^2 - \|v_{i,t} - \mu_-\|_2^2, \qquad \hat{A}_i < 0.$$

Hence, $\Delta^-_{i,t} > 0$ means that $v_{i,t}$ is closer to μ_- than to μ_+.

Positive-side assignment. For a token from a positive-advantage response, Eq. (5) can be written compactly as

$$\alpha_{i,t} = \arg\max_{\alpha \in [0,1]} \alpha\Delta^+_{i,t} + \gamma_+ h(\alpha),$$

where

$$h(\alpha) = -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha)$$

is the binary entropy, with the standard convention $0 \log 0 = 0$. Since the centroids and temperature are fixed, this is a one-dimensional optimization problem. To simplify notation, write $\Delta = \Delta^+_{i,t}$ and $\gamma = \gamma_+$. We solve

$$\max_{\alpha \in [0,1]} f(\alpha) := \alpha\Delta + \gamma h(\alpha).$$

For $\alpha \in (0, 1)$, the derivative is

$$f'(\alpha) = \Delta + \gamma h'(\alpha) = \Delta + \gamma \log \frac{1 - \alpha}{\alpha}.$$

Setting $f'(\alpha) = 0$ gives

$$\log \frac{1 - \alpha}{\alpha} = -\frac{\Delta}{\gamma},$$

or equivalently

$$\frac{\alpha}{1 - \alpha} = \exp\left(\frac{\Delta}{\gamma}\right).$$

Solving for α yields

$$\alpha = \frac{1}{1 + \exp(-\Delta/\gamma)} = \sigma\left(\frac{\Delta}{\gamma}\right),$$

where $\sigma(z) = 1/(1 + \exp(-z))$.

The solution is the unique maximizer. Indeed,

$$f''(\alpha) = \gamma h''(\alpha) = -\gamma \left(\frac{1}{\alpha} + \frac{1}{1 - \alpha}\right) = -\frac{\gamma}{\alpha(1 - \alpha)} < 0$$

for all $\alpha \in (0, 1)$, since $\gamma > 0$. Therefore, f is strictly concave on $(0, 1)$, and the stationary point above is the unique optimum. Substituting back $\Delta = \Delta^+_{i,t}$ and $\gamma = \gamma_+$, we obtain

$$\alpha_{i,t} = \sigma\left(\frac{\|v_{i,t} - \mu_-\|_2^2 - \|v_{i,t} - \mu_+\|_2^2}{\gamma_+}\right), \qquad \hat{A}_i > 0.$$

Negative-side assignment. For tokens from negative-advantage responses, DelTA uses the symmetric objective obtained by swapping the positive and negative centroids:

$$\alpha_{i,t} = \arg\max_{\alpha\in[0,1]} \alpha\Delta^{-}_{i,t} + \gamma_{-}h(\alpha), \qquad \hat{A}_i < 0.$$

Repeating the same one-dimensional derivation with $\Delta = \Delta^{-}_{i,t}$ and $\gamma = \gamma_{-}$ gives

$$\alpha_{i,t} = \sigma\left(\frac{\|v_{i,t} - \mu_{+}\|_2^2 - \|v_{i,t} - \mu_{-}\|_2^2}{\gamma_{-}}\right), \qquad \hat{A}_i < 0.$$

Combining the two sides, the DelTA soft assignment score is

$$\alpha_{i,t} = \begin{cases} \sigma\left(\dfrac{\|v_{i,t} - \mu_{-}\|_2^2 - \|v_{i,t} - \mu_{+}\|_2^2}{\gamma_{+}}\right), & \hat{A}_i > 0, \\ \sigma\left(\dfrac{\|v_{i,t} - \mu_{+}\|_2^2 - \|v_{i,t} - \mu_{-}\|_2^2}{\gamma_{-}}\right), & \hat{A}_i < 0, \end{cases}$$

which matches Eq. (6).

The score $\alpha_{i,t}$ is a raw stop-gradient assignment score. As described in Appendix F, DelTA subsequently remaps these scores to a bounded coefficient range and normalizes them before using them as loss coefficients in the weighted DAPO surrogate.

Connection to the inner-product discriminator. For fixed centroids μ_{+} and μ_{-}, the positive-side squared-distance margin used by DelTA can be rewritten as

$$\|v - \mu_{-}\|_2^2 - \|v - \mu_{+}\|_2^2 = 2v^{\top}(\mu_{+} - \mu_{-}) + \|\mu_{-}\|_2^2 - \|\mu_{+}\|_2^2.$$

Thus, up to a centroid-dependent offset, the distance margin scores v by its alignment with the centroid contrast direction $\mu_{+} - \mu_{-}$. The negative-side margin is obtained symmetrically by swapping μ_{+} and μ_{-}. This shows that the squared-distance comparison used for coefficient estimation is consistent with the positive-negative inner-product discriminator view in Section 3.1, while matching the weighted least-squares geometry of the side-wise centroids. In the original local discriminator, alignment with each centroid is weighted by its total advantage mass, as in Eq. (4); the main-text alignment intuition should be understood in this mass-weighted sense. DelTA's distance margin uses the side-wise centroid contrast to estimate token coefficients, while the resulting policy update remains advantage- and coefficient-weighted.

F DelTA Implementation Details

In this appendix, we describe how DelTA is computed in practice on each rollout batch. The raw assignment scores depend on the side-wise centroids, while the centroids are themselves estimated using these scores. We therefore use a small fixed number of lagged alternating refinement steps rather than solving the coupled problem to convergence. For each prompt, the group-normalized advantages are computed from the responses sampled for that prompt. DelTA does not estimate separate centroids for each prompt; instead, the side-wise centroids, temperatures, and token coefficients are computed jointly over responses from multiple prompts.

Let $\{v_{i,t}\}$ denote the token-gradient vectors in the rollout batch, and let $\{\hat{A}_i\}$ denote the corresponding sequence-level advantages. Throughout this appendix, $i \in \{1, \ldots, G\}$ indexes sampled responses and $t \in \{1, \ldots, |o_i|\}$ indexes valid response tokens. We use $\alpha_{i,t} \in [0, 1]$ for raw assignment scores, and reserve $\lambda_{i,t}$ for the final training coefficient used in the weighted DAPO surrogate. All distances below are standard squared Euclidean distances.

Initialization of side-wise centroids. We initialize the positive and negative centroids from the original advantage-weighted token-gradient means. Define

$$M_{+} = \sum_{i:\hat{A}_i>0} \sum_{t=1}^{|o_i|} \hat{A}_i, \qquad M_{-} = \sum_{i:\hat{A}_i<0} \sum_{t=1}^{|o_i|} |\hat{A}_i|.$$

The initial centroids are

$$\mu_+^{(0)} = \frac{\sum_{i:\hat{A}_i>0}\sum_{t=1}^{|o_i|}\hat{A}_i\, v_{i,t}}{M_+}, \qquad \mu_-^{(0)} = \frac{\sum_{i:\hat{A}_i<0}\sum_{t=1}^{|o_i|}|\hat{A}_i|\, v_{i,t}}{M_-}.$$

In implementation, the denominators are clamped by a small numerical constant $\varepsilon > 0$ for stability.

Squared-distance differences and lagged adaptive temperatures. For centroids $\mu_+^{(k)}$ and $\mu_-^{(k)}$, define the positive-side and negative-side squared-distance differences as

$$\Delta_{i,t}^{+,(k)} = \|v_{i,t} - \mu_-^{(k)}\|_2^2 - \|v_{i,t} - \mu_+^{(k)}\|_2^2, \qquad \hat{A}_i > 0,$$

and

$$\Delta_{i,t}^{-,(k)} = \|v_{i,t} - \mu_+^{(k)}\|_2^2 - \|v_{i,t} - \mu_-^{(k)}\|_2^2, \qquad \hat{A}_i < 0.$$

A larger value means that the token-gradient vector is closer to its own-side centroid than to the opposite-side centroid.

The side-specific temperatures are set from the empirical standard deviations of these squared-distance differences. For the initial centroids, we compute

$$\gamma_+^{(0)} = \sqrt{\max\Big(\mathrm{Var}\Big(\{\Delta_{i,t}^{+,(0)} : \hat{A}_i > 0,\ 1 \le t \le |o_i|\}\Big), \varepsilon_\gamma\Big)},$$

$$\gamma_-^{(0)} = \sqrt{\max\Big(\mathrm{Var}\Big(\{\Delta_{i,t}^{-,(0)} : \hat{A}_i < 0,\ 1 \le t \le |o_i|\}\Big), \varepsilon_\gamma\Big)},$$

where $\varepsilon_\gamma > 0$ is a small numerical constant.

The superscript of $\gamma_\pm^{(k)}$ denotes the cached temperature used when computing $\alpha_{i,t}^{(k)}$. In implementation, the temperatures are updated in a lagged manner. During the pass that computes $\alpha_{i,t}^{(k)}$, we also accumulate the empirical variances of $\Delta_{i,t}^{+,(k)}$ and $\Delta_{i,t}^{-,(k)}$, and store the resulting temperatures as $\gamma_+^{(k+1)}$ and $\gamma_-^{(k+1)}$ for the next score-computation step. Thus, for $k > 0$, $\gamma_\pm^{(k)}$ is the cached temperature produced by the previous margin-statistics pass. This avoids an additional proxy forward pass solely for temperature estimation.

Alternating refinement. Starting from $\mu_+^{(0)}$, $\mu_-^{(0)}$, $\gamma_+^{(0)}$, and $\gamma_-^{(0)}$, DelTA runs K stop-gradient refinement iterations. At iteration $k = 0, \ldots, K-1$, the raw assignment scores are

$$\alpha_{i,t}^{(k)} = \begin{cases} \sigma\Big(\Delta_{i,t}^{+,(k)}/\gamma_+^{(k)}\Big), & \hat{A}_i > 0, \\ \sigma\Big(\Delta_{i,t}^{-,(k)}/\gamma_-^{(k)}\Big), & \hat{A}_i < 0, \end{cases}$$

where $\sigma(\cdot)$ is the sigmoid function. Equivalently, this is the closed-form solution of the entropy-regularized assignment objective in Section 3.2.

The centroids are then updated as score-weighted within-side averages:

$$\mu_+^{(k+1)} = \frac{\sum_{i:\hat{A}_i>0}\sum_{t=1}^{|o_i|}\hat{A}_i\,\alpha_{i,t}^{(k)} v_{i,t}}{\sum_{i:\hat{A}_i>0}\sum_{t=1}^{|o_i|}\hat{A}_i\,\alpha_{i,t}^{(k)}},$$

$$\mu_-^{(k+1)} = \frac{\sum_{i:\hat{A}_i<0}\sum_{t=1}^{|o_i|}|\hat{A}_i|\,\alpha_{i,t}^{(k)} v_{i,t}}{\sum_{i:\hat{A}_i<0}\sum_{t=1}^{|o_i|}|\hat{A}_i|\,\alpha_{i,t}^{(k)}}.$$

Again, denominators are clamped by a small numerical constant for stability. The same pass also accumulates the statistics of $\Delta_{i,t}^{+,(k)}$ and $\Delta_{i,t}^{-,(k)}$, producing $\gamma_+^{(k+1)}$ and $\gamma_-^{(k+1)}$ for the next refinement iteration.

Final coefficient computation. After K refinement iterations, we obtain refined centroids $\mu_+^{(K)}$ and $\mu_-^{(K)}$. A final pass over the rollout batch recomputes the squared-distance differences using these refined centroids:

$$\Delta_{i,t}^{+,(\star)} = \|v_{i,t} - \mu_-^{(K)}\|_2^2 - \|v_{i,t} - \mu_+^{(K)}\|_2^2, \qquad \hat{A}_i > 0,$$

$$\Delta_{i,t}^{-,(\star)} = \|v_{i,t} - \mu_+^{(K)}\|_2^2 - \|v_{i,t} - \mu_-^{(K)}\|_2^2, \qquad \hat{A}_i < 0.$$

The final raw scores are computed using the latest cached temperatures:

$$\alpha_{i,t}^{\star} = \begin{cases} \sigma\left(\Delta_{i,t}^{+,(\star)} / \gamma_+^{(K)}\right), & \hat{A}_i > 0, \\ \sigma\left(\Delta_{i,t}^{-,(\star)} / \gamma_-^{(K)}\right), & \hat{A}_i < 0. \end{cases}$$

When $K = 0$, this reduces to using the initial centroids $\mu_+^{(0)}$, $\mu_-^{(0)}$ and initial temperatures $\gamma_+^{(0)}$, $\gamma_-^{(0)}$. The raw scores are then mapped to bounded loss coefficients:

$$\lambda_{i,t} = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\alpha_{i,t}^{\star}, \qquad \hat{A}_i \neq 0.$$

For zero-advantage responses, which do not belong to either side, we assign

$$\lambda_{i,t} = \lambda_{\min}.$$

Self-normalized implementation with the standard DAPO token average. The self-normalized DelTA objective in Eq. (8) can be implemented while keeping the standard DAPO token-count normalizer. Let

$$N = \sum_{i=1}^{G} |o_i|, \qquad Z = \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \lambda_{i,t}.$$

We define the implementation-facing coefficient

$$\bar{\lambda}_{i,t} = \lambda_{i,t} \frac{N}{Z}.$$

By construction,

$$\frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \bar{\lambda}_{i,t} = 1.$$

Therefore, for the clipped DAPO token loss

$$\ell_{i,t}(\theta) = \min\Big(r_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}\big(r_{i,t}(\theta), 1 - \epsilon_{\mathrm{low}}, 1 + \epsilon_{\mathrm{high}}\big)\hat{A}_i \Big),$$

we have

$$\frac{1}{N} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \bar{\lambda}_{i,t} \ell_{i,t}(\theta) = \frac{1}{Z} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \lambda_{i,t} \ell_{i,t}(\theta).$$

Thus, using $\bar{\lambda}_{i,t}$ under the standard DAPO token average is exactly equivalent to the self-normalized DelTA objective. The normalization changes only the global scale of the coefficients and preserves their relative token reweighting.

Use in training. The coefficients $\lambda_{i,t}$ and $\bar{\lambda}_{i,t}$ are treated as stop-gradient quantities. They are computed once from the rollout batch, held fixed across repeated optimization passes over that batch, and recomputed only when new trajectories are sampled. In the actual loss computation, $\bar{\lambda}_{i,t}$ is used as the per-token multiplier, and the rest of the DAPO training pipeline remains unchanged.

Summary. For each rollout batch, DelTA proceeds as follows:

1. Compute layer-restricted token-gradient proxies $v_{i,t}$ for all valid rollout tokens.
2. Initialize $\mu_+^{(0)}$ and $\mu_-^{(0)}$ from the original advantage-weighted side-wise centroids.
3. Compute initial temperatures $\gamma_+^{(0)}$ and $\gamma_-^{(0)}$ from the squared-distance differences under the initial centroids.
4. For $k = 0, \ldots, K-1$:
   (a) compute raw assignments $\alpha_{i,t}^{(k)}$ using $\mu_+^{(k)}$, $\mu_-^{(k)}$, and the cached temperatures $\gamma_+^{(k)}$, $\gamma_-^{(k)}$;
   (b) update $\mu_+^{(k+1)}$ and $\mu_-^{(k+1)}$ as score-weighted within-side centroids;
   (c) store $\gamma_+^{(k+1)}$ and $\gamma_-^{(k+1)}$ from the same squared-distance-difference statistics for the next assignment computation.
5. Compute final raw assignments $\alpha_{i,t}^{\star}$ using the refined centroids $\mu_+^{(K)}$, $\mu_-^{(K)}$, and latest cached temperatures $\gamma_+^{(K)}$, $\gamma_-^{(K)}$.
6. Map $\alpha_{i,t}^{\star}$ to bounded coefficients $\lambda_{i,t}$, normalize them into $\bar{\lambda}_{i,t}$, and use $\bar{\lambda}_{i,t}$ inside the standard DAPO token average.

G Detailed Settings

The training settings are presented in Table 5. We train the Qwen3-8B-Base model for 220 steps and the Qwen3-14B-Base model for 300 steps. For checkpoint selection, we evaluate checkpoints during training and select the checkpoint with the highest AIME25 avg@8 score. This selection rule is fixed in advance and applied uniformly to all methods and backbone sizes. Thus, our main comparison reflects the best checkpoint performance achieved by each method under the same training budget and the same model-selection protocol.

We report the average performance over all samples. Unless otherwise specified, Avg. denotes the question-count weighted average across benchmarks.

Appendix J.13 examines the robustness of this comparison by extending the training horizon and using an alternative checkpoint-selection benchmark.

All existing assets used in this work are publicly available research assets. We cite their original sources and follow their corresponding licenses and terms of use.

All experiments were conducted on $8\times$ NVIDIA B200 GPUs. For the main DelTA training experiments, the wall-clock time per training step was approximately 6 minutes at the beginning of Qwen3-8B-Base training and increased to approximately 10 minutes as response lengths grew. For Qwen3-14B-Base, the wall-clock time per training step was approximately 8 minutes at the beginning of training and increased to approximately 18 minutes as response lengths grew.

We use a simple binary verifiable reward: a response receives reward 1 if its final answer is correct, and 0 otherwise. Answer correctness is determined by `math-verify`[1]. Evaluation hyperparameters are presented in Table 6.

H Baseline Details

In this section, we briefly introduce the baseline methods used in our experiments.

DAPO. DAPO (Yu et al., 2025) extends GRPO by using asymmetric clipping, and token-level loss normalization to improve long-CoT RL training. We use DAPO as our main strong baseline.

[1] https://github.com/huggingface/Math-Verify

Table 5: Training settings.

Hyper-parameter	Value
Train Batch Size	128
Micro Batch Size	32
Rollout n	16
Maximum Prompt Length	2048
Maximum Response Length	20,480
Clip Ratio Low	0.2
Clip Ratio High	0.28
Rollout Engine	SGLang
Temperature	1.0
Top-p	1.0
LR	1×10^{-6}
KL Coefficient	0.0

Table 6: Evaluation hyper-parameters.

Hyper-parameter	Value
Max Length	30,000
Temperature	1.0
Top-p	0.7
Inference Engine	SGLang

Table 7: Unique training hyper-parameters of SAPO.

Hyper-parameter	Value
Gae Gamma	1.0
Gae Lam	0.95
Tau Pos	1.0
Tau Neg	1.05

Table 8: Unique training hyper-parameters of FIPO.

Hyper-parameter	Value
Decay Rate	32.0
Chunk Size	128
Future KL Start	include current
Future KL Window	-1
Future KL Average	False
Future KL Clip Ratio	0.2
Future KL Clip High Only	True
Safety Thresh	10.0

DAPO w/ Forking Tokens. DAPO w/ Forking Tokens (Wang et al., 2025) (DAPO w/ FT for short) is a token-filtered variant of DAPO that keeps only the top 20% high-entropy tokens, referred to as forking tokens, in the policy-gradient loss. The remaining low-entropy tokens are masked out, based on the observation that high-entropy tokens often correspond to critical reasoning forks.

SAPO. SAPO (Gao et al., 2025) replaces hard clipping with a smooth, temperature-controlled soft gate to attenuate off-policy updates. It further uses different temperatures for positive- and negative-advantage tokens to improve training stability.

FIPO. FIPO (Ma et al., 2026) reweights token-level advantages using discounted Future-KL, which estimates how much each token influences the subsequent trajectory. The resulting influence weights are clipped to control variance during policy optimization.

The method-specific hyperparameters of SAPO and FIPO are provided in Table 7 and Table 8, respectively.

I Significance Test Details

We describe the statistical significance tests used in our main comparison. Due to the high cost of RLVR training, we do not repeat full training runs with multiple random seeds. Instead, we perform an evaluation-run-level significance test that captures stochasticity from repeated generation.

For each method, we perform $S = 16$ independent evaluation runs over the full evaluation suite. In each run, we sample one response per problem and compute one question-count-weighted aggregate score. Thus, the 16 evaluation runs correspond to the 16 independently sampled responses per problem used in the main evaluation. Since different methods are evaluated with independently

sampled runs, we treat the 16 aggregate scores from DelTA and the 16 aggregate scores from the baseline as two independent samples.

We use a one-sided Mann–Whitney U test as the primary non-parametric test, with the pre-specified alternative hypothesis that DelTA outperforms the baseline. We regard the improvement as statistically significant when the one-sided p-value is below 0.05.

For the main significance claim, we compare DelTA with the strongest same-scale baseline for each backbone size, namely SAPO for the 8B backbone and FIPO for the 14B backbone. Under this evaluation-run-level testing protocol, DelTA significantly outperforms the strongest same-scale baseline at both model scales.

J SUPPLEMENTARY EXPERIMENTS

Overview. This appendix complements the main experiments with computational analysis, broader evaluations, and additional diagnostics. Appendices J.1 to J.6 report computational overhead, generalization results, hyperparameter sensitivity, and token-weight analysis. Appendices J.7 to J.9 examine the local discriminator, gradient-proxy fidelity, refinement depth, and centroid dynamics. Appendices J.10 to J.11 provide additional baselines and controlled comparisons, including trajectory-level reweighting and the extension to GRPO. Appendices J.12 to J.13 evaluate performance under approximately matched wall-clock budgets and assess robustness to training horizon and checkpoint selection.

J.1 COMPUTATIONAL OVERHEAD

In this subsection, we discuss the computational overhead introduced by DelTA.

The DelTA proxy is computed from final-layer hidden states, which are already produced when evaluating old log-probabilities. Therefore, if all hidden states for the rollout batch could be cached, the centroid refinement itself would not require additional actor forward passes: the same fixed token-gradient proxies could be reused across refinement iterations.

In practice, caching all hidden states for long-response RLVR rollouts is memory intensive. We therefore recompute the proxy whenever it is needed. Relative to the standard old-log-probability computation, the additional actor forward passes for DelTA with K refinement iterations are:

1. one pass to estimate the initial temperature scale γ;
2. one pass for each refinement iteration $k = 1, \ldots, K$, which computes token scores and updates the centroids;
3. one final pass to compute the final token coefficients used in the weighted DAPO objective.

Thus, DelTA requires $K + 2$ additional actor forward passes for coefficient estimation. In our main experiments, we use $K = 1$, which already gives strong performance. These additional passes are only used to compute stop-gradient coefficients; the resulting weighted DAPO objective is optimized in the same way as the baseline objective.

To empirically measure the overhead, we compare the execution time of the first training step between DelTA and DAPO. We focus on the first step because response lengths can diverge during training, as shown in Figure 2. Since rollout generation dominates wall-clock time in long-response RLVR, later-step end-to-end comparisons would be confounded by differences in generated response length rather than isolating the overhead of DelTA's coefficient computation.

On 8 NVIDIA B200 GPUs, the first step of DelTA takes 37 seconds longer than DAPO. Since rollout generation dominates long-response RLVR, this corresponds to approximately 10.2% of the total first-step time of DelTA. These results indicate that, when isolating the optimization phase from length-induced rollout variation, the overhead of DelTA's centroid refinement is modest for RLVR workloads.

Appendix J.12 complements this overhead analysis with performance comparisons under approximately matched end-to-end wall-clock budgets, accounting for both rollout generation and policy optimization.

Table 9: Main results on seven mathematical reasoning benchmarks for Olmo3-7B-Base. The best results are in **bold**.

Method	AIME24	AIME25	AIME26	HMMT25 (Feb.)	HMMT25 (Nov.)	HMMT26 (Feb.)	Brumo25	Avg.
Olmo3-7B-Base								
DAPO	25.62	21.88	22.08	14.58	8.75	14.02	26.67	19.01
DelTA	**30.83**	**24.79**	**27.08**	**16.67**	**13.96**	**16.29**	**30.63**	**22.80**

J.2 Other Model Architectures

To further examine the generality of DelTA beyond the Qwen3 backbones, we also conduct experiments on Olmo3-7B-Base. We train DelTA and the DAPO baseline with the same hyperparameter settings as in the main experiments, without architecture-specific tuning. This setting allows us to test whether the proposed token-level credit assignment remains effective on a different model family.

As shown in Table 9, DelTA consistently outperforms DAPO on all seven mathematical reasoning benchmarks. The average score improves from 19.01 to 22.80, corresponding to a gain of 3.79 points. The improvement is particularly large on AIME24, AIME26, HMMT25-Nov, and Brumo25, while DelTA also maintains clear gains on the remaining benchmarks. These results suggest that DelTA is not tied to a specific Qwen3 backbone, and that discriminative token-level reweighting can provide consistent benefits across different base model families.

J.3 Code Generation

To further validate the effectiveness of DelTA beyond mathematical reasoning, we conduct experiments on code generation tasks. We use Eurus2-RL-Code (Cui et al., 2025) as the training dataset and adopt DAPO as the baseline. All models are trained for two epochs under the same training recipe. We evaluate the trained models on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench (Jain et al., 2024). For each problem, we sample 5 rollouts and report the average accuracy. We also report a weighted average across benchmarks, where each benchmark is weighted by its number of evaluation problems.

Table 10: Code generation results. DelTA consistently improves performance across all benchmarks.

Method	HumanEval+	MBPP+	LCB	Avg.
DAPO	83.0	72.1	33.4	47.7
DelTA	**84.6**	**73.2**	**35.6**	**49.5**

As shown in Table 10, DelTA consistently improves over DAPO on all three code generation benchmarks, increasing the weighted average score from 47.7 to 49.5. These results suggest that the benefit of DelTA is not limited to mathematical reasoning, but also transfers to code generation tasks, where effective token-level credit assignment remains important under sequence-level supervision.

J.4 Q4: Is DelTA sensitive to its hyperparameters?

We further study the sensitivity of DelTA to its method-specific hyperparameters. We focus on two factors: the coefficient range $[\lambda_{\min}, \lambda_{\max}]$, which controls the strength of token reweighting, and the number of centroid refinement iterations K, which controls how many times the side-wise centroids are updated before computing the final coefficients. For the coefficient range, we evaluate several configurations, including $[0.5, 1.2]$, $[0.8, 1.5]$, and $[0.5, 1.5]$. For the refinement step, we vary K over different values, including $K = 2$ and $K = 3$.

As shown in Table 11, DelTA is relatively robust to the coefficient range across the tested settings. All experiments in this table use the default LM-head proxy. Under this proxy, increasing the refinement depth from $K = 1$ to $K = 2$ or $K = 3$ reduces the average score from 23.27 to 22.15 and 22.20, respectively. We therefore retain $[\lambda_{\min}, \lambda_{\max}] = [0.8, 1.2]$ with $K = 1$ as the lightweight default. However, this trend does not hold for all proxies: with the richer hidden-gradient proxy, increasing

Table 11: Hyperparameter sensitivity study of DelTA. The best results are in **bold**, and the second-best results are underlined.

Method	AIME25	AIME26	HMMT25	HMMT26	Avg.
Base DelTA	**26.46**	28.12	18.54	**20.27**	**23.27**
$\lambda_{\min} = 0.5$	25.42	**28.96**	**19.17**	19.70	23.22
$\lambda_{\max} = 1.5$	26.46	26.25	18.75	19.89	22.77
$\lambda_{\min} = 0.5, \lambda_{\max} = 1.5$	25.62	27.92	18.96	20.08	23.07
$K = 2$	25.00	27.08	18.33	18.56	22.15
$K = 3$	26.04	26.67	18.12	18.37	22.20

the depth from $K = 1$ to $K = 2$ improves the average score from 24.29 to 26.17 as shown in Appendix J.8. These results suggest that the benefit of deeper refinement depends on the proxy used for coefficient estimation.

J.5 Q5: Does DelTA generalize to out-of-domain evaluation?

To examine whether DelTA generalizes beyond the in-domain mathematical reasoning benchmarks, we further evaluate DAPO and DelTA on two out-of-domain benchmarks: GPQA-Diamond and MMLU-Pro. Since MMLU-Pro contains a large number of questions, we randomly sample 600 questions for evaluation. For each question, we sample 5 responses and report Avg@5. We also report a question-count weighted average across the two benchmarks.

Table 12: OOD evaluation on GPQA-D and MMLU-Pro. The best results are in **bold**.

Method	GPQA-D	MMLU-Pro	Avg.
Qwen3-8B-Base			
DAPO	43.43	63.97	58.87
DelTA	**50.00**	**66.47**	**62.38**
Qwen3-14B-Base			
DAPO	54.55	70.80	66.77
DelTA	**56.67**	**72.27**	**68.40**

As shown in Table 12, DelTA consistently improves over DAPO on both OOD benchmarks and both backbone sizes. On Qwen3-8B-Base, DelTA improves the weighted average from 58.87 to 62.38. On Qwen3-14B-Base, DelTA further improves the weighted average from 66.77 to 68.40. These results suggest that DelTA does not merely overfit to the main mathematical evaluation suite, but also transfers to broader scientific and general knowledge reasoning tasks.

J.6 Token weight analysis

To understand what DelTA emphasizes, we visualize high- and low-weight token clouds in Figure 5. The visualization is based on selected generated tokens collected during training at a generation scale of roughly 10^8 tokens. For each token type, we compute its average DelTA coefficient over its occurrences; larger tokens indicate higher average coefficients in the high-weight cloud and lower average coefficients in the low-weight cloud. This context-agnostic visualization is qualitative, since DelTA assigns coefficients to tokens based on their gradient vectors in specific rollout contexts rather than to token strings in isolation. The high-weight cloud contains several tokens related to transformations and verification, such as `scaffold`, `prime`, `=y`, `forward`, and `backward`. In contrast, the low-weight cloud is dominated by more background-like or entity-specific tokens, such as `Seat`, `domain`, `players`, `Vander`, and `Hamilton`.



(a) High-weight tokens



(b) Low-weight tokens

Figure 5: Token clouds of high-weight and low-weight tokens.

This pattern is consistent with DelTA's design. Rather than reflecting semantic importance alone, $\lambda_{i,t}$ reflects whether the gradient vector of a token is more representative of its own advantage side than of the opposite side. DelTA

therefore tends to assign larger coefficients to more discriminative reasoning-related tokens and smaller coefficients to shared or less informative background tokens.

J.7 Predictive Validity of the Local Discriminator

Unless otherwise specified, the additional experiments below use Qwen3-8B-Base and follow the four-benchmark analysis protocol in Section 5: AIME25, AIME26, HMMT25-Nov, and HMMT26-Feb. We abbreviate the latter two benchmarks as HMMT25 and HMMT26, respectively. Avg. denotes the question-count-weighted average across these benchmarks. Training settings and checkpoint selection follow Appendix G, except for the explicitly varied factors.

Our discriminator view characterizes a local, first-order update around θ_{old}, where importance ratios equal one and clipping is locally inactive. Practical RLVR training additionally involves repeated optimization, optimizer preconditioning, and gradient clipping. We therefore examine whether the local discriminator retains predictive power after these transformations.

Before optimization, we compute the full-parameter discriminator score for each sampled token. Up to a shared positive normalizer, this score is

$$s_{i,t} = v_{i,t}^{\top} \left(\sum_{j} \sum_{u=1}^{|o_j|} \hat{A}_j v_{j,u} \right), \tag{16}$$

where all token-gradient vectors are evaluated at the policy parameters before optimization. We then run one standard DAPO optimization cycle over the rollout batch, including multiple optimization steps with the clipped objective and the standard training pipeline, and measure the actual log-probability change of each sampled token under its original context.

Since the signs of near-zero changes can be sensitive to small optimization perturbations, we evaluate the top 20% of tokens ranked by the absolute magnitude of their *actual* log-probability changes. Within this subset, 52.0% of tokens have positive changes. The discriminator achieves 65.8% sign accuracy and 67.3% AUROC, compared with a 52.0% majority-class sign baseline and a 50.0% random-ranking AUROC baseline.

These results indicate that the local discriminator retains nontrivial predictive information after practical optimization, particularly for tokens that undergo substantive probability changes. This diagnostic is conditional on the selected high-change subset; it does not establish the same accuracy over all tokens or an exact characterization of the full nonlinear training trajectory.

J.8 Gradient-Proxy Fidelity and Refinement Depth

What fidelity is required from the proxy? The full-parameter token gradients in Section 3.2 define the exact geometric objects underlying DelTA, but the practical proxy is not used as an approximate policy gradient. It is used only to compute stop-gradient token coefficients; once these coefficients are fixed, the weighted RLVR objective is still differentiated with respect to all policy parameters. The relevant approximation target is therefore not exact reconstruction of full-parameter gradient vectors, but preservation of the discriminative comparisons that determine the relative token weights.

This distinction is particularly important for DelTA because the raw assignment scores are mapped to a bounded coefficient range, $[\lambda_{\min}, \lambda_{\max}] = [0.8, 1.2]$, followed by self-normalization. Consequently, moderate perturbations to the fine-grained ordering of nearby tokens cannot produce arbitrarily large changes in their loss weights. What matters more directly is whether the proxy preserves the separation between relatively high- and low-weight tokens and, more generally, the ranking structure used to construct the weighted centroids.

We therefore compare the default LM-head proxy, the top-K hidden-gradient proxy, gradients restricted to the last 25% and 50% of model layers, and full-parameter gradients. Using coefficients computed from full-parameter gradients as the reference, we report two complementary metrics: Spearman rank correlation, which measures agreement over the complete coefficient ranking, and top-50% overlap, which measures preservation of the high-weight token set. We use the 50% split consistently with the token-selection diagnostic in Section 5.2. Since both selected sets contain half of the tokens, the expected overlap under independent random selection is 50%.

Table 13: Agreement between proxy-derived DelTA coefficients and coefficients computed from full-parameter gradients. Spearman correlation evaluates the full ranking, whereas top-50% overlap evaluates preservation of the high-weight token set used in our token-selection analysis.

Gradient representation	Spearman	Top-50% overlap
Default LM-head proxy	0.470	0.888
Top-K hidden-gradient proxy	0.661	0.922
Last 25% of layers	0.885	0.970
Last 50% of layers	0.900	0.974
Full-parameter gradients	1.000	1.000

Table 14: Refinement depth with the top-K hidden-gradient proxy. Both rows use the same gradient representation and differ only in the number of centroid-refinement iterations.

Refinement depth	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.
$K = 1$	27.08	27.71	20.83	21.78	24.29
$K = 2$	**28.33**	**31.04**	**22.29**	**23.30**	**26.17**

Table 13 reveals a useful distinction between coarse and fine-grained fidelity. Although the default proxy has a moderate full-ranking correlation of 0.470, it preserves 88.8% of the high-weight tokens selected by the full-gradient coefficients. Thus, much of the ranking disagreement occurs within the same high- or low-weight region rather than across the high/low partition. Richer representations improve both notions of fidelity: the top-K hidden-gradient proxy reaches 0.661 Spearman correlation and 92.2% overlap, while gradients from the last half of the model reach 0.900 and 97.4%, respectively.

Importantly, this increase in coefficient fidelity is also consistent with the downstream proxy ablation in Appendix D. Replacing the default proxy with the more informative top-K hidden-gradient proxy improves the four-benchmark average from 23.27 to 24.29, whereas random coefficients with the same bounded range and normalization obtain only 18.34. Thus, the effectiveness of DelTA does not arise from arbitrary perturbation of token weights, and a proxy that more faithfully preserves the full-gradient coefficient structure can provide a stronger training signal. At the same time, computing layer-restricted or full-parameter gradients for individual rollout tokens is substantially more expensive, motivating the lightweight default for long-response RLVR training.

Interaction between proxy fidelity and refinement depth. Appendix J.4 studies refinement depth using the default LM-head proxy. Under this lightweight representation, increasing the number of refinement iterations from $K = 1$ to $K = 2$ or $K = 3$ changes the average score from 23.27 to 22.15 and 22.20, respectively. This result raises the question of whether deeper refinement is intrinsically unfavorable, or whether its usefulness depends on the fidelity of the gradient representation.

We therefore repeat the comparison using the more informative top-K hidden-gradient proxy from Appendix D. Here, K denotes the number of centroid-refinement iterations, rather than the vocabulary truncation parameter used by the proxy.

With the richer proxy, a second refinement iteration improves all four benchmarks and increases the average score from 24.29 to 26.17. Deeper refinement is therefore not intrinsically detrimental. Instead, the benefit of additional centroid updates depends on the gradient representation used to estimate the coefficients. The results are consistent with a fidelity–computation trade-off: the lightweight proxy already provides useful coarse token discrimination with one refinement step, whereas a richer representation supports more accurate coefficient estimation and can benefit from further refinement.

Accordingly, we use the default LM-head proxy with $K = 1$ throughout the main experiments as the lightweight configuration. The richer proxy with $K = 2$ is reported here as a diagnostic rather than as a replacement for the main setting, because it increases both proxy-computation cost and refinement

Table 15: Cosine similarity between positive- and negative-advantage centroids at different training stages.

Method	Step 0	Step 100	Step 200
DAPO	0.9880	0.9942	0.9965
DelTA, before refinement	0.9880	0.9938	0.9730
DelTA, after refinement	0.9856	0.9934	0.9579

cost. These results also show that DelTA is not tied to a particular proxy: more informative gradient representations can improve both coefficient fidelity and downstream performance when additional computation is available.

J.9 Centroid Dynamics and Relation to Entropy-Based Token Selection

Centroid contrast across training stages. We measure the cosine similarity between positive- and negative-advantage centroids at Steps 0, 100, and 200. For DelTA, we report both the original advantage-weighted centroids before refinement and the refined centroids used to compute token coefficients. A lower cosine similarity indicates greater angular separation between the two centroids.

As shown in Table 15, DAPO's centroids become more aligned over the measured stages. DelTA exhibits greater separation at Step 200, both before and after refinement, and refinement reduces cosine similarity at each measured stage. These observations are consistent with DelTA's goal of constructing more contrastive reference directions.

We emphasize that centroid similarity is only a geometric diagnostic and does not by itself determine the final policy update. In the idealized case where the two sides contain exactly the same shared gradient component with matched side-wise masses, that component cancels in the positive-negative update. In finite rollout batches, however, the two sides can differ in advantage mass, token frequency, and context-dependent gradient contributions, so shared patterns need not contribute identically to the two aggregates. The resulting residual can interfere with more discriminative directions.

Relationship to forking tokens. Entropy-based forking-token selection (Wang et al., 2025) emphasizes positions with high predictive uncertainty. DelTA instead scores contextualized token-gradient vectors by their relative relationship to the two advantage sides. The same surface token can have different hidden states, predictive distributions, and gradient directions in different contexts. Consequently, appearing in both successful and unsuccessful responses does not by itself imply a low DelTA coefficient.

We compare the top 20% highest-entropy tokens with the top 20% highest-weight DelTA tokens. Their overlap is 38.42%, measured as the intersection size divided by the size of either selected set, compared with 20% expected under independent random selection. This overlap indicates related but distinct selection signals: DelTA shares some high-weight positions with entropy-based selection while also identifying tokens through positive-negative gradient contrast.

J.10 Additional Baseline and Controlled Comparisons

Comparison with VIMPO. Our main baselines cover a strong base objective (DAPO), entropy-based token selection (DAPO w/ Forking Tokens), soft-clipping-based optimization (SAPO), and future-influence-based token reweighting (FIPO). They can be evaluated with outcome-only rewards without a separately trained critic or additional process annotations. Methods requiring dense process supervision or additional learned evaluators address related questions under different supervision or component budgets.

To extend this comparison, we additionally evaluate VIMPO (Kang et al., 2026), which derives a policy-implied value function and token-level actor advantages from KL-regularized optimality conditions without training a separate critic. We evaluate it under the same backbone, data, outcome-reward, training-budget, and evaluation settings as DelTA. Table 16 shows that DelTA outperforms VIMPO on all four benchmarks, with average scores of 23.27 and 20.98, respectively.

Table 16: Additional baseline and controlled comparisons. Unless otherwise stated, the learning rate is 1×10^{-6}. All rows follow the same four-benchmark evaluation protocol.

Method	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.
DAPO	23.33	24.17	12.08	16.86	19.05
VIMPO	24.17	25.83	15.83	18.37	20.98
Trajectory-level DelTA	23.96	25.83	14.17	16.29	19.97
DAPO, LR=1.5×10^{-6}	22.71	25.62	12.08	15.34	18.85
Token-level DelTA	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**

Trajectory-level reweighting. To isolate the benefit of within-response discrimination, we construct a trajectory-level DelTA variant. Each response is represented by its mean token-gradient proxy,

$$g_i = \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} v_{i,t}.$$

We construct the positive and negative trajectory-level centroids using weights $|o_i||\hat{A}_i|$, preserving the response masses induced by DAPO's token-level aggregation. Refinement is performed over these response representations, and one final coefficient is shared by all tokens in each response. The remaining training settings and coefficient normalization are unchanged.

The trajectory-level variant achieves an average score of 19.97, compared with 19.05 for DAPO and 23.27 for token-level DelTA. The 3.30-point advantage of token-level weighting supports the benefit of distinguishing token contributions within the same response.

Larger-learning-rate control. DelTA's coefficient self-normalization keeps the mean token multiplier equal to one. Thus, its weighting is not equivalent to a uniform increase in the learning rate, although reweighting can change both gradient direction and magnitude. As an empirical control, we increase DAPO's learning rate from 1×10^{-6} to 1.5×10^{-6}, keeping the remaining settings fixed. This produces an average score of 18.85, below the original DAPO score of 19.05 and DelTA's 23.27. The tested learning-rate increase therefore does not reproduce DelTA's improvement.

J.11 Response-Length Weighting and Extension to GRPO

How response length enters the centroid estimate. Let $\bar{v}_i = |o_i|^{-1} \sum_t v_{i,t}$ denote the mean token-gradient vector of response o_i. Under the local unclipped view, the DAPO and GRPO update directions satisfy

$$\Delta\theta_{\mathrm{DAPO}} \propto \sum_i |o_i| \hat{A}_i \bar{v}_i, \qquad \Delta\theta_{\mathrm{GRPO}} \propto \sum_i \hat{A}_i \bar{v}_i, \tag{17}$$

where shared global normalizers are omitted. DAPO therefore includes an explicit response-length factor, whereas GRPO averages token contributions within each response.

This difference does not imply that DAPO unconditionally favors longer outputs: the length factor amplifies both positive- and negative-advantage contributions. It can nevertheless affect the update when length correlates with advantage or token-gradient direction. Normalizing each side by M_+ or M_- removes a trivial scaling with total token mass, but does not eliminate the greater influence of longer responses on the centroid direction. The two masses also retain side-wise length imbalance in the final update.

DelTA with response-balanced aggregation. To test whether DelTA's improvement depends on DAPO-specific length weighting, we also apply DelTA to GRPO. This variant uses response-balanced weights for both centroid construction and token reweighting, consistent with the GRPO weighting $\rho_{i,t} = 1/|o_i|$ described in Appendix C.

Table 17 shows that DelTA improves the average score by 4.22 points on DAPO and 2.79 points on GRPO. Its gains therefore persist under response-balanced aggregation. DAPO produces longer

Table 17: DelTA under token-balanced DAPO and response-balanced GRPO. Length is the average sampled response length on AIME25, measured in tokens. Bold values indicate the better accuracy within each base-objective pair.

Method	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.	Length
DAPO	23.33	24.17	12.08	16.86	19.05	10,193
DAPO+DelTA	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**	11,745
GRPO	23.13	22.08	12.50	17.05	18.65	5,934
GRPO+DelTA	**25.62**	**25.00**	**16.88**	**18.56**	**21.44**	7,822

Table 18: End-to-end wall-clock comparisons under two approximately matched budgets. Steps denote the maximum training horizon, while accuracy is reported for the checkpoint selected within that horizon.

Method	Steps	Hours	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.
Budget of the original 220-step SAPO run							
SAPO	220	38.8	24.37	26.25	16.04	17.42	20.93
DelTA	200	38.6	**26.46**	**27.08**	**17.08**	**20.08**	**22.61**
Budget of the original 220-step DelTA run							
SAPO	236	43.3	25.62	27.08	16.04	17.80	21.54
DelTA	220	43.1	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**

responses than GRPO in these experiments, and DelTA increases average response length under both objectives. This comparison establishes transfer across the two aggregation schemes; it does not isolate performance at a fixed generated length.

J.12 Comparison under Matched Wall-Clock Budgets

The coefficient-computation overhead in Appendix J.1 does not by itself establish performance under equal end-to-end training time. We therefore compare DelTA with SAPO on the same $8\times$ NVIDIA B200 GPUs using two approximately matched wall-clock budgets. We use SAPO because it is the strongest non-DelTA baseline on Qwen3-8B-Base in the main experiments.

Reported wall-clock times include rollout generation, policy optimization, all additional evaluations, checkpoint saving, and other runtime overhead. Since response lengths evolve differently across methods, the per-step runtime ratio is not constant. We therefore match the observed elapsed times in the training logs. Within each budget, checkpoints are selected using the same pre-specified AIME25 avg@8 criterion.

Under the original SAPO budget, the 220-step SAPO run takes 38.8 hours and the shortened 200-step DelTA run takes 38.6 hours. DelTA achieves an average score of 22.61, exceeding SAPO's 20.93 by 1.68 points. Conversely, extending SAPO to 236 steps takes 43.3 hours, close to the 43.1 hours of the original 220-step DelTA run. DelTA retains a 1.73-point advantage, with average scores of 23.27 and 21.54.

The improvement thus persists when either shortening DelTA to match SAPO's original runtime or extending SAPO to match DelTA's original runtime. These comparisons support DelTA's effectiveness under matched practical training budgets, beyond comparisons using the same number of training steps.

J.13 Robustness to Training Horizon and Checkpoint Selection

Longer training on Qwen3-8B-Base. The original maximum training horizons, 220 steps for Qwen3-8B-Base and 300 steps for Qwen3-14B-Base, were fixed before the method comparison as

Table 19: Effect of extending Qwen3-8B-Base training from 220 to 300 steps. Each row reports the best checkpoint under AIME25 avg@8 selection within the indicated maximum horizon.

Method	Steps	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.
DAPO	220	23.33	24.17	12.08	16.86	19.05
DAPO	300	23.33	24.17	12.08	16.86	19.05
DelTA	220	26.46	28.12	18.54	20.27	23.27
DelTA	300	**27.71**	**30.00**	**19.58**	**20.45**	**24.34**

backbone-specific compute allocations. All methods within each backbone use the same maximum horizon and checkpoint-selection rule. To examine whether the 220-step horizon misses later improvements on the 8B backbone, we extend both DAPO and DelTA to 300 steps and retain the pre-specified AIME25 avg@8 selection criterion.

DAPO does not achieve a higher AIME25 avg@8 score after Step 220, so its selected checkpoint and reported scores remain unchanged. DelTA continues to improve, increasing its average score from 23.27 to 24.34. Under the evaluated horizon and selection rule, the original budget therefore does not omit a later improvement of the selected DAPO checkpoint, while additional training further improves DelTA.

Alternative checkpoint-selection benchmark. We also reselect checkpoints from the original training runs using AIME24 avg@8 instead of AIME25 avg@8. This changes only the selection criterion; the training runs and the four-benchmark reporting suite remain fixed.

Table 20: Robustness to checkpoint selection using AIME24 or AIME25. Both selection criteria use avg@8.

Method	Selection	AIME25	AIME26	HMMT25 (Nov.)	HMMT26 (Feb.)	Avg.
DAPO	AIME25	23.33	24.17	12.08	16.86	19.05
DAPO	AIME24	23.33	23.96	12.08	17.05	19.05
DelTA	AIME25	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**
DelTA	AIME24	**26.46**	**28.12**	**18.54**	**20.27**	**23.27**

Using AIME24 selects a different checkpoint for DAPO, but its average score remains 19.05 at the reported precision. For DelTA, AIME24 and AIME25 select the same checkpoint, yielding an average score of 23.27. DelTA therefore retains its 4.22-point advantage under this alternative selection benchmark.

Interpretation of increasing entropy. The entropy increase observed for DAPO in Figure 2 is not accompanied by an abrupt reward collapse: training completes normally, and the reward curve plateaus with a mild late-stage decrease. Increasing entropy alone is therefore insufficient evidence of training collapse. The longer-training comparison above further shows that extending the run does not yield a better DAPO checkpoint under the pre-specified selection criterion.