

Balanced Aggregation: Understanding and Fixing Aggregation Bias in GRPO

Zhiyuan Zeng^{1,3} Jiameng Huang² Zhangyue Yin¹ Jiashuo Liu³ Ziniu Li³
Bingrui Li⁴ Yuhao Wu⁶ Yining Zheng¹
Ge Zhang^{3*} Wenhao Huang^{3*} Xipeng Qiu^{1,5*}

¹Fudan University ²Peking University ³M-A-P ⁴Tsinghua University
⁵Shanghai Innovation Institute ⁶Singapore University of Technology and Design
cengzy23@m.fudan.edu.cn gezhang@umich.edu rubio8741@gmail.com xpqiu@fudan.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a central paradigm for improving reasoning and code generation in large language models, and GRPO-style training is widely adopted for its simplicity and effectiveness. However, an important design choice remains underexplored: how token-level policy gradient terms are aggregated within each sampled group. Standard GRPO uses sequence aggregation, while recent work has advocated token aggregation as a better alternative. We show that these two rules induce different optimization biases: token aggregation introduces sign-length coupling, while sequence aggregation implicitly downweights longer responses through sequence-level equal weighting. To address this tension, we propose **Balanced Aggregation (BA)**, a simple drop-in replacement that computes token-level means separately within the positive and negative subsets and then combines them with sequence-count-based weights. We further derive BA as the unique minimum-KL correction to uniform token aggregation under a hard advantage-balance constraint. Experiments with Qwen2.5-Math-7B and Qwen3-1.7B on DAPO-17k and Polaris show that BA is robust across model-dataset settings. Independent late-stage diagnostics of entropy, gradient norm, and PPO-KL rank BA first or second in all 12 matched comparisons, although these temporal statistics do not replace multi-seed uncertainty estimates.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central paradigm for improving the reasoning and code generation abilities of large language models (LLMs). By replacing learned reward models with programmatically verifiable signals such as exact-match correctness or unit-test pass rate, RLVR provides a simple and

scalable way to optimize models on tasks with objective outcomes.

Among recent RLVR methods, GRPO-style training is particularly attractive in practice due to its simplicity and effectiveness. For each prompt, the policy samples multiple responses, assigns rewards based on verifiable outcomes, and computes normalized group-wise advantages to optimize a PPO-style objective. This design has been widely adopted in reasoning and coding settings because it avoids training a separate critic while still providing useful relative learning signals within each sampled group.

Despite the growing adoption of GRPO-style RLVR, an important design choice remains underexplored: *how token-level policy gradient terms are aggregated within each sampled group*. In standard GRPO, the default choice is *sequence aggregation*, which first averages over tokens within each response and then averages across responses. Recent works such as DAPO and Dr.GRPO highlighted limitations of this design and accordingly advocated *token aggregation*, which directly averages the clipped objective over all tokens in the sampled group, as a better alternative (Yu et al., 2025; Liu et al., 2025b). In this paper, we show that these two rules induce systematically different optimization biases and can lead to substantially different training dynamics and final performance.

We show that token aggregation introduces a *sign-length coupling bias*: the relative contribution of positive and negative samples on the policy gradient depends not only on their normalized advantages, but also on their average response lengths. Therefore, when positive and negative responses have different length distributions, token aggregation can systematically amplify one side of the update.

Sequence aggregation removes this positive-negative length coupling by assigning equal weight to each response. However, this introduces a

*Corresponding authors: Ge Zhang, Wenhao Huang, and Xipeng Qiu.

different bias: longer responses are implicitly downweighted because each sequence contributes equally regardless of how many tokens it contains (Yu et al., 2025; Liu et al., 2025b).

These two biases matter in practice. We find that token aggregation can be favorable when response length variance is large, since it avoids overly suppressing long responses. However, it is also more sensitive to positive-negative length imbalance and often leads to less stable optimization. This tension suggests that a better aggregation rule should preserve the sign-balance property of sequence aggregation without inheriting its strong sequence-level equal-weighting effect.

To this end, we propose **Balanced Aggregation (BA)**. The key idea is simple: we first split responses within each group into positive and negative subsets according to the sign of their normalized advantages, compute token-level means separately within each subset, and then combine the two subset losses using weights proportional to the number of sequences in each subset. This construction removes the positive-negative length coupling induced by token aggregation, while retaining token-level averaging within each sign group. As a result, BA preserves the same inter-sign balancing principle as sequence aggregation, but does not force every response to have equal weight within a sign group.

We evaluate BA on GRPO-style RLVR training using Qwen2.5-Math-7B and Qwen3-1.7B across DAPO and Polaris training sets, and report results on six evaluation benchmarks including Math-500, AIME 2024, AIME 2025, OlympicBench, Minerva-MATH, and LiveCodeBench. Across both weak and strong model regimes, BA delivers robust final performance and avoids the unstable extremes exhibited by either standard aggregation rule. Our analysis further shows that the relative effectiveness of token and sequence aggregation can be largely explained by two factors: response length variance and the response-length gap between positive and negative samples.

Our contributions are as follows:

- We show that loss aggregation in GRPO-style RLVR is not a benign implementation detail, and provide a unified analysis of the sign-length coupling bias in token aggregation and the sequence equal-weighting bias in sequence aggregation.
- We derive Balanced Aggregation (BA) as the

unique minimum-KL correction to uniform token aggregation under a hard advantage-balance constraint. The solution preserves uniform token weights within each sign subset and changes only their total masses.

- We provide empirical evidence that BA is robust across models, datasets, and benchmarks. Independent entropy, gradient-norm, and PPO-KL diagnostics rank BA in the top two in all 12 matched comparisons, and our length analysis clarifies when token or sequence aggregation is preferable.

2 Related Work

RLVR and GRPO-Style Post-Training Recent progress in reasoning-oriented LLM post-training has highlighted the importance of reinforcement learning, as reflected by the success of systems such as OpenAI’s o1, DeepSeek-R1 (OpenAI et al., 2024; Guo et al., 2025; Zeng et al., 2024). In tasks with programmatically verifiable outcomes, reinforcement learning with verifiable rewards (RLVR) has emerged as a particularly attractive paradigm because it avoids learned reward modeling and provides a scalable training signal for reasoning and code generation. On the optimization side, PPO (Schulman et al., 2017) has long served as the standard policy optimization backbone, while GRPO, introduced in DeepSeekMath (Shao et al., 2024), further reduces training cost by replacing the critic with group-relative reward normalization. This critic-free formulation has made GRPO-style training a practical foundation for large-scale RLVR.

RLVR Training Tricks A growing line of work has improved RLVR training from multiple angles. Some methods focus on reducing train-infer mismatch and improving training stability (Yao et al., 2025; Liu et al., 2025a; Ma et al., 2025; DeepSeek-AI et al., 2025; Zhiyuan et al., 2025). Another line studies clipping and trust-region design, including asymmetric clipping (Yu et al., 2025) and soft clipping (MiniMax et al., 2025). A related direction directly improves importance sampling, with methods such as GSPO and ASPO (Zheng et al., 2025; Wang et al., 2025). Some works also examine the role of advantage estimation and normalization, showing that the standard-deviation normalization can introduce nontrivial optimization bias (Liu et al., 2025b; DeepSeek-AI et al., 2025). Recent empirical studies further revisit a wide range

of RLVR training heuristics and scaling choices, highlighting that many commonly used tricks can have subtle effects (Liu et al., 2025c; Khatri et al., 2025). Together, these studies show that RLVR performance depends heavily on a collection of low-level optimization choices rather than on the policy objective alone.

Aggregation in GRPO-Style RL Among these design choices, how token-level policy gradient terms are aggregated has received less systematic attention. Standard GRPO uses sequence aggregation, which first averages token-level contributions within each response and then averages across responses. Recent works such as DAPO and Dr.GRPO identified limitations of this design in long-form reasoning and accordingly advocated token-level alternatives (Yu et al., 2025; Liu et al., 2025b). GMPO improves optimization stability in a different way, by replacing the arithmetic mean with a geometric mean (Zhao et al., 2025). By contrast, our focus is the bias induced by the aggregation rule itself, so we center the analysis and experiments on sequence aggregation and token aggregation, and position Balanced Aggregation as a simple alternative that directly addresses their respective biases.

3 Method

3.1 GRPO-style RLVR

Given a prompt x , GRPO-style RLVR samples a group of G responses from the current policy,

$$o_1, o_2, \dots, o_G \sim \pi_\theta(\cdot | x). \quad (1)$$

Each response receives a verifiable scalar reward r_i , and rewards are normalized within the group to form sequence-level advantages:

$$\mu = \frac{1}{G} \sum_{i=1}^G r_i, \quad \sigma = \sqrt{\frac{1}{G} \sum_{i=1}^G (r_i - \mu)^2 + \epsilon}, \quad (2)$$

$$\hat{A}_i = \frac{r_i - \mu}{\sigma}. \quad (3)$$

The key point for our analysis is that $\hat{A}_i$ is shared by all tokens in response o_i .

3.2 Token-Level PPO Objective

Let T_i denote the length of response o_i . For token t , define

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} | x, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | x, o_{i,<t})}. \quad (4)$$

The clipped token-level PPO contribution is

$$\phi_{i,t}(\theta) = \min\left(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i\right). \quad (5)$$

The aggregation rule specifies how the token-level terms $\phi_{i,t}$ are reduced to a group-level objective.

3.3 Aggregation rules

Two common reductions are token aggregation and sequence aggregation. Token aggregation averages over all generated tokens:

$$\mathcal{J}_{\text{token}}(\theta) = \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^G \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right], \quad (6)$$

$$N = \sum_{i=1}^G T_i.$$

Sequence aggregation first averages within each response and then averages across responses:

$$\mathcal{J}_{\text{seq}}(\theta) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{T_i} \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right]. \quad (7)$$

Both use the same token-level objective, but they induce different length-dependent weights.

3.4 Motivation: Aggregation Bias in GRPO

To expose the bias, partition responses by the sign of their advantage:

$$S_+ = \{i \mid \hat{A}_i > 0\}, \quad S_- = \{i \mid \hat{A}_i < 0\}. \quad (8)$$

Let $k = |S_+|$, so $|S_-| = G - k$. For analysis, we factor the sequence-level advantage out of the token-level PPO contribution and define $\delta_{i,t}(\theta) := \phi_{i,t}(\theta) / \hat{A}_i$ for nonzero-advantage responses:

$$\phi_{i,t}(\theta) = \hat{A}_i \delta_{i,t}(\theta). \quad (9)$$

Under binary rewards, group normalization gives

$$\hat{A}_i = \begin{cases} \sqrt{\frac{G-k}{k}}, & i \in S_+, \\ -\sqrt{\frac{k}{G-k}}, & i \in S_-. \end{cases} \quad (10)$$

Substituting this into token aggregation yields

$$\mathcal{J}_{\text{token}} \propto \frac{\sqrt{k(G-k)}}{N} \left(\bar{T}_+ \bar{\delta}_+^{\text{tok}} - \bar{T}_- \bar{\delta}_-^{\text{tok}} \right), \quad (11)$$

where

$$\bar{T}_+ = \frac{1}{k} \sum_{i \in S_+} T_i, \quad \bar{T}_- = \frac{1}{G-k} \sum_{i \in S_-} T_i, \quad (12)$$

and $\bar{\delta}_{\pm}^{\text{tok}}$ denote token-level averages within each sign group. More explicitly, with $N_+ = \sum_{i \in S_+} T_i$ and $N_- = \sum_{i \in S_-} T_i$,

$$\bar{\delta}_{\pm}^{\text{tok}} = \frac{1}{N_{\pm}} \sum_{i \in S_{\pm}} \sum_{t=1}^{T_i} \delta_{i,t}. \quad (13)$$

Thus token aggregation creates a **sign-length coupling bias**: when $\bar{T}_+ \neq \bar{T}_-$, the positive and negative gradient contributions are reweighted by their average lengths. This is not merely a normalization detail; it changes the effective balance between reward-increasing and reward-decreasing updates whenever response length correlates with reward sign.

By contrast, sequence aggregation removes this cross-sign length coupling:

$$\mathcal{J}_{\text{seq}} \propto \frac{\sqrt{k(G-k)}}{G} (\bar{\delta}_+^{\text{seq}} - \bar{\delta}_-^{\text{seq}}), \quad (14)$$

where

$$\begin{aligned} \bar{\delta}_+^{\text{seq}} &= \frac{1}{k} \sum_{i \in S_+} \left(\frac{1}{T_i} \sum_{t=1}^{T_i} \delta_{i,t} \right), \\ \bar{\delta}_-^{\text{seq}} &= \frac{1}{G-k} \sum_{i \in S_-} \left(\frac{1}{T_i} \sum_{t=1}^{T_i} \delta_{i,t} \right). \end{aligned} \quad (15)$$

This equalizes the positive and negative signs at the group level, but introduces a **sequence equal-weighting bias**: each response has equal weight regardless of its token count. This can be desirable when the dominant issue is sign-length coupling, but it may discard useful token-level averaging when response lengths vary substantially for reasons unrelated to correctness. The goal of BA is to preserve the sign balance of sequence aggregation while retaining token-level averaging within each sign group.

3.5 Balanced Aggregation

We propose **Balanced Aggregation (BA)**, a simple aggregation rule that separates positive and negative samples before averaging.

We first compute token-level mean losses within the positive and negative subsets:

$$\begin{aligned} \mathcal{L}_+ &= \frac{1}{N_+} \sum_{i \in S_+} \sum_{t=1}^{T_i} \phi_{i,t}(\theta), \\ \mathcal{L}_- &= \frac{1}{N_-} \sum_{i \in S_-} \sum_{t=1}^{T_i} \phi_{i,t}(\theta). \end{aligned} \quad (16)$$

We then combine them using sequence-count-based weights:

$$\mathcal{J}_{\text{BA}}(\theta) = \mathbb{E} \left[\frac{k}{G} \mathcal{L}_+ + \frac{G-k}{G} \mathcal{L}_- \right]. \quad (17)$$

Equivalently,

$$\begin{aligned} \mathcal{J}_{\text{BA}}(\theta) &= \mathbb{E} \left[\frac{k}{G} \cdot \frac{1}{N_+} \sum_{i \in S_+} \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right. \\ &\quad \left. + \frac{G-k}{G} \cdot \frac{1}{N_-} \sum_{i \in S_-} \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right]. \end{aligned} \quad (18)$$

BA as minimum-distortion sign-balanced aggregation. BA is not a heuristic interpolation between token and sequence aggregation. Consider one mixed-reward group and let $p_{i,t} = 1/N$ be the uniform token weights used by token aggregation. Among normalized nonnegative token weights q , impose the hard advantage-balance constraint $\sum_{i,t} q_{i,t} \hat{A}_i = 0$, and otherwise perturb p as little as possible:

$$\min_{q \in \Delta^N} D_{\text{KL}}(q \| p) \quad \text{s.t.} \quad \sum_{i=1}^G \sum_{t=1}^{T_i} q_{i,t} \hat{A}_i = 0. \quad (19)$$

For binary rewards with $0 < k < G$, the unique solution is

$$q_{i,t}^* = \begin{cases} \frac{k}{GN_+}, & i \in S_+, \\ \frac{G-k}{GN_-}, & i \in S_-. \end{cases} \quad (20)$$

Applying q^* to the token-level terms $\phi_{i,t}$ gives exactly $\mathcal{J}_{\text{BA}}$. Thus, BA assigns the required total masses k/G and $(G-k)/G$ to the two signs while preserving uniform relative token weights within each sign subset. The complete derivation and uniqueness argument appear in [Appendix A](#).

3.6 Connection to Sequence Aggregation

BA is closely related to sequence aggregation, but the two are not equivalent.

Under the same binary-reward GRPO setting, substituting the normalized advantages into BA yields

$$\mathcal{J}_{\text{BA}} \propto \frac{\sqrt{k(G-k)}}{G} (\bar{\delta}_+^{\text{BA}} - \bar{\delta}_-^{\text{BA}}), \quad (21)$$

where

$$\begin{aligned} \bar{\delta}_+^{\text{BA}} &= \frac{1}{N_+} \sum_{i \in S_+} \sum_{t=1}^{T_i} \delta_{i,t}, \\ \bar{\delta}_-^{\text{BA}} &= \frac{1}{N_-} \sum_{i \in S_-} \sum_{t=1}^{T_i} \delta_{i,t}. \end{aligned} \quad (22)$$

By contrast, sequence aggregation has exactly the same inter-sign form as in Eq. (14), with within-sign averages defined in Eq. (15).

Therefore, BA and sequence aggregation share the same *inter-sign balancing* structure: both remove the sign-length coupling of token aggregation and induce the same positive-negative prefactor $\sqrt{k(G-k)}/G$. However, they differ in their *within-sign averaging* rule. Sequence aggregation first averages tokens within each response and then averages responses within each sign group, as shown in Eq. (15). BA directly averages over all tokens within each sign group, as shown in Eq. (22).

In general,

$$\bar{\delta}_{\pm}^{\text{seq}} \neq \bar{\delta}_{\pm}^{\text{BA}}, \quad (23)$$

unless all responses within a sign group have the same length. Thus, BA should be understood as preserving the sign-balance property of sequence aggregation without inheriting its strong per-sequence equal-weighting effect.

BA is a simple drop-in replacement for the aggregation step in GRPO-style RLVR. It removes the sign-length coupling bias of token aggregation while avoiding the strong sequence equal-weighting bias of sequence aggregation. Although the current formulation of BA is derived under the binary-reward setting, BA can naturally extend to non-binary rewards, which is shown in Appendix B.

4 Experiments

4.1 Experimental Settings

Training Data We conduct RLVR training on two datasets: **DAPO-17k** (approximately 17,000

mathematical reasoning problems) and **Polaris** (approximately 53,000 mathematical problems) (Yu et al., 2025; Group et al., 2025). Both consist of problem-answer pairs, where answers are used to compute verifiable rewards for generated responses.

Evaluation Benchmarks We evaluate on six benchmarks covering both difficult reasoning and coding tasks: Math-500, AIME-2024, AIME-2025, OlympicBench, Minerva-MATH, and LivecodeBench (Lightman et al., 2023; Lewkowycz et al., 2022; He et al., 2024; Jain et al., 2024).

Compared Methods We compare three aggregation rules applied within the DAPO algorithm:

- **token-agg**: Token-level averaging, where the clipped PPO objective is averaged over all tokens in the sampled group. This is used in DAPO and Dr.GRPO (Yu et al., 2025; Liu et al., 2025b).
- **seq-agg**: Sequence-level averaging, where token-level contributions are first averaged within each response and then averaged across responses. This is the default in GRPO (Shao et al., 2024).
- **balanced-agg**: Our proposed balanced aggregation, which splits responses by advantage sign, computes token-level means separately within positive and negative subsets, and combines them with sequence-count-based weights.

All other components (advantage normalization, PPO clipping, sampling) are kept identical across methods.

Training Details We train **Qwen2.5-Math-7B** and **Qwen3-1.7B** (Yang et al., 2024, 2025) with maximum response lengths of 2,048 and 8,192 tokens. Training is implemented in the verl framework, using group size $G = 16$, learning rate 10^{-6} , and 500 total steps. We apply PPO clipping bounds of 0.2 and 0.28 (the standard DAPO setting). The global batch size is 128 prompts, each generating 16 responses via vLLM (temperature 1.0). Other hyper-parameters follow the standard DAPO configuration (Yu et al., 2025).

Table 1: Main experimental results on DAPO-17k and Polaris datasets using Qwen2.5-Math-7B and Qwen3-1.7B models. We report the average peak and last-step accuracy across six evaluation benchmarks.

Dataset	Model	Aggregation	Peak Performance		Last-Step	
			Acc@8	Best@8	Acc@8	Best@8
DAPO-17k	Qwen2.5-Math-7B	seq-agg	0.3578	0.4821	0.3446	0.4598
		token-agg	0.3595	0.4869	0.3364	0.4434
		balanced-agg	0.3634	0.4896	0.3424	0.4627
	Qwen3-1.7B	seq-agg	0.4947	0.5990	0.4481	0.5522
		token-agg	0.4876	0.6058	0.4360	0.5608
		balanced-agg	0.4928	0.6086	0.4695	0.5861
Polaris	Qwen2.5-Math-7B	seq-agg	0.3405	0.4697	0.3172	0.4360
		token-agg	0.3539	0.4768	0.3292	0.4527
		balanced-agg	0.3423	0.4750	0.3319	0.4443
	Qwen3-1.7B	seq-agg	0.4812	0.5950	0.4614	0.5730
		token-agg	0.4842	0.6053	0.4349	0.5557
		balanced-agg	0.4879	0.6094	0.4640	0.5550

Evaluation Protocol We sample 8 responses per prompt (temperature 1.0). For math benchmarks, correctness is determined using OpenCompass’s rule-based verifier (Team, 2024); for Live-codeBench, we execute the generated code against unit tests (Jain et al., 2024). We report three metrics: **peak accuracy** (highest accuracy observed during training), **peak best@8 accuracy**, and **last-step accuracy** (accuracy at the final training step).

4.2 Main Results

To evaluate **Balanced Aggregation** (denoted as **balanced-agg**), we compare it against **token-agg** and **seq-agg** baselines on two training datasets: DAPO-17k and Polaris. We benchmark the aggregation methods on two base models: Qwen2.5-Math-7B and Qwen3-1.7B.

Table 1 presents the average scores across six evaluation benchmarks. Since a full breakdown would necessitate an overly large table, detailed per-benchmark results are shown in Figure 1. Furthermore, because RLVR training dynamics can be highly volatile in later stages, the highest peak performance does not guarantee the best last-step performance. We therefore explicitly report both **peak** and **last-step** metrics to comprehensively evaluate each method’s training stability.

Overall results on DAPO-17k For Qwen2.5-Math-7B, **token-agg** yields better peak performance than **seq-agg**, but **balanced-agg** surpasses both to establish the highest peak metrics. For Qwen3-1.7B, the relationship flips: **seq-agg** becomes superior to **token-agg**. This is highly rele-

vant since **token-agg** is the default in frameworks like *verl*, yet it is clearly not universally better than **seq-agg**. More crucially, while **balanced-agg** achieves peak metrics comparable to **seq-agg**, it successfully prevents the severe degradation often observed in later stages of RLVR, maintaining much higher last-step accuracies.

Overall results on Polaris Similar performance dynamics are observed on Polaris. For Qwen2.5-Math-7B, **token-agg** exhibits the highest peak metrics but suffers noticeable degradation toward the end of training. In contrast, **balanced-agg** achieves the most robust last-step accuracy and strictly outperforms **seq-agg** across all evaluated metrics. For Qwen3-1.7B, **balanced-agg** achieves the highest peak metrics compared to both baselines. In the final training stages, while **token-agg** suffers a severe collapse, both **seq-agg** and **balanced-agg** maintain much more robust last-step accuracies.

Cross-setting summary Across both datasets, a consistent dynamic emerges: **token-agg** performs better on Qwen2.5-Math-7B, whereas **seq-agg** is more stable and accurate on Qwen3-1.7B. This suggests neither standard aggregation provides a consistently reliable optimization signal across different base models. We delve into the reasons behind this performance flip in Section 4.3. **Balanced-agg** bridges this gap, ranking as the best or a competitive method across the evaluated settings. The independent diagnostics in Section 4.3.3 further show that it consistently avoids the least stable rank.

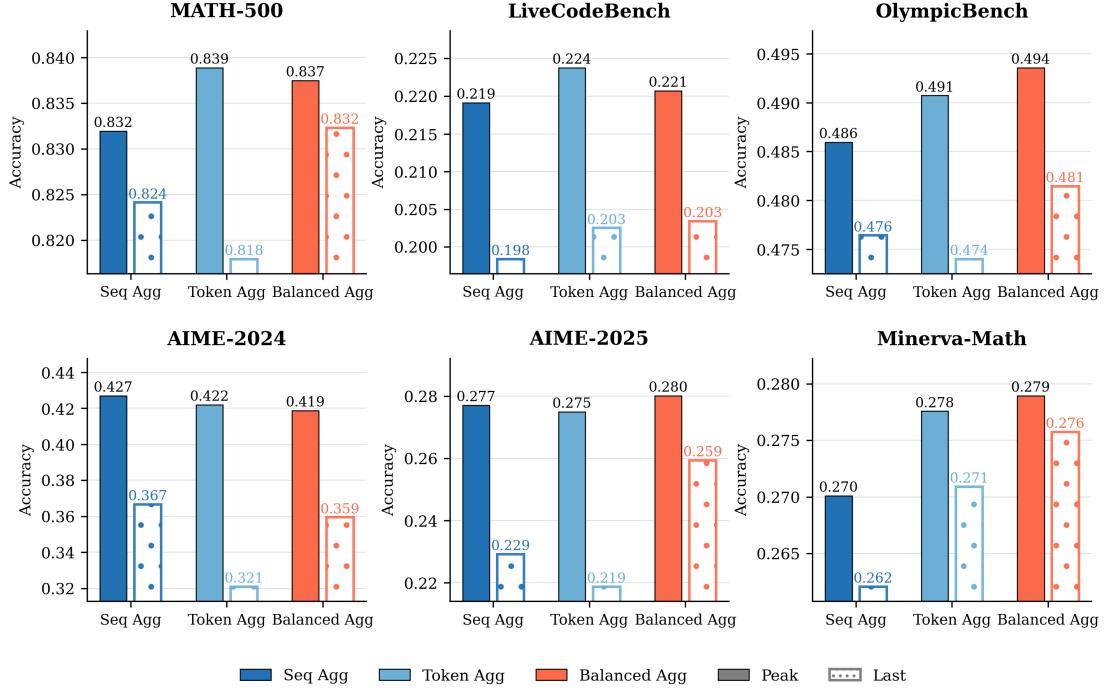


Figure 1: Comparison of peak and last-step accuracy across evaluation benchmarks. For each benchmark, values are averaged over four training settings: Qwen2.5-Math-7B and Qwen3-1.7B, each trained on DAPO-17k and Polaris. Solid bars indicate peak performance, while dashed bars indicate last-step performance.

4.3 Analysis

The main results in Table 1 show that aggregation rules interact strongly with the base model and training corpus, and that peak accuracy alone can be misleading when training is volatile. In this subsection, we unpack these findings along four complementary axes. First, we compare peak versus last-step accuracy at the per-benchmark level. Second, we use policy-gradient loss trajectories only to visualize the aggregation imbalance predicted by our analysis. Third, we quantify late-stage training variability using entropy, gradient norm, and PPO-KL, which are independent of the scalar loss definition. Finally, we connect the theory in Section 3.4 to the model-dependent flip in Section 4.2 and to length statistics over training (Figure 3).

4.3.1 Peak vs. Last-Step Performance

Figure 1 compares peak and last-step accuracy on each benchmark, where each bar is averaged over four training settings: Qwen2.5-Math-7B and Qwen3-1.7B, each trained on DAPO-17k and Polaris. At peak performance, **token-agg** and **balanced-agg** are very close, and on most benchmarks both outperform **seq-agg**. The difference emerges at the final checkpoint: **token-agg** exhibits the largest peak-to-last drop on nearly all

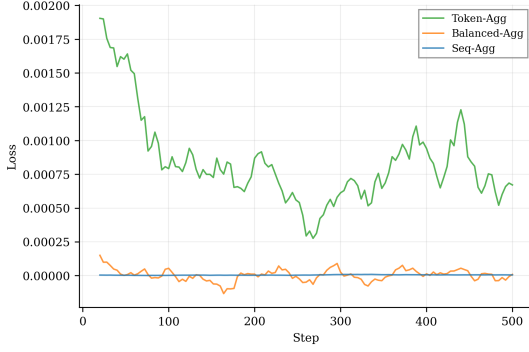
benchmarks, whereas **balanced-agg** preserves its gains much better and achieves the best or tied-best last-step result on five of the six benchmarks. The largest peak-to-last gaps appear on AIME-2024 and AIME-2025, likely due in part to the higher variance of their small evaluation sets. Overall, BA shows stronger late-stage retention than token aggregation while remaining competitive at peak performance.

4.3.2 Policy-gradient loss dynamics and aggregation bias

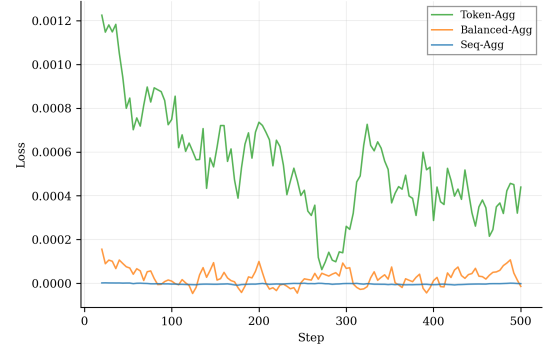
Figure 2 illustrates the loss decomposition predicted by Section 3.4. When the positive–negative response-length gap is large, token aggregation overweights the longer sign group and produces a persistent offset, whereas sequence aggregation and BA remove this cross-sign factor. Because this behavior partly follows from the respective normalizations, we do not interpret a near-zero scalar loss as evidence of better optimization. We instead evaluate training variability with independent signals below.

4.3.3 Independent Late-Stage Stability Diagnostics

We measure the temporal standard deviation of policy entropy, gradient norm, and absolute PPO-KL



(a) Qwen2.5 on DAPO



(b) Qwen2.5 on Polaris

Figure 2: Policy-gradient loss for Qwen2.5-Math-7B on DAPO-17k and Polaris. Because negative responses are systematically longer, **token-agg** places disproportionate mass on negative samples and its scalar loss shifts away from zero. **Seq-agg** and **balanced-agg** remove this cross-sign length coupling. This figure diagnoses the predicted aggregation imbalance; proximity to zero is not itself evidence of policy quality or optimization stability.

Table 2: Late-stage stability diagnostics over the final 100 training steps. Lower is better. PPO-KL standard deviations are reported in units of 10^{-5} (e.g., 3.333 denotes 3.333×10^{-5}). Bold and italic denote the best and second-best aggregation rules within each matched model–dataset–metric comparison. The rank summary counts positions over all 12 comparisons.

Model	Dataset	Temporal std	Token-agg	Seq-agg	BA
Qwen2.5	DAPO-17k	Entropy	.0284	.0237	.0223
		Grad norm	.0288	.0234	.0245
		PPO-KL	3.333	4.381	3.911
Qwen2.5	Polaris	Entropy	.0532	.0381	.0339
		Grad norm	.0155	.1011	.0236
		PPO-KL	3.610	4.895	3.732
Qwen3	DAPO-17k	Entropy	.0491	.0380	.0485
		Grad norm	.0045	.0035	.0042
		PPO-KL	.864	.605	.794
Qwen3	Polaris	Entropy	.0562	.0249	.0517
		Grad norm	.0036	.0033	.0030
		PPO-KL	.857	.846	.716
Rank summary (12 comparisons)			Token-agg	Seq-agg	BA
First			3	5	4
Second			0	4	8
Last			9	3	0

over the final 100 training steps (steps 401–500). These quantities characterize fluctuations in policy uncertainty, update magnitude, and policy drift, respectively, and do not depend on the scalar policy-loss aggregation. Lower values indicate smoother late-stage dynamics.

Table 2 reveals different leaders for different diagnostics. For entropy, BA has lower variability than token-agg in all four settings: it ranks first for both Qwen2.5 runs and second to seq-agg for both Qwen3 runs. For gradient norm, BA remains in the top two in every setting, ranking first on Qwen3–Polaris and second in the other three. The Qwen2.5–Polaris run also illustrates why no

method dominates universally: token-agg has the lowest gradient-norm variation there, while seq-agg is substantially more variable.

The PPO-KL results show a complementary model-dependent pattern. Token-agg is best on the two Qwen2.5 runs, seq-agg is best on Qwen3–DAPO, and BA is best on Qwen3–Polaris; nevertheless, BA ranks second in the other three comparisons. Aggregating all metrics and settings, BA obtains 4 first-place and 8 second-place ranks, with no last-place outcomes. Seq-agg obtains more wins (5) but ranks last 3 times, while token-agg ranks last 9 times. We therefore interpret BA as the most consistent method across diagnostics, not as

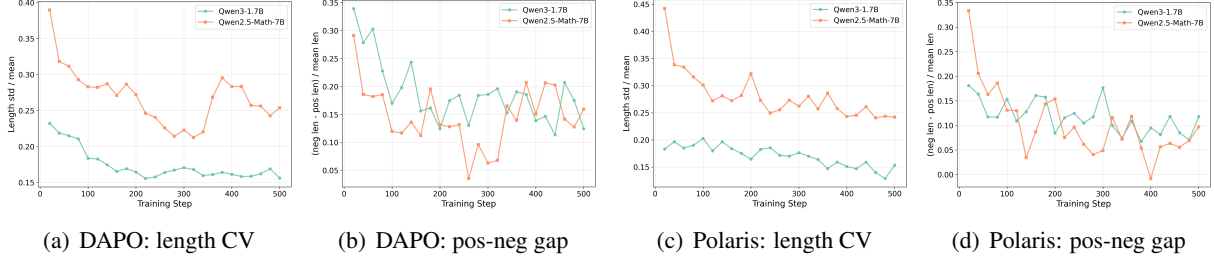


Figure 3: Length-distribution statistics related to the relative behavior of token and sequence aggregation. We report response-length CV, where CV denotes the coefficient of variation, and the pos-neg gap, defined as the average length of positive responses minus that of negative responses, over training on DAPO-17k and Polaris.

a method that minimizes every variability measure. Because each result comes from one training trajectory, these temporal statistics do not establish multi-seed significance.

4.3.4 Understanding the Model-Dependent Flip

Our theory in Sections 3.3 and 3.4 suggests a simple criterion for when each aggregation rule should be preferred. **Token-agg** should be more favorable when response-length variation is large but the positive–negative length gap is relatively mild, because in that regime the weakness of **seq-agg** is more pronounced than the weakness of **token-agg**. Conversely, **seq-agg** should be more favorable when response-length variation is small but the positive–negative length gap is large, because in that regime the sign-length coupling bias in **token-agg** becomes the dominant issue. In Section 4.2, we observe exactly such a model-dependent flip: **token-agg** performs better on Qwen2.5-Math-7B, whereas **seq-agg** performs better on Qwen3-1.7B. Figure 3 tests this explanation by tracking over training the two quantities highlighted by the theory on DAPO-17k and Polaris.

When is Token-Agg Better than Seq-Agg?

Qwen2.5-Math-7B falls into the regime favorable to **token-agg**. As shown in Figure 3, it exhibits substantially larger response-length variation than Qwen3-1.7B, while its positive–negative length gap is generally milder. This means the sequence-level equal-weighting effect in **seq-agg** is more harmful than the sign-length coupling bias in **token-agg**, which explains why **token-agg** can outperform **seq-agg** on Qwen2.5-Math-7B in Table 1.

When is Seq-Agg Better than Token-Agg?

Qwen3-1.7B shows the opposite pattern. Its response-length variation is much smaller, so the

weakness of **seq-agg** is reduced. Meanwhile, its positive–negative length gap is consistently larger, making the sign-length coupling bias in **token-agg** more severe. This is why **seq-agg** outperforms **token-agg** on Qwen3-1.7B in Table 1. The same comparison also helps explain why **balanced-agg** is robust across both regimes: it mitigates the positive–negative coupling that hurts **token-agg** without paying the full cost of sequence-level equal weighting.

5 Conclusion

We studied how aggregation in GRPO-style RLVR shapes optimization. Token aggregation suffers from sign-length coupling, while sequence aggregation removes this coupling but imposes strong per-sequence equal weighting. BA is the unique minimum-KL correction to uniform token aggregation that enforces advantage balance under binary rewards: it preserves uniform token weighting within each sign group while correcting only their total masses. Across models and datasets, BA remains competitive in performance and ranks first or second in all 12 independent late-stage stability comparisons. These results establish aggregation as a first-class design choice rather than a benign implementation detail.

Limitations

Our experiments cover two models and two corpora in math/code RLVR; broader settings and multi-seed runs are needed to establish generality and statistical significance. The minimum-KL result assumes binary rewards and mixed-sign groups, and the non-binary extension in Appendix B remains untested.

References

- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. [Deepseek-v3.2: Pushing the frontier of open large language models](#). *Preprint*, arXiv:2512.02556.
- HKU NLP Group, ByteDance Seed, and Fudan University. 2025. [Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models](#).
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shitong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida I. Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. [Live-codebench: Holistic and contamination free evaluation of large language models for code](#). *Preprint*, arXiv:2403.07974.
- Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. 2025. [The art of scaling reinforcement learning compute for llms](#). *Preprint*, arXiv:2510.13786.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. [Solving quantitative reasoning problems with language models](#). *Preprint*, arXiv:2206.14858.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *Preprint*, arXiv:2305.20050.
- Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. 2025a. [When speed kills stability: Demystifying RL collapse from the training-inference mismatch](#).
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. [Understanding r1-zero-like training: A critical perspective](#). *Preprint*, arXiv:2503.20783.
- Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, Shengyi Huang, Johan Obando-Ceron, Siran Yang, Jiamang Wang, Wenbo Su, and Bo Zheng. 2025c. [Part i: Tricks or traps? a deep dive into rl for llm reasoning](#). *Preprint*, arXiv:2508.08221.
- Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. 2025. [Stabilizing moe reinforcement learning by aligning training and inference routers](#). *Preprint*, arXiv:2510.11370.
- MiniMax, :, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, and 109 others. 2025. [Minimax-m1: Scaling test-time compute efficiently with lightning attention](#). *Preprint*, arXiv:2506.13585.
- OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *Preprint*, arXiv:1707.06347.
- Yiping Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, Daya Guo, Dejian Yang, Dejian Yang, and Ruoyu Zhang. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- OpenCompass Team. 2024. [Opencompass: A universal evaluation platform for foundation models](#). *Preprint*, arXiv:2410.16256.
- Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. 2025. [Aspo: Asymmetric importance sampling policy optimization](#). *Preprint*, arXiv:2510.06062.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 42 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. [Qwen2.5-math technical report: Toward mathematical expert model via self-improvement](#). *Preprint*, arXiv:2409.12122.
- Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. 2025. [Your efficient rl framework secretly brings you off-policy rl training](#).
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.
- Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. 2024. [Scaling of search and learning: A roadmap to reproduce ol from reinforcement learning perspective](#). *Preprint*, arXiv:2412.14135.
- Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. 2025. [Geometric-mean policy optimization](#). *Preprint*, arXiv:2507.20673.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. [Group sequence policy optimization](#). *Preprint*, arXiv:2507.18071.
- Zeng Zhiyuan, Jiashuo Liu, Zhangyue Yin, Ge Zhang, Wenhao Huang, and Xipeng Qiu. 2025. [Rloop: An self-improving framework for reinforcement learning with iterative policy initialization](#). *Preprint*, arXiv:2511.04285.

Appendix A BA as a Minimum-KL Sign-Balanced Aggregation

We derive the solution of Eq. (19). Let $\mathcal{I}_+$ and $\mathcal{I}_-$ denote the positive- and negative-advantage token sets, with cardinalities N_+ and N_- . Under binary rewards, all tokens in each set share one of two normalized advantage values:

$$a_+ = \sqrt{\frac{G-k}{k}}, \quad a_- = -\sqrt{\frac{k}{G-k}}. \quad (24)$$

Let $Q_+ = \sum_{(i,t) \in \mathcal{I}_+} q_{i,t}$ and $Q_- = 1 - Q_+$ be the total token-weight masses assigned to the two signs. The balance constraint is

$$a_+ Q_+ + a_- Q_- = 0. \quad (25)$$

Together with $Q_+ + Q_- = 1$, this uniquely determines

$$\begin{aligned} Q_+ &= \frac{-a_-}{a_+ - a_-} = \frac{k}{G}, \\ Q_- &= \frac{a_+}{a_+ - a_-} = \frac{G-k}{G}. \end{aligned} \quad (26)$$

It remains to determine how each fixed mass should be distributed within its sign set. Since $p_{i,t} = 1/N$,

$$D_{\text{KL}}(q||p) = \sum_{i,t} q_{i,t} \log q_{i,t} + \log N. \quad (27)$$

For a fixed Q_+ , strict convexity of $u \log u$ (or the KKT conditions) implies that the positive-set contribution is uniquely minimized by distributing Q_+ uniformly over its N_+ tokens. The same argument applies to the negative set. Hence

$$q_{i,t}^* = \begin{cases} \frac{Q_+}{N_+} = \frac{k}{GN_+}, & (i,t) \in \mathcal{I}_+, \\ \frac{Q_-}{N_-} = \frac{G-k}{GN_-}, & (i,t) \in \mathcal{I}_-. \end{cases} \quad (28)$$

The feasible set is convex and $D_{\text{KL}}(q||p)$ is strictly convex on it, so this solution is unique. Finally,

$$\sum_{i,t} q_{i,t}^* \phi_{i,t} = \frac{k}{G} \mathcal{L}_+ + \frac{G-k}{G} \mathcal{L}_- = \mathcal{J}_{\text{BA}}, \quad (29)$$

which proves that BA is exactly the minimum-distortion sign-balanced correction to token aggregation. The sequence-count weights are therefore derived from the balance constraint rather than chosen heuristically.

Appendix B Extension to Non-Binary Rewards

Our main presentation of Balanced Aggregation (BA) focuses on the standard binary-reward GRPO setting, where the normalized advantages are constant within each sign subset. In that case, the positive and negative groups can be characterized solely by their sequence counts, which leads to the simple sequence-count weights k/G and $(G-k)/G$.

For general real-valued rewards, however, the normalized advantages $\hat{A}_i$ are no longer constant even within the same sign subset. As a result, sequence counts alone are no longer sufficient to characterize the relative contribution of the positive and negative subsets. In particular, if we write the token-level PPO contribution as

$$\phi_{i,t}(\theta) = \hat{A}_i \delta_{i,t}(\theta),$$

then the total contribution of each sign subset depends not only on the number of responses and their lengths, but also on the magnitudes of their advantages.

To generalize BA to this setting, we first define the positive and negative subsets

$$S_+ = \{i \mid \hat{A}_i > 0\}, \quad S_- = \{i \mid \hat{A}_i < 0\}.$$

We then define the corresponding sign-wise advantage masses

$$M_+ = \sum_{i \in S_+} \hat{A}_i, \quad M_- = \sum_{i \in S_-} (-\hat{A}_i),$$

and the sign-wise advantage-weighted token masses

$$Z_+ = \sum_{i \in S_+} \hat{A}_i T_i, \quad Z_- = \sum_{i \in S_-} (-\hat{A}_i) T_i.$$

Using these quantities, we define the generalized BA objective as

$$\begin{aligned} \mathcal{J}_{\text{BA-gen}}(\theta) = & \mathbb{E} \left[\frac{M_+}{G} \cdot \frac{1}{Z_+} \sum_{i \in S_+} \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right. \\ & \left. + \frac{M_-}{G} \cdot \frac{1}{Z_-} \sum_{i \in S_-} \sum_{t=1}^{T_i} \phi_{i,t}(\theta) \right]. \end{aligned}$$

Since $\phi_{i,t}(\theta)$ is positive-advantage-weighted on S_+ and negative-advantage-weighted on S_- , this construction preserves the original policy-gradient weighting induced by $\hat{A}_i$, while normalizing the

positive and negative subsets separately. Equivalently, substituting $\phi_{i,t}(\theta) = \hat{A}_i \delta_{i,t}(\theta)$ gives

$$\mathcal{J}_{\text{BA-gen}}(\theta) = \mathbb{E} \left[\frac{M_+}{G} \bar{\delta}_+^{\text{gen}} - \frac{M_-}{G} \bar{\delta}_-^{\text{gen}} \right],$$

where

$$\begin{aligned} \bar{\delta}_+^{\text{gen}} &= \frac{1}{Z_+} \sum_{i \in S_+} \hat{A}_i \sum_{t=1}^{T_i} \delta_{i,t}(\theta), \\ \bar{\delta}_-^{\text{gen}} &= \frac{1}{Z_-} \sum_{i \in S_-} (-\hat{A}_i) \sum_{t=1}^{T_i} \delta_{i,t}(\theta). \end{aligned}$$

Therefore, the generalized BA retains the same core principle as the binary version: positive and negative samples are first normalized within their own sign subsets and are then recombined in a sign-balanced manner. The key difference is that, in the non-binary case, sign balance is determined by advantage mass rather than sequence count.

Moreover, under the usual group-normalization condition

$$\sum_{i=1}^G \hat{A}_i = 0,$$

we have

$$M_+ = M_- = \frac{1}{2} \sum_{i=1}^G |\hat{A}_i|.$$

Hence, the positive and negative subsets remain symmetric at the inter-sign level, so the generalized BA still removes the cross-sign sign-length coupling induced by standard token aggregation.

Finally, under binary rewards, $\hat{A}_i$ is constant within each sign subset. In that case,

$$\begin{aligned} M_+ &= k a_+, & M_- &= (G - k) a_-, \\ Z_+ &= a_+ N_+, & Z_- &= a_- N_-, \end{aligned}$$

for some constants $a_+ > 0$ and $a_- > 0$. Substituting these expressions into $\mathcal{J}_{\text{BA-gen}}$ recovers exactly the original BA objective defined in the main text. Therefore, the generalized formulation is a strict extension of BA rather than a different objective.