On the Hidden Objective Biases of Group-based Reinforcement Learning

Aleksandar Fontana[1,2*], **Marco Simoni**[2,3*], **Giulio Rossolini**[1], **Paolo Mori**[2], **Andrea Saracino**[1]
[1] Department of Excellence in Robotics and AI, TeCIP, Scuola Superiore Sant'Anna, Pisa
[2] Institute of Informatics and Telematics, National Research Council of Italy, Pisa
[3] National Doctorate on Artificial Intelligence, Sapienza Università di Roma

Abstract

Group-based reinforcement learning methods, like Group Relative Policy Optimization (GRPO), are widely used nowadays to post-train large language models. Despite their empirical success, they exhibit structural mismatches between reward optimization and the underlying training objective. In this paper, we present a theoretical analysis of GRPO style methods by studying them within a unified surrogate formulation. This perspective reveals recurring properties that affect all the methods under analysis: (i) non-uniform group weighting induces systematic gradient biases on shared prefix tokens; (ii) interactions with the AdamW optimizer make training dynamics largely insensitive to reward scaling; and (iii) optimizer momentum can push policy updates beyond the intended clipping region under repeated optimization steps. We believe that these findings highlight fundamental limitations of current approaches and provide principled guidance for the design of future formulations.

1 Introduction

Recent advances in Large Language Model (LLM) post-training have shown that reinforcement learning methods based on group-level feedback can effectively improve reasoning performance while avoiding the cost of explicit value-function estimation, as used in previous works (Ouyang et al., 2022; Yao et al., 2023). Among these approaches, Group Relative Policy Optimization (GRPO) and related methods have gained widespread adoption due to their simplicity and scalability, and are now commonly used in post-training pipelines for reasoning-oriented models (Shao et al., 2024; Liu et al., 2025a; Zheng et al., 2025; Yu et al., 2025).

Despite their empirical success, GRPO style methods rely on a surrogate objective whose optimization dynamics remain only partially understood. Several recent works have reported unexpected behaviors during training, including length-related biases (Liu et al., 2025b), sensitivity to formatting tokens (Simoni et al., 2025), reward hacking in multi-objective settings (Ichihara et al., 2025), and instability across different optimization regimes However, these findings represent fragmented empirical observations, and a unified formal framework that systematically connects and further extends them to the surrogate objective's implicit inductive biases is lacking.

This work offers a unified critical analysis of group-based optimization methods. We propose a general formulation of GRPO style methods, showing ten recent approaches as special cases. This view reveals shared issues, showing that the surrogate objective is often dominated by weighting schemes, regularization, and importance sampling, rather than by pure reward maximization. Building on this formulation, we identify three recurring properties of GRPO style training dynamics: (i) we analyze token-level gradients to demonstrate that non-uniform weighting induces systematic biases on shared prefix tokens; (ii) we study the interaction with AdamW (Loshchilov and Hutter, 2017), demonstrating that the training process remains invariant to global reward scaling across various scenarios; (iii) we show that optimizer momentum can drive policy updates beyond the intended clipping boundaries during multi-step optimization. Beyond empirical performance, our analysis offers theoretical insights exposing a divergence between the surrogate objective and the true training goal. By characterizing these dynamics, our findings provide the community with a reference for the design and interpretation of LLM post-training strategies.

2 Related Work

Recent work has started investigating the problems arising during GRPO style post-training. Several

*Equal contribution.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 109–121
July 2-7, 2026 ©2026 Association for Computational Linguistics

GRPO style Objective

$$\mathcal{J}_{\text{GRPO-L}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\sum_{i=1}^{G}\left(\sum_{t=1}^{|o_i|}\alpha_{i,t}\min\left(s_{i,t}(\theta)\,A_i,\ \text{clip}\left(s_{i,t}(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{up}\right)A_i\right)\right) - \beta R(\theta)\right] \quad (1)$$

studies report optimization issues, like systematic biases where correct answers are shorter than negative ones (Liu et al., 2025b). Other analyses propose simple stabilization techniques, including masking strategies, to improve robustness across different training regimes (Mroueh et al., 2025). In multi-objective settings, GRPO is vulnerable to reward hacking, motivating the use of normalization-based mitigations (Ichihara et al., 2025). Additional work focuses on issues that emerge at the token level: formatting tokens often dominate optimization (Simoni et al., 2025), and simple cues like sequence length can drive learning (Xin et al., 2025). Clipping mechanisms used in PPO and GRPO have also been shown to introduce systematic entropy biases (Park et al., 2025). Complementary to analyses of clipping and instability, SFPO introduces a reposition-before-update scheme to control off-policy drift induced by repeated inner updates (Wang et al., 2025).

Based on these observations, our work provides a unified analysis of why the surrogate loss can be misleading, how shared prefixes bias token-level gradients, and how optimizer dynamics interact with clipping under repeated updates.

3 Unified Formulation

In the following, we introduce a generalized surrogate objective that serves as a unified framework for a broad class of recent group-based policy optimization methods, including GRPO (R1 (Shao et al., 2024) and v3.2 (Liu et al., 2025a)), GSPO[1] (Zheng et al., 2025), GTPO (Simoni et al., 2025), DAPO (Yu et al., 2025), CPPO (Lin et al., 2025), Dr. GRPO (Liu et al., 2025b), GPG (Chu et al., 2025), CISPO (Chen et al., 2025), and GCPO (Wu and Liu, 2025). For a group of G outputs $\{o_i\}_{i=1}^{G}$ generated from the same prompt q, the advantage A_i for the i-th output is calculated by standardizing the reward r_i against the group's distribution:

$$A_i = r_i - \left(\frac{1}{G}\sum_{j=1}^{G} r_j\right) \quad (2)$$

This advantage term drives the *GRPO style objective* (Eq. 1). A_i usually determines the direction of the token-level policy updates weighting coefficients $\alpha_{i,t}$. Optimization typically involves μ gradient updates on a fixed group of samples, which progressively induces off-policy drift. To mitigate this, it is employed a token-level importance ratio

$$s_{i,t}(\theta) \propto \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})} \quad (3)$$

clipped to $[1-\varepsilon_{\text{low}},\ 1+\varepsilon_{\text{up}}]$ following PPO (Schulman et al., 2017). Finally, a regularization term $R(\theta)$, generally the KL divergence from a reference policy weighted by β, is added for training stability. As detailed in Table 1, each method represents a distinct configuration of Eq. 1 regarding the three core components: the weighting coefficients $\alpha_{i,t}$, the importance ratio $s_{i,t}(\theta)$, and the regularization term $R(\theta)$. Eq. 1 acts strictly as an optimization mechanism, not as a performance metric. Since advantages are group-centered ($\sum_i A_i = 0$), the loss value does not exclusively reflect reward improvement. Instead, the loss magnitude is dominated by nuisance factors, like importance sampling fluctuations ($s_{i,t} \neq 1$) during multi-step updates. Consequently, the surrogate loss offers no monotonic or reliable signal for policy improvement and should not be used to monitor training progress (Achiam, 2018) (see Appendix A for a formal analysis demonstrating why the loss leads to an unreliable proxy measure).

4 Biases in Token-level Gradients

When multiple answers share a starting prefix, GRPO couples the gradient updates for those early tokens. Because left-to-right autoregressive generation dictates that early parameter updates exert a global influence on the entire sequence, analyzing the aggregate gradient at these shared positions is critical to understanding the model's behavior.

[1]We report GSPO-token, as it yields the same gradients and optimization trajectory as standard GSPO.

Table 1: Instantiation of the unified objective in Eq. 1 for representative GRPO style methods. Weights α, importance ratios $s_{i,t}(\theta)$, and regularization terms $R(\theta)$ are reported for each algorithm. The definitions are: $\alpha_i^S := \frac{1}{G \cdot |o_i| \cdot \sigma(r)}$, $I := \frac{\pi_\theta}{\pi_{\theta_{old}}}$, and $\mathcal{D}_{KL} := \frac{\pi_{ref}}{\pi_\theta} - \log \frac{\pi_{ref}}{\pi_\theta} - 1$. Unless otherwise specified, dependence on $(y_{i,t} \mid x, y_{i,<t})$ is implicit.

Algorithm	$\alpha_{i,t}$	$s_{i,t}(\theta)$	$R(\theta)$	**Algorithm**	$\alpha_{i,t}$	$s_{i,t}(\theta)$	$R(\theta)$
GRPO R1	α_i^S	I	$\mathcal{D}_{KL}$	CPPO	$\alpha_i^S 1_{\{\lvert A_i \rvert > \gamma\}}$	I	$\mathcal{D}_{KL}$
GRPO v3.2	$\frac{M_{i,t}}{G\lvert o_i \rvert}$	I	$I \cdot \mathcal{D}_{KL}$	Dr GRPO	$\frac{1}{G}$	I	$\mathcal{D}_{KL}$
GSPO	α_i^S	$sg\left[\frac{\frac{\pi_\theta(y_i \lvert x_i)}{\pi_{\theta_{old}}(y_i \lvert x_i)}^{\frac{1}{\lvert o_i \rvert}}}{\pi_\theta}\right]\pi_\theta$	$\times$	GPG	$\frac{\hat{\alpha}}{F_{norm} \sum \lvert o_i \rvert}$	$\log(\pi_\theta)$	$\times$
GTPO	$\frac{\delta_i \lambda_{i,t}}{G\lvert o_i \rvert}$	I	$\frac{1}{G} \sum_i \frac{\delta_i \langle H \rangle_i}{\lvert o_i \rvert} \sum_t I \lambda_{i,t}$	CISPO	$\frac{M_{i,t}}{\sigma(r) \sum \lvert o_i \rvert}$	$sg[I] \log \pi_\theta$	$\times$
DAPO	$\frac{1}{\sigma(r) \sum \lvert o_i \rvert}$	I	$\times$	GCPO	$\frac{1}{\sigma(r) G}$	$\frac{\pi_\theta(y_i \lvert x_i)}{\pi_{\theta_{old}}(y_i \lvert x_i)}$	$\times$

Specifically, for the first k tokens that are identical across a subset of answers, the policy probability $\pi_\theta(y_{i,t} \mid x, y_{i,<t})$ remains identical for all answers within that group. Consequently, the gradient contributions derived from Equation 1 for these shared positions diverge solely based on their sequence-specific weighting terms and associated advantages. We formalize the exact form of this aggregate gradient contribution in the following proposition:

Proposition 1. *Consider a policy π_θ optimized with Eq. 1 via centered advantages (Eq. 2). For any subset of answers $\tilde{G} \subseteq G$ sharing a common prefix $y_{i,1:|k|}$, the gradient with respect to this prefix is modulated by the aggregate term $\mathcal{W}_{agg} = \sum_{i \in \tilde{G}} \omega_i A_i$, where $\omega_i = \alpha_i * s_i(\theta)$.*

Proposition 1 reveals a source of structural bias in token-level gradients, particularly when tokens are a shared prefix across all sequences. Consider the behavior under a uniform weighting scheme $\omega_i = c$. In this baseline scenario, the aggregate gradient term $\mathcal{W}_{agg}$ would vanish where prefix are shared across all sequences ($\tilde{G} = G$). Because the centered advantages (Eq. 2) sum to zero by definition ($\sum_{i \in G} A_i = 0$), the conflicting gradient signals from the group perfectly cancel each other out. However, this theoretical cancellation fails under sequence dependent weighting, such as GRPO R1's length-inverse weighting (Table 1), where the loss is averaged over tokens ($\omega_i \propto \frac{1}{|o_i|}$). Instead of canceling, the aggregate gradient becomes a weighted sum that assigns larger coefficients to shorter sequences. Consequently, a short completion with a positive advantage contributes a gradient vector to the common prefix with a significantly larger magnitude than a long completion with an equivalent positive advantage. This structural disparity creates a distinct training signal that implicitly encourages the model to favor shorter outputs for correct answers, and longer outputs for incorrect ones (Liu et al., 2025b). From an optimization perspective, the induced bias on shared prefix tokens constitutes a distinct training signal. Depending on the application, this signal can be exploited, for example, to control verbosity, or it may need to be mitigated to avoid unwanted structural preferences (Simoni et al., 2025).

5 Effects of AdamW Optimizer

We now turn our attention to the AdamW optimizer (Loshchilov and Hutter, 2017), the standard choice for GRPO training setups (Simoni et al., 2025; Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b). Analyzing AdamW is particularly relevant in this setting because, as shown in the following, it exhibits invariance to reward scaling and can drive model updates beyond the boundaries imposed by policy clipping. The AdamW update rule is formally defined as follows:

$$\theta_t = \theta_{t-1} + \xi \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \xi \lambda \theta_{t-1} \tag{4}$$

$$m_t = \frac{\beta_1 m_{t-1}}{1 - \beta_1^{t-1}} + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t} \tag{5}$$

$$v_t = \frac{\beta_2 v_{t-1}}{1 - \beta_2^{t-1}} + \frac{(1 - \beta_2)(g_t)^2}{1 - \beta_2^t} \tag{6}$$

where $g_t = \nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta)$ denotes the gradient of the GRPO style objective (the full derivation is reported in Appendix Eq. 15). Unlike standard gradient descent, the update depends not only on the current gradient, but also on exponentially smoothed estimates of its first- and second-order moments.

Reward Scaling. Despite the extensive literature emphasizing the criticality of reward scaling for stabilizing reinforcement learning algorithms (van Hasselt et al., 2016; Engstrom et al., 2020), the adaptive nature of AdamW fundamentally neutralizes this effect, in GRPO style algorithm, under specific conditions. In particular, when regularization is omitted ($\beta = 0$) - a configuration empirically shown to enhance performance in reasoning domains (Liu et al., 2025a) - and assuming a sufficiently large second-order momentum for Adam $\frac{\epsilon}{\phi\sqrt{\hat{v}_t}} \to 0$, AdamW renders the optimization trajectory strictly invariant to uniform reward scaling. If we scale a reward signal by a factor $\phi \in \mathbb{R}^+$, such that $r_i^* = \phi r_i$ (e.g., scaling a reward set of $\{0, 0.5, 1\}$ to $\{0, 50, 100\}$, unlike a non-linear shift to $\{0, 49, 100\}$), the gradient magnitude is proportionally altered. However, the adaptive normalization of AdamW perfectly compensates for this shift; the second moment estimate ($\hat{v}_t$) scales accordingly, preserving the exact update direction. We formalize this scale invariance in the following proposition:

Proposition 2. *Assume $\beta = 0$ in Eq. 1 and $\hat{v}_t \neq 0$ for the Adam update, define a scaled reward $r_i^* = \phi r_i$, with $\phi \in \mathbb{R}^+$. In the limit where the numerical stability term $\frac{\epsilon}{\phi\sqrt{\hat{v}_t}} \to 0$, the Adam update in Eq. 4 is invariant to the scaling factor ϕ.*

Proposition 2, formally derived in Appendix C, demonstrates that the intuitively expected amplification or reduction of the feedback signal is entirely canceled out by the optimizer's internal mechanics. However, this theoretical invariance breaks down in two critical scenarios, at which point reward scaling once again becomes a vital design decision. First, if a regularization term is active ($\beta \neq 0$)), scaling the reward modifies the relative strength between the reward driven gradient and the regularization penalty, forcing the optimization dynamics to explicitly depend on the scale ϕ. Second, even when $\beta = 0$, the invariance described in Proposition 2 relies on the numerical stability constant ϵ being negligible compared to $\phi\sqrt{\hat{v}_t}$. Although ϵ is typically set to a small value (10^{-8} in PyTorch implementation[2]), some reinforcement learning implementations adopt larger values such as 10^{-5} (Huang et al., 2022). In these cases, ϵ may become comparable to small gradient magnitudes, reintroducing sensitivity to reward scaling. Despite its potential impact on convergence, the value of ϵ is often omitted from reported hyperparameters.

The breakdown of this invariance introduces an unintended bias into the GRPO style objective. The policy update becomes dictated by hyperparameters that theoretically should remain entirely orthogonal to the objective function. This phenomenon conflates the true reward signal with optimizer-specific artifacts, rendering the algorithm's behavior and ultimate success highly dependent on low-level implementation details rather than the fundamental reward design itself.

Momentum Overshoot. The clipping mechanism in GRPO style objectives is designed to enforce a strict trust region when performing multiple optimization steps on the same batch (Schulman et al., 2017). However, we demonstrate below that optimizers relying on momentum, as AdamW, fundamentally undermine this constraint. Conceptually, while the momentum mechanism smooths the optimization trajectory by aggregating historical gradients, it inherently decouples the parameter update from the strict bounds of the instantaneous gradient. Consequently, when the model reaches the clipping boundary and the advantage-based gradient vanishes, the optimizer's internal dynamics do not cease. Instead, the accumulated momentum causes the model to continue along its previously established trajectory, driving updates deep into untrusted regions where the policy has diverged too far from the reference model.

We formalize this overshoot behavior specifically for AdamW in the following proposition:

Proposition 3. *Let θ_T denote a parameter state at iteration T that lies on the boundary of the clipped region. Even if the instantaneous gradient of the advantage term becomes zero for all $t > T$, the Adam update $\Delta\theta_{T+k}$ continues to move the parameters further into the clipped region.*

In this analysis, we implicitly assume the regularization penalty coefficient $\beta = 0$ to isolate the

[2] https://docs.pytorch.org/docs/stable/generated/torch.optim.AdamW.html

effects of the clipping mechanism. While $\beta \neq 0$ generally prevents the gradient from vanishing entirely, it serves as a regularization force rather than the primary learning signal, preserving the relevance of analyzing the isolated advantage gradient.

For GRPO style algorithms, this behavior induces a bias in the form of *unidirectional drift*. As a result, the model progressively deviates from the trust region until new data is generated in the subsequent iteration. However GRPO style algorithms converge even when clipping is inactive ($\mu = 1$) (Shao et al., 2024; Simoni et al., 2025; Chu et al., 2025), this implies that the clipping mechanism may be unnecessary, making its complete omission a promising direction for future work.

Note that, Proposition 3 focuses primarily on AdamW, with the full derivation provided in Appendix D, this overshoot phenomenon manifests broadly across optimizers that incorporate momentum. We provide an analogous theoretical analysis for Stochastic Gradient Descent (SGD) with momentum in Appendix E.

6 Conclusion

In this work, we established a unified formulation for Group Relative Policy Optimization and its variants, revealing disconnects between heuristics and theory. Our analysis identified distinct properties: first, that specific weighting schemes introduce structural gradient biases into shared prefixes; second, the interaction between AdamW momemntum and GRPO style objective, in absence of regularization term, makes the objective insensitive to the global reward scaling; and third, that the interaction between AdamW momentum and the objective clipping mechanisms causes parameters to overshoot trust regions, undermining the stability of multi-step updates. These findings suggest that the empirical scalability of GRPO style methods is achieved at the expense of optimization transparency, necessitating a re-evaluation of current post-training strategies to ensure rigorous alignment between surrogate objectives and desired policy outcomes.

Limitations

Our theoretical analysis relies on the assumption of standard autoregressive generation and may not fully generalize to non-standard attention mechanisms or bidirectional architectures. Additionally, while we identified the momentum-induced drift in AdamW, we did not propose a closed-form correction for the optimizer itself, leaving the development of momentum-aware clipping strategies for future work. Finally, our empirical validation of the "overshoot" phenomenon (Proposition 3) focuses on the standard GRPO style implementation and may vary under aggressive regularization regimes or alternative optimizer choices such as RMSProp or SGD.

7 Acknowledgements

This paper has been supported by the European Union's Horizon Europe research and innovation actions under grant agreement No 101215032.

References

Joshua Achiam. 2018. Spinning Up in Deep Reinforcement Learning.

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, Da Chen, Deming Ding, and 80 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *CoRR*, abs/2506.13585.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. GPG: A simple and strong reinforcement learning baseline for model reasoning. *CoRR*, abs/2504.02546.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation matters in deep policy gradients: A case study on PPO and TRPO. *CoRR*, abs/2005.12729.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. 2022. The 37 implementation details of proximal policy optimization. In *ICLR Blog Track*.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. 2025. Mo-grpo: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *arXiv preprint arXiv:2509.22047*.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. 2025. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

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. *arXiv preprint arXiv:2503.20783*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. 2025. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jaesung R Park, Junsu Kim, Gyeongman Kim, Jinyoung Jo, Sean Choi, Jaewoong Cho, and Ernest K Ryu. 2025. Clip-low increases entropy and clip-high decreases entropy in reinforcement learning of large language models. *arXiv preprint arXiv:2509.26114*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, and Andrea Saracino. 2025. Gtpo: Trajectory-based policy optimization in large language models. *arXiv preprint arXiv:2508.03772*.

Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. 2016. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4287–4295.

Ziyan Wang, Zheng Wang, Jie Fu, Xingwei Qu, Qi Cheng, Shengpu Tang, Minjia Zhang, and Xiaoming Huo. 2025. Slow-fast policy optimization: Reposition-before-update for llm reasoning. *arXiv preprint arXiv:2510.04072*.

Hao Wu and Wei Liu. 2025. GCPO: when contrast fails, go gold. *CoRR*, abs/2510.07790.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. 2025. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Inadequacy of the Surrogate Loss as a Performance Proxy

This section provides a detailed analysis of why the GRPO style surrogate objective suffers from limitations in representing a reliable performance proxy (an intermediate signal intended to estimate the underlying objective). While the objecitve is well-defined as an optimization signal, its numerical value does not admit a consistent or monotonic relationship with reward improvement, even under idealized conditions. We formalize this limitation in Proposition 4 and explicitly characterize the mechanisms that decouple the surrogate loss from true policy quality.

Proposition 4. *Consider the surrogate objective $\mathcal{J}_{GRPO\text{-}L}(\theta)$ defined in Eq. 1. Assume that importance weights are computed with respect to a fixed reference policy π_{old} sampled at the initial iteration, i.e.,*

$$s_{i,t} \propto \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{old}(o_{i,t} \mid q, o_{i,<t})}.$$

Under group-standardized advantages $\sum_{i=1}^{G} \mathcal{A}_i = 0$, the value of $\mathcal{J}_{GRPO\text{-}L}(\theta)$ is an inconsistent proxy for policy performance.

General form of the objective. Ignoring the clipping operation for analytical clarity, the GRPO style surrogate objective can be written as:

$$\mathcal{J}_{\text{GRPO-L}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\mathcal{A}_i \sum_{t=1}^{|o_i|} \omega_{i,t}\rho_{i,t}(\theta)\right) - \beta R(\pi_\theta)\right] \quad (7)$$

where: $\mathcal{A}_i = r_i - \frac{1}{G}\sum_{j=1}^{G} r_j$ is the group-centered advantage; $|o_i|$ is the length of the i-th completion; $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q,o_{i,<t})}$ is the token-level importance sampling ratio; $\omega_{i,t}$ aggregates algorithm-specific weighting choices (e.g., $\alpha_{i,t}$, length normalization, masking strategies); $\beta R(\pi_\theta)$ denotes the regularization term.

The central question addressed in this section is whether the scalar value of $\mathcal{J}_{\text{GRPO-L}}(\theta)$ can be interpreted as a meaningful indicator of training progress or policy quality. To answer this question, we analyze two scenarios: (A) the first optimization step, where the current policy coincides with the sampling policy, and (B) later iterations, where the two policies diverge.

A.1 Scenario A: First optimization step ($\rho_{i,t}(\theta) = 1$)

At the first update, the policy has not yet changed, so $\pi_\theta = \pi_{\text{old}}$ and therefore $\rho_{i,t}(\theta) = 1$ for all i, t. In this case, all importance sampling effects vanish.

We can absorb the remaining per-token design choices into a single effective weight $\tilde{\omega}_{i,t}$. The objective simplifies to:

$$\mathcal{J}_{\text{align}}(\theta) = \mathbb{E}\left[\sum_{i=1}^{G} \mathcal{A}_i\,\Omega_i - \beta R(\pi_\theta)\right] \quad (8)$$

where

$$\Omega_i = \sum_{t=1}^{|o_i|} \tilde{\omega}_{i,t}$$

is the cumulative weight assigned to trajectory i.

This formulation makes explicit that the surrogate objective depends only on the interaction between advantages $\mathcal{A}_i$ and cumulative weights Ω_i. We now examine three representative weighting regimes.

Case 1: Length-normalized weights

Many GRPO style methods normalize updates by sequence length, using weights of the form $\tilde{\omega}_{i,t} = \frac{C}{|o_i|}$. In this case,

$$\Omega_i = \sum_{t=1}^{|o_i|} \frac{C}{|o_i|} = C,$$

which is constant across all trajectories. Substituting into Eq. 8 yields:

$$\mathcal{J}_{\text{align}}(\theta) = \mathbb{E}\left[C \underbrace{\sum_{i=1}^{G} \mathcal{A}_i}_{=0} - \beta R(\pi_\theta)\right] = -\beta\,\mathbb{E}[R(\pi_\theta)]. \quad (9)$$

Thus, the entire reward-driven component of the objective cancels out. The surrogate loss is fully dominated by the regularization term and contains no information about relative reward improvement. In this regime, the loss value is fundamentally uninformative as a measure of policy performance.

Case 2: Constant token-wise weights

If weights are constant per token, $\tilde{\omega}_{i,t} = C$ (e.g., Dr. GRPO), then the cumulative weight scales linearly with output length:

$$\Omega_i = C\,|o_i|.$$

The objective becomes:

$$\mathcal{J}_{\text{align}}(\theta) = \mathbb{E}\left[C\sum_{i=1}^{G}\mathcal{A}_i|o_i| - \beta R(\pi_\theta)\right] \quad (10)$$

In this case, the loss no longer cancels, but its sign and magnitude reflect whether positively advantaged completions tend to be longer or shorter than negatively advantaged ones. The objective therefore acts as a proxy for sequence length statistics, not for reward maximization or task correctness.

Case 3: General parametric weighting

More complex methods (e.g., GTPO, CPPO) define $\tilde{\omega}_{i,t}$ as a non-trivial function of i and t. Here, the reward-weighted sum does not vanish, but instead satisfies:

$$\mathcal{J}_{\text{align}}(\theta) \propto \mathrm{Cov}(\mathcal{A}, \Omega) \quad (11)$$

Although the loss is non-zero, its value is entirely determined by the interaction between the advantage distribution and the chosen weighting scheme. Unless the weights are explicitly designed to encode task-relevant structure, the loss magnitude is an artifact of hyperparameterization, not a measure of learning progress.

Conclusion of Scenario A. Across all weighting regimes, the surrogate loss fails to maintain a consistent or monotonic relationship with true policy quality. Its numerical value is therefore an unreliable indicator of performance, even in the absence of importance sampling effects.

A.2 Scenario B: Multiple optimization steps ($\rho_{i,t}(\theta) \neq 1$)

After the first update, π_θ diverges from π_{old} and importance sampling ratios $\rho_{i,t}(\theta) \neq 1$ appear. While this breaks the exact cancellations observed in Scenario A, it does not restore interpretability.

The loss value now depends on two independent sources of variability: the structural biases induced by the weighting scheme $\tilde{\omega}_{i,t}$ and stochastic fluctuations of the importance ratios $\rho_{i,t}(\theta)$.

As a result, changes in the surrogate loss primarily reflect off-policy drift and optimizer dynamics, rather than genuine reward improvement. A decreasing loss does not imply better policies, nor does a stable loss indicate convergence.

B Derivation of the Gradient for $\mathcal{J}_{\textbf{GRPO-L}}(\theta)$

This appendix derives the gradient of the GRPO style surrogate objective and makes explicit the token-level structure that later induces shared-prefix biases. For clarity, we derive the gradient in the region where the *unclipped* term is active; when the clipped branch is active, the gradient through the advantage term is zero (up to boundary measure-zero cases).

B.1 Gradient of the GRPO style objective

Recall the GRPO style objective in Eq. 1.

$$\mathcal{J}_{\text{GRPO-L}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\alpha_{i,t}\,\min\Big(s_{i,t}(\theta)\,A_i,\right.$$
$$\left.\text{clip}(s_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{up}})\,A_i\Big) - \beta R(\theta)\right]$$

We define the token-level importance ratio as

$$s_{i,t}(\theta) := k_{i,t}\cdot\pi_\theta(y_{i,t} \mid x, y_{i,<t}) \quad (12)$$

Here, $k_{i,t}$ is a term that depends on i and t, but is independent of θ. We now apply the gradient and move ∇_θ inside expectation and sums. By linearity,

$$\nabla_\theta\mathcal{J}_{\text{GRPO-L}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\alpha_{i,t}\,\nabla_\theta\,\min(\cdot)\right.$$
$$\left. -\ \beta\nabla_\theta R(\theta)\right] \quad (13)$$

The gradient depends on the active branch. When the unclipped term is active,

$$\nabla_\theta\min(\cdot) = \nabla_\theta(s_{i,t}(\theta)A_i) = A_i\,\nabla_\theta s_{i,t}(\theta)$$

while when the clipped term is active, its value is constant w.r.t. θ in the interior of the clipped region, hence the advantage gradient is zero (ignoring boundary non-differentiability).

Since $\pi_{\theta_{\text{old}}}$ does not depend on current θ,

$$\begin{aligned}\nabla_\theta s_{i,t}(\theta) &= \nabla_\theta\left(k_{i,t}\cdot\pi_\theta(y_{i,t} \mid x, y_{i,<t})\right)\\ &= k_{i,t}\cdot\nabla_\theta\pi_\theta(y_{i,t} \mid x, y_{i,<t})\end{aligned}$$

Using the log-derivative trick, $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$, we get

$$\begin{aligned}\nabla_\theta s_{i,t}(\theta) &= [k_{i,t} \cdot \pi_\theta(y_{i,t} \mid x, y_{i,<t})] \cdot \\ &\cdot \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \\ &= s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\,. \end{aligned} \tag{14}$$

Substituting Eq. 14 into Eq. 13 yields:

$$\nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta) = \mathbb{E}_{q,\{o_i\}} \left[\sum_{i=1}^{G} A_i \sum_{t=1}^{|o_i|} \alpha_{i,t}\, s_{i,t}(\theta) \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) - \beta \nabla_\theta R(\theta) \right] \tag{15}$$

B.2 First token issues

We now isolate the gradient contribution on tokens that belong to a prefix shared by multiple completions in the same group. Let $|k|$ be the length of a prefix shared by a subset of $\tilde{G} \leq G$ completions.

Prefix/deviation decomposition. Splitting the inner sum over time gives:

$$\begin{aligned}\nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta) = \\ \mathbb{E}_{q,\{o_i\}} \Bigg[\sum_{i=1}^{G} A_i \Bigg(\sum_{t=1}^{|k|} \alpha_{i,t}\, s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \\ + \sum_{t=|k|+1}^{|o_i|} \alpha_{i,t}\, s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \Bigg) \\ - \beta \nabla_\theta R(\theta) \Bigg] \end{aligned} \tag{16}$$

For all $t \leq |k|$ and all completions i in the subset that shares the prefix, both $y_{i,t}$ and its context $y_{i,<t}$ are identical. Hence,

$$\nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) = \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})\,, \quad \forall\, i \in \{1, \ldots, \tilde{G}\},\ t \leq |k| \tag{17}$$

Define the aggregated coefficient

$$\omega_{i,t} \ := \ \alpha_{i,t}\, s_{i,t}(\theta) \tag{18}$$

Then the gradient restricted to the shared prefix (denoted $\nabla_\theta \tilde{\mathcal{J}}_{\text{GRPO-L}}(\theta)$) becomes:

$$\nabla_\theta \tilde{\mathcal{J}}_{\text{GRPO-L}}(\theta) = \sum_{t=1}^{|k|} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \sum_{i=1}^{\tilde{G}} A_i\, \omega_{i,t}$$

In the following, when it does not change the qualitative argument, we suppress the explicit dependence on t and write ω_i for simplicity.

Case 1: Constant token-wise weights

Assume uniform weights over completions: $\omega_i = C$. Then:

$$\nabla_\theta \tilde{\mathcal{J}}_{\text{GRPO-L}}(\theta) = C \sum_{t=1}^{|k|} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \sum_{i=1}^{\tilde{G}} A_i$$

Since $A_i = R_i - \bar{R}$ is group-centered, the behavior depends on which completions share the prefix: (i) if the prefix occurs only in $A_i > 0$ completions, it is reinforced; (ii) in mixed regimes, the net update is the algebraic sum; (iii) if the prefix is ubiquitous across all G completions, $\sum_{i=1}^{G} A_i = 0$ and the update cancels.

Case 2: Non-uniform weighting over i

If weights depend on the completion index, $\omega_i \neq$ const, then:

$$\nabla_\theta \tilde{\mathcal{J}}_{\text{GRPO-L}}(\theta) = \sum_{t=1}^{|k|} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \sum_{i=1}^{\tilde{G}} \omega_i\, A_i$$

In this regime, cancellations generally do not hold: shared-prefix tokens can receive a net update dominated by the completions with larger ω_i, which can induce systematic biases unrelated to semantic quality (e.g., length preferences when ω_i depends on $|o_i|$).

C Reward magnitude and Adam

This section analyzes how scaling the reward signal affects GRPO style training when optimization is performed with Adam/AdamW. We first show that group-centered advantages scale linearly with the reward. We then propagate this scaling through (i) the GRPO style gradient, (ii) Adam's first and second moments, and (iii) the final parameter update. The key takeaway is that, when the regularization term is absent (or negligible), Adam is approximately invariant to global reward scaling.

C.1 Scaling properties of the advantage term

We start by characterizing how the GRPO style advantage behaves under a linear transformation of the reward.

Proposition 5 (Advantage scaling)**.** *Let the group-centered advantage be* $A_i = R_i - \frac{1}{G}\sum_{j=1}^{G} R_j$. *If rewards are scaled by a constant* $\phi \in \mathbb{R}$, $R_i^* = \phi R_i$, *then the transformed advantage satisfies*

$$A_i^* = \phi A_i \tag{19}$$

Proof. First compute the transformed group baseline:

$$\bar{R}^* = \frac{1}{G}\sum_{j=1}^{G} R_j^* = \frac{1}{G}\sum_{j=1}^{G} \phi R_j = \phi\left(\frac{1}{G}\sum_{j=1}^{G} R_j\right) = \phi\bar{R}$$

Then the transformed advantage is

$$A_i^* = R_i^* - \bar{R}^* = \phi R_i - \phi\bar{R} = \phi(R_i - \bar{R}) = \phi A_i \tag{20}$$

□

C.2 Gradient decomposition and scaling properties

We decompose the GRPO style gradient into an advantage-driven term and a regularization term. Using Eq. 15, define:

$$g_A(\theta) := \sum_{i=1}^{G} A_i \sum_{t=1}^{|o_i|} \alpha_{i,t}\, s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \tag{21}$$

$$g_R(\theta) := \beta \nabla_\theta R(\theta) \tag{22}$$

so that the total gradient is $g(\theta) = g_A(\theta) - g_R(\theta)$.

By Proposition 5, scaling the rewards by ϕ implies $A_i^* = \phi A_i$. Therefore the advantage-driven component scales linearly:

$$\begin{aligned} g_A^*(\theta) &= \sum_{i=1}^{G} A_i^* \sum_{t=1}^{|o_i|} \alpha_{i,t}\, s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \\ &= \phi \sum_{i=1}^{G} A_i \sum_{t=1}^{|o_i|} \alpha_{i,t}\, s_{i,t}(\theta)\, \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) = \\ &= \phi\, g_A(\theta) \end{aligned} \tag{23}$$

Conversely, $g_R(\theta)$ is unaffected by reward scaling because it depends only on the regularizer and β.

C.3 Adam moments under gradient scaling

We now study how Adam's moments scale when the gradient is multiplied by ϕ. Let g_t be the gradient at optimization step t, and assume

$$g_t^* = \phi g_t \qquad \text{for all } t \geq 1$$

Adam maintains exponential moving averages:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned}$$

with bias-corrected versions

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

Proposition 6 (Moment scaling)**.** *If* $g_t^* = \phi g_t$ *for all t, then for all* $t \geq 1$*:*

$$m_t^* = \phi m_t, \qquad v_t^* = \phi^2 v_t, \tag{24}$$

and equivalently $\hat{m}_t^* = \phi \hat{m}_t$ *and* $\hat{v}_t^* = \phi^2 \hat{v}_t$.

Proof. We prove by induction.

Base case ($t = 1$)*.* With $m_0 = v_0 = 0$,

$$\begin{aligned} m_1^* &= (1-\beta_1) g_1^* = (1-\beta_1)\phi g_1 = \phi m_1 \\ v_1^* &= (1-\beta_2)(g_1^*)^2 = (1-\beta_2)\phi^2 g_1^2 = \phi^2 v_1 \end{aligned}$$

Inductive step. Assume $m_{t-1}^* = \phi m_{t-1}$ and $v_{t-1}^* = \phi^2 v_{t-1}$. Then

$$\begin{aligned} m_t^* &= \beta_1 m_{t-1}^* + (1-\beta_1) g_t^* \\ &= \beta_1(\phi m_{t-1}) + (1-\beta_1)(\phi g_t) = \\ &= \phi\big(\beta_1 m_{t-1} + (1-\beta_1) g_t\big) = \phi m_t \end{aligned} \tag{25}$$

$$\begin{aligned} v_t^* &= \beta_2 v_{t-1}^* + (1-\beta_2)(g_t^*)^2 \\ &= \beta_2(\phi^2 v_{t-1}) + (1-\beta_2)\phi^2 g_t^2 = \\ &= \phi^2\big(\beta_2 v_{t-1} + (1-\beta_2) g_t^2\big) = \phi^2 v_t \end{aligned} \tag{26}$$

Bias correction divides by $(1-\beta_1^t)$ and $(1-\beta_2^t)$, hence it preserves the same scaling. □

C.4 Adam update invariance under reward scaling

We now analyze when Adam becomes invariant to global reward scaling. Assume the regularization term is absent or negligible, i.e., $g_R(\theta) \approx 0$, and the second momentum $\hat{v}_t$ is different from 0. Then g_t is driven only by the advantage term and scales as $g_t^* = \phi g_t$.

AdamW updates parameters as

$$\Delta\theta_t = -\xi \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \; - \; \xi\lambda\theta_{t-1}$$

where ξ is the learning rate, ϵ the numerical stabilizer, and λ the weight decay coefficient.

Using Proposition 6, we have $\hat{m}_t^* = \phi\hat{m}_t$ and $\hat{v}_t^* = \phi^2\hat{v}_t$, hence

$$\begin{aligned}\Delta\theta_t^* &= -\xi \frac{\hat{m}_t^*}{\sqrt{\hat{v}_t^*} + \epsilon} \; - \; \xi\lambda\theta_{t-1} \\ &= -\xi \frac{\phi\hat{m}_t}{\sqrt{\phi^2\hat{v}_t} + \epsilon} \; - \; \xi\lambda\theta_{t-1} = \\ &= -\xi \frac{\phi\hat{m}_t}{\phi\sqrt{\hat{v}_t} + \epsilon} \; - \; \xi\lambda\theta_{t-1}\end{aligned}$$

Factor ϕ out of the denominator (assuming $\phi > 0$):

$$\Delta\theta_t^* = -\xi \frac{\hat{m}_t}{\sqrt{\hat{v}_t}\left(1 + \frac{\epsilon}{\phi\sqrt{\hat{v}_t}}\right)} \; - \; \xi\lambda\theta_{t-1}$$

Therefore, in the regime where $\epsilon \ll \phi\sqrt{\hat{v}_t}$, we obtain the approximate invariance:

$$\lim_{\frac{\epsilon}{\phi\sqrt{\hat{v}_t}} \to 0} \Delta\theta_t^* = -\xi \frac{\hat{m}_t}{\sqrt{\hat{v}_t}} \; - \; \xi\lambda\theta_{t-1} = \Delta\theta_t. \quad (27)$$

This shows that when the optimization signal is purely reward-driven, Adam's adaptive normalization cancels global reward scaling. However, if a regularization term is present ($\beta \neq 0$), then the total gradient becomes $g_t = g_{A,t} - g_{R,t}$ and scaling the rewards changes the relative strength between the two components, breaking invariance.

D Adam overly moves your model

This section analyzes the interaction between GRPO style clipping and Adam's momentum. The key point is that clipping can zero out the instantaneous advantage gradient once the policy ratio exits the trust region, but Adam's first-moment accumulator can continue to move parameters in the same direction, causing overshoot into the clipped region.

D.1 Gradient discontinuity induced by clipping

Let $\mathcal{R}_{\text{out}}$ denote the subset of parameter space where the importance ratio exceeds the clip bounds in the direction favored by A_i (e.g., $s_{i,t} > 1 + \epsilon_{\text{up}}$ with $A_i > 0$, or $s_{i,t} < 1 - \epsilon_{\text{low}}$ with $A_i < 0$). Inside this region, the advantage term is clipped and its gradient is zero.

Equivalently, the gradient takes the piecewise form:

$$\nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta) = \begin{cases} \nabla_\theta \mathcal{J}_{\text{ADV}}(\theta) - \beta\nabla_\theta R(\theta), & \theta \notin \mathcal{R}_{\text{out}}, \\ -\beta\nabla_\theta R(\theta), & \theta \in \mathcal{R}_{\text{out}}. \end{cases} \quad (28)$$

Intuitively, if β is small, entering $\mathcal{R}_{\text{out}}$ should dramatically reduce the gradient magnitude and stop motion in that direction. The next subsection shows why Adam can violate this intuition.

D.2 Proposition: momentum overshoot

Proposition 7 (Momentum overshoot)**.** *Let θ_T be a parameter iterate lying on the boundary of the clipped region. Assume that for $t > T$ the advantage gradient becomes zero due to clipping, i.e., $g_{A,t} = 0$. Then, even if the instantaneous advantage gradient remains zero for subsequent inner-loop steps, Adam can continue to update parameters in the same direction, pushing the iterate deeper into $\mathcal{R}_{out}$.*

Proof. For steps $t < T$, assume the advantage gradient points consistently toward the upper clip boundary, i.e., $g_{A,t}$ has a persistent sign that increases $s_{i,t}(\theta)$. Adam accumulates these gradients in the first moment:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$$

At $t = T$, the iterate enters $\mathcal{R}_{\text{out}}$ and the advantage gradient is suppressed: $g_{A,T} = 0$ (and similarly for all $t > T$). Neglecting regularization for exposition, the new first-moment update becomes:

$$m_T = \beta_1 m_{T-1} + (1 - \beta_1) \underbrace{g_T}_{=0} = \beta_1 m_{T-1}$$

$$m_{T+k} = \beta_1^{k+1} m_{T-1}, \qquad k \geq 0.$$

Thus, even though the instantaneous gradient is zero, m_{T+k} remains non-zero for many steps when β_1 is close to one (e.g., $\beta_1 = 0.9$). Since the Adam update depends on $\hat{m}_t$, the parameter update remains non-zero:

$$\Delta\theta_{T+k} = -\xi \frac{\hat{m}_{T+k}}{\sqrt{\hat{v}_{T+k}} + \epsilon} \; - \; \xi\lambda\theta_{T+k-1} \quad (29)$$

Therefore, the iterate continues to move in the direction encoded by the pre-clipping momentum, pushing the ratio further beyond the clip boundary. Clipping acts as a "hard stop" for the instantaneous gradient, but Adam's momentum makes it a "soft brake" for the parameter trajectory. □

Practical implication. When multiple optimization steps are applied on the same sampled group (inner loop), the overshoot effect becomes more pronounced: the policy can drift further into the clipped region before new samples are generated, weakening the intended trust-region interpretation of clipping.

Quantifying overshoot (Adam canonical form). We now quantify how large the Adam step can remain *after* entering the clipped region, even when the instantaneous advantage gradient becomes zero.

Assume that at step T the iterate enters $\mathcal{R}_{\text{out}}$, so that the advantage gradient is suppressed for all subsequent inner-loop steps, i.e., $g_{A,t} = 0$ for $t \geq T$. For clarity, we first ignore weight decay and regularization and focus on the Adam preconditioned direction $\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$.

Under $g_T = 0$, Adam moment recurrences reduce to pure exponential decay:

$$m_T = \beta_1 m_{T-1} + (1-\beta_1)\underbrace{g_T}_{0} = \beta_1 m_{T-1}$$

$$v_T = \beta_2 v_{T-1} + (1-\beta_2)\underbrace{g_T^2}_{0} = \beta_2 v_{T-1}$$

Using bias correction,

$$\hat{m}_T = \frac{m_T}{1-\beta_1^T} = \frac{\beta_1 m_{T-1}}{1-\beta_1^T}, \quad \hat{v}_T = \frac{v_T}{1-\beta_2^T} = \frac{\beta_2 v_{T-1}}{1-\beta_2^T} \tag{30}$$

Similarly,

$$\hat{m}_{T-1} = \frac{m_{T-1}}{1-\beta_1^{T-1}}, \qquad \hat{v}_{T-1} = \frac{v_{T-1}}{1-\beta_2^{T-1}}$$

A C_T-like coefficient. Define the ratio between the (magnitude of the) preconditioned update immediately after clipping and the one immediately before clipping:

$$C_T := \frac{\left\| \dfrac{\hat{m}_T}{\sqrt{\hat{v}_T} + \epsilon} \right\|}{\left\| \dfrac{\hat{m}_{T-1}}{\sqrt{\hat{v}_{T-1}} + \epsilon} \right\|}. \tag{31}$$

In the common regime where $\epsilon \ll \sqrt{\hat{v}_{T-1}}$ and $\epsilon \ll \sqrt{\hat{v}_T}$, we can approximate

$$C_T \approx \frac{\left\| \dfrac{\hat{m}_T}{\sqrt{\hat{v}_T}} \right\|}{\left\| \dfrac{\hat{m}_{T-1}}{\sqrt{\hat{v}_{T-1}}} \right\|} = \frac{\|\hat{m}_T\|}{\|\hat{m}_{T-1}\|} \cdot \frac{\sqrt{\hat{v}_{T-1}}}{\sqrt{\hat{v}_T}}$$

Substituting Eq. 30 yields

$$C_T \approx \left[\beta_1 \frac{1-\beta_1^{T-1}}{1-\beta_1^T}\right] \cdot \left[\sqrt{\frac{1}{\beta_2}\frac{1-\beta_2^T}{1-\beta_2^{T-1}}}\right]. \tag{32}$$

This coefficient captures how much "inertia" remains *exactly at the first step* after the advantage gradient is clipped out. In the limit $T \to \infty$, bias-correction saturates and we obtain:

$$\lim_{T\to\infty} C_T = \frac{\beta_1}{\sqrt{\beta_2}}. \tag{33}$$

For typical values $(\beta_1, \beta_2) = (0.9, 0.95)$,

$$\frac{\beta_1}{\sqrt{\beta_2}} = \frac{0.9}{\sqrt{0.95}} \approx 0.923$$

meaning that the *first* post-clipping step can still be on the order of $\sim 92\%$ of the previous preconditioned step once training is past the early bias-correction transient.

Overshoot across k inner-loop steps. The same reasoning extends to subsequent clipped steps. For $k \geq 0$, when $g_{T+k} = 0$ we have

$$m_{T+k} = \beta_1^{k+1} m_{T-1}, \quad v_{T+k} = \beta_2^{k+1} v_{T-1}. \tag{34}$$

Define an extension of Eq. 31:

$$C_{T,k} := \frac{\left\| \dfrac{\hat{m}_{T+k}}{\sqrt{\hat{v}_{T+k}} + \epsilon} \right\|}{\left\| \dfrac{\hat{m}_{T-1}}{\sqrt{\hat{v}_{T-1}} + \epsilon} \right\|}. \tag{35}$$

Again for ϵ negligible, we obtain the closed form

$$C_{T,k} \approx \left[\beta_1^{k+1} \frac{1-\beta_1^{T-1}}{1-\beta_1^{T+k}}\right] \cdot \left[\sqrt{\frac{1}{\beta_2^{k+1}}\frac{1-\beta_2^{T+k}}{1-\beta_2^{T-1}}}\right]. \tag{36}$$

For large T (where bias correction is stable), Eq. 36 simplifies to an exponential decay:

$$C_{T,k} \approx \left(\frac{\beta_1}{\sqrt{\beta_2}}\right)^{k+1}. \tag{37}$$

With $(\beta_1, \beta_2) = (0.9, 0.95)$ this gives $C_{T,4} \approx 0.923^5 \approx 0.66$, i.e., even after *five* clipped inner-loop steps the update magnitude can still be around $\sim 66\%$ of the pre-clipping step, which explains why the policy can drift substantially deeper into the clipped region before new samples are generated.

Effect of ϵ. When ϵ is not negligible (e.g., for very small $\hat{v}_t$), the ratios in Eq. 32–36 are further modulated by

$$\frac{\sqrt{\hat{v}_{T-1}} + \epsilon}{\sqrt{\hat{v}_{T+k}} + \epsilon}, \tag{38}$$

which can either dampen or amplify the residual step depending on the scale of $\hat{v}_t$ relative to ϵ.

E Overshoot in SGD with momentum

This section extends the analysis to Stochastic Gradient Descent (SGD) with momentum, demonstrating that a similar overshoot phenomenon occurs when the gradient is zeroed out by clipping.

Momentum decay and parameter update. Consider an SGD optimizer with a momentum buffer v_t, a momentum decay coefficient μ, and learning parameters λ and ν. The update rules for the momentum and the parameters are:

$$\begin{aligned} v_t &= \mu v_{t-1} + \nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta_t) \\ \theta_{t+1} &= \theta_t - \lambda v_t - \nu \nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta_t) \end{aligned} \tag{39}$$

Assume that at step T, the parameter iterate enters the clipped region $\mathcal{R}_{\text{out}}$, causing the instantaneous advantage gradient to become zero. Neglecting weight decay and regularization for clarity, the gradient vanishes: $\nabla_\theta \mathcal{J}_{\text{GRPO-L}}(\theta_{T+k}) = 0$ for all $k \geq 0$.

Without new gradient information, the momentum accumulator transitions to a pure exponential decay:

$$v_{T+k} = \mu^{k+1} v_{T-1}$$

Consequently, the parameter update $\Delta\theta_{T+k}$ for any step $k \geq 0$ inside the clipped region simplifies to rely entirely on the residual momentum:

$$\Delta\theta_{T+k} = -\lambda v_{T+k} = -\lambda \mu^{k+1} v_{T-1}$$

Quantifying the SGD overshoot. To compare this behavior with the Adam analysis, we define $C_{T,k}^{\text{SGD}}$ to represent the ratio of the update magnitude k steps after clipping against the momentum-driven portion of the step immediately before clipping:

$$C_{T,k}^{\text{SGD}} := \frac{\|\Delta\theta_{T+k}\|}{\|\lambda v_{T-1}\|} = \tag{40}$$

$$= \frac{\left\|\lambda \mu^{k+1} v_{T-1}\right\|}{\|\lambda v_{T-1}\|} = \mu^{k+1} \tag{41}$$

Practical implication. Unlike Adam, which depends on a preconditioned direction and transient bias-correction terms, SGD with momentum experiences a straightforward exponential decay dictated strictly by μ.

For a standard momentum value of $\mu = 0.9$, the first step entirely inside the clipped region still retains 90% of the previous momentum magnitude. After five consecutive clipped inner-loop steps ($k = 4$), the coefficient is still roughly 59% ($\mu^5 \approx 0.59$). This confirms that standard SGD momentum also acts as a "soft brake," carrying the policy deeper into $\mathcal{R}_{\text{out}}$ long after the instantaneous gradient has signaled a hard stop.