DGPO: DECOUPLED GRADIENT POLICY OPTIMIZATION FOR RLVR IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has catalyzed a leap in Large Language Model (LLM) reasoning, yet its optimization dynamics remain fragile. Standard algorithms like GRPO enforce stability via "hard clipping", which inadvertently stifles exploration by discarding gradients of tokens outside the trust region. While recent "soft clipping" methods attempt to recover these gradients, they suffer from a critical challenge: relying on *log-probability gradient* ($\nabla_\theta \log \pi_\theta$) yields divergent weights as probabilities vanish, destabilizing LLM training. We rethink this convention by establishing *probability gradient* ($\nabla_\theta \pi_\theta$) as the superior optimization primitive. Accordingly, we propose **D**ecoupled **G**radient **P**olicy **O**ptimization (DGPO), which employs a decoupled decay mechanism based on importance sampling ratios. By applying asymmetric, continuous decay to boundary tokens, DGPO resolves the conflict between stability and sustained exploration. Extensive experiments across DeepSeek-R1-Distill-Qwen series models (1.5B/7B/14B) demonstrate that DGPO consistently outperforms strong baselines on various mathematical benchmarks, offering a robust solution for RLVR.

1 INTRODUCTION

Reinforcement Learning (RL) has become a cornerstone for aligning LLMs, particularly in reasoning domains where Verifiable Rewards (RLVR) provide ground-truth feedback Ouyang et al. (2022); Lightman et al. (2023); Shao et al. (2024); Guo et al. (2025). However, optimizing RLVR is challenging due to the conflict between exploration and stability.

Standard algorithms like PPO Schulman et al. (2017) and GRPO Shao et al. (2024) enforce trust regions via "hard clipping", which inadvertently zeros out gradients for outlier tokens, stifling exploration and causing entropy collapse. While recent "soft clipping" methods Chen et al. (2025); Su et al. (2025a;b) attempt to preserve these gradients, they predominantly operate on *log-probability gradients* ($\nabla_\theta \log \pi_\theta$). We identify a critical flaw in this convention: as token probability approaches zero, the gradient weight grows divergently, causing catastrophic instability at the left boundary (Figure 1).

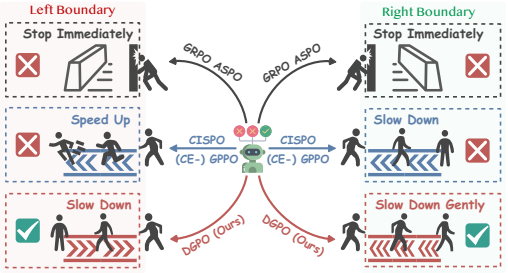
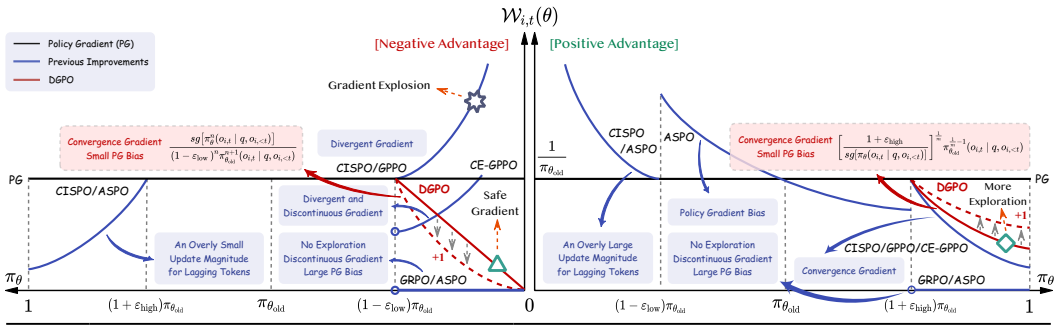


Figure 1: DGPO vs. Baselines. DGPO employs "Slow Down" for stability and "Slow Down Gently" for exploration, avoiding the gradient divergence of soft-clipping methods.

We propose **D**ecoupled **G**radient **P**olicy **O**ptimization (DGPO), shifting the optimization primitive from log-probability to *probability* ($\nabla_\theta \pi_\theta$). DGPO replaces hard clipping with a decoupled decay mechanism applied to the probability gradient weight. It ensures convergent weights for stability while sustaining exploration for high-reward tokens. Our contributions are: (1) establishing $\nabla_\theta \pi_\theta$ as the superior primitive for LLMs; (2) proposing DGPO to reconcile exploration and stability via adaptive decay; (3) demonstrating superior performance and scalability across 1.5B, 7B, and 14B models.



Policy Optimization Algorithm	Left Boundary			Right Boundary			Gradient Estimator Bias					Lagging Token Update Magnitude	
	Exploration Capability	Gradient Convergence	Gradient Continuity	Exploration Capability	Gradient Convergence	Gradient Continuity	In Boundary	Left Boundary	Right Boundary	Reverse Left Boundary	Reverse Right Boundary	Reverse Left Boundary	Reverse Right Boundary
GRPO	✗	✓	✗	✗	✓	✗	Unbiased	Large	Large	Unbiased	Unbiased	Normal	Normal
CISPO	✓	✗	✓	✓	✓	✓	Unbiased	Small	Moderate	Biased	Biased	Large	Small
GPPO	✓	✗	✓	✓	✓	✓	Unbiased	Small	Moderate	Unbiased	Unbiased	Normal	Normal
CE-GPPO	✓	✗	✗	✓	✓	✓	Unbiased	Moderate	Moderate	Unbiased	Unbiased	Normal	Normal
ASPO	✗	✓	✗	✗	✓	✗	Biased	Large	Large	Biased	Biased	Large	Small
DGPO	✓	✓	✓	✓	✓	✓	Unbiased	Small	Small	Unbiased	Unbiased	Normal	Normal

Figure 2: Comparative analysis of gradient dynamics. We systematically contrast DGPO with the standard GRPO, prior "soft clipping" enhancements (CISPO, GPPO, and CE-GPPO), and importance sampling improvements (ASPO). The visualization highlights the theoretical properties regarding the exploration capability of clipped tokens and the alignment with the true policy gradient, demonstrating DGPO's superior stability and gradient consistency.

2 Preliminary

Problem Definition. Given a query q, a policy π_θ generates response o. A group of outputs $\{o_i\}_{i=1}^{G}$ is sampled from $\pi_{\theta_{\text{old}}}$. The advantage $\hat{A}_i$ is computed by normalizing reward Shao et al. (2024).

Unified Gradient Formulation. Most RLVR algorithms can be unified under a single gradient estimator. Let $w_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ be the Importance Sampling (IS) ratio:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}\frac{1}{\sum|o_i|}\sum_{i,t}\mathcal{F}_{i,t}(\theta)\hat{A}_i\nabla_\theta \log \pi_\theta(o_{i,t}|q,o_{i,<t}). \tag{1}$$

Here, $\mathcal{F}_{i,t}(\theta)$ defines the clipping strategy based on five regions (Table 1). Prior methods like GRPO, CISPO, GPPO, CE-GPPO, and ASPO are specific instantiations of $\mathcal{F}_{i,t}(\theta)$ Chen et al. (2025); Su et al. (2025a;b); Wang et al. (2025) (detailed in Appendix B).

Table 1: Definition of clipping regions based on IS ratio $w_{i,t}(\theta)$ and advantage $\hat{A}_i$.

Case	Abbrev.	Condition
Left Boundary (Low ratio, Neg. adv.)	LN	$w_{i,t}(\theta) < 1-\varepsilon_{\text{low}} \wedge \hat{A}_i < 0$
Right Boundary (High ratio, Pos. adv.)	HP	$w_{i,t}(\theta) > 1+\varepsilon_{\text{high}} \wedge \hat{A}_i > 0$
Reverse Left (Low ratio, Pos. adv.)	LP	$w_{i,t}(\theta) < 1-\varepsilon_{\text{low}} \wedge \hat{A}_i > 0$
Reverse Right (High ratio, Neg. adv.)	HN	$w_{i,t}(\theta) > 1+\varepsilon_{\text{high}} \wedge \hat{A}_i < 0$
In-Boundary	M	Otherwise

3 Methodology

3.1 Shifting Focus: From $\log \pi$ to π

Classical policy gradient methods Williams (1992) prioritize $\nabla_\theta \log \pi_\theta$. We argue that *probability gradient* is superior for LLMs. Comparing Supervised Fine-Tuning (SFT) and RL objectives:

$$\nabla_\theta \mathcal{J}_{\text{SFT}} = \mathbb{E}\sum \nabla_\theta \log \pi_\theta, \qquad \nabla_\theta \mathcal{J}_{\text{RL}} = \mathbb{E}\sum \nabla_\theta \pi_\theta. \tag{2}$$

Table 2: Comparison results of different methods on various benchmarks. Avg@32 (%) and Pass@32 (%) are abbreviated as A@32 and P@32. The best results are **bold**, and the second-best results are underlined, respectively.

Method	AIME24		AIME25		AMC23		MATH500		Minerva		Olympiad		Avg.	
	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32
DEEPSEEK-R1-DISTILL-QWEN-1.5B														
GRPO	33.2	71.8	27.7	49.9	79.5	94.8	77.6	90.8	26.1	48.8	46.3	64.7	48.4	70.1
CISPO	34.8	69.1	25.8	53.3	76.9	94.9	76.8	**91.8**	26.5	**54.2**	45.8	65.8	47.8	71.5
GPPO	29.6	60.5	23.5	51.9	73.5	94.1	76.3	89.1	26.6	50.0	43.9	64.2	45.6	68.3
CE-GPPO	35.1	70.2	27.7	55.1	82.5	**95.0**	76.7	90.2	27.8	50.5	45.6	63.1	49.2	70.7
ASPO	36.4	73.2	28.3	51.5	83.1	94.7	74.6	90.5	26.0	49.8	44.9	63.7	48.9	70.6
Ours	**43.3**	**79.3**	**32.8**	**56.1**	**86.0**	**95.0**	**77.9**	91.0	**28.2**	50.4	**48.0**	**66.4**	**52.7**	**73.0**
DEEPSEEK-R1-DISTILL-QWEN-7B														
GRPO	48.2	**82.5**	37.4	60.5	88.1	96.6	84.8	92.4	37.4	57.2	57.2	73.9	58.9	77.2
CISPO	51.6	76.6	38.2	65.4	**90.6**	96.6	82.1	91.6	38.7	56.5	54.3	69.9	59.3	76.1
GPPO	43.1	72.5	31.7	62.5	85.6	94.9	83.2	**95.4**	33.1	**59.3**	53.2	**74.3**	55.0	76.5
CE-GPPO	48.7	76.9	36.4	60.4	90.5	95.0	84.3	93.3	39.0	55.4	54.9	72.5	59.0	75.6
ASPO	51.8	79.6	37.1	54.1	90.0	**97.2**	83.8	94.9	37.0	59.2	54.1	72.8	59.0	76.3
Ours	**55.5**	81.9	**43.1**	**68.0**	**90.6**	96.6	**85.4**	92.0	**39.8**	56.7	**57.7**	72.0	**62.0**	**77.9**

SFT maximizes log-probability, while RL maximizes probability directly (see derivation in Appendix C.1). Since SFT is a lower bound of RL Qin & Springenberg (2025), aligning the gradient design with probability (π) rather than log-probability ($\log \pi$) better serves the RL objective. Furthermore, probability lies in a bounded symmetric interval $(0, 1)$, facilitating stable gradient design compared to the unbounded $(-\infty, 0)$ of log-probability.

3.2 DECOUPLED GRADIENT POLICY OPTIMIZATION (DGPO)

Instability in Soft Clipping. Prior soft clipping methods (e.g., GPPO) maintain constant weights for $\nabla \log \pi$. In probability space, this implies the effective weight $\mathcal{W} \propto 1/\pi$. As $\pi \to 0$ (Left Boundary), the weight diverges, causing instability.

DGPO Formulation. We propose DGPO to: (1) preserve gradients for exploration, (2) stabilize via adaptive decay, and (3) minimize bias. We define the weighting function $\mathcal{W}_{i,t}^{\text{DGPO}}(\theta)$ applied to the *probability gradient*:

$$\mathcal{W}_{i,t}^{\text{DGPO}}(\theta) = \begin{cases} C_{\text{left}} \cdot sg^{n}[\pi_\theta], & \text{if LN,} \\ C_{\text{right}} \cdot sg^{-\frac{1}{m}}[\pi_\theta], & \text{if HP,} \\ 1/\pi_{\theta_{\text{old}}}, & \text{otherwise.} \end{cases} \tag{3}$$

The objective function is $\mathcal{J}_{\text{DGPO}}(\theta) = \mathbb{E} \sum \mathcal{W}_{i,t}^{\text{DGPO}}(\theta) \hat{A}_i \pi_\theta(o_{i,t})$. Here, $n, m \in \mathbb{Z}^+$ control decay rates. C_{left} and C_{right} are constants ensuring continuity (derived in Appendix C.2). DGPO applies a polynomial decay (n) to low-probability tokens for stability and a reciprocal radical decay (m) to high-probability tokens to foster exploration. This mathematically guarantees gradient continuity and prevents the divergence seen in prior methods. Theoretical analysis (Appendix C.3) confirms DGPO achieves minimal bias relative to the true policy gradient compared to baselines.

4 EXPERIMENTS

We use DeepSeek-R1-Distill-Qwen (1.5B/7B/14B) models Guo et al. (2025) on the DAPO-Math-17K dataset Yu et al. (2025). Baselines include GRPO, CISPO, GPPO, CE-GPPO, and ASPO. We report Avg@32 and Pass@32 on benchmarks like AIME and MATH MAA (2024; 2025; 2023); Hendrycks et al. (2021); Lewkowycz et al. (2022); He et al. (2024). Full details are in Appendix D.

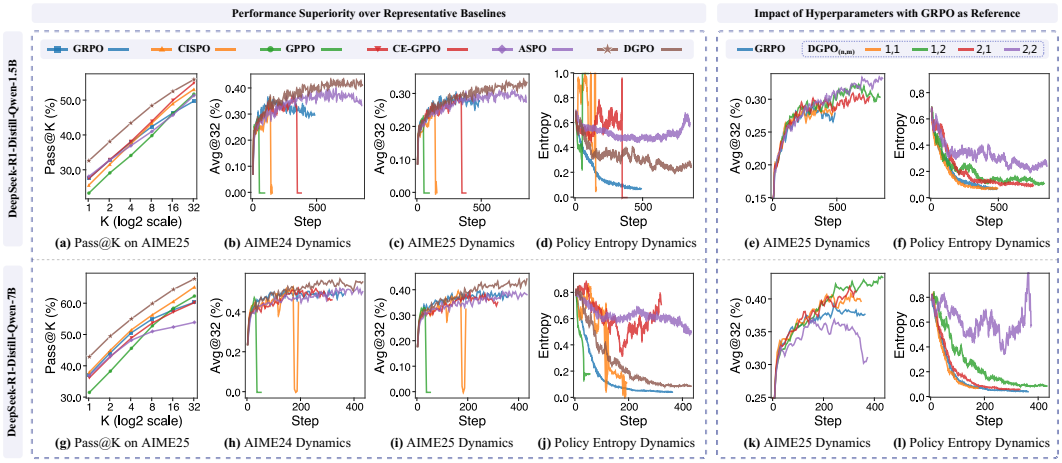


Figure 3: Training dynamics. DGPO (red) avoids the collapse seen in soft-clipping methods and the premature convergence of GRPO, maintaining healthy entropy and higher accuracy.

Performance & Dynamics. Table 2 shows DGPO significantly outperforms baselines. On the 1.5B model, DGPO surpasses GRPO by +4.3% in Avg@32. On 7B, it leads by +3.1%. Figure 3 illustrates training dynamics. Soft-clipping methods (CISPO, GPPO) suffer from collapse due to divergent weights. GRPO shows premature convergence (entropy drop). DGPO maintains stable entropy and sustained performance gains.

Scalability. We extended experiments to the 14B model (details in Appendix E.1). DGPO achieves 56.7% Avg@32 vs. GRPO's 53.6%, confirming that the benefits of decoupled gradients scale effectively.

Ablation. We analyzed hyperparameters n (stability) and m (exploration). Increasing m boosts exploration but risks instability if unchecked. We recommend $n = 1, m = 2$ as a robust configuration. Mechanistic analysis (Appendix E.3) confirms DGPO produces a wider, yet controlled, distribution of importance sampling ratios compared to GRPO's narrow and GPPO's divergent distributions, aligning with observations that effective exploration requires broader coverage Yang et al. (2025b).

5 Conclusion

We introduce DGPO, rethinking RLVR optimization by prioritizing probability gradients. By decoupling decay mechanisms for stability and exploration, DGPO resolves the fragility of prior hard and soft clipping methods. Empirical results across multiple model scales validate DGPO as a robust and scalable solution for reasoning LLMs.

References

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

MAA. American mathematics competitions - amc, 2023. URL `https://maa.org/`. Accessed: 2023.

MAA. American invitational mathematics examination - aime, 2024. URL `https://maa.org/`. Accessed: 2024.

MAA. American invitational mathematics examination - aime, 2025. URL `https://maa.org/`. Accessed: 2025.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Doina Precup, Richard S Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. 2000.

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*, 2025a.

Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *arXiv preprint arXiv:2509.20712*, 2025b.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. Aspo: Asymmetric importance sampling policy optimization. *arXiv preprint arXiv:2510.06062*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, et al. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*, 2025.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*, 2025a.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in rl for llms. *arXiv preprint arXiv:2505.12929*, 2025b.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6672–6679, 2020.

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

A Extended Related Works

A.1 RLVR in LLMs

RLVR utilizes deterministic signals (e.g., correct answers) rather than learned reward models to enhance LLM reasoning Uesato et al. (2022). Recent advancements like DeepSeek-Math Shao et al. (2024) popularized GRPO, which efficiently normalizes rewards within a sampled group, eliminating the need for a critic. However, GRPO inherits the clipping-induced exploration limitations of PPO.

A.2 Importance Sampling and Clipping

IS enables off-policy training by correcting distribution shifts via the ratio $\pi_\theta/\pi_{\theta_{\text{old}}}$ Precup et al. (2000); Schulman et al. (2015). To prevent variance explosion, PPO Schulman et al. (2017) clips this ratio within $[1-\varepsilon, 1+\varepsilon]$. This "hard clipping" mechanism zeros out gradients for outlier tokens, prioritizing exploitation while neglecting low-probability tokens essential for exploration, causing rapid entropy decay O'Donoghue et al. (2016); Yu et al. (2025). To mitigate exploration losses, several methods dynamically adjust clipping bounds Yu et al. (2025); Yang et al. (2025a), allowing more updates for specific tokens. However, they still rely on hard boundaries, inevitably discarding gradient information for tokens beyond the adjusted thresholds.

A.3 Soft Clipping Policy Optimization

More recent approaches replace "hard clipping" with soft schemes. For instance, CISPO Chen et al. (2025) combines "soft clipping" with soft dual clip Ye et al. (2020), while GPPO Su et al. (2025a) retains a constant log-probability gradient weight for out-of-bound tokens. Crucially, both CISPO and GPPO suffer from left boundary instability: as $\pi_\theta \to 0$, the gradient grows indefinitely. Without proper decay mechanisms, this results in divergent updates that destabilize training. CE-GPPO Su et al. (2025b) attempts to scale boundary gradients via hyperparameters but fails to resolve the underlying divergence. Distinctly, ASPO Wang et al. (2025) proposes a reversed ratio to balance updates. Compared with these approaches, DGPO redefines the optimization target in probability space to ensure theoretical continuity and stability.

B Detailed Instantiations of Baselines

Instantiations of Clipping Strategies. Various existing methods can be interpreted as specific instantiations of $\mathcal{F}_{i,t}(\theta)$. Standard **PPO** Schulman et al. (2017) and **GRPO** Shao et al. (2024) applies "hard clipping" to penalize excessive updates:

$$\mathcal{F}_{i,t}^{\text{GRPO}}(\theta) = \begin{cases} 0, & \text{if LN} \vee \text{HP}, \\ w_{i,t}(\theta), & \text{otherwise}. \end{cases} \tag{4}$$

CISPO Chen et al. (2025) preserves exploration for these tokens by the following gradient weight:

$$\mathcal{F}_{i,t}^{\text{CISPO}}(\theta) = \begin{cases} 1-\varepsilon_{\text{low}}, & \text{if LN} \vee \text{LP}, \\ 1+\varepsilon_{\text{high}}, & \text{if HP} \vee \text{HN}, \\ w_{i,t}(\theta), & \text{otherwise}. \end{cases} \tag{5}$$

This method inherently employs soft dual clip Ye et al. (2020), which clips the loss values for LP and HN (reverse cases) while preserving gradients. Subsequently, **GPPO** Su et al. (2025a) introduces a gradient weight more aligned with PPO, focusing solely on LN and HP cases:

$$\mathcal{F}_{i,t}^{\text{GPPO}}(\theta) = \begin{cases} 1-\varepsilon_{\text{low}}, & \text{if LN}, \\ 1+\varepsilon_{\text{high}}, & \text{if HP}, \\ w_{i,t}(\theta), & \text{otherwise}. \end{cases} \tag{6}$$

CE-GPPO Su et al. (2025b) further refines GPPO by introducing hyperparameters (β_1 and β_2) to control gradient scaling at both boundaries:

$$\mathcal{F}_{i,t}^{\text{CE}}(\theta) = \begin{cases} \beta_1(1-\varepsilon_{\text{low}}), & \text{if LN}, \\ \beta_2(1+\varepsilon_{\text{high}}), & \text{if HP}, \\ w_{i,t}(\theta), & \text{otherwise}. \end{cases} \tag{7}$$

To address the imbalance in updating positive-advantage tokens, **ASPO** Wang et al. (2025) reverses gradient weight and adopts soft dual clip:

$$\mathcal{F}_{i,t}^{\text{ASPO}}(\theta) = \begin{cases} 0, & \text{if LN} \vee \text{HP}, \\ 1 - \varepsilon'_{\text{low}}, & \text{if LP}', \\ 1 + \varepsilon'_{\text{high}}, & \text{if HN}', \\ \frac{1}{w_{i,t}(\theta)}, & \text{otherwise.} \end{cases} \tag{8}$$

The values of $1 - \varepsilon'_{\text{low}}$ and $1 + \varepsilon'_{\text{high}}$ for cases LP′ and HN′ in ASPO may be more extreme.

C Derivation and Proof

C.1 Derivation of RL Estimator in SFT Form

The token-level objective of RL is to maximize the arithmetic mean expectation of the estimated advantage, formulated as follows:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_\theta(\cdot|q)} \left[\sum_{t=1}^{|o|} A_t \right] \tag{9}$$

Assume that $\pi_e(\cdot|q)$ represents the ideal expert distribution, which satisfies two key properties: (1) its output follows a one-hot distribution; (2) every sampled token o_t achieves the maximum advantage A_t. By applying Importance Sampling, we obtain:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_e(\cdot|q)} \left[\sum_{t=1}^{|o|} \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_e(o_t \mid q, o_{<t})} \cdot A_t \right] \tag{10}$$

Since $\pi_e(\cdot|q)$ is a one-hot distribution, the probability $\pi_e(o_t \mid q, o_{<t})$ is always 100%, leading to:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_e(\cdot|q)} \left[\sum_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}) \cdot A_t \right] \tag{11}$$

Given that each action o_t attains the highest possible A_t, under the binary advantage setting, A_t is consistently 1, resulting in:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{(q,o)\sim\mathcal{D}_e} \left[\sum_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}) \right] \tag{12}$$

Consequently, the gradient is derived as:

$$\nabla_\theta \mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{(q,o)\sim\mathcal{D}_e} \left[\sum_{t=1}^{|o|} \nabla_\theta \pi_\theta(o_t \mid q, o_{<t}) \right] \tag{13}$$

C.2 Derivation of Continuity Constants

In this section, we derive the constants C_{left} and C_{right} used in the DGPO weighting function. The primary objective is to ensure that the gradient estimator is continuous with respect to the policy probability π_θ at the clipping boundaries.

From equation equation **??**, the gradient equation of DGPO can be explicitly expressed as:

$$\nabla_\theta \mathcal{J}_{\text{DGPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \mathcal{W}_{i,t}^{\text{DGPO}}(\theta) \hat{A}_i \nabla_\theta \pi_\theta(o_{i,t}|q, o_{i,<t}), \tag{14}$$

Using the identity $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$, the effective coefficient applied to the standard score function $\nabla_\theta \log \pi_\theta$ is $\mathcal{F}_{i,t} = \mathcal{W}_{i,t}^{\text{DGPO}} \cdot \pi_\theta$. Since π_θ is continuous, ensuring the continuity of $\mathcal{W}_{i,t}^{\text{DGPO}}$ at the boundaries is sufficient to ensure the continuity of the entire gradient estimator.

Let $w_{i,t} = \frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}$ denote the importance sampling ratio. The In-Boundary (M) weight is given by:

$$\mathcal{W}_{i,t}^{\text{M}} = \frac{1}{\pi_{\theta_{\text{old}}}}. \tag{15}$$

C.2.1 Left Boundary Derivation

The transition between the Left Boundary (LN) and the In-Boundary (M) region occurs when the importance sampling ratio is $w_{i,t} = 1 - \varepsilon_{\text{low}}$. At this boundary, the current policy probability is:

$$\pi_\theta = (1 - \varepsilon_{\text{low}})\pi_{\theta_{\text{old}}}. \tag{16}$$

We then equate the weighting functions for the LN and M regions at this point:

$$\begin{aligned} \mathcal{W}_{i,t}^{\text{LN}}\Big|_{w_{i,t}=1-\varepsilon_{\text{low}}} &= \mathcal{W}_{i,t}^{\text{M}}\Big|_{w_{i,t}=1-\varepsilon_{\text{low}}} \\ C_{\text{left}} \cdot \pi_\theta^n &= \frac{1}{\pi_{\theta_{\text{old}}}} \\ C_{\text{left}} \cdot [(1 - \varepsilon_{\text{low}})\pi_{\theta_{\text{old}}}]^n &= \frac{1}{\pi_{\theta_{\text{old}}}}. \end{aligned} \tag{17}$$

Solving for C_{left}:

$$C_{\text{left}} = \frac{1}{\pi_{\theta_{\text{old}}} \cdot (1 - \varepsilon_{\text{low}})^n \pi_{\theta_{\text{old}}}^n} = \frac{1}{(1 - \varepsilon_{\text{low}})^n \pi_{\theta_{\text{old}}}^{n+1}}. \tag{18}$$

Thus, $C_{\text{left}} = (1 - \varepsilon_{\text{low}})^{-n} \pi_{\theta_{\text{old}}}^{-(n+1)}$.

Table 3: Definition of bias relative to Policy Gradient

Bias Type	**Mathematical Definition**
In-Boundary Bias	$\text{Bias}_{\text{Algo}}^{\text{M}} = \left\|\nabla_\theta \mathcal{J}_{\text{Algo}}^{\text{M}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{M}}(\theta)\right\|$
Left-Boundary Bias	$\text{Bias}_{\text{Algo}}^{\text{LN}} = \left\|\nabla_\theta \mathcal{J}_{\text{Algo}}^{\text{LN}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{LN}}(\theta)\right\|$
Right-Boundary Bias	$\text{Bias}_{\text{Algo}}^{\text{HP}} = \left\|\nabla_\theta \mathcal{J}_{\text{Algo}}^{\text{HP}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{HP}}(\theta)\right\|$
Reverse Left-Boundary Bias	$\text{Bias}_{\text{Algo}}^{\text{LP}} = \left\|\nabla_\theta \mathcal{J}_{\text{Algo}}^{\text{LP}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{LP}}(\theta)\right\|$
Reverse Right-Boundary Bias	$\text{Bias}_{\text{Algo}}^{\text{HN}} = \left\|\nabla_\theta \mathcal{J}_{\text{Algo}}^{\text{HN}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{HN}}(\theta)\right\|$

C.2.2 Right Boundary Derivation

Similarly, the transition between the Right Boundary (HP) and the In-Boundary (M) region occurs when $w_{i,t} = 1 + \varepsilon_{\text{high}}$. At this boundary:

$$\pi_\theta = (1 + \varepsilon_{\text{high}})\pi_{\theta_{\text{old}}}. \tag{19}$$

We similarly equate the weighting functions for the HP and M regions at this point:

$$\begin{aligned} \mathcal{W}_{i,t}^{\text{HP}}\Big|_{w_{i,t}=1+\varepsilon_{\text{high}}} &= \mathcal{W}_{i,t}^{\text{M}}\Big|_{w_{i,t}=1+\varepsilon_{\text{high}}} \\ C_{\text{right}} \cdot \pi_\theta^{-\frac{1}{m}} &= \frac{1}{\pi_{\theta_{\text{old}}}} \\ C_{\text{right}} \cdot [(1 + \varepsilon_{\text{high}})\pi_{\theta_{\text{old}}}]^{-\frac{1}{m}} &= \frac{1}{\pi_{\theta_{\text{old}}}}. \end{aligned} \tag{20}$$

Solving for C_{right}:

$$C_{\text{right}} = \frac{(1 + \varepsilon_{\text{high}})^{\frac{1}{m}} \pi_{\theta_{\text{old}}}^{\frac{1}{m}}}{\pi_{\theta_{\text{old}}}} = (1 + \varepsilon_{\text{high}})^{\frac{1}{m}} \pi_{\theta_{\text{old}}}^{\frac{1}{m}-1}. \tag{21}$$

This concludes the derivation of the constants.

C.3 Proof of Policy Gradient Bias

C.3.1 Standard Policy Gradient Estimator

The gradient estimator for the standard policy gradient method is formally given by:

$$\nabla_\theta \mathcal{J}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_\theta(\cdot|q)} \left[A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \tag{22}$$

By introducing Importance Sampling with the old policy, we obtain the following form:

$$\nabla_\theta \mathcal{J}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D}, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})} A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t}) \right] \tag{23}$$

We adopt Equation equation 23 as the standard policy gradient estimator in the subsequent proofs.

C.3.2 Decomposed Gradient Estimation

To distinguish gradient estimators under different conditions, we define the following binary variables based on the importance sampling ratio $r_t(\theta) = \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}$ and the advantage A_t. Note that while r_t is a function of θ, for the purpose of defining the estimator's functional form, these regions are treated as piecewise conditions:

$$\begin{cases} v_t^{\text{LN}} = \mathbb{I}\left[r_t(\theta) < 1 - \varepsilon_{\text{low}} \wedge A_t < 0\right] \\ v_t^{\text{HP}} = \mathbb{I}\left[r_t(\theta) > 1 + \varepsilon_{\text{high}} \wedge A_t > 0\right] \\ v_t^{\text{LP}} = \mathbb{I}\left[r_t(\theta) < 1 - \varepsilon_{\text{low}} \wedge A_t > 0\right] \\ v_t^{\text{HN}} = \mathbb{I}\left[r_t(\theta) > 1 + \varepsilon_{\text{high}} \wedge A_t < 0\right] \\ v_t^{\text{M}} = 1 - \left(v_t^{\text{LN}} + v_t^{\text{HP}} + v_t^{\text{LP}} + v_t^{\text{HN}}\right) \end{cases} . \tag{24}$$

The policy gradient estimator is formally decomposed and expressed as a sum of five distinct terms:

$$\nabla_\theta \mathcal{J}_{\text{PG}}(\theta) = \sum_{X\in\mathcal{X}} \nabla_\theta \mathcal{J}_{\text{PG}}^X(\theta), \tag{25}$$

where $\mathcal{X} \in \{\text{M, LN, LP, HP, HN}\}$, and the general mathematical form for each term is defined as:

$$\nabla_\theta \mathcal{J}_{\text{PG}}^X(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[v_t^X \cdot r_t A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t}) \right]. \tag{26}$$

Similarly, gradient estimators for GRPO, CISPO, GPPO, CE-GPPO, ASPO, and DGPO can be consistently decomposed into five-term sums. The bias is defined as the magnitude (L_2 norm) of the difference vector between the algorithm's gradient estimator and the standard policy gradient under identical conditions, as shown in Table 3.

C.3.3 In-Boundary Bias

The gradient estimates for each algorithm under in-boundary conditions are analyzed below. While most algorithms (GRPO, CISPO, GPPO, CE, DGPO) maintain an unbiased estimator locally ($v_t^M = 1$), ASPO introduces a non-linear regularization term specifically when the advantage is positive ($A_t > 0$). The magnitude of the bias is then computed as shown in Table 4.

Table 4: In-Boundary bias among various policy optimization algorithms

Algorithm	**In-Boundary Bias Magnitude**
GRPO, CISPO, GPPO, CE-GPPO DGPO	$\text{Bias}_{\text{X}}^{\text{M}} = \left\|\nabla_\theta \mathcal{J}_{\text{X}}^{\text{M}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{M}}(\theta)\right\| = 0$
ASPO	$\text{Bias}_{\text{ASPO}}^{\text{M}} = \left\| \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot\|q)} \left[v_t^{\text{M}} \cdot \left(\frac{1}{r_t} - r_t \right) \cdot A_t \nabla_\theta \log \pi_\theta(o_t\|q,o_{<t}) \right] \right\|$ $\approx \left\| \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot\|q)} \left[v_t^{\text{M}} \cdot \underbrace{-2(r_t-1)}_{\lambda_t} \cdot A_t \nabla_\theta \log \pi_\theta(o_t\|q,o_{<t}) \right] \right\| \neq 0^*$

The relationship of in-boundary bias magnitudes is determined to be:

$$0 = \text{Bias}_{\text{DGPO}}^{\text{M}} = \text{Bias}_{\text{Others}}^{\text{M}} < \text{Bias}_{\text{ASPO}}^{\text{M}}.$$

$$
\begin{cases}
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot 0 \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{CISPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot (1-\varepsilon_{\text{low}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{GPPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot (1-\varepsilon_{\text{low}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{CE}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot \beta_1(1-\varepsilon_{\text{low}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{ASPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot 0 \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{DGPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot \frac{r_t^{n+1}}{(1-\varepsilon_{\text{low}})^n} \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]
\end{cases}
\tag{27}
$$

Table 5: Left-Boundary bias among various policy optimization algorithms

Algorithm	**Left-Boundary Bias Magnitude**
GRPO	$\text{Bias}^{\text{LN}}_{\text{GRPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{GRPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]\right\|$
CISPO	$\text{Bias}^{\text{LN}}_{\text{CISPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{CISPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot (r_0 - r_t) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]\right\|$
GPPO	$\text{Bias}^{\text{LN}}_{\text{GPPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{GPPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot (r_0 - r_t) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]\right\|$
CE-GPPO	$\text{Bias}^{\text{LN}}_{\text{CE}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{CE}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left\{v_t^{\text{LN}} \cdot [\beta_1 r_0 - r_t] \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right\}\right\|$
ASPO	$\text{Bias}^{\text{LN}}_{\text{ASPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{ASPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]\right\|$
DGPO	$\text{Bias}^{\text{LN}}_{\text{DGPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LN}}_{\text{DGPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LN}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LN}} \cdot (r_t - \frac{r_t^{n+1}}{r_0^n}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\right]\right\|$

C.3.4 LEFT-BOUNDARY BIAS

The gradient estimates for each algorithm under specific left-boundary conditions ($r_t < 1-\varepsilon_{\text{low}}, A_t < 0$) are presented in Eq. equation 27. Let $r_0 = 1 - \varepsilon_{\text{low}}$. The magnitude of the bias is calculated as shown in Table 5.

Since $0 < r_t < r_0 < 1$, it follows universally that $\text{Bias}^{\text{LN}}_{\text{DGPO}} < \text{Bias}^{\text{LN}}_{\text{GRPO}}$ and $\text{Bias}^{\text{LN}}_{\text{DGPO}} < \text{Bias}^{\text{LN}}_{\text{ASPO}}$. However, ranking other algorithms requires a quantitative analysis of the integrals.

Decoupling Assumption and Quantitative Analysis. To rigorously compare the bias magnitudes, we apply a **decoupling assumption**: we assume the gradient norm is locally independent of the importance sampling ratio r_t within the small boundary region. Let $\delta = \mathbb{E}_{q\sim\mathcal{D}}[\|\nabla_\theta \log \pi_\theta(o_t|q, o_{<t})\|]$ denote the average gradient magnitude.

The explicit expressions analytically are derived as follows (integrals are over r_t):

$$
\text{Bias}^{\text{LN}}_{\text{GRPO}} = \text{Bias}^{\text{LN}}_{\text{ASPO}} = \int_0^{r_0} \underbrace{|0 - r_t|}_{\text{Coeff. Diff}} \cdot \delta \cdot k r_t^\gamma \, \mathrm{d}r_t = \frac{k\delta r_0^{\gamma+2}}{\gamma+2} \tag{28}
$$

$$
\text{Bias}^{\text{LN}}_{\text{CISPO}} = \text{Bias}^{\text{LN}}_{\text{GPPO}} = \int_0^{r_0} \underbrace{|r_0 - r_t|}_{\text{Coeff. Diff}} \cdot \delta \cdot k r_t^\gamma \, \mathrm{d}r_t = k\delta r_0^{\gamma+2}\left(\frac{1}{\gamma+1} - \frac{1}{\gamma+2}\right) \tag{29}
$$

$$\text{Bias}_{\text{CE}}^{\text{LN}} = \int_0^{r_0} \underbrace{|\beta_1 r_0 - r_t|}_{\text{Coeff. Diff}} \cdot \delta \cdot k r_t^{\gamma} \, \mathrm{d}r_t = k\delta r_0^{\gamma+2} \left| \frac{\beta_1}{\gamma+1} - \frac{1}{\gamma+2} \right| \tag{30}$$

$$\text{Bias}_{\text{DGPO}}^{\text{LN}} = \int_0^{r_0} \underbrace{\left| \frac{r_t^{n+1}}{r_0^n} - r_t \right|}_{\text{Coeff. Diff}} \cdot \delta \cdot k r_t^{\gamma} \, \mathrm{d}r_t = k\delta r_0^{\gamma+2} \left(\frac{1}{\gamma+2} - \frac{1}{n+\gamma+2} \right) \tag{31}$$

where we utilize the property $r_t > r_t^{n+1}/r_0^n$ for $r_t < r_0$. In the extreme limiting case ($\gamma \to +\infty$), the resulting bias ratios are:

$$\lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{DGPO}}^{\text{LN}}}{\text{Bias}_{\text{CISPO}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{DGPO}}^{\text{LN}}}{\text{Bias}_{\text{GPPO}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{n(\gamma+1)}{\gamma+2+n} = n \tag{32}$$

$$\lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{DGPO}}^{\text{LN}}}{\text{Bias}_{\text{CE}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{n(\gamma+1)(\gamma+2+n)^{-1}}{|(1-\beta_1)\gamma+1-2\beta_1|} = 0 \tag{33}$$

$$\lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{CISPO}}^{\text{LN}}}{\text{Bias}_{\text{CE}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{GPPO}}^{\text{LN}}}{\text{Bias}_{\text{CE}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{1}{|(1-\beta_1)\gamma+1-2\beta_1|} = 0 \tag{34}$$

$$\lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{GRPO}}^{\text{LN}}}{\text{Bias}_{\text{CE}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{\text{Bias}_{\text{ASPO}}^{\text{LN}}}{\text{Bias}_{\text{CE}}^{\text{LN}}} = \lim_{\gamma\to\infty} \frac{\gamma+1}{|(1-\beta_1)\gamma+1-2\beta_1|} = \frac{1}{|1-\beta_1|} \tag{35}$$

Assuming $n = 1$ (linear decay), we obtain the strict ranking:

$$0 < \text{Bias}_{\text{DGPO}}^{\text{LN}} < \text{Bias}_{\text{CISPO}}^{\text{LN}} = \text{Bias}_{\text{GPPO}}^{\text{LN}} < \text{Bias}_{\text{CE}}^{\text{LN}} < \text{Bias}_{\text{GRPO}}^{\text{LN}} = \text{Bias}_{\text{ASPO}}^{\text{LN}}. \tag{36}$$

C.3.5 Right-Boundary Bias

The gradient estimates for the right-boundary case across algorithms are shown in Eq. equation 37. The bias magnitude is computed as shown in Table 6.

$$\begin{cases} \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{GRPO}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot 0 \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{CISPO}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot (1+\varepsilon_{\text{high}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{GPPO}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot (1+\varepsilon_{\text{high}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{CE}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot \beta_2(1+\varepsilon_{\text{high}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{ASPO}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot 0 \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \\ \nabla_\theta \mathcal{J}_{\text{DGPO}}^{\text{HP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}} \left[v_t^{\text{HP}} \cdot (1+\varepsilon_{\text{high}})^{\frac{1}{m}} r_t^{1-\frac{1}{m}} \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \right] \end{cases} \tag{37}$$

Given the right-boundary condition $1 < 1 + \varepsilon_{\text{high}} < r_t$, and adopting the CE-GPPO configuration $\beta_2 = 1$, we derive the ranking:

$$0 < \text{Bias}_{\text{DGPO}}^{\text{HP}} \leq \text{Bias}_{\text{CISPO}}^{\text{HP}} = \text{Bias}_{\text{GPPO}}^{\text{HP}} = \text{Bias}_{\text{CE}}^{\text{HP}} < \text{Bias}_{\text{GRPO}}^{\text{HP}} = \text{Bias}_{\text{ASPO}}^{\text{HP}}. \tag{38}$$

C.3.6 Reverse Left-Boundary Bias

Based on the definition of ASPO, if the reverse left-boundary is treated as a clipping region $1 - \varepsilon_{\text{low}}$, the gradient estimates are listed in Eq. equation 39. The bias magnitude is shown in Table 7.

The reverse left-boundary bias magnitude relationship is directly derived as:

$$0 = \text{Bias}_{\text{DGPO}}^{\text{LP}} = \text{Bias}_{\text{GRPO}}^{\text{LP}} = \text{Bias}_{\text{GPPO}}^{\text{LP}} = \text{Bias}_{\text{CE}}^{\text{LP}} < \text{Bias}_{\text{CISPO}}^{\text{LP}} = \text{Bias}_{\text{ASPO}}^{\text{LP}}. \tag{40}$$

Table 6: Right-Boundary bias among various policy optimization algorithms

Algorithm	Right-Boundary Bias Magnitude
GRPO	$\text{Bias}^{\text{HP}}_{\text{GRPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{GRPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]\right\|$
CISPO	$\text{Bias}^{\text{HP}}_{\text{CISPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{CISPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left\{v_t^{\text{HP}} \cdot [r_t - (1+\varepsilon_{\text{high}})] \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right\}\right\|$
GPPO	$\text{Bias}^{\text{HP}}_{\text{GPPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{GPPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left\{v_t^{\text{HP}} \cdot [r_t - (1+\varepsilon_{\text{high}})] \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right\}\right\|$
CE-GPPO	$\text{Bias}^{\text{HP}}_{\text{CE}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{CE}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left\{v_t^{\text{HP}} \cdot [r_t - \beta_2(1+\varepsilon_{\text{high}})] \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right\}\right\|$
ASPO	$\text{Bias}^{\text{HP}}_{\text{ASPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{ASPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]\right\|$
DGPO	$\text{Bias}^{\text{HP}}_{\text{DGPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{HP}}_{\text{DGPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{HP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HP}} \cdot [r_t - (1+\varepsilon_{\text{high}})^{\frac{1}{m}} r_t^{1-\frac{1}{m}}] \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]\right\|$

$$
\begin{cases}
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{CISPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot (1-\varepsilon_{\text{low}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{GPPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{CE}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{ASPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot (1-\varepsilon_{\text{low}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{DGPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]
\end{cases}
\tag{39}
$$

Table 7: Reverse Left-Boundary bias among various policy optimization algorithms

Algorithm	Reverse Left-Boundary Bias Magnitude
GRPO, GPPO CE-GPPO, DGPO	$\text{Bias}^{\text{LP}}_{\text{X}} = \left\|\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{X}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LP}}_{\text{PG}}(\theta)\right\| = 0$
CISPO, ASPO	$\text{Bias}^{\text{LP}}_{\text{CISPO}} = \left\|\nabla_\theta \mathcal{J}^{\text{LP}}_{\text{CISPO}}(\theta) - \nabla_\theta \mathcal{J}^{\text{LP}}_{\text{PG}}(\theta)\right\| = \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{LP}} \cdot (1-\varepsilon_{\text{low}} - r_t) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]\right\| \neq 0^*$

$$
\begin{cases}
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{CISPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot (1+\varepsilon_{\text{high}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{GPPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{CE}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{ASPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot (1+\varepsilon_{\text{high}}) \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
\nabla_\theta \mathcal{J}^{\text{HN}}_{\text{DGPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot r_t \cdot A_t \nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]
\end{cases}
\tag{41}
$$

C.3.7 Reverse Right-Boundary Bias

Assuming the reverse right-boundary of ASPO is $1+\varepsilon_{\text{high}}$, the gradient estimates are listed in Eq. equation 41. The bias magnitude is shown in Table 8.

Table 8: Reverse Right-Boundary bias among various policy optimization algorithms

Algorithm	Reverse Left-Boundary Bias Magnitude
GRPO, GPPO CE-GPPO, DGPO	$\text{Bias}_{\text{X}}^{\text{HN}} = \left\|\nabla_\theta \mathcal{J}_{\text{X}}^{\text{HN}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{HN}}(\theta)\right\| = 0$
CISPO, ASPO	$\text{Bias}_{\text{CISPO}}^{\text{HN}} = \left\|\nabla_\theta \mathcal{J}_{\text{CISPO}}^{\text{HN}}(\theta) - \nabla_\theta \mathcal{J}_{\text{PG}}^{\text{HN}}(\theta)\right\|$ $= \left\|\mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}}\left[v_t^{\text{HN}} \cdot [r_t - (1+\varepsilon_{\text{high}})] \cdot A_t \nabla_\theta \log \pi_\theta(o_t\|q, o_{<t})\right]\right\| \neq 0^*$

The reverse right-boundary bias magnitude relationship is directly derived as:

$$0 = \text{Bias}_{\text{DGPO}}^{\text{HN}} = \text{Bias}_{\text{GRPO}}^{\text{HN}} = \text{Bias}_{\text{GPPO}}^{\text{HN}} = \text{Bias}_{\text{CE}}^{\text{HN}} < \text{Bias}_{\text{CISPO}}^{\text{HN}} = \text{Bias}_{\text{ASPO}}^{\text{HN}}. \tag{42}$$

C.4 Derivation of Learning Rate Scaling

To ensure consistent training dynamics across models of varying sizes (1.5B, 7B, and 14B), we employ a learning rate scaling rule based on the principle of *Constant Total Gradient Variance*. Below, we provide the formal derivation of this scaling law.

Assumptions. Consider two models with identical architecture but different parameter counts, denoted by N. We make the following standard assumptions for large-scale model training:

- The training data, batch size, and optimizer configuration remain constant.
- The gradient variance of a single parameter, denoted as σ^2, is constant and independent of the total model size (assuming high redundancy in LLM parameters).
- The gradients of individual parameters are approximately independent.

Total Gradient Variance. Let $\nabla_\theta \mathcal{L}$ represent the gradient vector of the loss function with respect to the model parameters $\theta \in \mathbb{R}^N$. Based on the independence assumption, the variance of the total gradient norm is the sum of the variances of individual parameter gradients:

$$\text{Var}(\nabla_\theta \mathcal{L}) = \sum_{i=1}^{N} \text{Var}(g_i) = N \cdot \sigma^2 \tag{43}$$

Stability Condition. To maintain a consistent convergence speed across different scales, we require the variance of the parameter update step (the "step size" in the parameter space) to remain constant. Let η be the learning rate. The update step is $\Delta\theta = \eta \cdot \nabla_\theta \mathcal{L}$. The variance of this update step is:

$$\text{Var}(\Delta\theta) = \text{Var}(\eta \cdot \nabla_\theta \mathcal{L} = \eta^2 \cdot \text{Var}(\nabla_\theta \mathcal{L}) = \eta^2 N \sigma^2 \tag{44}$$

We define a stability constant C such that:

$$\eta^2 N \sigma^2 = C \tag{45}$$

Scaling Law Derivation. Let (η_1, N_1) be the configuration for the base model (1.5B) and (η_2, N_2) be the configuration for the target model (e.g., 7B or 14B). From Eq. equation 45, we have:

$$\eta_1^2 N_1 \sigma^2 = \eta_2^2 N_2 \sigma^2 = C \tag{46}$$

Eliminating constant terms σ^2 and C, we obtain the relationship:

$$\eta_1^2 N_1 = \eta_2^2 N_2 \implies \eta_2 = \eta_1 \sqrt{\frac{N_1}{N_2}} \tag{47}$$

This establishes the **Inverse Square Root Scaling Law** relative to the number of parameters.

Numerical Verification. Applying this rule to our experimental settings with the base learning rate $\eta_{1.5B} = 1.0 \times 10^{-6}$:

- **For 7B Model:**

$$\eta_{7B} = 1.0 \times 10^{-6} \times \sqrt{\frac{1.5}{7}} \approx 1.0 \times 10^{-6} \times 0.4629 \approx 4.63 \times 10^{-7}$$

- **For 14B Model:**

$$\eta_{14B} = 1.0 \times 10^{-6} \times \sqrt{\frac{1.5}{14}} \approx 1.0 \times 10^{-6} \times 0.3273 \approx 3.27 \times 10^{-7}$$

These calculated values correspond exactly to the learning rates reported in Table 10.

D Implementation Details

D.1 Infrastructure and Environment

Computational Resources. Our experiments are conducted on a high-performance computing cluster consisting of 30 nodes. Each node is equipped with 8 × NVIDIA A100 GPUs (80GB VRAM), interconnected via NVLink for high-bandwidth intra-node communication. The multi-node training relies on a robust Ethernet/InfiniBand fabric to ensure synchronization efficiency. The total effective training time for the largest model (14B) was approximately 500 hours.

Software Stack. We build our reinforcement learning pipeline upon VeRL Sheng et al. (2025), a flexible framework designed for post-training.

- **Training Backend:** We utilize Fully Sharded Data Parallel (FSDP) for distributed training. This setup manages memory efficiency through parameter sharding and offloading, allowing us to train 14B models with full-parameter updates without memory overflow.
- **Inference Engine:** To maximize rollout throughput, we integrate vLLM as the inference backend. We leverage its **PagedAttention** mechanism to efficiently manage Key-Value (KV) cache memory, significantly reducing fragmentation during the generation of long reasoning chains. We align the environment configurations (CUDA version 12.4, PyTorch version 2.6.0) across all nodes to prevent numerical discrepancies.

D.2 Data and Evaluation Benchmarks

Training Dataset. We utilize the DAPO-Math-17k dataset for training. We employ the official tokenizer corresponding to the Qwen2.5-Math series to ensure consistent token mapping between the pre-trained backbone and the RL fine-tuning stage. The maximum response length is set to 8192 tokens. This length is empirically chosen to sufficiently accommodate the chain-of-thought reasoning steps required by benchmarks like AIME and MATH, while maintaining high training throughput.

Benchmark Details To rigorously evaluate the mathematical reasoning capabilities of our models, we selected a suite of diverse benchmarks. These datasets cover a spectrum of difficulty levels, from foundational high school mathematics to expert-level olympiad problems. Table 9 provides a comprehensive overview of their characteristics.

The datasets used for evaluation can be accessed via the following repositories:

- **AIME 2024:** `https://huggingface.co/datasets/math-ai/aime24`
- **AIME 2025:** `https://huggingface.co/datasets/math-ai/aime25`
- **AMC 2023:** `https://huggingface.co/datasets/math-ai/amc23`
- **MATH-500:** `https://huggingface.co/datasets/HuggingFaceH4/MATH-500`
- **Minerva:** `https://huggingface.co/datasets/math-ai/minervamath`
- **OlympiadBench:** `https://huggingface.co/datasets/math-ai/olympiadbench`

Dataset	Core Description	Key Characteristics
AIME 2024	The 2024 edition of the American Invitational Mathematics Examination, serving as a bridge between the AMC and the USAMO.	• Focuses on arithmetic precision and number theory. • Requires answers in a strict integer format (000–999). • Tests robustness against recent contamination.
AIME 2025	The most recent iteration of the AIME competition, representing a strictly "held-out" set for evaluating generalization to unseen problems.	• Zero data contamination risk due to recency. • High complexity requiring multi-step logical chains. • Validates the model's potential for future reasoning.
AMC 2023	Selected problems from the 2023 American Mathematics Competitions (AMC 10/12), representing the entry-level olympiad difficulty.	• Covers broad topics: Algebra, Geometry, Counting. • Functions as a baseline for competitive math ability. • Requires mapping multiple-choice logic to open-ended generation.
MATH-500	A curated subset of 500 representative problems from the widely used MATH dataset, designed by OpenAI for efficient evaluation.	• Spans 7 categories including Calculus and Probability. • Reduces evaluation costs while maintaining distribution fidelity. • Heavily relies on LaTeX understanding.
Minerva	A collection of technical mathematics problems derived from scientific papers and undergraduate-level coursework.	• Involves higher-order symbolic reasoning. • Contains domain-specific vocabulary and notation. • Tests capabilities beyond standard competition math.
OlympiadBench	A comprehensive aggregate of international mathematics competitions (e.g., IMO, CMO) spanning multiple languages and formats.	• Represents the upper bound of mathematical reasoning. • Includes theorem proving and fill-in-the-blank types. • Challenges the model's cross-lingual mathematical logic.

Table 9: Detailed comparison of the mathematical reasoning benchmarks used in this study.

Reward Function. We employ a strict rule-based reward mechanism to rigorously verify the correctness of the generated solutions.

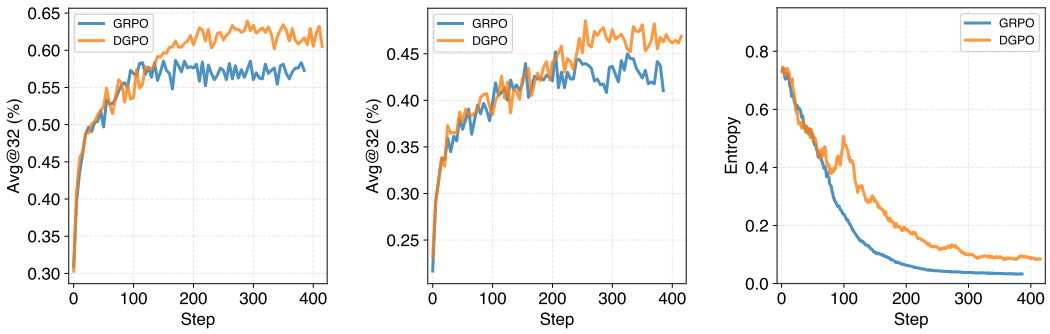
- **Format Verification:** We first check if the output follows the required format (e.g., enclosing the answer in `\boxed{}`).
- **Correctness Check:** Using the `math_verify` toolkit, we compare the extracted answer against the ground truth. A reward of $r = 1$ is assigned for a correct match, and $r = -1$ otherwise. This binary reward setting poses a significant challenge for exploration, as no partial rewards are applied.

D.3 Training Protocols

Optimization Strategy. We employ the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a small weight decay. Unlike standard pre-training schedules, we adopt a **constant learning rate** strategy without warm-up or cosine decay. This design choice eliminates the confounding factors of learning

Table 10: Detailed hyperparameter configurations for all experiments. To ensure a fair comparison, we maintain identical training settings across all algorithms, varying only the learning rate according to the model scale.

Model Scale	Algorithm	Learning Rate	Specific Hyperparameters
Common Settings: Mini-batch Size = 32, Rollout Batch Size = 512, Max Length = 8192			
DeepSeek-R1-Distill-Qwen-1.5B	GRPO	1.0×10^{-6}	$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CISPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CE-GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \beta_1 = 0.75, \beta_2 = 1$
	ASPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \varepsilon'_{\text{low}} = 0.33, \varepsilon'_{\text{high}} = 3$
	DGPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, n = 2, m = 2$
DeepSeek-R1-Distill-Qwen-7B	GRPO	4.63×10^{-7}	$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CISPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CE-GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \beta_1 = 0.75, \beta_2 = 1$
	ASPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \varepsilon'_{\text{low}} = 0.33, \varepsilon'_{\text{high}} = 3$
	DGPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, n = 1, m = 2$
DeepSeek-R1-Distill-Qwen-14B	GRPO	3.27×10^{-7}	$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CISPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$
	CE-GPPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \beta_1 = 0.75, \beta_2 = 1$
	ASPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, \varepsilon'_{\text{low}} = 0.33, \varepsilon'_{\text{high}} = 3$
	DGPO		$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2, n = 1, m = 2$



(a) AIME 2024 (Avg@32) (b) AIME 2025 (Avg@32) (c) Policy Entropy

Figure 4: Training dynamics of DeepSeek-R1-Distill-Qwen-14B comparing GRPO and DGPO.

rate scheduling, allowing us to attribute performance gains solely to the algorithmic improvements. To ensure consistent convergence dynamics across different model scales (1.5B, 7B, 14B), we calibrate the learning rate based on the *Constant Total Gradient Variance* principle.

Numerical Precision and Stability. All models are trained using `bfloat16` precision. We apply gradient clipping with a norm threshold of 1.0 to mitigate gradient explosion. Random seeds for model initialization, data shuffling, and sampling are fixed to 42 to ensure reproducibility.

D.4 Hyperparameter Specifications

Table 10 summarizes the key hyperparameters used across all experiments.

Controlled Variables. To ensure a fair comparison and strictly isolate the impact of different gradient weighting strategies, we unify the clipping thresholds across all experiments. Specifically, for all methods involving trust region clipping or boundary definitions (including GRPO, CISPO, GPPO, CE-GPPO, ASPO and our DGPO), we fix:

$$\varepsilon_{\text{low}} = \varepsilon_{\text{high}} = 0.2$$

This rigorous control ensures that any observed performance differences are primarily attributable to the gradient dynamics rather than minor variations in the trust region size.

Common Configurations. We set the KL coefficient $\beta_{\text{KL}} = 0$. By removing the explicit KL penalty, we rely implicitly on the trust region constraints imposed by the clipping mechanisms to prevent policy collapse. The rollout generation uses a temperature of 1.0 and a top-p of 1.0.

Method-Specific Settings. The method-specific configurations are as follows:

- **CE-GPPO:** Adopts scaled soft clipping with $\beta_1 = 0.75$ and $\beta_2 = 1.0$, as recommended in the original paper.
- **ASPO:** Use $\varepsilon'_{\text{low}} = 0.33$ and $\varepsilon'_{\text{high}} = 3$ as soft dual clip threshold.
- **DGPO:** We introduce the decoupled decay parameters n and m:
  - **1.5B:** $n = 2, m = 2$.
  - **7B & 14B:** $n = 1, m = 2$.

  The continuity constants C_{left} and C_{right} are automatically calculated based on $\pi_{\theta_{\text{old}}}$.

E Additional Experimental Results

E.1 14B Model Training Dynamics

We visualize the training trajectories of the DeepSeek-R1-Distill-Qwen-14B model in Figure 4. Similar to the 1.5B and 7B models, DGPO on 14B scale demonstrates faster convergence and higher asymptotic performance on AIME benchmarks while maintaining a stable entropy reduction curve, avoiding the collapse issues seen in RLVR.

Table 11 provides comprehensive evaluation results for 14B model, including both Avg@32 (expected performance) and Pass@32 (potential capability).

E.2 Pass@K Statistics on AIME Benchmarks

We report Pass@K ($k \in \{1, 2, 4, 8, 16, 32\}$) metrics specifically for the **AIME 2024** and **AIME 2025** benchmarks across 1.5B, 7B, and 14B scales (Table 12). The results demonstrate that DGPO consistently achieves higher coverage of the solution space (higher Pass@K) compared to baselines, particularly as k increases.

E.3 Hyperparameter Analysis

We investigate the impact of hyperparameters n and m by expanding from the baseline ($n = 1$, $m = 1$) to four configurations: $(1, 1)$, $(1, 2)$, $(2, 1)$, and $(2, 2)$. Figure 3(e,k) shows their performance on AIME25 across 1.5B and 7B models.

Robustness and Patterns. All DGPO configurations outperform GRPO (with the exception of $n = 2$ and $m = 2$ on the 7B model), demonstrating our algorithm's robustness. However, the optimal configuration varies by scale: $(2, 2)$ for 1.5B and $(1, 2)$ for 7B. Analyzing Figure 3(e,f,k,l), which reveals a consistent pattern: increasing n or m generally yields: (1) improved performance, (2) elevated overall entropy levels, but (3) reduced entropy stability. Notably, significant entropy volatility is observed only in the 7B model under the $(2, 2)$ setting that negates the performance benefits.

Tuning Guideline. Based on these observations, we propose a heuristic for hyperparameter tuning: *Enhance exploration by increasing n and m as long as entropy remains stable. Upon observing instability, revert to the preceding stable configuration.* Empirically, we recommend $n = 1$ and $m = 2$ as a robust and conservative baseline configuration.

Table 11: Full performance comparison on DeepSeek-R1-Distill-Qwen-14B.

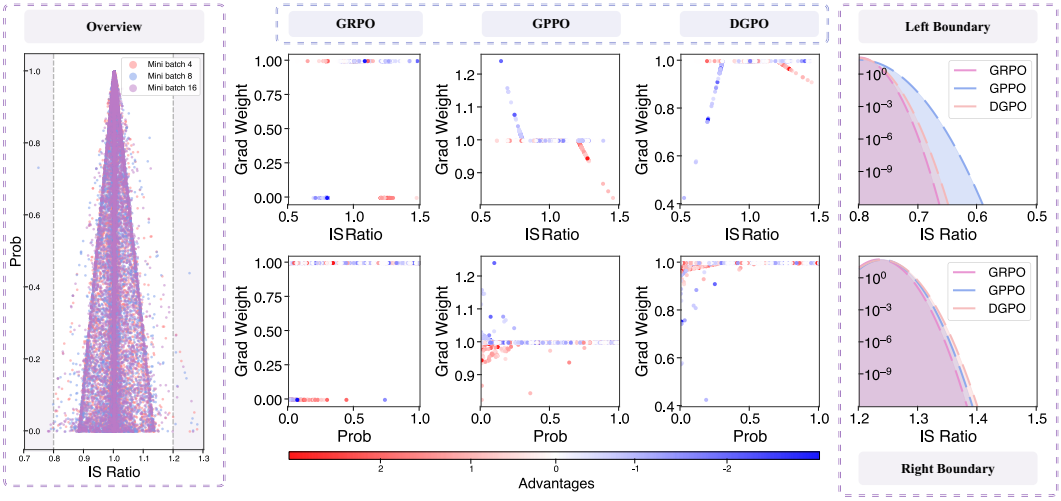
Method	AIME24		AIME25		AMC23		MATH500		Minerva		Olympiad		Avg.	
	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32	A@32	P@32
DEEPSEEK-R1-DISTILL-QWEN-14B														
GRPO	56.6	82.9	40.5	**67.8**	92.2	97.4	66.5	70.4	22.6	35.3	43.2	50.7	53.6	67.4
DGPO	**63.3**	**86.2**	**47.6**	66.4	**93.9**	**99.1**	**67.0**	**75.1**	**23.0**	**37.9**	**45.1**	**57.4**	**56.7**	**70.4**

Table 12: Detailed Pass@K performance on AIME 2024 and AIME 2025 across all model scales.

Method	P@1	P@2	P@4	P@8	P@16	P@32
DeepSeek-R1-Distill-Qwen-1.5B						
Dataset: AIME 2024						
GRPO	33.2	43.2	53.2	61.6	67.9	71.8
CISPO	34.8	44.1	52.0	58.7	64.4	69.1
GPPO	29.6	38.4	47.1	53.6	57.9	60.5
CE-GPPO	35.1	44.4	52.6	60.0	66.2	70.2
ASPO	36.4	44.4	53.1	61.7	68.5	73.2
DGPO	**43.3**	**53.4**	**63.1**	**70.6**	**75.6**	**79.3**
Dataset: AIME 2025						
GRPO	27.7	33.0	38.1	42.6	46.5	49.9
CISPO	25.8	31.8	37.6	43.6	49.2	53.3
GPPO	23.5	29.3	34.3	40.0	46.4	51.9
CE-GPPO	27.7	33.1	38.3	44.1	50.1	55.1
ASPO	28.3	33.0	37.1	41.2	45.9	51.5
DGPO	**32.8**	**38.3**	**43.6**	**48.6**	**52.7**	**56.1**
DeepSeek-R1-Distill-Qwen-7B						
Dataset: AIME 2024						
GRPO	48.2	58.0	66.6	73.4	78.6	**82.5**
CISPO	51.6	60.9	68.5	73.5	75.8	76.6
GPPO	43.1	52.1	59.3	65.0	69.7	72.5
CE-GPPO	48.7	55.7	62.3	68.1	73.0	76.9
ASPO	51.8	61.2	68.5	73.4	76.7	79.6
DGPO	**55.5**	**64.1**	**71.6**	**77.0**	**80.2**	81.9
Dataset: AIME 2025						
GRPO	37.4	44.4	50.8	55.2	58.1	60.5
CISPO	38.2	45.2	51.7	56.5	60.8	65.4
GPPO	31.7	38.5	45.8	53.0	58.6	62.5
CE-GPPO	36.4	43.1	49.1	53.9	57.4	60.4
ASPO	37.1	43.3	48.4	51.2	52.6	54.1
DGPO	**43.1**	**49.7**	**55.2**	**60.1**	**64.6**	**68.0**
DeepSeek-R1-Distill-Qwen-14B						
Dataset: AIME 2024						
GRPO	56.6	69.3	76.0	79.8	81.9	82.9
DGPO	**63.3**	**70.4**	**76.4**	**81.1**	**84.6**	**86.2**
Dataset: AIME 2025						
GRPO	40.5	48.4	54.2	60.1	**64.7**	**67.8**
DGPO	**47.6**	**53.7**	**58.7**	**61.7**	63.9	66.4

E.4 MECHANISTIC ANALYSIS VIA VISUALIZATION

Mechanism of Stability (Left Boundary). *Why does DGPO prevent collapse?* Figure 5(a) visualizes the joint distribution of probability and IS ratios in the final mini-batch. Crucially, tokens at both boundaries are predominantly low-probability tokens. Figure 5(b) further details the relationship



(a) Prob vs IS Ratio **(b)** IS Ratio vs Grad Weight and Prob vs Grad Weight **(c)** IS Ratio

Figure 5: Comparison of gradient weight distributions (Prob: Probability, Grad: Gradient). (a) Overall distribution. (b) Detailed scatter plots of grad weight vs. IS ratio (top row) and prob (bottom row) across three methods: GRPO (left), GPPO (middle), and DGPO (right). Points are colored by advantage. (c) Boundary distribution analysis.

between probability, IS ratio, advantage, and relative gradient weight (normalized by $\pi_{\theta_{\text{old}}}$ probability) for GRPO (zero weight), GPPO (divergent weight), and DGPO (convergent weight). As illustrated, GRPO's zero gradients at the left boundary lead to the narrowest ratio distribution and insufficient exploration. Conversely, GPPO's divergent weights induce an excessively broad distribution, eventually precipitating training collapse due to instability (Figure 3(b,c,h,i)) Yang et al. (2025b). In contrast, DGPO maintains convergent relative weights, resulting in a ratio distribution only slightly wider than GRPO that successfully balances stability with effective exploration.

Mechanism of Improvement (Right Boundary). *Why does DGPO perform better?* Compared with GRPO (where right-boundary tokens have zero gradient and the narrowest ratio distribution), DGPO maintains gradients to foster exploration. Compared with GPPO (which uses a reciprocal standard weight equivalent to DGPO's $m = 1$), DGPO with $m = 2$ employs a reciprocal radical weight. This design induces the widest ratio distribution on the right boundary (Figure 5(c)), significantly enhancing performance. This finding aligns with prior research suggesting that increasing the contribution of positive samples improves performance Yang et al. (2025b); Xi et al. (2025), and is consistent with our hyperparameter analysis in Section **??** where increasing m boosts results.