Exploration v.s. Exploitation: Rethinking RLVR through Clipping, Entropy, and Spurious Reward

Peter Chen[1] **Xiaopeng Li**[1,2] **Ziniu Li**[2] **Wotao Yin**[3] **Xi Chen**[4] **Tianyi Lin**[1]

[1]Columbia [2]CUHK SZ [3]Alibaba Group US [4]NYU Stern

Abstract

This paper examines the exploration–exploitation trade-off in reinforcement learning with verifiable rewards (RLVR), a framework for improving the reasoning of Large Language Models (LLMs). Recent studies suggest that RLVR can elicit strong mathematical reasoning in LLMs through two seemingly paradoxical mechanisms: *spurious rewards*, which suppress exploitation by rewarding outcomes unrelated to the ground truth, and *entropy minimization*, which suppresses exploration by pushing the model toward more confident and deterministic outputs, highlighting a puzzling dynamic: both discouraging exploitation and discouraging exploration improve reasoning performance, yet the underlying principles that reconcile these effects remain poorly understood. We focus on two fundamental questions: (i) how policy entropy relates to performance, and (ii) whether spurious rewards yield gains, potentially through the interplay of clipping bias and model contamination. Our results show that clipping bias under spurious rewards reduces policy entropy, leading to more confident and deterministic outputs, while entropy minimization alone is insufficient for improvement. We further propose a reward-misalignment model explaining why spurious rewards can enhance performance beyond contaminated settings. Our findings clarify the mechanisms behind spurious-reward benefits and provide principles for more effective RLVR training.

1 Introduction

The recent emergence of Large AI Reasoning Models (e.g., Kimi-K2, OpenAI-o1, and DeepSeek-R1 [21, 27, 29]) has been driven by reinforcement learning with verifiable rewards (RLVR). In RLVR, a verifier compares the model's rollout against a deterministic ground-truth solution, especially in mathematics and other STEM domains, providing outcome rewards. This verifiability has enabled models to achieve competitive and human-level performance on challenging benchmarks [26].

In traditional reinforcement learning, the exploration–exploitation trade-off is framed within a Markov decision process with per-step or shaped rewards. Exploration is typically promoted through stochastic policies or explicit bonus terms for underexplored actions, while exploitation reinforces high-return actions via accurate value estimation. RLVR for LLMs departs from this paradigm in three respects: (i) rewards are outcome-level, extremely sparse, and verifiable only at the end of long rollouts, rendering all intermediate token-level actions reward-equivalent; (ii) exploration unfolds in sequence space and is governed by decoding temperature rather than state-local bonuses; and (iii) policy updates rely on ratio clipping with group-normalized advantages, making them more sensitive to importance ratios and relative ranks than to absolute reward values.

These properties give RLVR a distinctive exploration–exploitation regime. In classical RL, spurious rewards, which are misaligned with the true outcome reward (e.g., random noise), would be expected to hinder exploitation by injecting randomness that encourages suboptimal actions. Yet in RLVR, they have been observed to improve performance in `Qwen-Math` models [61], a phenomenon

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

attributed to upper-clipping bias that disproportionately amplifies high-prior responses, consistent with contamination effects reported on MATH500 [80]. Conversely, entropy minimization, which reduces policy entropy to yield more deterministic, high-confidence rollouts, has been widely adopted in RLVR and empirically linked to consistent gains [15, 18, 91, 95]. Notably, Agarwal et al. [2] and Gao et al. [19] directly optimize entropy as an objective and report substantial improvements even without verifiable feedback. These findings point to an RLVR-specific paradox: discouraging exploitation through spurious rewards and discouraging exploration through entropy minimization can both enhance validation accuracy, underscoring learning dynamics that depart from classical RL.

In this paper, we investigate how clipping, policy entropy, and spurious (random) rewards jointly shape model performance in RLVR. We show, both theoretically and empirically, that under random rewards, which discourage exploitation, clipping bias alone provides no meaningful learning signal and cannot directly improve performance. Instead, we establish a direct connection between clipping and policy entropy: clipping reduces entropy and drives the policy toward more deterministic, higher-confidence outputs, thereby inducing an entropy-minimization effect. Importantly, reduced entropy by itself does not guarantee performance gains. To clarify when spurious rewards can be beneficial, we introduce a simple reward-misalignment model. Our analysis overturns the prevailing view that improvements under spurious rewards are limited to potentially contaminated Qwen-Math models; similar gains also arise in the Llama and QwQ families, revealing a more nuanced exploration–exploitation dynamic that cannot be explained by contamination alone.

Contributions. We focus on two fundamental questions: (i) how policy entropy relates to performance, and (ii) whether spurious rewards yield gains, potentially through the interplay of clipping bias and model contamination. Our contributions can be summarized as follows: **(1)** We advance the theoretical foundations of RLVR by deriving explicit bounds on clipping bias and showing, under spurious rewards, this bias does not constitute a meaningful learning signal. To capture its effect more precisely, we introduce a novel one-step policy-entropy shift formulation, which establishes a deterministic link between clipping and policy entropy: clipping systematically reduces entropy and drives the policy toward more deterministic, higher-confidence rollouts; **(2)** We conduct extensive experiments across multiple model families (Qwen-Math, Llama, QwQ) and sizes (7B, 8B, 32B), including both base and distilled variants. These results reconcile conflicting reports in the literature, demonstrating that performance improvements under spurious rewards are robust and not tied to any single model or dataset; **(3)** We show that these gains cannot be attributed to clipping bias or to causal effects of policy entropy, thereby overturning the prevailing view that improvements under spurious rewards are confined to potentially contaminated Qwen-Math models. Instead, our findings reveal a broader and more nuanced exploration–exploitation dynamic unique to RLVR.

2 Main Results Overview

Given the rapid evolution of recent RLVR findings, we provide a technical review for preliminaries and related works in Appendix A and Appendix B, respectively. We provide a detailed derivation and proof for the theoretical results in Appendix D and present empirical evaluation in Appendix C.

2.1 Clipping and Model Performance

Upper-clipping bias under random reward. For the improvements observed in Qwen2.5-Math, which are not present in other base model families (e.g., Llama), Shao et al. [61] attribute this effect to *upper-clipping bias*, a qualitative mechanism that favors higher-probability tokens (which may correlate with the contaminated benchmark) under the old policy, as formalized in Remark 2.1.

Remark 2.1 (Clipping-induced up-bias for high-probability tokens)**.** *When the GRPO upper clip is active (i.e. $|r-1|>\varepsilon$), the largest admissible increase of the probability ratio for a token $\mathbf{y}$ is*

$$\Delta_{\max}(\mathbf{y}) = \varepsilon\, \pi_{\text{old}}(\mathbf{y}),$$

where $\Delta_{\max}$ denotes the threshold, or tolerance, of the clipping deactivation. Hence, if $\pi_{\text{old}}(\mathbf{y}_1) \geq \pi_{\text{old}}(\mathbf{y}_2)$, then $\Delta_{\max}(\mathbf{y}_1) \geq \Delta_{\max}(\mathbf{y}_2)$. That is, tokens that were already likely under the old policy enjoy a wider tolerance before clipping, while low-probability tokens are clipped more aggressively, implicitly reinforcing high-probability choices.

Surrogate decomposition for upper-clipping. To theoretically analyze the effect of clipping bias, we decompose the upper-clipping surrogate and define raw (the raw gradient itself without

being affected by the clipping) and clipping-correction part as follows: $N_t := r_t\hat{A}_t, N_t^{\text{clip}} := \bar{r}_t\hat{A}_t$. Then, given rollout $\mathbf{y}_i$ with length L, the total *clipping correction* C_{tot} can be written as $C_{\text{tot}} = \sum_{t=1}^{L}(N_t^{\text{clip}} - N_t) = \sum_{t=1}^{L}(\bar{r}_t - r_t)\hat{A}_t$. Through the one-step exponential update derived in Proposition D.1, we introduce the following theoretical results:

Theorem 2.2. *Fix group rollout number G, rollout length L, clipping threshold $\varepsilon > 0$ and let $I_t = \mathbf{1}_{\{r_t > 1+\varepsilon\}}$ be the activation indicator with activation rate $\mathbb{E}[I_t] = p$. Define $D_t := (\bar{r}_t - r_t)I_t$ and $C_{\text{tot}} := \sum_{t=1}^{L} D_t\,\hat{A}_t$, for all learning rate $\eta > 0$, we have*

$$\mathbb{E}[|C_{\text{tot}}|] \le M\sqrt{2LR_\eta^{\max}\phi(R_\eta^{\max})p} + M(R_\eta^{\max} - 1)L\min\left\{\sqrt{p}, \frac{\phi(R_\eta^{\max})}{\phi(1+\varepsilon)}\right\}, \tag{1}$$

where $R_\eta^{\max} := e^{2M\eta}$, $M = \sqrt{G-1}$, and $\phi(u) = u\log u - u + 1$. Furthermore, for small η,

$$\mathbb{E}[|C_{\text{tot}}|] \le \mathcal{O}\left(\eta\sqrt{L} + \min\{\eta\sqrt{p}L, \eta^3 L\}\right). \tag{2}$$

Theorem 2.3 (Law of Clipping)**.** *Under the same settings as Theorem 2.2, the lower bound on the expected ratio between the magnitude of the raw surrogate $|N_{\text{raw}}|$ and that of clipping bias $|C_{\text{tot}}|$ is*

$$\frac{\mathbb{E}\big[|N_{\text{raw}}|\big]}{\mathbb{E}\big[|C_{\text{tot}}|\big]} \ge \frac{(1-2^{1-G})\,\eta(1-\eta^2)}{L^{-1/2}M\sqrt{2R_\eta^{\max}\phi(R_\eta^{\max})p} + M(R_\eta^{\max}-1)\min\left\{\sqrt{p}, \frac{\phi(R_\eta^{\max})}{\phi(1+\varepsilon)}\right\}}.$$

In addition, $\mathbb{E}\big[|N_{\text{raw}}|\big] \gg \mathbb{E}\big[|C_{\text{tot}}|\big]$ under practical parameter settings (details in Remark C.2). In Appendix C.1 and Figure 2, we provide extensive ablation analysis of clipping, which supports our theoretical results showing that removing clipping still consistently improves the model performance.

2.2 Clipping and Policy Entropy

While clipping does not directly determine performance, we show a deterministic link between clipping and policy entropy (see Definition D.3). The random-reward setting provides a clean testbed to isolate this effect and its impact on validation performance. Recent work provides theoretical results for estimating one-step policy entropy change in GRPO training. Cui et al. [15] show that

$$\mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}}) \approx -\operatorname{Cov}_{a\sim\pi_{\text{old}}(\cdot|h)}\big(\log\pi_{\text{old}}(a \mid h),\, A(a,h)\big). \tag{3}$$

Under spurious rewards setup, Eq. (3) yields zero entropy change (see Appendix D.6 for Eq. (3)'s theoretical limitation), deviating from the actual training results (see Appendix C.2). We therefore provide Theorem 2.4 and Remark 2.5 to accurately analyze entropy dynamics under spurious rewards.

Theorem 2.4 (Entropy collapse under clipped training)**.** *Let π_{old} be a policy on a finite action set $\mathcal{A}$, fix a clipping ratio $\varepsilon \in (0,1)$ and a small step size η. Define the clipped-advantage reparameterization $A_*(a) = \big(\text{Clip}_\varepsilon\{r(a)\} - 1\big)/\eta$, (see Lemma D.4 for proof). Then, under sufficiently small step size $\eta > 0$, the one-step update admits the exact log form $\log\pi_{\text{new}}(a) = \log\pi_{\text{old}}(a) + \log\big(1+\eta A_*(a)\big) - \log\big(1+\eta\mu_*\big)$, with $\mu_* := \mathbb{E}_{a\sim\pi_{\text{old}}}[A_*(a)]$. Thus, the expected one-step entropy change is*

$$\mathbb{E}\big[\mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}})\big] = -\tfrac{1}{2}\,\eta^2\,\mathbb{E}\big[\text{Var}_{a\sim\pi_{\text{old}}}\big(A_*(a)\big)\big] + \mathcal{O}(\eta^3) \;<\; 0.$$

Remark 2.5 (Entropy increase under unclipped training)**.** *Using the same notation as in Theorem 2.4, under the one-step unclipped GRPO update we have*

$$\mathbb{E}[\mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}})] = -\tfrac{1}{G}(1-2^{1-G})\,\Phi(\pi_{\text{old}})\,\eta^2 \;+\; \mathcal{O}(\eta^3),$$

where $\Phi(\cdot)$ is a third-order polynomial functional measuring the skewness of the policy π_{old} (coefficients given in Theorem D.8. Consequently, the one-step entropy change under unclipped training depends on the initial policy distribution; in particular, more skewed policies can exhibit entropy increases during training. We provide a detailed numerical example in Remark D.9 and validate the result via simulations comparing more- and less-skewed policies in Figure 5, along with actual training result in Appendix C.2. For a less-skewed policy (Figure 5, Left), spurious rewards do not increase policy entropy even under unclipped training. For skewed policy (Figure 5, Right), policy entropy can increase during training, which explains the entropy growth in Figure 3 (Left). Whereas remedies from previous works merely slow early entropy collapse via regularization, we show it can be deliberately increased under spurious rewards while validation performance also improves, offering a complementary way to modulate entropy and to balance exploration–exploitation in RLVR.

Revisiting the role of clipping. Under random rewards, removing clipping drives policy entropy upward (more exploration), whereas clipping keeps entropy controlled and typically decreasing. Clipping acts as a trust-region–like regularizer: by capping per-token ratios, it limits step size and prevents gradient explosion. Without it, oversize updates can destabilize training—e.g., for `R1-Distill-Llama-8B`, validation on `MATH500` rises from 65.6% to 76.6% in 100 steps, then collapses around step 150 due to exploding gradients (Figure 3, Right). Clipped results for `DeepSeek-R1-Distill-Llama-8B` are shown in Figure 1.

Entropy and model performance. Under random rewards, entropy is the main quantity that changes, largely controlled by whether clipping is applied. Both higher (more exploration) and lower (more confident) entropy can coincide with better validation performance, but pushing entropy down (i.e., entropy minimization)—effectively what clipping does—helps only in regimes where the initial policy already concentrates on correct trajectories (strong model on easy data). On harder data or with weaker models, entropy minimization can entrench wrong modes and stall or degrade performance. We present further experiment results and discussions on entropy minimization in Appendix C.3.

2.3 Reward Misalignment: Who can Benefit from Random Rewards?

From empirical observations in this and prior work, we note two regularities under random-reward training. First, consistent with Shao et al. [61], weaker models tend to improve less. Crucially, *model strength* is training-dataset-dependent: a model that performs well on an easier benchmark may perform poorly on a harder one. Second, shown in Figure 2, as baseline accuracy rises (e.g., toward 70%), training curves exhibit fewer oscillations and become more stable; at moderate accuracy (e.g., around 50%), they fluctuate markedly. To account for why a model may improve under random rewards, we analyze the phenomenon from the reward misalignment perspective.

We develop a theoretical framework to analyze the varying degrees of damage to correct rollouts under false-positive and false-negative reward misalignment schemes. A warm-up setup is provided in Appendix D.9, and we formalize the notion of asymmetric damage to correct rollouts from strong and weak base models in Definition D.10. Our theoretical results in Proposition D.11 and Theorem D.12 justify that correct rollouts from stronger models are more likely to benefit from random rewards, leading to more stable training with reduced variance.

Beyond model contamination. Our theory indicates that the observed performance gains should not be attributed to clipping bias or validation-set contamination. While prior work [61, 80] reports improvements under random rewards for potentially contaminated `Qwen-Math` models, our results show that these gains arise from a subtler interaction between policy entropy and reward misalignment. As reported by Shao et al. [61], base `Llama` models consistently degrade during training across trials. Under the reward-misalignment view, stronger models should be more likely to benefit, tending to improve during training. We test this by using a stronger distilled `Llama` variant with long chain-of-thought reasoning; neither its base nor teacher model exhibits contamination on `MATH500`. As shown in Figure 1, with a rollout length of 8192 tokens and all other hyperparameters matched to the `Qwen-Math` setup, we observe improvements comparable to those in Figure 2. This suggests that contamination of the validation set is unlikely to explain the gains under random rewards; moreover, the effect is not unique to `Qwen-Math` models, as evidenced by both Figure 4 (Left) and Figure 1.

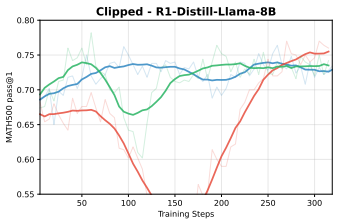

Figure 1: Distilled `Llama` model's performance on `MATH500` under random reward with DeepScaleR for training.

3 Conclusion

In this work, we examine the learning dynamics of RLVR under random reward setup. We prove that clipping bias is negligible under practical GRPO settings, reveal a link between clipping and policy entropy, and provide insight into policy entropy and model performance. Experiments across multiple model setup confirm that gains under random rewards arise from reward-entropy interplay and model-data regime, not contamination alone. We hope our results will facilitate the community's understanding and further development of techniques for RLVR training.

References

[1] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. In *Journal of Machine Learning Research*, 2021.

[2] S. Agarwal, Z. Zhang, L. Yuan, J. Han, and H. Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *ArXiv Preprint: 2505.15134*, 2025.

[3] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *ACL*, pages 12248–12267, 2024.

[4] M. G. Azar, Z. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, pages 4447–4455, 2024.

[5] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. In *ICLR*, 2019.

[6] H. Chen, G. He, L. Yuan, G. Cui, H. Su, and J. Zhu. Noise contrastive alignment of language models with explicit rewards. In *NeurIPS*, pages 117784–117812, 2024.

[7] L. Chen, M. Prabhudesai, K. Fragkiadaki, H. Liu, and D. Pathak. Self-questioning language models. *arXiv preprint arXiv:2508.03682*, 2025.

[8] P. Chen, Y. Xie, and Q. Zhang. Sicnn: Sparsity-induced input convex neural network for optimal transport. In *OPT 2024: Optimization for Machine Learning*, 2024.

[9] P. Chen, X. Chen, W. Yin, and T. Lin. Compo: Preference alignment via comparison oracles. *arXiv preprint arXiv:2505.05465*, 2025.

[10] P. Chen, X. Li, Z. Li, X. Chen, and T. Lin. Spectral policy optimization: Coloring your incorrect reasoning in grpo. *ArXiv Preprint: 2505.11595*, 2025.

[11] P Chen, Y Xie, and Q Zhang. Displacement-sparse neural optimal transport. *arXiv preprint arXiv:2502.01889*, 2025.

[12] D. Cheng, S. Huang, X. Zhu, B. Dai, W. X. Zhao, Z. Zhang, and F. Wei. Reasoning with exploration: An entropy perspective on reinforcement learning for llms. *arXiv preprint arXiv:2506.14758*, 2025.

[13] X. Chu, H. Huang, X. Zhang, F. Wei, and Y. Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. *ArXiv Preprint: 2504.02546*, 2025.

[14] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *ArXiv Preprint: 2110.14168*, 2021.

[15] G. Cui, Y. Zhang, J. Chen, L. Yuan, Z. Wang, Y. Zuo, H. Li, Y. Fan, H. Chen, W. Chen, Z. Liu, H. Peng, L. Bai, W. Ouyang, Y. Cheng, B. Zhou, and N. Ding. The entropy mechanism of reinforcement learning for reasoning language models. *ArXiv Preprint: 2505.22617*, 2025.

[16] H. Dong, W. Xiong, D. Goyal, Y. Zhang, W. Chow, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang. RAFT: Reward ranked fine-tuning for generative foundation model alignment. In *Transactions on Machine Learning Research*, 2023.

[17] H. Dong, W. Xiong, B. Pang, H. Wang, H. Zhao, Y. Zhou, N. Jiang, D. Sahoo, C. Xiong, and T. Zhang. RLHF workflow: From reward modeling to online RLHF. In *Transactions on Machine Learning Research*, 2024.

[18] Y. Fu, X. Wang, Y. Tian, and J. Zhao. Deep think with confidence. *ArXiv Preprint: 2508.15260*, 2025.

[19] Z. Gao, L. Chen, H. Luo, J. Zhou, and B. Dai. One-shot entropy minimization. *arXiv preprint arXiv:2505.20282*, 2025.

[20] D. Guo, D. Yang, H. Zhang, and et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. In *Nature*, volume 645, 2025.

[21] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *ArXiv Preprint: 2501.12948*, 2025.

[22] J. Guo, Z. Li, J. Qiu, Y. Wu, and M. Wang. On the role of preference variance in preference optimization. *arXiv preprint arXiv:2510.13022*, 2025.

[23] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS*, 2021.

[24] J. Hong, N. Lee, and J. Thorne. ORPO: Monolithic preference optimization without reference model. In *EMNLP*, pages 11170–11189, 2024.

[25] A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-STar: Training verifiers for self-taught reasoners. In *COLM*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

[26] Y. Huang and L. Yang. Gemini 2.5 pro capable of winning gold at imo 2025. *ArXiv Preprint: 2507.15855*, 2025.

[27] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. OpenAI o1 system card. *ArXiv Preprint: 2412.16720*, 2024.

[28] J. Kay, G. Van Horn, S. Maji, D. Sheldon, and S. Beery. Consensus-driven active model selection. *ArXiv Preprint: 2507.23771*, 2025.

[29] Team Kimi. Kimi k2: Open agentic intelligence. *ArXiv Preprint: 2507.20534*, 2025.

[30] J. Koch, L. Langosco, J. Pfau, J. Le, and L. Sharkey. Objective robustness in deep reinforcement learning. *arXiv Preprint: 2105.14111*, 2021.

[31] L. Langosco Di Langosco, J. Koch, L. Sharkey, J. Pfau, and D. Krueger. Goal misgeneralization in deep reinforcement learning. 2022.

[32] G. Li, Y. Wei, Y. Chi, and Y. Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. In *Operation Research*, volume 72, 2023.

[33] Z. Li, T. Xu, Y. Zhang, Z. Lin, Y. Yu, R. Sun, and Z-Q. Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *ICML*, pages 29128–29163, 2024.

[34] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. In *ICLR*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

[35] T. Lin, C. Jin, and M. Jordan. Two-timescale gradient descent ascent algorithms for nonconvex minimax optimization. In *Journal of Machine Learning Research*, 2025.

[36] J. Liu. How does rl policy entropy converge during iteration? 2025. URL https://zhuanlan.zhihu.com/p/28476703733.

[37] M. Liu, G. Farina, and A. Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. *NeurIPS*, 2025.

[38] T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization. In *ICLR*, 2024. URL https://openreview.net/forum?id=xbjSwwrQOe.

[39] T. Liu, Z. Qin, J. Wu, J. Shen, M. Khalman, R. Joshi, Y. Zhao, M. Saleh, S. Baumgartner, J. Liu, et al. LiPO: Listwise preference optimization through learning-to-rank. In *NAACL*, page To appear, 2025.

[40] Z. Liu, M. Lu, S. Zhang, B. Liu, H. Guo, Y. Yang, J. Blanchet, and Z. Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *NeurIPS*, pages 138663–138697, 2024.

[41] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding R1-zero-like training: A critical perspective. *ArXiv Preprint: 2503.20783*, 2025.

[42] M. Luo, S. Tan, J. Wong, X. Shi, W. Y. Tang, M. Roongta, C. Cai, J. Luo, L. E. Li, R. A. Popa, and I. Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. 2025. Notion Blog.

[43] H. Ma, G. Fu, Z. Luo, J. Wu, and T.-Y. Leong. Exploration by random reward perturbation. *ArXiv Preprint: 2506.08737*, 2025.

[44] S. Mahankali, Z. Hong, and P. Agrawal. Does novelty-based exploration maximize novelty? 2022. URL https://srinathm1359.github.io/data/Does_Novelty-Based_Exploration_Maximize_Novelty.pdf.

[45] Y. Meng, M. Xia, and D. Chen. SimPO: Simple preference optimization with a reference-free reward. In *NeurIPS*, pages 124198–124235, 2024.

[46] R. Munos, M. Valko, D. Calandriello, M. Gheshlaghi Azar, M. Rowland, Z. Guo, Y. Tang, M. Geist, T. Mesnard, C. Fiegel, A. Michi, M. Selvi, S. Girgin, N. Momchev, O. Bachem, D. J. Mankowitz, D. Precup, and B. Piot. Nash learning from human feedback. In *ICML*, 2024.

[47] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999.

[48] O. Oertell, W. Zhan, G. Swamy, Z. S. Wu, K. Brantley, J. Lee, and W. Sun. Heuristics considered harmful: Rl with random rewards should not make llms reason. *Notion Blog*, 2025.

[49] I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *UAI*, 2013.

[50] A. Pal, D. Karkhanis, S. Dooley, M. Roberts, S. Naidu, and C. White. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *ArXiv Preprint: 2402.13228*, 2024.

[51] A. Pan, K. Bhatia, and J. Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *ICLR*, 2022.

[52] R. Y. Pang, W. Yuan, H. He, K. Cho, S. Sukhbaatar, and J. Weston. Iterative reasoning preference optimization. In *NeurIPS*, pages 116617–116637, 2024.

[53] C. Park, S. Han, X. Guo, A. Ozdaglar, K. Zhang, and J. K. Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *ACL*, 2025.

[54] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[55] M. Prabhudesai, L. Chen, A. Ippoliti, K. Fragkiadaki, H. Liu, and D. Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

[56] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pages 53728–53741, 2023.

[57] R. Rafailov, J. Hejna, R. Park, and C. Finn. From r to q^*: Your language model is secretly a Q-function. In *COLM*, 2024. URL https://openreview.net/forum?id=kEVcNxtqXk.

[58] N. Razin, S. Malladi, A. Bhaskar, D. Chen, S. Arora, and B. Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *ICLR*, 2025. URL https://openreview.net/forum?id=uaMSBJDnRv.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. 2017. URL https://arxiv.org/abs/1707.06347.

[60] A. Setlur, C. Nagpal, A. Fisch, X. Geng, J. Eisenstein, R. Agarwal, A. Agarwal, J. Berant, and A. Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *ICLR*, 2025. URL https://openreview.net/forum?id=A6Y7AqlzLW.

[61] R. Shao, S. S. Li, R. Xin, S. Geng, Y. Wang, S. Oh, S. S. Du, N. Lambert, S. Min, R. Krishna, Y. Tsvetkov, H. Hajishirzi, P. W. Koh, and L. Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

[62] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv Preprint: 2402.03300*, 2024.

[63] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *ArXiv Preprint: 2402.03300*, 2024.

[64] H. Shen. On entropy control in llm-rl algorithms. *arXiv preprint arXiv:2509.03493*, 2025.

[65] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[66] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. ProgPrompt: Generating situated robot task plans using large language models. In *ICRA*, pages 11523–11530. IEEE, 2023.

[67] F. Song, B. Yu, M. Li, H. Yu, F. Huang, Y. Li, and H. Wang. Preference ranking optimization for human alignment. In *AAAI*, pages 18990–18998, 2024.

[68] Y. Song, J. Kempe, and Munos R. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*, 2025.

[69] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT Press, 1998.

[70] F. Tajwar, A. Singh, A. Sharma, R. Rafailov, J. Schneider, T. Xie, S. Ermon, C. Finn, and A. Kumar. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In *ICML*, pages 47441–47474, 2024.

[71] J. Tien, J. Z. He, Z. Erickson, A. Dragan, and D. S. Brown. Causal confusion and reward misidentification in preference-based reward learning. In *ICLR*, 2023.

[72] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving math word problems with process-and outcome-based feedback. *ArXiv Preprint: 2211.14275*, 2022.

[73] C. van Niekerk, R. Vukovic, B. M. Ruppik, H. Lin, and M. Gašić. Post-training large language models via reinforcement learning from self-feedback. *arXiv preprint arXiv:2507.21931*, 2025.

[74] J. Wang, J. Liu, Y. Fu, Y. Li, X. Wang, Y. Lin, Y. Yue, L. Zhang, Y. Wang, and K. Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents. *arXiv preprint arXiv:2509.09265*, 2025.

[75] P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *ACL*, pages 9426–9439, 2024.

[76] S. Wang, L. Yu, C. Gao, C. Zheng, S. Liu, R. Lu, K. Dang, X. Chen, J. Yang, Z. Zhang, Y. Liu, A. Yang, A. Zhao, Y. Yue, S. Song, B. Yu, G. Huang, and J. Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *ArXiv Preprint: 2506.01939*, 2025.

[77] Y. Wang, M. Yang, R. Dong, B. Sun, F. Liu, and L. H. U. Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. In *NeurIPS*, 2023.

[78] L. Weng. Reward hacking in reinforcement learning. *lilianweng.github.io*, 2024. URL https://lilianweng.github.io/posts/2024-11-28-reward-hacking/.

[79] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[80] M. Wu, Z. Zhang, Q. Dong, Z. Xi, J. Zhao, S. Jin, X. Fan, Y. Zhou, Y. Fu, Q. Liu, S. Zhang, and Q. Zhang. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. *ArXiv Preprint: 2507.10532*, 2025.

[81] J. Xiao, Z. Li, X. Xie, E. Getzen, C. Fang, Q. Long, and W. J. Su. On the algorithmic bias of aligning large language models with RLHF: Preference collapse and matching regularization. *ArXiv Preprint: 2405.16455*, 2024.

[82] W. Xiong, H. Dong, C. Ye, Z. Wang, H. Zhong, H. Ji, N. Jiang, and T. Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *ICML*, pages 54715–54754, 2024.

[83] R. Xu, K. Li, H. Wang, G. Kementzidis, W. Zhu, and Y. Deng. Rl-qesa: Reinforcement-learning quasi-equilibrium simulated annealing. In *ICML Workshop*, 2025.

[84] J. Yao, R. Cheng, X. Wu, J. Wu, and K. C. Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

[85] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *ArXiv Preprint: 2503.14476*, 2025.

[86] H. Yuan, Z. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. RRHF: Rank responses to align language models with human feedback. In *NeurIPS*, pages 10935–10950, 2023.

[87] L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, B. Shan, Z. Liu, J. Deng, H. Chen, R. Xie, Y. Lin, Z. Liu, B. Zhou, H. Peng, Z. Liu, and M. Sun. Advancing LLM reasoning generalists with preference trees. In *ICLR*, 2025. URL https://openreview.net/forum?id=2ea5TNVR0c.

[88] E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. STaR: self-taught reasoner bootstrapping reasoning with reasoning. In *NeurIPS*, pages 15476–15488, 2022.

[89] A. Zhang, N. Ballas, and J. Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv Preprint: 1806.07937*, 2018.

[90] K. Zhang, Y. Hong, J. Bao, H. Jiang, Y. Song, D. Hong, and H. Xiong. GVPO: Group variance policy optimization for large language model post-training. *ArXiv Preprint: 2504.19599*, 2025.

[91] Q. Zhang, H. Wu, C. Zhang, P. Zhao, and Y. Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[92] T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian. Noveld: A simple yet effective exploration criterion. In *NeurIPS*, 2021.

[93] Z. Zhang, C. Zheng, Y. Wu, B. Zhang, R. Lin, B. Yu, D. Liu, J. Zhou, and J. Lin. The lessons of developing process reward models in mathematical reasoning. *ArXiv Preprint: 2501.07301*, 2025.

[94] W. Zhao, P. Aggarwal, S. Saha, A. Celikyilmaz, J. Weston, and I. Kulikov. The majority is not always right: Rl training for solution aggregation. *ArXiv Preprint: 2509.06870*, 2025.

[95] X. Zhao, Z. Kang, A. Feng, S. Levine, and D. Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

[96] Y. Zhao, R. Joshi, T. Liu, M. Khalman, M. Saleh, and P. J. Liu. SLiC-HF: Sequence likelihood calibration with human feedback. *ArXiv Preprint: 2305.10425*, 2023.

[97] T. Zheng, T. Xing, Q. Gu, T. Liang, X. Qu, X. Zhou, Y. Li, Z. Wen, C. Lin, W. Huang, Q. Liu, G. Zhang, and Z. Ma. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025.

Contents for Appendix

A Preliminaries

RLVR & GRPO. RLVR assigns a binary outcome-based reward $\mathbf{r}(\mathbf{x}, \mathbf{y}')$ to a sampled response $\mathbf{y}'$ from prompt $\mathbf{x}$ by comparing it against the ground-truth answer $\mathbf{y}$. To learn an optimized policy via these reward, policy–gradient methods [69, 79] aim to maximize

$$J(\theta) = \mathbb{E}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[\,\mathbf{r}(\mathbf{x}, \mathbf{y})\,],$$

where ρ is the prompt distribution and π_θ denotes the LLM policy. The parameter update at each iteration is $\theta \leftarrow \theta + \eta\,\nabla_\theta J(\theta)$. In practice, trajectories are generated by an older policy $\pi_{\theta_{\text{old}}} \neq \pi_{\theta_{\text{new}}}$, but we still wish to estimate the gradient for π_θ. Using importance sampling, it can be rewritten as

$$J(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho,\ \mathbf{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathbf{x})}\left[\frac{\pi_\theta(\mathbf{y}\mid\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}\mid\mathbf{x})}\,\mathbf{r}(\mathbf{x}, \mathbf{y})\right].$$

Importance sampling can suffer from large variance when π_θ drifts from $\pi_{\theta_{\text{old}}}$. To stabilize training, we instead optimize the clipped surrogate objective

$$J(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho,\ \mathbf{y}\sim\pi_{\theta_{\text{old}}}(\cdot|\mathbf{x})}\left[\min\left\{\frac{\pi_\theta(\mathbf{y}\mid\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}\mid\mathbf{x})}\,\mathbf{r}(\mathbf{x}, \mathbf{y}),\ \texttt{clip}\left\{\frac{\pi_\theta(\mathbf{y}\mid\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}\mid\mathbf{x})},\ 1-\epsilon,\ 1+\epsilon\right\}\mathbf{r}(\mathbf{x}, \mathbf{y})\right\}\right].$$

Within this framework, GRPO [62] and related variants [10, 13, 41, 85, 90] estimate policy gradients using groups of samples. For each prompt $\mathbf{x}$, GRPO draws a set $\{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ from $\pi_{\theta_{\text{old}}}$ and maximizes

$$J(\theta) = \frac{1}{G}\sum_{i=1}^{G}\min\left\{\frac{\pi_\theta(\mathbf{y}_i\mid\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}_i\mid\mathbf{x})}\,A_i,\ \texttt{clip}\left\{\frac{\pi_\theta(\mathbf{y}_i\mid\mathbf{x})}{\pi_{\theta_{\text{old}}}(\mathbf{y}_i\mid\mathbf{x})},\ 1-\epsilon,\ 1+\epsilon\right\}A_i\right\},$$

where $\epsilon \in (0, 1)$ is a hyperparameter and the advantage A_i is computed from the group rewards as

$$A_i = \frac{\mathbf{r}(\mathbf{x}, \mathbf{y}_i) - \texttt{mean}\big(\{\mathbf{r}(\mathbf{x}, \mathbf{y}_1), \ldots, \mathbf{r}(\mathbf{x}, \mathbf{y}_G)\}\big)}{\texttt{std}\big(\{\mathbf{r}(\mathbf{x}, \mathbf{y}_1), \ldots, \mathbf{r}(\mathbf{x}, \mathbf{y}_G)\}\big)}, \tag{4}$$

with $\mathbf{r}(\mathbf{x}, \mathbf{y}_i) = 1$ if $\mathbf{y}_i$ matches the ground-truth final answer and $\mathbf{r}(\mathbf{x}, \mathbf{y}_i) = 0$ otherwise.

Random rewards for RLVR. Shao et al. [61] report striking gains on `MATH500` for the `Qwen-Math` family when models are fine-tuned with *purely random* rewards, a pattern not observed for several other model families. Wu et al. [80] likewise find substantial contamination of `Qwen-Math` via overlap with validation benchmarks such as `MATH500`, suggesting that RLVR may predominantly reinforce memorized solutions. Shao et al. [61] attribute these gains to PPO-style upper-clipping bias, which preferentially amplifies high-prior responses and thus exploits latent knowledge rather than teaching new reasoning skills. Nevertheless, Oertell et al. [48] contest this account, attributing purported gains to algorithmic heuristics and evaluation artifacts; in their experiments, random-reward fine-tuning does not reliably improve reasoning and can even degrade it. These conflicting results underscore how little is understood about RLVR learning dynamics and motivate two central questions: **(i)** ***Can random rewards improve model performance, and under what conditions?*** **(ii)** ***Does clipping bias furnish a genuine learning signal, and if not, what is its true role?*** Following prior work, our empirical validation primarily focuses on `MATH500`. We also provide further discussion of broader usage of random reward in classic RL in Appendix B.1.

Policy entropy. Recent studies on RLVR probe how policy entropy can be leveraged to improve model performance. Intuitively, policy entropy measures the diversity of a policy's action distribution: a high-entropy policy spreads probability more evenly and samples a wider variety of responses, whereas a low-entropy policy concentrates probability on fewer actions, yielding more deterministic behavior. The prevailing view guards against "entropy collapse" to prevent premature convergence to a suboptimal, low-diversity policy [85]; at the token level, Wang et al. [76] likewise emphasize that minority high-entropy tokens are crucial for reasoning. Yet several papers report the opposite trend: lower entropy can help. Agarwal et al. [2] optimize an explicit entropy-minimization objective and observe gains, and Cui et al. [15] claim a monotonic law: better performance at lower entropy. These conflicting results leave the core question unresolved: **(iii)** ***Is there a direct relationship between policy entropy and policy performance?***

B Related Works

We provide a technical review to explain the difference across experiment setup and insight from the recent advancements in RLVR for LLM post training.

B.1 Spurious Reward for Reinforcement Learning

Spurious reward in general reinforcement learning. In this section, we provide broader context on how previous work in reinforcement learning for classic settings (non-LLM) used spurious rewards to facilitate the training process. First, spurious reward signals are closely tied to issues of generalization in RL, shown in [30, 31, 78, 89]. While the above works showed intentional uses of such rewards, spurious signals can also arise unintentionally, leading to reward misspecification and the phenomenon of reward hacking [51, 78], but note that such spurious-based reward hacking is shown in LLM RLHF learning as well [71].

On the other hand, the design of deliberated misaligned reward shaping can be traced back to *potential-based reward shaping* (PBRS) [47]. The key is to inject additional reward signals in principled ways (like PBRS) or with careful tuning so that the intended behavior is still optimal. After that, *Random Network Distillation* (RND) introduced by [5] became a state-of-the-art exploration method, being extended in its follow-up work [43, 44]. [32, 54, 77, 92] also proposes spurious rewards that encourage the agent or model to traverse state-space in ways that eventually uncover actual rewards. Spurious reward is also largely applied to directly improve agent exploration. One prominent theoretical idea is *Posterior Sampling for Reinforcement Learning* (PSRL) [49]. Chen et al. [8, 11], Xu et al. [83] also extends the similar heuristic-driven search into broader areas to encourage exploration.

Spurious reward for RLVR. In this section, we zoom into recent works on spurious rewards for RLVR. Beyond the headline empirical findings, the experimental setups in prior work differ in important ways. In Shao et al. [61]'s released code, the prompt does not include the usual `Qwen`-style instruction to place the final answer in a box; as they note, `Qwen-Math` is sensitive to prompt formatting, and the prompt composition will largely affect its baseline performance. In our experiments, we instead follow `verl` [65]'s default `Qwen` prompt, which explicitly asks the model to place the final answer in a boxed expression. This aligns with the RLVR verifier in `verl`, which extracts the boxed answer for scoring and reward provision. Aside from the prompt, we match Shao et al. [61] on all RLVR hyperparameters.

By contrast, Oertell et al. [48] use a different configuration: (i) a rollout-length cap of 1024 tokens (well below `Qwen-Math`'s 4096-token context window), (ii) a different training set (`MATH` [23] rather than DeepScaleR [42]), (iii) a substantially smaller learning rate (1×10^{-7} versus 5×10^{-7} in Shao et al. [61]), and (iv) a reduced batch size (64 versus 128 in [61]). The smaller learning rate changes the effective step size and can materially alter policy updates; the smaller batch size yields noisier estimates of the underlying random reward provision distribution. Given these differences, the reported experiment results in these two works are not directly comparable at least from empirical level.

B.2 LLM Post-Training

LLM entropy. Agarwal et al. [2] show that simply minimizing token-level entropy can substantially enhance an LLM's reasoning ability *without* verifiable, labeled feedback. They argue that entropy reduction makes models more confident in their highest-probability answers, thereby unlocking latent reasoning capability. We note that this mechanism closely resembles *clipped training under random rewards*, where updates primarily modulate entropy rather than exploit informative reward signals. However, we show that entropy minimization alone can drive the policy to a low-entropy yet suboptimal solution; thus, entropy should be treated as a stabilizing regularizer, and it should be cautious when using as the substitute for genuine RLVR learning signal.

Similarly entropy-minimizing idea has also in relate to model confidence, Prabhudesai et al. [55] connect lower policy entropy to higher model confidence and use the model's own low-entropy (high-confidence) rollouts as a reward signal, reporting notable gains across multiple benchmarks and base model families. More surprisingly, Gao et al. [19] demonstrate that even a single unlabeled example

can boost a model's reasoning via entropy minimization. In a similar vein, Entropy-Minimized Policy Optimization [91] and Zhao et al. [95] improve performance in an unsupervised setting by increasing the model's self-confidence. Relatedly, van Niekerk et al. [73] construct ranked preference datasets from the model's own confidence over answers and obtain comparable improvements without human feedback or external verification, suggesting that self-confidence can serve as a weak training signal.

Cui et al. [15] claim a simple but insightful empirical relation between policy entropy $\mathcal{H}$ and model performance R, with fitting coefficient a and b,

$$R = -a \exp\left(\mathcal{H}\right) + b, \qquad a > 0,$$

estimated from extensive experiments. This fit suggests that performance increases monotonically as entropy decreases, but also plateaus once entropy collapses early. Intuitively, as the model overreinforces certain token sequences, its output distribution becomes overconfident and loses exploratory capacity, creating a performance ceiling.

Still, multiple works propose different perspectives to avoid earlier-stage entropy collapse. [64] examines why standard entropy regularization often provides little benefit in RLVR training of LLMs, attributing it to the extremely large response space of LLMs and the sparsity of optimal outputs, and proposes an adaptive entropy control technique that uses a clamped entropy bonus with an automatically tuned coefficient. [68] shows that GRPO-style ORM yields strong accuracy gains but induces a systematic drop in output entropy and diversity, evident in lower `pass@n` scores compared to the base model. To counter this, a outcome-based exploration that introduces entropy-promoting bonuses at the level of final outcomes is proposed. Similarly, previous works [12, 74, 84, 97] also applied different techniques to control the entropy during RLVR training.

Reinforcement learning for LLM. Proximal Policy Optimization [59] has emerged as the foundation for using reward-based policy updates to enhance LLM capabilities and serves as a key component of RLHF. However, due to the computational and memory inefficiency of loading four models, many lightweight and adapted policy gradient update methods have been proposed [3, 20, 33, 37, 62]. Along with the development of verifiable reward methods [14, 25, 34, 60, 66, 72, 75, 88, 93], reinforcement learning has greatly facilitated the reasoning capabilities of LLMs, especially in solving mathematical problems.

Apart from training methods, recent works have also advanced post-processing and collaborative approaches to improve reasoning effectiveness. Kay et al. [28], Zhao et al. [94] propose consensus- and answer-aggregation-based methods to reinforce results under multi-model frameworks. Chen et al. [7] introduce a novel self-questioning framework, while Park et al. [53] present a practical framework for advancing online multi-agent collaborative reinforcement learning.

Further offline practices. *Direct preference alignment* approaches such as DPO [56] provide a simple, stable offline alternative to online RLHF. Numerous variants extend DPO with different objectives, including ranking formulations beyond pairwise comparisons [6, 16, 39, 67, 86] and lightweight methods that remove the reference model [24, 45]. Because DPO avoids training a reward model, the limited supply of human labels becomes a key bottleneck; to address this, subsequent work augments preference data using an SFT policy [96] or a refined SFT policy with rejection sampling [38]. The DPO loss has also been generalized to token-level MDPs with deterministic transitions—covering standard LLM fine-tuning [57]—and to broader RL settings [4]. Complementary work elicits human feedback online to mitigate distribution shift and over-parameterization [17, 82], improving performance on complex reasoning tasks [52]. A parallel line studies *unintentional alignment* and proposes remedies [9, 22, 40, 50, 58, 70, 81, 87]; for example, Razin et al. [58] introduce the CHES similarity to filter near-duplicate preference pairs, and Chen et al. [9] leverage comparison oracles (ComPO), showing that combining them with DPO alleviates unintentional alignment in practice. Attributed to *Gradient Descent-Ascent* (GDA) scheme [35], many recent works over Nash Learning from Human Feedback (NLHF) [46] arises along with RLHF.

C Empirical Evaluations and Further Discussions

C.1 Clipping and Model Performance

Setup. Following the exact same hyperparameter setup from Shao et al. [61], we apply random rewards with $\text{Bernoulli}\left(\frac{1}{2}\right)$ for `Qwen2.5-Math-7B` on the DeepScaleR dataset [42], with a batch

size of 128, group size of 16, decoding temperature of 1.0, clipping ratio of 0.2, learning rate of 5×10^{-7}, and KL coefficient of 0.

We conduct multiple consecutive runs with and without clipping using the `verl` framework [65]. The resulting training trajectories on the `MATH500` validation set, along with the clipping activation fraction during training, are presented in Figure 2. We use the default training prompt from `verl`, which instructs the model to enclose its final answer in a box for verifier validation (see further discussions for this in Appendix B.1). Notably, for `Qwen2.5-Math-7B`, the clipping activation ratio is far lower than typical levels in other base models:

Remark C.1 (Clipping activation frequency)**.** *Empirically, the clipping activation ratio is usually below* 1% *for general GRPO training. For specific* `Qwen2.5-Math-7B` *training, the clipping activation ratio never exceeds* 0.2%*, with expected activation probability* $\mathbb{E}[I_t] \approx 0.001$.

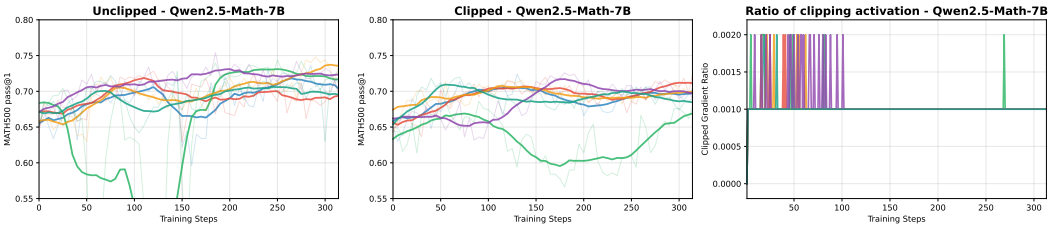

Figure 2: Training trajectories across multiple runs of `MATH500` validation accuracy over unclipped training (**Left**), clipped training (**Middle**), and clipping activation ratio during training (**Right**).

Results. As shown in Figure 2, enabling clipping can still cause the validation curve to decline. Instead, even without clipping, the validation performance still improves in most cases. These results indicate that upper-clipping bias is not the factor driving model improvement under random reward. We present a numerical instantiation of Theorem 2.3 using specific `Qwen-Math` training hyperparameters:

Remark C.2. *Numerically, we have* $\eta = 5 \times 10^{-7}$, $\varepsilon = 0.2$, $p = 0.001$, $G = 16$, $M = \sqrt{G-1}$, $R_\eta^{\max} \approx 1 + 3.87 \times 10^{-6}$, *and* $\phi(R_\eta^{\max}) \approx 7.5 \times 10^{-12}$. *With rollout length* $L = 4096$, *the off-diagonal term becomes negligible compared to the diagonal term. Substituting the above values into Theorem 2.3, we obtain*

$$\frac{\mathbb{E}\big[|N_{\text{raw}}|\big]}{\mathbb{E}\big[|C_{\text{tot}}|\big]} \geq \frac{(1-2^{1-G})\,\eta(1-\eta^2)}{L^{-1/2}M\sqrt{2R_\eta^{\max}\phi(R_\eta^{\max})p} + M(R_\eta^{\max}-1)\min\left\{\sqrt{p}, \frac{\phi(R_\eta^{\max})}{\phi(1+\varepsilon)}\right\}} \approx 67.45.$$

This justifies that $\mathbb{E}\big[|N_{\text{raw}}|\big] \gg \mathbb{E}\big[|C_{\text{tot}}|\big]$ *in magnitude for hyperparameters used in practice. Therefore, we argue that clipping bias does not provide a meaningful learning signal even under the contaminated models and benchmarks, which is verified through both empirical results and theoretical analysis.*

C.2 Clipping and Policy Entropy

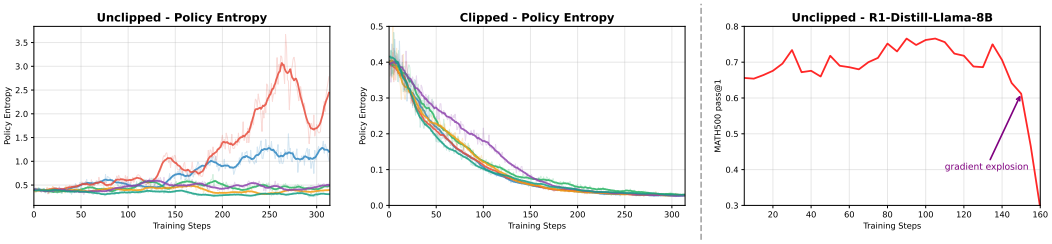

Figure 3: Policy entropy of `Qwen2.5-Math-7B` under random-reward training, with results for unclipped training (**Left**) and clipped training (**Middle**); Unclipped training with `R1-Distill-Llama-8B`, an example that leads to the gradient explosion (**Right**).

Clipping and entropy. Figure 3 (Left & Middle) shows that, under random rewards, disabling clipping can cause the policy entropy to rise throughout training, indicating increased exploration. Conversely, enabling clipping keeps the entropy in check and makes it decrease monotonically. This behavior confirms that clipping serves primarily as a *regularizer*: by capping each per-token ratio it restricts the effective step size, preventing abrupt updates that would otherwise push the policy far from its previous distribution. Apart from serving as a regularizer, the original role of clipping is to prevent gradient explosion, thereby providing additional stability during training.

Revisiting the role of clipping. When gradients grow large across different base-model trainings, clipping helps protect learning by preventing sudden explosions that could otherwise lead to significant drops in performance. When clipping is removed, this safeguard disappears; the optimizer can take oversize steps, injecting excessive exploration and destabilizing training. Thus, clipping contributes no additional learning signal, and its main purpose is to maintain optimization stability by enforcing a local trust region. Models with relatively large single-step gradient norms can collapse. A typical example is shown in Figure 3 (Right), training `R1-Distill-Llama-8B` without clipping initially improves the `MATH500` validation accuracy from 65.6% to 76.6% over the first 100 steps. Around step 150, however, the gradients explode, producing a cliff-like drop in performance. We show the clipped-training results for `DeepSeek-R1-Distill-Llama-8B` in Figure 1.

C.3 Policy Entropy and Model Performance

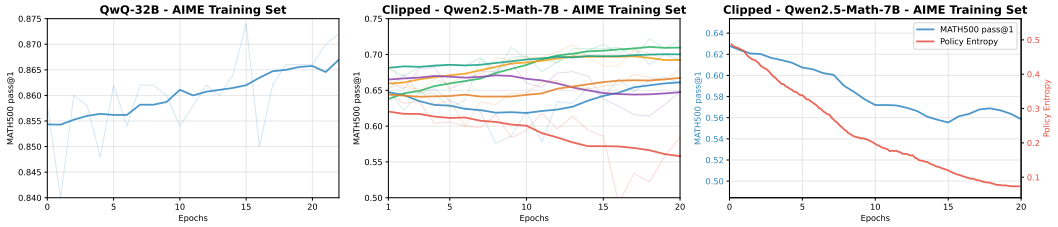

Figure 4: Results on AIME series as training set on `QwQ-32B` (**Left**) and `Qwen2.5-Math-7B` (**Middle**). With one specific example that shows entropy minimization would lead to sub-optimal policy under noisier and more difficult training environment (**Right**).

Random rewards provide a clean setting to study the relationship between policy entropy and performance: because rewards are independent of rollouts and have zero mean, there is no genuine learning signal, leaving entropy as the primary quantity that changes during training, largely governed by whether clipping is applied. Recall that in Figure 3, both higher and lower entropy can accompany improved performance. In practice, higher entropy corresponds to stronger *exploration*: the policy distribution is flatter and, from a general reinforcement learning perspective, better able to discover new trajectories. Conversely, lower entropy reflects greater confidence: the policy becomes more peaked and deterministic, concentrating probability mass on correct trajectories; in the RLVR setting, such concentration can also improve performance. However, we challenge this latter claim: convergence to a highly skewed, low-entropy policy does not necessarily improve performance, as shown in Figure 4 (Right). Hence, methods that directly minimize policy entropy or explicitly encourage a more confident policy should be cautiously applied.

Entropy minimization. Under random rewards, clipping effectively serves as a proxy for entropy minimization, pushing the policy toward a more peaked distribution with probability mass concentrated on a few trajectories. The utility of this effect depends critically on the model's initial distribution and the training data. For a strong model on a relatively easy dataset, the initial policy is already heavily concentrated on correct trajectories; further concentration can suffice, and entropy minimization appears beneficial. We provide a simple theoretical treatment of this case in § 2.3.

By contrast, as the training data become more difficult, incorrect trajectories may occupy the peak of the policy distribution. This yields noisy rollouts and updates and can drive convergence to an erroneous low-entropy solution. To illustrate, for `Qwen2.5-Math-7B`, we replace the milder `DeepScaleR` curriculum with the harder `AIME Past` series. Figure 4 (Middle) demonstrates results after 20 epochs of training, with all other hyperparameters matched to Figure 3. The training

result resembles a random walk with less meaningful improvement in validation accuracy relative to Figure 3. By contrast, the stronger `QwQ-32B` model (with rollout length set to 8192, all other hyperparameters match the 7B setup exactly) trained on the same `AIME` training set shows steady early-epoch gains (Figure 4, Left). This finding suggests that entropy minimization is regime-dependent and potentially risky given uncertainty in the initial policy distribution: it can help strong models on easier data by further concentrating mass on correct trajectories, but on harder data or for weaker models it may entrench incorrect modes and stall—or even degrade—performance. Accordingly, entropy-minimizing procedures (including clipping under random rewards) should be treated as regularizers rather than universal learning signals.

D Full Proofs and Technical Details

D.1 Proof Setup & Notations

One-step exponential gradient update. We note that such NPG-style update is recent used in previous works for GRPO analysis (e.g., [15]) can be derived from *Natural Policy Gradient* (NPG) [1], given objective with history denoted as s for past states:

$$\pi(\cdot|s) \in \arg\max_{\pi} \mathbb{E}_{a\sim\pi_{\text{old}}}\left[Q^{\pi_{\text{old}}}(s,a)\right] - \frac{1}{\eta} D_{\text{KL}}\big(\pi(\cdot|s), \pi_{\text{old}}(\cdot|s)\big),$$

which has been shown in [36] can be reformulated into the following exponential-step update:

$$\pi(a \mid s) \propto \pi_{\text{old}}(a \mid s)\exp\{\eta A(s,a)\} = \frac{\pi_{\text{old}}(a \mid s)\exp\{\eta A(s,a)\}}{\mathbb{E}_{a'\sim\pi_{\text{old}}(\cdot|s)}\left[\exp\{\eta A(s,a')\}\right]}$$

To further facilitate the analysis, we derive the following exponential-step update proposition:

Proposition D.1 (Single-step exponentiated-gradient update)**.** *Let $\pi_{\text{old}}(\cdot \mid h)$ be a policy and let $\hat{A}(a,h)$ denote the action (token)-level advantage from rollout $\mathbf{y}_i$. Define the new policy $\pi_{\text{new}}(\cdot \mid h)$ by the NPG-style exponentiated-gradient step (see Appendix D.3 for proof):*

$$\pi_{\text{new}}(a \mid h) = \frac{\pi_{\text{old}}(a \mid h)\,\exp\big(\eta\hat{A}(a,h)\big)}{Z(h)}, \quad Z(h) = \sum_{a'} \pi_{\text{old}}(a' \mid h)\exp\big(\eta\hat{A}(a',h)\big),$$

for a small learning rate $\eta > 0$. Then, for each action a and context h, the following holds:

$$\log \pi_{\text{new}}(a \mid h) = \log \pi_{\text{old}}(a \mid h) + \eta\hat{A}(a,h) - \tfrac{1}{2}\eta^2 + \mathcal{O}(\eta^3).$$

Furthermore, the token-level importance ratio $r(a \mid h) = \frac{\pi_{\text{new}}(a|h)}{\pi_{\text{old}}(a|h)}$ can be written as

$$r(a \mid h) = \exp\big(\eta\hat{A}(a,h) - \tfrac{1}{2}\eta^2 + \mathcal{O}(\eta^3)\big). \tag{5}$$

Because prior analyses of GRPO's policy entropy largely use an NPG-style approximation, we first provide a technical justification for its validity and then apply it to analyze GRPO's clipping bias.

NPG-update for Clipping Analysis. Such NPG-update has widely used in previous works for GRPO entropy analysis. As the first work to study the clipping effect in GRPO, we briefly review technical details in GRPO that motivates our reduction to an NPG-style update for analyzing clipping. Algorithm 1 summarizes the iterative procedure from Shao et al. [62]. In the outer loop (line 2), a *reference policy* is set once per iteration (line 3), and the per-step objective may include a KL penalty that constrains the updated policy π_θ to stay close to π_{ref}, thereby controlling step size and preventing excessive drift.

Recent "zero-RL" setups (e.g., DAPO [85]), which is also adopted in the empirical evaluation setup from Shao et al. [61], set the KL coefficient to zero, effectively removing the explicit KL term from the objective. Matching this setting, we likewise drop the KL penalty in our analysis. In this regime, the outer loop would not affect the following analysis.

In the middle loop (line 4), which is for standard GRPO training step, the model samples each macro-batch from dataset, which is update-style agnostic. The key difference between exact-GRPO- and NGP-style update happens in the inner loop (line 10). First, μ is a constant hyperparameter

Algorithm 1 Iterative Group Relative Policy Optimization

Require: initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; hyperparameters ε, β, μ

$\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration $= 1, \ldots, I$ **do** ▷ *Outer loop: for KL penalty calculation*
  $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for step $= 1, \ldots, M$ **do** ▷ *Middle loop: for macro-batch sampling*
    Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
    Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
    Compute rewards $\{r_i\}_{i=1}^{G}$ for each sampled output o_i by running r_φ
    Compute $\hat{A}_{i,t}$ for the t-th token of o_i via group-relative advantage estimation
    for GRPO iteration $= 1, \ldots, \mu$ **do** ▷ *Inner loop: policy update via micro-batches*
      Update the policy model π_θ by maximizing the GRPO objective
    end for
  end for
  Update r_φ through continuous training using a replay mechanism
end for

Ensure: π_θ

for the number of actual updates per macro batch, used to improve sample efficiency and better optimize the surrogate while clipping limits drift from π_{old}. Therefore, the statement *for GRPO iteration* $= 1, \ldots, \mu$ performs μ optimizer steps on the *same* mini-batch to maximize the clipped GRPO surrogate. At each step, importance ratio $r_{i,t} = \pi_\theta(\mathbf{y}_{i,t} \mid \mathbf{x}) / \pi_{\text{old}}(\mathbf{y}_{i,t} \mid \mathbf{x})$ are recomputed and the loss $\frac{1}{G} \sum_{i,t} \min\{r_{i,t}\hat{A}_{i,t},\ \text{clip}(r_{i,t}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\}$ is backpropagated.

In GRPO, the μ-step inner loop produces a chain of micro-updates whose importance ratios r evolve across steps, making the *expected* contribution of clipping analytically intractable unless one specifies the per-step *clip-activation rate* (the expected fraction of tokens/micro-batches with $r \notin [1-\varepsilon,\ 1+\varepsilon]$). This rate is model- and dataset-dependent and is only available empirically. Conditioning on the empirically measured activation rate, we collapse the μ clipped micro-steps into a single NPG-update with actual model-specific token-level expected clipping activation ratio. This surrogate preserves the first-order effect of clipping and enables tractable bounds for our theoretical results. Comparing to recent works that directly used NPG for GRPO analysis, our setup for clipping analysis is validly justified, facilitating the later theoretical derivation and without unjustified oversimplification.

Notation. Throughout the proofs, we use several notational variants for the same underlying quantities. The GRPO (action- or response-level) advantage is denoted by A, A_i, or $A(a, h)$, following standard GRPO formulations. The token-level advantage is denoted by $\hat{A}_t$ or $\hat{A}(a, h)$ and is used in the analysis of § 2.1. Note that both response level A and token level $\hat{A}$ are numerically the same. However, the clipping-based, reparameterized advantage is denoted by A_* and is used in Theorem 2.4; it differs from the general GRPO advantage A as specified in Lemma D.4.

For policy notation in GRPO, we abbreviate $\pi_{\theta_{\text{old}}}$ and π_θ to π_{old} and π_{new}, respectively, in the following theoretical statements and proofs.

D.2 GRPO Advantage Distribution under Random Rewards

In this subsection, we study the distribution of GRPO advantage and its basic statistics. Recall the definition of GRPO advantage in Eq. (4), we notice that A_i is not well-defined if all G samples in a group receive the same reward because the standard deviation in the denominator is 0. In practice, these two cases lead to zero gradient update. Based on this, we set $A_i = 0$ if all samples receive the same reward, which occurs with probability 2^{1-G}.

Lemma D.2. *Fix $G \geq 2$. Let $(R_1, \ldots, R_G)$ be i.i.d.* Bernoulli($\frac{1}{2}$). *Define*

$$\overline{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad S := \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j - \overline{R})^2}, \qquad A_i := \frac{R_i - \overline{R}}{S}.$$

Then the following holds:

(a) A_i *has symmetric distribution around* 0 *and thus* $\mathbb{E}[A_i^{2k-1}] = 0$ *for all* $k \in \mathbb{N}^+$*;*

(b) $|A_i| \le M := \sqrt{G-1}$*;*

(c) $\mathbb{E}[A_i^{2k}] \ge 1 - 2^{1-G}$ *for all* $k \in \mathbb{N}^+$ *with equality holding when* $k = 1$*.*

Proof. We prove three statements one by one as follows.

(a) Let $\tau : \{0,1\}^G \to \{0,1\}^G$ be $\tau(r_1, \dots, r_G) = (1 - r_1, \dots, 1 - r_G)$. If $\bar{r} = \frac{1}{G}\sum_j r_j$ and $r' = \tau(r)$, then $\bar{r}' = 1 - \bar{r}$ and

$$r'_j - \bar{r}' = (1 - r_j) - (1 - \bar{r}) = -(r_j - \bar{r}).$$

Hence $S(r') = S(r)$ and $A_i(r') = -(A_i(r))$. Since $(R_1, \dots, R_G)$ is i.i.d. Bernoulli($\frac{1}{2}$), its law is invariant under τ, so $A_i \overset{d}{=} -A_i$ and thus $\mathbb{E}[A_i^{2k-1}] = 0$.

(b) Write $x_j := R_j - \bar{R}$ so that $\sum_{j=1}^G x_j = 0$ and $S^2 = \frac{1}{G}\sum_{j=1}^G x_j^2$. Since $\sum_{j \ne i} x_j = -x_i$, Cauchy-Schwarz gives

$$(G-1)\sum_{j \ne i} x_j^2 \ge \left(\sum_{j \ne i} x_j\right)^2 = x_i^2.$$

Therefore

$$GS^2 = \sum_{j=1}^G x_j^2 \ge x_i^2 + \frac{1}{G-1}x_i^2 = \frac{G}{G-1}x_i^2,$$

and hence $|A_i| = |x_i|/S \le \sqrt{G-1}$.

(c) Let $K := \sum_{j=1}^G R_j \sim \text{Binomial}(G, \frac{1}{2})$ and $p := K/G$. On $\{1 \le K \le G-1\}$ we have

$$S = \frac{\sqrt{K(G-K)}}{G}, \qquad A_i = \begin{cases} \sqrt{\frac{1-p}{p}}, & R_i = 1, \\ -\sqrt{\frac{p}{1-p}}, & R_i = 0. \end{cases}$$

Hence for $m \in \mathbb{N}^+$,

$$\mathbb{E}\left[A_i^{2m} \mid K\right] = p\left(\frac{1-p}{p}\right)^m + (1-p)\left(\frac{p}{1-p}\right)^m =: f_m(p).$$

Write $x := \frac{p}{1-p} > 0$ so $f_m(p) = \frac{x^m + x^{-(m-1)}}{1+x}$. Define $h_m(x) := x^m + x^{-(m-1)} - x - 1$. Then

$$h_m''(x) = m(m-1)\,x^{m-2} + m(m-1)\,x^{-(m+1)} \;\ge\; 0, \qquad x > 0,$$

and $h_m(1) = h_m'(1) = 0$. By convexity, $h_m(x) \ge 0$ for all $x > 0$, hence $f_m(p) \ge 1$ for all $p \in (0,1)$. Taking expectations and using construction $A_i = 0$ on $\{K \in \{0, G\}\}$ yields

$$\mathbb{E}[A_i^{2m}] = \sum_{k=1}^{G-1} \binom{G}{k} 2^{-G} f_m\left(\frac{k}{G}\right) \ge \sum_{k=1}^{G-1} \binom{G}{k} 2^{-G} = 1 - 2^{1-G}.$$

For $m = 1$ we have $f_1(p) \equiv 1$, so equality holds. This completes the proof.

□

D.3 Proposition D.1

Proof. By definition,

$$\pi_{\text{new}}(a \mid h) = \frac{\pi_{\text{old}}(a \mid h)\, e^{\left(\eta\, \hat{A}(a,h)\right)}}{\sum_{a'} \pi_{\text{old}}(a' \mid h)\, e^{\left(\eta\, \hat{A}(a',h)\right)}} = \frac{\pi_{\text{old}}(a \mid h) e^{\left(\eta\, \hat{A}(a,h)\right)}}{Z(h)}.$$

Taking logarithms yields

$$\log \pi_{\text{new}}(a \mid h) = \log \pi_{\text{old}}(a \mid h) + \eta\, \hat{A}(a,h) - \log Z(h). \tag{6}$$

It remains only to expand $\log Z(h)$ for small η. We have

$$Z(h) = \sum_{a'} \pi_{\text{old}}(a' \mid h)\, e^{\left(\eta\, \hat{A}(a',h)\right)}.$$

Therefore, for small η we can expand each exponential to

$$e^{\left(\eta\, \hat{A}(a',h)\right)} = 1 + \eta\, \hat{A}(a',h) + \tfrac{1}{2}\eta^2 \left[\hat{A}(a',h)\right]^2 + \mathcal{O}(\eta^3).$$

Substituting into $Z(h)$ gives

$$\begin{aligned} Z(h) &= \sum_{a'} \pi_{\text{old}}(a' \mid h) \Big[1 + \eta\, \hat{A}(a',h) + \tfrac{1}{2}\eta^2\, \hat{A}(a',h)^2 + \mathcal{O}(\eta^3) \Big] \\ &= \Big(\sum_{a'} \pi_{\text{old}}(a' \mid h) \Big) + \eta \sum_{a'} \pi_{\text{old}}(a' \mid h)\, \hat{A}(a',h) + \tfrac{1}{2}\eta^2 \sum_{a'} \pi_{\text{old}}(a' \mid h)\, \hat{A}(a',h)^2 + \mathcal{O}(\eta^3) \\ &= 1 + \tfrac{1}{2}\, \eta^2 + \mathcal{O}(\eta^3). \end{aligned}$$

Hence

$$\log Z(h) = \tfrac{1}{2}\, \eta^2 + \mathcal{O}(\eta^3).$$

Substituting back into Eq. (6) yields

$$\log \pi_{\text{new}}(a \mid h) = \log \pi_{\text{old}}(a \mid h) + \eta\, \hat{A}(a,h) - \tfrac{1}{2}\eta^2 + \mathcal{O}(\eta^3),$$

Combining $\log \pi_{\text{old}}(a \mid h)$ with $\log \pi_{\text{new}}(a \mid h)$ then yields Eq. (5). □

D.4 Theorem 2.2

Proof. First, expanding the second moment $\mathbb{E}[C_{\text{tot}}^2] = \sum_{s,t} \mathbb{E}[D_s \hat{A}_s\, D_t \hat{A}_t]$, where $D_t := (\bar{r}_t - r_t) I_t$ and $A_s = A_t$ given the token-level advantage in the same rollout. We can then decompose it into diagonal and off-diagonal parts:

$$\mathbb{E}[C_{\text{tot}}^2] = \sum_{t=1}^{L} \mathbb{E}[D_t^2 \hat{A}_t^2] + \sum_{s \neq t} \mathbb{E}[D_s \hat{A}_s D_t \hat{A}_t]. \tag{7}$$

Diagonal terms. On the activation event $I_t = 1$ we are in the upper-clip regime, so $r_t \geq 1 + \varepsilon$ and

$$D_t = (\bar{r}_t - r_t) I_t = -(r_t - 1 - \varepsilon) I_t, \qquad |D_t| \leq (r_t - 1) I_t.$$

Because the indicator enforces $r_t \geq 1 + \varepsilon > 1$, we may use the inequality valid for $u \geq 1$,

$$(u-1)^2 \leq 2u\, \phi(u), \qquad \phi(u) = u \log u - u + 1,$$

to obtain

$$D_t^2 \leq (r_t - 1)^2 I_t \leq 2 r_t \phi(r_t) I_t. \tag{8}$$

By Lemma D.2, we have $|\hat{A}_t|^2 \leq M^2 = G - 1$ and $r_t \leq R_\eta^{\max} = \exp\{2M\eta\}$. Thus,

$$\mathbb{E}[D_t^2 \hat{A}_t^2] \leq M^2 \mathbb{E}[D_t^2] \leq M^2 \mathbb{E}[2 r_t \phi(r_t) I_t] \leq 2M^2 R_\eta^{\max} \phi(R_\eta^{\max}) p,$$

where the last inequality uses the fact that ϕ is strictly increasing on $[1, \infty)$. Summing over $t = 1, \ldots, L$ yields the desired upper bound for the diagonal terms:

$$\mathbb{E}[C_{\text{tot}}^2] \leq 2LM^2 R_\eta^{\max} \phi(R_\eta^{\max}) p + \sum_{s \neq t} \mathbb{E}[D_s \hat{A}_s D_t \hat{A}_t].$$

Off-diagonal terms. Let $X_t := D_t\,\hat{A}_t$. Recall that when $I_t = 1$, we have $|D_t| = (r_t - 1 - \varepsilon) \le (r_t - 1)$, and $r_t \le R_\eta^{\max}$ with $|\hat{A}_t| \le M = \sqrt{G-1}$, hence

$$|X_t| \le |D_t|\,|\hat{A}_t| \le M(R_\eta^{\max} - 1) I_t.$$

Therefore

$$\sum_{s \neq t} \mathbb{E}[|X_s X_t|] \le M^2 (R_\eta^{\max} - 1)^2 \sum_{s \neq t} \mathbb{E}[I_s I_t] \le M^2\,(R_\eta^{\max} - 1)^2 \mathbb{E}[J^2] \tag{9}$$

where $J := \sum_{t=1}^{L} I_t$. Since ϕ is strictly increasing on $[1, \infty)$ and $\phi(u) \ge 0$ for $u \in (0, \infty)$ with equality holds only when $u = 1$, we have

$$I_t = \mathbf{1}_{\{r_t > 1+\varepsilon\}} \le \frac{\phi(r_t)}{\phi(1+\varepsilon)} \implies J \le \frac{1}{\phi(1+\varepsilon)} \sum_{t=1}^{L} \phi(r_t) \le \frac{L\phi(R_\eta^{\max})}{\phi(1+\varepsilon)}.$$

Notice that $\mathbb{E}[J^2] \le L\mathbb{E}[J] = L^2 p$ and $\mathbb{E}[J^2] \le L^2 \phi(R_\eta^{\max})/\phi(1+\varepsilon)$. Thus,

$$\sum_{s \neq t} \mathbb{E}[|X_s X_t|] \le M^2 (R_\eta^{\max} - 1)^2 L^2 \min\left\{ p, \frac{\phi(R_\eta^{\max})^2}{\phi(1+\varepsilon)^2} \right\}.$$

By Cauchy-Schwarz, $\mathbb{E}[|C_{\text{tot}}|] \le \sqrt{\mathbb{E}[C_{\text{tot}}^2]}$. Using $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$ for any $x, y \ge 0$, we have

$$\mathbb{E}[|C_{\text{tot}}|] \le M\sqrt{2LR_\eta^{\max}\phi(R_\eta^{\max})p} + M(R_\eta^{\max} - 1)L \min\left\{ \sqrt{p}, \frac{\phi(R_\eta^{\max})}{\phi(1+\varepsilon)} \right\}.$$

Finally, when $\eta > 0$ is small, we can directly see the order of magnitude of the above upper bound by using Taylor expansion of $R_\eta^{\max} = 1 + 2M\eta + \mathcal{O}(\eta^2)$ and $\phi(R_\eta^{\max}) = 2M^2\eta^2 + \mathcal{O}(\eta^3)$. Therefore, we obtain

$$\mathbb{E}[|C_{\text{tot}}|] \le M\sqrt{2Lp(2M^2\eta^2 + \mathcal{O}(\eta^3))} + 2M^2 L \min\left\{ \sqrt{p}\eta, \frac{2M^2\eta^3 + \mathcal{O}(\eta^4)}{\phi(1+\varepsilon)} \right\}.$$

This shows $\mathbb{E}[|C_{\text{tot}}|] \le \mathcal{O}\left(\eta\sqrt{L} + \min\{\eta\sqrt{p}L, \eta^3 L\}\right)$. □

D.5 Theorem 2.3

Proof. Following from Proposition D.1, we have

$$r_t = \frac{\pi_{\theta+\Delta\theta}(o_t \mid h_t)}{\pi_\theta(o_t \mid h_t)} = e^{\left(\eta\hat{A}_t - \frac{1}{2}\eta^2 + \mathcal{O}(\eta^3)\right)}. \tag{10}$$

Insert Eq. (10) into $N_{\text{raw}} = \sum_t r_t \hat{A}_t$:

$$\begin{aligned}
\mathbb{E}[|N_{\text{raw}}|] &\ge \left| \sum_{t=1}^{L} \mathbb{E}\left[\hat{A}_t e^{\eta\hat{A}_t - \frac{1}{2}\eta^2 + \mathcal{O}(\eta^3)} \right] \right| \\
&= \left| \sum_{t=1}^{L} \mathbb{E}\left[\hat{A}_t \left(1 + \eta\hat{A}_t + \frac{1}{2}\eta^2 + \mathcal{O}(\eta^3) + \frac{\eta^2\hat{A}_t^2}{2} - \eta^3\hat{A}_t + \frac{\eta^3\hat{A}_t^3}{6} \right) \right] + \mathcal{O}(\eta^4) \right| \\
&= \left| \sum_{t=1}^{L} \mathbb{E}\left[\hat{A}_t \left(\eta\hat{A}_t - \eta^3\hat{A}_t + \frac{\eta^3\hat{A}_t^3}{6} \right) \right] + \mathcal{O}(\eta^4) \right| \\
&= \left| \sum_{t=1}^{L} \left[\eta\,\mathbb{E}[\hat{A}_t^2] + \eta^3 \left(-\mathbb{E}[\hat{A}_t^2] + \frac{\mathbb{E}[\hat{A}_t^4]}{6} \right) \right] + \mathcal{O}(\eta^4) \right| \\
&= \eta\,L\,\mathbb{E}[\hat{A}_t^2] + \eta^3 L \left(-\mathbb{E}[\hat{A}_t^2] + \frac{\mathbb{E}[\hat{A}_t^4]}{6} \right) + \mathcal{O}(\eta^4 L),
\end{aligned}$$

where the second equality uses $\mathbb{E}[\hat{A}_t^{2k-1}] = 0$ for all $k \in \mathbb{N}^+$ from Lemma D.2. In addition, using the lower bound $\mathbb{E}[\hat{A}_t^{2k}] \geq 1 - 2^{1-G}$ for all $k \in \mathbb{N}^+$ from Lemma D.2, we have

$$\mathbb{E}[|N_{\text{raw}}|] \geq \left|(1-2^{1-G})L\eta(1-\eta^2) + \frac{(1-2^{1-G})L}{6}\eta^3 + \mathcal{O}(\eta^4 L)\right| \geq (1-2^{1-G})L\eta(1-\eta^2). \tag{11}$$

for all $\eta > 0$ small enough. Therefore, we have the following lower bound

$$\frac{\mathbb{E}[|N_{\text{raw}}|]}{\mathbb{E}[|C_{\text{tot}}|]} \geq \frac{(1-2^{1-G})\eta(1-\eta^2)}{L^{-1/2}M\sqrt{2R_\eta^{\max}\phi(R_\eta^{\max})p} + M(R_\eta^{\max}-1)\min\left\{\sqrt{p}, \frac{\phi(R_\eta^{\max})}{\phi(1+\varepsilon)}\right\}}.$$

If $\eta > 0$ is small enough, $\frac{\mathbb{E}[|N_{\text{raw}}|]}{\mathbb{E}[|C_{\text{tot}}|]} \geq \mathcal{O}\left(\frac{1-\eta^2}{L^{-1/2}+\min\{\sqrt{p},\eta^2\}}\right)$. We named this lower bound as the *Law of Clipping* between the magnitude of raw and clipped part. □

We present a numerical evaluation from actual parameter setup from the experiment of the derived bound in Remark C.2.

D.6 Detail for Eq. (3) under Random Rewards

Note that the policy entropy $\mathcal{H}(\pi)$ is defined as

Definition D.3 (Policy entropy)**.** For a policy π, its entropy over action set $a \in \mathcal{A}$ is defined as

$$\mathcal{H}(\pi) := -\mathbb{E}_{a\sim\pi(\cdot|s)}\big[\log \pi\,(a \mid h)\big] = -\sum_a \pi\,(a \mid h)\log \pi\,(a \mid h).$$

We first demonstrate why Eq. (3) fails under random reward:

$$\begin{aligned}\text{Cov}_{a\sim\pi_{\text{old}}(\cdot|s)}\big(\log \pi_{\text{old}}(a \mid s), A(s,a)\big) &= \mathbb{E}_a\big[\log \pi_{\text{old}}(a \mid s)\, A(s,a)\big] - \mathbb{E}_a\big[\log \pi_{\text{old}}(a \mid s)\big]\underbrace{\mathbb{E}_a[A(s,a)]}_{=0} \\ &= \mathbb{E}_a\big[\log \pi_{\text{old}}(a \mid s)\big]\underbrace{\mathbb{E}_a[A(s,a)]}_{=0} = 0.\end{aligned}$$

However, this prediction deviates from our empirical results, which show a clear relationship between clipping and policy entropy. Therefore, it is incorrect to apply Eq. (3) directly to entropy analysis for random-reward setup. The key limitation lies in the fact that Eq. (3) considers only first-order terms in the policy expansion while neglecting higher-order terms, and most importantly, under unclipped formulation. Our subsequent theoretical results then reveal a more comprehensive understanding of the effect of clipping and policy entropy.

D.7 Theorem 2.4

To establish the proof for Theorem 2.4, we first introduce the Lemma D.4 for advantage parameterization along with its proof:

Lemma D.4. *Consider action space $\mathcal{A}$ with current policy $\sum_{a\in\mathcal{A}}\pi_\theta(a) = 1$, under PPO/GRPO-style clipping with clipping ratio $\epsilon \in [0,1]$ and small step size $\eta > 0$. Denote the response-level importance ratio $r(a) = \frac{\pi_{\text{new}}(a)}{\pi_{\text{old}}(a)}$, we have following reparameterization of ratio in respect to the unclipped advantage: $r(a) \approx 1 + \eta A(a) + \mathcal{O}(\eta^2)$. We assume that there exists a function $A : \mathcal{A} \to \mathbb{R}$ and a constant $C < \infty$ such that*

$$r(a) = 1 + \eta A(a) + \delta(a), \qquad |\delta(a)| \leq C\eta^2 \quad \text{for all } a \in \mathcal{A},$$

for sufficiently small η. Moreover, let

$$\text{Clip}_\varepsilon(x) := \min\{\max\{x, 1-\varepsilon\}, 1+\varepsilon\}, \qquad A_*(a) := \frac{\text{Clip}_\varepsilon(r(a)) - 1}{\eta}.$$

With an $O(\eta)$ remainder that is uniform in a, fixing clipping threshold ε, we then have

$$A_*(a) = \text{clip}(A(a), -\varepsilon/\eta, \varepsilon/\eta) + \mathcal{O}(\eta).$$

We introduce following Lemmas before establish the proof.

Lemma D.5 (Clipping as 1-Lipschitz projection)**.** *The map $x \mapsto \mathrm{Clip}_\varepsilon(x)$ is the metric projection onto the closed interval $[1-\varepsilon,\, 1+\varepsilon]$. In particular,*

$$\big|\mathrm{Clip}_\varepsilon(x) - \mathrm{Clip}_\varepsilon(y)\,\big| \;\le\; |x-y| \qquad \textit{for all } x, y \in \mathbb{R}.$$

Proof. $\mathrm{Clip}_\varepsilon(\cdot)$ is the Euclidean projection onto a convex closed set, hence non-expansive with Lipschitz constant 1. □

Lemma D.6 (Exact centering and scaling identity)**.** *For any $\eta > 0$ and any $y \in \mathbb{R}$,*

$$\frac{\mathrm{Clip}_\varepsilon(1+\eta y) - 1}{\eta} \;=\; \mathrm{clip}(y, -\varepsilon/\eta, \varepsilon/\eta)\,.$$

Proof. We check the three following cases: (i) $y \le -\varepsilon/\eta$ gives $\mathrm{Clip}_\varepsilon(1+\eta y) = 1-\varepsilon$ and the quotient $-\varepsilon/\eta$; (ii) $-\varepsilon/\eta \le y \le \varepsilon/\eta$ yields no clipping and the quotient y; (iii) $y \ge \varepsilon/\eta$ gives $+\varepsilon/\eta$. These coincide with the definition of $\mathrm{clip}(y, -\varepsilon/\eta, \varepsilon/\eta)$. □

In Lemma D.4, we assumed that $r(a) = 1 + \eta A(a) + \delta(a)$; we elaborate it in Remark D.7:

Remark D.7. *If $r(a) = \exp\{\eta A(a) + \eta^2 R_2(a)\}$ with $|R_2(a)| \le \tilde{C}$ uniformly, then $r(a) = 1 + \eta A(a) + \delta(a)$ with $\delta(a) = \frac{1}{2}\eta^2 A(a)^2 + \eta^2 R_2(a) + \mathcal{O}(\eta^3)$, so the assumption holds.*

Now, we establish the proof for Lemma D.4:

Proof of Lemma D.4. We consider the single-step GRPO update in logits-scale. Let the logits under old policy π_θ be $w(a) = \log \pi_\theta(a)$, we have

$$w_{\theta'}(a) = w_\theta(a) + \eta A(a),$$

which implies the unclipped policy update:

$$\pi_{\theta'}(a) \propto \pi_\theta(a) e^{\eta A(a)}.$$

Thus the unclipped ratio is:

$$r(a) = e^{\eta A(a)}.$$

Using the approximation $e^x \approx 1 + x + \mathcal{O}(x^2)$:

$$r(a) \approx 1 + \eta A(a) + \mathcal{O}(\eta^2)$$

This gives the reparameterization of importance ratio. By Lemma D.5,

$$\big|r_{\mathrm{clip}}(a) - \mathrm{Clip}_\varepsilon(1+\eta A(a))\,\big| \le |r(a) - (1+\eta A(a))| = |\delta(a)| \le C\eta^2.$$

Divide both sides by η and subtract $1/\eta$ inside the absolute value, we have

$$\left|A_*(a) - \frac{\mathrm{Clip}_\varepsilon(1+\eta A(a)) - 1}{\eta}\right| \le C\eta, \qquad \text{for all } a.$$

Following from Lemma D.6, we then have

$$\Big|A_*(a) - \mathrm{clip}(A(a), -\varepsilon/\eta, \varepsilon/\eta)\,\Big| \le C\eta \qquad \text{for all } a.$$

This establish the reparameterization of clipped advantage in terms of raw advantage surrogate $A(a)$. □

Proof of Theorem 2.4. Let $\zeta(a) = \log\big(1+\eta A_*(a)\big)$ and $\psi = \log(\mathbb{E}_{\pi_{\mathrm{old}}}\big[e^\zeta\big]) = \log(1+\eta\mu_*)$, then the clipped one-step update satisfies $\pi_{\mathrm{new}}(a) = \pi_{\mathrm{old}}(a)\, e^{\zeta(a)-\psi}$. Notice that

$$\pi_{\mathrm{new}}(a) - \pi_{\mathrm{old}}(a) = \frac{\pi_{\mathrm{old}}(a)(1+\eta A_*(a))}{1+\eta\mu_*} - \pi_{\mathrm{old}}(a) = \pi_{\mathrm{old}}(a)\frac{\eta(A_*(a)-\mu_*)}{1+\eta\mu_*}.$$

For convenience, we define a matrix $J := \mathrm{diag}(\pi_{\mathrm{old}}) - \pi_{\mathrm{old}}\pi_{\mathrm{old}}^{\mathrm{T}}$, where we treat the policy π_{old} as a vector of length equal to the size of action space. Then we have the above relation in compact form

$$\pi_{\mathrm{new}} - \pi_{\mathrm{old}} = \frac{\eta}{1+\eta\mu_*} JA_* = \eta JA_* + \mathcal{O}(\eta^2).$$

where we also regard A_* as a vector of length equal to the size of action space. In addition,

$$\begin{aligned}
\mathcal{H}(\pi_{\mathrm{new}}) &= -\sum_a \pi_{\mathrm{new}}(a) \log\left[\pi_{\mathrm{old}}(a) e^{\zeta(a)-\psi}\right] \\
&= -\sum_a \pi_{\mathrm{new}}(a) \left(\log \pi_{\mathrm{old}}(a) + \zeta(a) - \psi\right) \\
&= -\langle \pi_{\mathrm{new}}, \log \pi_{\mathrm{old}}\rangle - \langle \pi_{\mathrm{new}}, \zeta\rangle + \psi.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\mathcal{H}(\pi_{\mathrm{new}}) - \mathcal{H}(\pi_{\mathrm{old}}) &= -\langle \pi_{\mathrm{new}} - \pi_{\mathrm{old}}, \log \pi_{\mathrm{old}}\rangle - \langle \pi_{\mathrm{new}}, \zeta\rangle + \psi \\
&= -\eta\langle JA_*, \log \pi_{\mathrm{old}}\rangle - \frac{\eta^2}{2}\langle A_*, JA_*\rangle + \eta^2\mu_*\langle JA_*, \log \pi_{\mathrm{old}}\rangle + \mathcal{O}(\eta^3).
\end{aligned}$$

where we use the Taylor expansion $\zeta = \eta A_* - \eta^2 A_*^2/2 + \mathcal{O}(\eta^3)$ and $\psi = \eta\mu_* - \eta^2\mu_*^2/2 + \mathcal{O}(\eta^3)$. Therefore, by taking $\mathbb{E}[\cdot]$ on both sides, we have

$$\mathbb{E}[\mathcal{H}(\pi_{\mathrm{new}}) - \mathcal{H}(\pi_{\mathrm{old}})] = -\frac{\eta^2}{2}\mathbb{E}[\mathrm{Var}_{\pi_{\mathrm{old}}}(A_*)] + \mathcal{O}(\eta^3),$$

where we use the fact that $\mathbb{E}[\mu_*\langle JA_*, \log \pi_{\mathrm{old}}\rangle] = \mathcal{O}(\eta^2)$. □

D.8 Remark 2.5

D.8.1 GRPO in the Context of SGD

In § 2.2, we consider a special case of GRPO algorithm [63, Algorithm 1] under the setting of stochastic gradient descent. Since we study the algorithm without KL regularization, the outermost loop disappears. Now we consider the middle loop, to simplify the analysis, we assume batch size is 1 and the length of each sample is 1. In this case, we will simply call the middle loop the outer loop and the innermost loop the inner loop.

Under our simplification, for each outer loop, we generate G samples $y_1, \ldots, y_G \sim \pi_{\mathrm{old}} := \pi_{\theta_{\mathrm{old}}}$, where each sample is essentially a token, or we can call it an action. Now for each sample y_i we generate an independent reward $\mathbf{r}_i \sim \mathrm{Bernoulli}(\frac{1}{2})$. Then we compute A_i according to our convention in Appendix D.2. We run μ inner steps by using the update:

$$\theta_{t+1} = \theta_t + \eta\hat{g}_t, \quad \hat{g}_t = \frac{1}{G}\sum_{i=1}^{G} r_i^{(t)} A_i \mathbf{1}_{\{(A_i\geq 0 \wedge r_i^{(t)} \leq 1+\epsilon)\vee(A_i<0\wedge r_i^{(t)}\geq 1-\epsilon)\}} \nabla_\theta \log \pi_{\theta_t}(y_i).,$$

where $r_i^{(t)} := \pi_{\theta_t}(y_i)/\pi_{\theta_0}(y_i)$, $\theta_0 := \theta_{\mathrm{old}}$ and $\theta_{\mathrm{new}} := \theta_\mu$. For convenience, we can also use the abbreviation $\pi_{\mathrm{new}} := \pi_{\theta_{\mathrm{new}}}$ and $\pi_{t+1} := \pi_{\theta_t}$. Taking into account that we use softmax parameterization of the policy, we have

$$\begin{aligned}
\Delta\theta_t(a) &:= \theta_{t+1}(a) - \theta_t(a) \\
&= \frac{\eta}{G}\sum_{i=1}^{G} \hat{g}_t \\
&= \frac{1}{G}\sum_{i=1}^{G} r_i^{(t)} A_i \mathbf{1}_{\{(A_i\geq 0 \wedge r_i^{(t)} \leq 1+\epsilon)\vee(A_i<0\wedge r_i^{(t)}\geq 1-\epsilon)\}} \big(\mathbf{1}_{\{y_i=a\}} - \pi_t(a)\big).
\end{aligned}$$

To help the analysis, we write the update rule in policy space as

$$\pi_{t+1}(a) = \frac{\pi_t(a)\exp(\Delta\theta_t(a))}{\sum_b \pi_t(b)\exp(\Delta\theta_t(b))}.$$

We first consider the special case when $\mu = 1$, in this case, $r_i^{(t)} = 1$ and the clipping will never be activated for any $\epsilon > 0$. Therefore, we don't need to distinguish clipped or unclipped case.

D.8.2 Entropy Analysis for Unclipped Training

Theorem D.8. *Define the per-action advantage* $A(a) := \frac{1}{G}\sum_{i=1}^{G} A_i \mathbf{1}_{\{y_i=a\}}$. *Then, for small enough* $\eta > 0$, *we have*

$$\mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}}) = -\eta \mathbb{E}_{\pi_{\text{old}}}[(A - \mathbb{E}_{\pi_{\text{old}}}[A]) \log \pi_{\text{old}}] - \frac{\eta^2}{2}\left[\text{Var}_{\pi_{\text{old}}}(A) + \text{Cov}_{\pi_{\text{old}}}((A - \mathbb{E}_{\pi_{\text{old}}}[A])^2, \log \pi_{\text{old}})\right] + \mathcal{O}(\eta^3).$$

Furthermore, define $s_p := \sum_a \pi_{\text{old}}(a)^p$, $h_p := \sum_a \pi_{\text{old}}(a)^p \log \pi_{\text{old}}(a)$ *for all* $p \in \mathbb{N}^+$, *and*

$$\Phi(\pi) := (3s_3 - s_2 - 2s_2^2)h_1 + (1 + 2s_2)h_2 - 3h_3 + s_2 - 2s_3 + s_2^2.$$

Then we have

$$\mathbb{E}[\mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}})] = -\frac{(1 - 2^{1-G})\Phi(\pi_{\text{old}})}{G}\eta^2 + \mathcal{O}(\eta^3).$$

Proof. The update rule can be simplified as

$$\pi_{\text{new}}(a) = \frac{\pi_{\text{old}}(a)e^{\eta A(a)}}{Z(\eta)}, \quad Z(\eta) := \sum_b \pi_{\text{old}}(b)e^{\eta A(b)}.$$

Let $\psi(\eta) := \log Z(\eta)$ and $u(a) := \eta A(a) - \psi(\eta)$. Then $\pi_{\text{new}}(a) = \pi_{\text{old}}(a)e^{u(a)}$ and

$$\mathbb{E}_{\pi_{\text{old}}}[e^u] = \sum_a \pi_{\text{old}}(a)e^{\eta A(a) - \psi(\eta)} = 1.$$

The entropy change can be computed as

$$\Delta\mathcal{H} := \mathcal{H}(\pi_{\text{new}}) - \mathcal{H}(\pi_{\text{old}}) = -\mathbb{E}_{\pi_{\text{old}}}[e^u(\log \pi_{\text{old}} + u)] + \mathbb{E}_{\pi_{\text{old}}}[\log \pi_{\text{old}}].$$

Using Taylor expansion of cumulant generating function $\log \mathbb{E}_\pi[e^{\eta A}]$, we have

$$\psi(\eta) = \eta \mathbb{E}_\pi[A] + \frac{\eta^2}{2}\text{Var}_\pi(A) + \mathcal{O}(\eta^3).$$

$$u(a) = \eta(A(a) - \mathbb{E}_\pi[A]) - \frac{\eta^2}{2}\text{Var}_\pi(A) + \mathcal{O}(\eta^3).$$

Also using $e^u = 1 + u + u^2/2 + \mathcal{O}(u^3)$, we have

$$e^{u(a)} = 1 + \eta(A(a) - \mathbb{E}_\pi[A]) - \frac{\eta^2}{2}\text{Var}_\pi(A) + \frac{\eta^2}{2}(A(a) - \mathbb{E}_\pi[A])^2 + \mathcal{O}(\eta^3),$$

$$u(a)e^{u(a)} = \eta(A(a) - \mathbb{E}_\pi[A]) - \frac{\eta^2}{2}\text{Var}_\pi(A) + \eta^2(A(a) - \mathbb{E}_\pi[A])^2 + \mathcal{O}(\eta^3).$$

Combining the above expansion, we have

$$\begin{aligned}\Delta\mathcal{H} &= -\eta \mathbb{E}_{\pi_{\text{old}}}[(A - \mathbb{E}_{\pi_{\text{old}}}[A]) \log \pi_{\text{old}}] - \frac{\eta^2}{2}\mathbb{E}_{\pi_{\text{old}}}[(A - \mathbb{E}_{\pi_{\text{old}}}[A])^2 \log \pi_{\text{old}}] \\ &\quad - \frac{\eta^2}{2}\text{Var}_{\pi_{\text{old}}}(A)(1 - \mathbb{E}_{\text{old}}[\log \pi_{\text{old}})] + \mathcal{O}(\eta^3) \\ &= -\eta \mathbb{E}_{\pi_{\text{old}}}[(A - \mathbb{E}_{\pi_{\text{old}}}[A]) \log \pi_{\text{old}}] - \frac{\eta^2}{2}\left[\text{Var}_{\pi_{\text{old}}}(A) + \text{Cov}_{\pi_{\text{old}}}((A - \mathbb{E}_{\pi_{\text{old}}}[A])^2, \log \pi_{\text{old}})\right] + \mathcal{O}(\eta^3).\end{aligned}$$

Notice that $\mathbb{E}\left[\mathbb{E}_{\pi_{\text{old}}}[(A - \mathbb{E}_{\pi_{\text{old}}}[A]) \log \pi_{\text{old}}]\right] = \mathbb{E}_{\pi_{\text{old}}}[(\mathbb{E}[A] - \mathbb{E}_{\pi_{\text{old}}}[\mathbb{E}[A]]) \log \pi_{\text{old}}] = 0$. We then compute $\mathbb{E}[\text{Var}_{\pi_{\text{old}}}(A)]$. Consider

$$\begin{aligned}\text{Var}_{\pi_{\text{old}}}(A) &= \sum_a \pi_{\text{old}}(a)A(a)^2 - \left(\sum_a \pi_{\text{old}}(a)A(a)\right)^2 \\ &= \sum_a \pi_{\text{old}}(a)\left(\frac{1}{G}\sum_{i=1}^{G} A_i \mathbf{1}_{\{y_i=a\}}\right)^2 - \left(\sum_a \pi_{\text{old}}(a)\frac{1}{G}\sum_{i=1}^{G} A_i \mathbf{1}_{\{y_i=a\}}\right)^2 \\ &= \frac{1}{G^2}\sum_{i,j} A_i A_j \sum_a \pi_{\text{old}}(a)\mathbf{1}_{\{y_i=a\}}\mathbf{1}_{\{y_j=a\}} - \frac{1}{G^2}\sum_{i,j} A_i A_j \pi_{\text{old}}(y_i)\pi_{\text{old}}(y_j).\end{aligned}$$

By independence of A_i and y_i, we have

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}_{\pi_{\mathrm{old}}}(A)] &= \frac{1}{G^2}\sum_{i=1}^{G}\mathbb{E}[A_i^2]\left(\mathbb{E}\left[\sum_a \pi_{\mathrm{old}}(a)\mathbf{1}_{\{y_i=a\}}\right] - \mathbb{E}\left[\pi_{\mathrm{old}}(y_i)^2\right]\right) \\
&+ \frac{1}{G^2}\sum_{i\neq j}\mathbb{E}[A_iA_j]\left(\mathbb{E}\left[\sum_a \pi_{\mathrm{old}}(a)\mathbf{1}_{\{y_i=y_j=a\}}\right] - \mathbb{E}^2[\pi_{\mathrm{old}}(y_i)]\right) \\
&= \frac{1}{G^2}\sum_{i=1}^{G}\mathbb{E}[A_i^2]\left(\sum_a \pi_{\mathrm{old}}(a)^2 - \sum_a \pi_{\mathrm{old}}(a)^3\right) \\
&+ \frac{1}{G^2}\sum_{i\neq j}\mathbb{E}[A_iA_j]\left(\sum_a \pi_{\mathrm{old}}(a)^3 - \left(\sum_a \pi_{\mathrm{old}}(a)^2\right)^2\right).
\end{aligned}$$

Notice that by construction, $\sum_{i=1}^{G} A_i = 0$, so we have $\sum_{i=1}^{G} A_i^2 = -\sum_{i\neq j} A_iA_j$. Furthermore, $\mathbb{E}[A_i^2] = 1 - 2^{1-G}$ by Lemma D.2, thus,

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}_{\pi_{\mathrm{old}}}(A)] &= \frac{1-2^{1-G}}{G}\left[\sum_a \pi_{\mathrm{old}}(a)^2 - 2\sum_a \pi_{\mathrm{old}}(a)^3 + \left(\sum_a \pi_{\mathrm{old}}(a)^2\right)^2\right] \\
&= \frac{1-2^{1-G}}{G}(s_2 - 2s_3 + s_2^2).
\end{aligned}$$

We next compute $\mathbb{E}[\mathrm{Cov}_{\pi_{\mathrm{old}}}((A - \mathbb{E}_{\pi_{\mathrm{old}}}[A])^2, \log \pi_{\mathrm{old}})]$. Similarly, consider

$$\begin{aligned}
\mathrm{Cov}_{\pi_{\mathrm{old}}}((A - \mathbb{E}_{\pi_{\mathrm{old}}}[A])^2, \log \pi_{\mathrm{old}}) &= \sum_a \pi_{\mathrm{old}}(a)\log\pi_{\mathrm{old}}(a)A(a)^2 \\
&- 2\mathbb{E}_{\pi_{\mathrm{old}}}[A]\sum_a \pi_{\mathrm{old}}(a)\log\pi_{\mathrm{old}}(a)A(a) \\
&+ \mathbb{E}^2_{\pi_{\mathrm{old}}}[A]\sum_a \pi_{\mathrm{old}}(a)\log\pi_{\mathrm{old}}(a) \\
&- \mathrm{Var}_{\pi_{\mathrm{old}}}(A)\sum_a \pi_{\mathrm{old}}(a)\log\pi_{\mathrm{old}}(a).
\end{aligned}$$

Taking $\mathbb{E}[\cdot]$ on both sides, similarly, we have

$$\mathbb{E}\left[\mathrm{Cov}_{\pi_{\mathrm{old}}}((A - \mathbb{E}_{\pi_{\mathrm{old}}}[A])^2, \log \pi_{\mathrm{old}})\right] = \frac{1-2^{1-G}}{G}[(3s_3 - s_2 - 2s_2^2)\mathcal{H}_1 + (1+2s_2)\mathcal{H}_2 - 3\mathcal{H}_3].$$

In conclusion,

$$\mathbb{E}[\mathcal{H}(\pi_{\mathrm{new}}) - \mathcal{H}(\pi_{\mathrm{old}})] = -\frac{(1-2^{1-G})\Phi(\pi_{\mathrm{old}})}{G}\eta^2 + \mathcal{O}(\eta^3).$$

□

Remark D.9. *Theorem D.8 shows how the entropy changes after taking one inner step of GRPO starting from a given policy π_{old}. The sign of η^2 term depends on how skewed the given policy π_{old}. For example, we can consider the case when there are only two actions, namely, $\pi_{\mathrm{old}} = (p, 1-p)$ where $p \in (0,1)$. In this case we can easily compute $\Phi(\pi) := 2p^2(1-p)^2[2 - \log^2(p/(1-p))]$ and we know $\Phi(\pi) \geq 0$ if and only if $p \in [(1+e^{\sqrt{2}})^{-1}, (1-e^{-\sqrt{2}})^{-1}] \approx [0.196, 0.804]$. Thus, we can conclude that, in expectation, the entropy decreases when $p \in [0.196, 0.804]$ (not very skewed) and increases if $p > 0.804$ or $p < 0.196$ (very skewed). The numerical simulation results in Figure 5 also show that entropy has very different evolution patterns under different initial policy.*

Specifically, the policy entropy growth pattern only occurs at the relatively skewed policy initialization. This further highlights the applicability of injecting spurious reward without clipping into GRPO training to protect entropy, typically when the entropy already collapsed or degraded to a relatively skewed distribution.

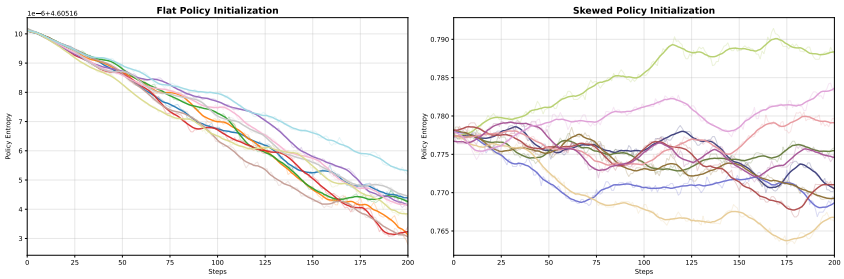

Figure 5: Simulation of policy entropy evolution over unclipped GRPO training. Each panel includes the result with 10 independent trails. Flat (relatively less-skewed) policy π initialization (**Left**); Skewed policy π initialization (**Right**).

D.9 Reward Misalignment: Setup and Theoretical Results

Consider binary outcome reward scheme (ORM), we establish the following reward misalignment setup:

Setup. For a prompt $\mathbf{x}$, draw G rollouts $\{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ from current policy π_θ. Partition the indices into correct and incorrect sets $C, I \subseteq \{1, \ldots, G\}$ with $|C| = n_c$, $|I| = n_i$, and $n_c + n_i = G$. We analyze two label errors:

- **(i) False positives (FP):** $R_j = 1$ for $j \in I$ (an incorrect rollout is rewarded).
- **(ii) False negatives (FN):** $R_k = 0$ for $k \in C$ (a correct rollout is not rewarded).

Specifically, we aim to explain:

- Why validation curves exhibit smaller fluctuations at higher accuracy and are more unstable at lower accuracy?
- Why stronger models are more likely to improve under random reward?

We first quantify reward misalignment as the loss of advantage mass that ought to accrue to correct rollouts but is diverted by random-reward mislabels:

Definition D.10 (Correct-response advantage loss)**.** Let $\{R_j\}_{j=1}^G$ be i.i.d. rewards with $R_j \sim \mathrm{Bernoulli}(\frac{1}{2})$, independent of correctness. Define the event counts $f := \#\mathrm{FP} = \sum_{j \in I} \mathbf{1}\{R_j = 1\}$ and $g := \#\mathrm{FN} = \sum_{k \in C} \mathbf{1}\{R_k = 0\}$, and let $T := \sum_{j=1}^G R_j = f + (n_c - g)$ be the total number of $+1$ rewards. Write $\bar{R} := T/G$ for the group-averaged reward. The GRPO class-wise centered reward sum over C is

$$\Sigma_C(f, g) := \sum_{k \in C} \big(R_k - \bar{R}\big) = (n_c - g) - \frac{n_c T}{G}.$$

As an "ideal" reference with no mislabels ($f = g = 0$), we have

$$\Sigma_C^{\mathrm{ideal}} = \sum_{k \in C} \Big(1 - \frac{n_c}{G}\Big) = n_c \Big(1 - \frac{n_c}{G}\Big).$$

Define the *damage* (advantage loss) as

$$\Delta(f, g) := \Sigma_C^{\mathrm{ideal}} - \Sigma_C(f, g). \tag{12}$$

Proposition D.11 (Unconditional global loss and variance)**.** *For any $n_c, n_i \geq 1$ and $G = n_c + n_i$, let $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$, $g \sim \mathrm{Binom}(n_c, \frac{1}{2})$ be independent, and $\Delta := \Delta(f, g)$ be defined in Eq. (12). Under i.i.d. $\mathrm{Bernoulli}(\frac{1}{2})$ rewards,*

$$\mathbb{E}[\Delta] = \frac{n_c(G - n_c)}{G}, \qquad \mathrm{Var}(\Delta) = \frac{n_c(G - n_c)}{4G}. \tag{13}$$

The expected damage decreases as the number of correct rollouts n_c increases, and the variance likewise decreases with n_c, indicating reduced variability for stronger models. Fluctuations are largest near the symmetric regime $n_c \approx n_i$, which aligns with our empirical observation in Figure 2. See Appendix D.10 for proof details. We next refine this picture by decomposing the damage via conditional means in Theorem D.12.

Theorem D.12. *Let $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$ and $g \sim \mathrm{Binom}(n_c, \frac{1}{2})$ be independent, and let Δ be defined in Eq. (12). For policy with more correct rollouts ($n_c > n_i$), we have*

$$\mathbb{E}[\Delta\, \mathbf{1}_{\{f>g\}}] \;\le\; \mathbb{E}[\Delta\, \mathbf{1}_{\{g>f\}}].$$

Moreover, as n_c increases on $[G/2, G]$, $\mathbb{E}[\Delta\, \mathbf{1}_{\{f>g\}}]$ will monotonically account for a smaller portion of $\mathbb{E}[\Delta]$.

D.10 Proposition D.11

Proof. Since $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$ and $g \sim \mathrm{Binom}(n_c, \frac{1}{2})$, we have $\mathbb{E}[f] = n_i/2$ and $\mathbb{E}[g] = n_c/2$. Plugging these into Eq. (12) yields the expectation:

$$\mathbb{E}[\Delta] = \frac{n_c}{G}\frac{n_i}{2} + \frac{n_i}{G}\frac{n_c}{2} = \frac{n_c(G-n_c)}{G}.$$

For the variance notice that $\mathrm{Var}(\mathrm{Binom}(n, \frac{1}{2})) = n/4$ and use independence of f and g:

$$\begin{aligned}\mathrm{Var}(\Delta) &= \left(\frac{n_c}{G}\right)^2 \mathrm{Var}(f) + \left(\frac{n_i}{G}\right)^2 \mathrm{Var}(g) \\ &= \left(\frac{n_c}{G}\right)^2 \frac{n_i}{4} + \left(\frac{n_i}{G}\right)^2 \frac{n_c}{4} \\ &= \frac{n_c n_i}{4G^2}(n_c + n_i) = \frac{n_c(G-n_c)}{4G}.\end{aligned}$$

This completes the proof. □

D.11 Theorem D.12

We first provide a conceptual analysis to under the counterintuitive results in Theorem D.12.

D.11.1 Conceptual proof through probabilistic method

Write $X := f$ and $Y := \#\{+1 \text{ in } C\}$. Then $Y \sim \mathrm{Binom}(n_c, \frac{1}{2})$, $g = n_c - Y$, and

$$\Delta = \frac{n_c}{G}X + \frac{n_i}{G}g = \frac{n_c}{G}X + \frac{n_i}{G}(n_c - Y) = \frac{n_i n_c}{G} + \frac{n_c}{G}X - \frac{n_i}{G}Y. \tag{14}$$

Let $Z := X + Y$ be the total number of $+1$'s over all G items; then $Z \sim \mathrm{Binom}(G, \frac{1}{2})$. Note that

$$\{f > g\} \iff \{X > (n_c - Y)\} \iff \{Z > n_c\}, \tag{15}$$

$$\{g > f\} \iff \{Z < n_c\}. \tag{16}$$

We now compute $\mathbb{E}[\Delta \mid Z]$. Condition on $Z = z$. Given $Z = z$, exactly z of the G positions carry a $+1$; by exchangeability, for each $j \in C$ we have

$$\Pr(\text{position } j \text{ is } +1 \mid Z = z) = \frac{z}{G}.$$

Hence

$$\mathbb{E}[Y \mid Z = z] = \sum_{j \in C} \Pr(j \text{ is } +1 \mid Z = z) = n_c\,\frac{z}{G},$$

and similarly

$$\mathbb{E}[X \mid Z = z] = n_i\,\frac{z}{G}.$$

Taking conditional expectations in Eq. (14) gives

$$\begin{aligned}\mathbb{E}[\Delta \mid Z = z] &= \frac{n_i n_c}{G} + \frac{n_c}{G}\,\mathbb{E}[X \mid Z = z] - \frac{n_i}{G}\,\mathbb{E}[Y \mid Z = z] \\ &= \frac{n_i n_c}{G} + \frac{n_c}{G}\cdot\frac{n_i z}{G} - \frac{n_i}{G}\cdot\frac{n_c z}{G} = \frac{n_i n_c}{G},\end{aligned}$$

which is constant in z.

Therefore, by the tower property and Eq. (15)–Eq. (16),

$$\mu_+ = \mathbb{E}[\Delta \mid f > g] = \mathbb{E}[\mathbb{E}[\Delta \mid Z] \mid Z > n_c] = \frac{n_i n_c}{G},$$
$$\mu_- = \mathbb{E}[\Delta \mid g > f] = \mathbb{E}[\mathbb{E}[\Delta \mid Z] \mid Z < n_c] = \frac{n_i n_c}{G}.$$

Re-writing it into conditional expectation:

$$\mathbb{E}[\Delta\, \mathbf{1}_{\{f>g\}}] = \mathbb{E}[\Delta \,|\, f > g] \Pr(f > g), \qquad \mathbb{E}[\Delta\, \mathbf{1}_{\{g<f\}}] = \mathbb{E}[\Delta \,|\, g > f] \Pr(g > f).$$

Given a strong model that generates more correct rollouts than incorrect ones, i.e., $n_c > n_i$, it is easy to see that $\Pr(f > g) < \Pr(f < g)$. We provide the proof below. First, consider the following combinatorial lemma:

Lemma D.13. *Fix integers $k > \ell \geq 0$. The map*

$$\Psi(n) = \frac{\binom{n}{k}}{\binom{n}{\ell}}, \qquad n \geq k,$$

is strictly increasing in n. In particular, if $n_c > n_i \geq k$, then

$$\frac{\binom{n_i}{k}}{\binom{n_i}{\ell}} < \frac{\binom{n_c}{k}}{\binom{n_c}{\ell}}.$$

Proof. Using falling factorials $\binom{n}{m} = \frac{n^m}{m!}$,

$$\Psi(n) = \frac{\ell!}{k!}\frac{n^k}{n^\ell} = \frac{\ell!}{k!}\prod_{j=0}^{k-\ell-1}(n-\ell-j).$$

The product has $k - \ell \geq 1$ strictly increasing linear factors in n, hence $Psi(n)$ is strictly increasing in n. □

Therefore, we have

Proposition D.14. *Let $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$ and $g \sim \mathrm{Binom}(n_c, \frac{1}{2})$ be independent with $n_c > n_i$. Then*

$$\Pr(f > g) < \Pr(g > f).$$

Proof. Write the probabilities in wedge form:

$$\Pr(f > g) = 2^{-(n_i+n_c)} \sum_{k>\ell} \binom{n_i}{k}\binom{n_c}{\ell}, \qquad \Pr(g > f) = 2^{-(n_i+n_c)} \sum_{\ell>k} \binom{n_i}{k}\binom{n_c}{\ell}.$$

Pair terms (k, ℓ) with $k > \ell$ against the swapped pair (ℓ, k) and compare weights

$$h_{k\ell} := \binom{n_i}{k}\binom{n_c}{\ell}, \qquad h_{\ell k} := \binom{n_i}{\ell}\binom{n_c}{k}.$$

By Theorem D.13,

$$\frac{h_{k\ell}}{h_{\ell k}} = \frac{\binom{n_i}{k}/\binom{n_i}{\ell}}{\binom{n_c}{k}/\binom{n_c}{\ell}} < 1, \qquad \text{as} \quad k > \ell, n_c > n_i,$$

so $h_{k\ell} < h_{\ell k}$ for every admissible pair. Summing over all $k > \ell$ gives $\Pr(f > g) < \Pr(g > f)$. □

This completes the proof for Theorem D.12. We further present the conditional variance analysis in Theorem D.15 to understand the accuracy oscillation during the training:

Remark D.15 (Conditional variance of damage and slice asymmetry)**.** *Let $G = n_c + n_i$ and let $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$, $g \sim \mathrm{Binom}(n_c, \frac{1}{2})$ be independent. Write $X := f$, $Y := \#\{+1 \text{ in } C\}$ so that $g = n_c - Y$, and let $Z := X + Y \sim \mathrm{Binom}(G, \frac{1}{2})$. For the damage Δ defined in Eq. (14), we have the exact identities*

$$\mathbb{E}[\Delta \mid Z = z] = \frac{n_i n_c}{G} \qquad \textit{and} \qquad \mathrm{Var}(\Delta \mid Z = z) = \frac{n_i(G - n_i)}{G - 1} \cdot \frac{z(G - z)}{G^2}.$$

Consequently, with $C := \dfrac{n_i(G - n_i)}{(G - 1)G^2}$ and $h(z) := z(G - z)$,

$$\mathrm{Var}(\Delta \mid f > g) = C\,\mathbb{E}\big[h(Z) \mid Z > n_c\big], \qquad \mathrm{Var}(\Delta \mid g > f) = C\,\mathbb{E}\big[h(Z) \mid Z < n_c\big].$$

Moreover, if $n_c > n_i$ (equivalently $n_c > G/2$), then

$$\mathrm{Var}(\Delta \mid f > g) \;<\; \mathrm{Var}(\Delta \mid g > f).$$

Proof. Define $X := f$ (false positives in I), $Y := \#\{+1 \text{ in } C\}$ so $g = n_c - Y$, and $Z := X + Y$. Since $f \sim \mathrm{Binom}(n_i, \frac{1}{2})$ and $Y \sim \mathrm{Binom}(n_c, \frac{1}{2})$ are independent, we have $Z \sim \mathrm{Binom}(G, \frac{1}{2})$ with $G = n_c + n_i$.

Conditional on $Z = z$, the z positive labels are uniformly scattered among G positions. The count X of positives falling inside the n_i indices of I is therefore

$$X \mid Z = z \;\sim\; \mathrm{Hypergeometric}\big(G,\; z,\; n_i\big),$$

so

$$\mathbb{E}[X \mid Z = z] = \frac{n_i z}{G}, \qquad \mathrm{Var}(X \mid Z = z) = n_i\,\frac{z}{G}\Big(1 - \frac{z}{G}\Big)\,\frac{G - n_i}{G - 1}.$$

Using Eq. (14) and $Y = Z - X$,

$$\Delta = \frac{n_c}{G} X + \frac{n_i}{G}(n_c - Y) = X - \frac{n_i}{G} Z + \frac{n_i n_c}{G}.$$

Hence, conditioned on $Z = z$,

$$\mathbb{E}[\Delta \mid Z = z] = \mathbb{E}[X \mid Z = z] - \frac{n_i}{G} z + \frac{n_i n_c}{G} = \frac{n_i n_c}{G},$$

which is constant in z, and

$$\mathrm{Var}(\Delta \mid Z = z) = \mathrm{Var}(X \mid Z = z) = \frac{n_i(G - n_i)}{G - 1} \cdot \frac{z(G - z)}{G^2}.$$

Let $C := \dfrac{n_i(G - n_i)}{(G - 1)G^2}$ and $h(z) := z(G - z)$. By total variance on any event A measurable w.r.t. Z and the constancy of $\mathbb{E}[\Delta \mid Z]$,

$$\mathrm{Var}(\Delta \mid A) = \mathbb{E}[\mathrm{Var}(\Delta \mid Z) \mid A] = C\,\mathbb{E}[h(Z) \mid A].$$

Since $\{f > g\} \iff \{Z > n_c\}$ and $\{g > f\} \iff \{Z < n_c\}$ (Eqs. (15) and (16)), the displayed slice formulas follow.

The binomial $Z \sim \mathrm{Binom}(G, \frac{1}{2})$ is symmetric about $G/2$, and $h(z) = z(G - z)$ is symmetric $h(G - z) = h(z)$ and strictly increasing on $\{0, 1, \ldots, \lfloor G/2 \rfloor\}$. Symmetry gives

$$\mathbb{E}[h(Z) \mid Z > n_c] = \mathbb{E}[h(Z) \mid Z < G - n_c].$$

When $n_c > G/2$, we have $0 \le G - n_c < n_c \le G$ and $G - n_c \le G/2$. For integers $0 \le a < b \le \lfloor G/2 \rfloor$ and strictly increasing h on $\{0, \ldots, \lfloor G/2 \rfloor\}$,

$$\mathbb{E}[h(Z) \mid Z < a] \;<\; \mathbb{E}[h(Z) \mid Z < b],$$

which follows from the convex combination decomposition of the latter and monotonicity on $[a, b)$. Taking $a := G - n_c$ and $b := n_c$ yields

$$\mathbb{E}[h(Z) \mid Z > n_c] = \mathbb{E}[h(Z) \mid Z < G - n_c] < \mathbb{E}[h(Z) \mid Z < n_c].$$

Multiplying by $C > 0$ proves $\mathrm{Var}(\Delta \mid f > g) < \mathrm{Var}(\Delta \mid g > f)$. Finally, unconditioning with $\mathbb{E}[h(Z)] = G(G - 1)/4$ recovers $\mathrm{Var}(\Delta) = \frac{n_c(G - n_c)}{4G}$ in agreement with Theorem D.11. □

D.11.2 Algebraic verification

Specifically, for the conditional expected damage part, we provide a rigorous algebraic derivation to verify the equivalence conclusion drawn from Appendix D.11.1.

First note that, writing $X := f$ and $Y := n_c - g$ (the number of $+1$ in C),

$$\Delta = \frac{n_c}{G}X + \frac{n_i}{G}g = \frac{n_c}{G}X + \frac{n_i}{G}(n_c - Y) = \frac{n_i n_c}{G} + \frac{n_c}{G}X - \frac{n_i}{G}Y.$$

Define the wedge weights

$$h_{k\ell} \;:=\; 2^{-G}\binom{n_i}{k}\binom{n_c}{\ell} \qquad (0 \le k \le n_i,\; 0 \le \ell \le n_c).$$

Then

$$\begin{aligned} p_+ &= \sum_{k>\ell} h_{k\ell}, \qquad p_- \;=\; \sum_{\ell>k} h_{k\ell}, \\ S_+ &:= \sum_{k>\ell} (n_c k + n_i \ell)\, h_{k\ell} = G\,\mathbb{E}\big[\Delta\, \mathbf{1}_{\{f>g\}}\big], \\ S_- &:= \sum_{\ell>k} (n_c k + n_i \ell)\, h_{k\ell} = G\,\mathbb{E}\big[\Delta\, \mathbf{1}_{\{g>f\}}\big]. \end{aligned}$$

Therefore

$$\mu_+ \;=\; \frac{S_+}{G\,p_+}, \qquad \mu_- \;=\; \frac{S_-}{G\,p_-}.$$

We claim the following *wedge proportionality* identities:

$$\sum_{k>\ell} (n_c k + n_i \ell)\binom{n_i}{k}\binom{n_c}{\ell} = n_i n_c \sum_{k>\ell} \binom{n_i}{k}\binom{n_c}{\ell}, \tag{17}$$

$$\sum_{\ell>k} (n_c k + n_i \ell)\binom{n_i}{k}\binom{n_c}{\ell} = n_i n_c \sum_{\ell>k} \binom{n_i}{k}\binom{n_c}{\ell}. \tag{18}$$

Proof of Eq. (17). Set $A_k := \binom{n_i}{k}$ and $B_\ell := \binom{n_c}{\ell}$, and write

$$A_{\le r} := \sum_{k=0}^{r} A_k, \qquad B_{\le r} := \sum_{\ell=0}^{r} B_\ell, \qquad B_{\le -1} := 0.$$

Two elementary transforms on the strict wedge $\{k > \ell\}$ are

$$\sum_{k>\ell} A_k B_\ell = \sum_{k\ge 0} A_k\, B_{\le k-1}, \tag{19}$$

$$\sum_{k>\ell} \ell\, A_k B_\ell = \sum_{\ell\ge 1} \ell\, B_\ell \sum_{k\ge \ell+1} A_k = \sum_{\ell\ge 1} \ell\, B_\ell\, (2^{n_i} - A_{\le \ell}). \tag{20}$$

Using $k\binom{n}{k} = n\binom{n-1}{k-1}$ and $\ell\binom{n}{\ell} = n\binom{n-1}{\ell-1}$, we compute

$$\sum_{k>\ell} n_c\, k\, A_k B_\ell = n_c n_i \sum_{k\ge 1} \binom{n_i - 1}{k-1} B_{\le k-1}, \tag{21}$$

$$\sum_{k>\ell} n_i\, \ell\, A_k B_\ell = n_i n_c \sum_{\ell\ge 1} \binom{n_c - 1}{\ell - 1} (2^{n_i} - A_{\le \ell}). \tag{22}$$

Summing Eq. (21)–Eq. (22) gives

$$\sum_{k>\ell} (n_c k + n_i \ell) A_k B_\ell = n_i n_c \left[\sum_{k\ge 1} \binom{n_i - 1}{k-1} B_{\le k-1} + \sum_{\ell\ge 1} \binom{n_c - 1}{\ell - 1} (2^{n_i} - A_{\le \ell})\right]. \tag{23}$$

We now show the bracket equals $\sum_{k\geq 0} A_k B_{\leq k-1}$ (the right-hand side of Eq. (19)). By Pascal's rule, $A_k = \binom{n_i-1}{k} + \binom{n_i-1}{k-1}$, hence

$$\sum_{k\geq 0} A_k B_{\leq k-1} = \sum_{k\geq 0} \binom{n_i-1}{k} B_{\leq k-1} + \sum_{k\geq 1} \binom{n_i-1}{k-1} B_{\leq k-1}. \tag{24}$$

Thus it suffices to prove

$$\sum_{k\geq 0} \binom{n_i-1}{k} B_{\leq k-1} = \sum_{\ell\geq 1} \binom{n_c-1}{\ell-1} (2^{n_i} - A_{\leq \ell}). \tag{25}$$

Expanding the right-hand side and swapping sums in the double sum,

$$\begin{aligned}\sum_{\ell\geq 1} \binom{n_c-1}{\ell-1} (2^{n_i} - A_{\leq \ell}) &= 2^{n_i} \sum_{\ell\geq 1} \binom{n_c-1}{\ell-1} - \sum_{\ell\geq 1} \binom{n_c-1}{\ell-1} \sum_{k\leq \ell} A_k \\ &= 2^{n_i} 2^{n_c-1} - \sum_{k\geq 0} A_k \sum_{\ell\geq \max\{1,k\}} \binom{n_c-1}{\ell-1}. \end{aligned} \tag{26}$$

Using the finite-tail identity $\sum_{\ell\geq r} \binom{n_c-1}{\ell-1} = 2^{n_c-1} - \sum_{m=0}^{r-2} \binom{n_c-1}{m}$, we get

$$\sum_{\ell\geq \max\{1,k\}} \binom{n_c-1}{\ell-1} = 2^{n_c-1} - \sum_{m=0}^{k-2} \binom{n_c-1}{m}. \tag{27}$$

Insert Eq. (27) into Eq. (26) and simplify:

$$\sum_{\ell\geq 1} \binom{n_c-1}{\ell-1} (2^{n_i} - A_{\leq \ell}) = \sum_{k\geq 0} A_k \sum_{m=0}^{k-2} \binom{n_c-1}{m}. \tag{28}$$

Finally, by another Pascal telescoping, $\sum_{m=0}^{k-2} \binom{n_c-1}{m} = \binom{n_c-1}{k-1}$, so Eq. (28) equals $\sum_{k\geq 0} \binom{n_i-1}{k} B_{\leq k-1}$, proving Eq. (25). Tracing back through Eq. (23)–Eq. (24)–Eq. (19) yields Eq. (17).

Proof of Eq. (18). The same argument applies on the strict wedge $\{\ell > k\}$, merely interchanging the roles of (n_i, A_k) and (n_c, B_ℓ). This gives Eq. (18).

This indicates that for a stronger model—one that produces more correct than incorrect rollouts—the asymmetry between false-positive and false-negative rewards can, by chance, still yield net improvement: because false positives are infrequent, most of the random mass transfer occurs within the correct set itself, so the advantage largely remains with correct trajectories.

By contrast, weaker models do not benefit from random rewards: when most rollouts are incorrect, the bulk of the mass transfer occurs within the incorrect set, making the model more likely to reinforce erroneous trajectories through randomly assigned rewards.