Mohamed bin Zayed
University of
Artificial Intelligence

NLP Reading Group

September 17th 2025

“GSPO & What is RL for LLMs”

Talk by: Nikolai Rozanov
(nikolai.rozanov@mbzuai.ac.ae)

Group Relative Policy Optimisation (GRPO)

GRPO: Policy Optimization History

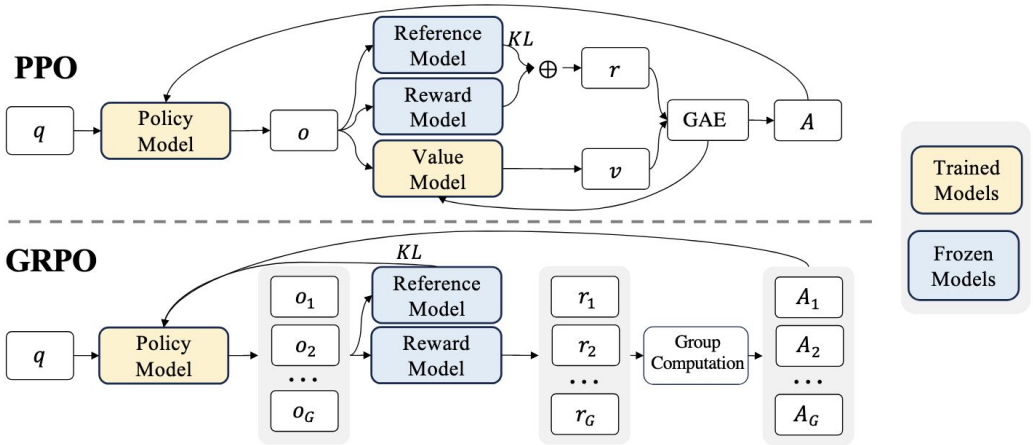


Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}, \quad (3)$$

GRPO: Policy Optimization History

1. Bellman, et al. ~1950s: **Dynamic Programming, Control Theory**
    a. Objective Function: E[R] = Sum (reward_t=0 + reward_t=1 + …)
    b. reward_t ~ Reward(given action a_t & states s_t, s_t-1)
    c. s_t ~ Transition(a_t, s_t-1)
    d. a_t ~ Policy(s_t-1)

GRPO: Policy Optimization History

1. Bellman, et al. ~1950s: **Dynamic Programming, Control Theory**
   a. Objective Function: E[R] = Sum (reward_t=0 + reward_t=1 + …)
   b. reward_t ~ Reward(given action a_t & states s_t, s_t-1)
   c. s_t ~ Transition(a_t, s_t-1)
   d. a_t ~ Policy(s_t-1)
2. Alexandrov ~1968, Williams 1992: **REINFORCE**
   a. What if Policy P(s, **theta**)? How do we optimize P?
   b. What is ∇E[R]?
   c. Answer: ∇E[R] = E[∇ logP(theta) * **R**]
   d. Better Estimator: ∇E[R] = E[∇ logP(theta) * **(R-b)**]
      i. (b… baseline)
      ii. A = R-b (A…Advantage)

GRPO: Policy Optimization History

1. Bellman, et al. ~1950s: **Dynamic Programming, Control Theory**
   a. Objective Function: E[R] = Sum (reward_t=0 + reward_t=1 + …)
   b. reward_t ~ Reward(given action a_t & states s_t, s_t-1)
   c. s_t ~ Transition(a_t, s_t-1)
   d. a_t ~ Policy(s_t-1)
2. Alexandrov ~1968, Williams 1992: **REINFORCE**
   a. What if Policy P(s, **theta**)? How do we optimize P?
   b. What is ∇E[R]?
   c. Answer: ∇E[R] = E[∇ logP(theta) * r] $\hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$
3. Trust Region Policy Optimization (TRPO)
   a.

$$\underset{\theta}{\text{maximize}}\, \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t - \beta \, \text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right]$$

GRPO: Policy Optimization History

1. Bellman, et al. ~1950s: **Dynamic Programming, Control Theory**
   a. Objective Function: E[R] = Sum (reward_t=0 + reward_t=1 + …)
   b. reward_t ~ Reward(given action a_t & states s_t, s_t-1)
   c. s_t ~ Transition(a_t, s_t-1)
   d. a_t ~ Policy(s_t-1)
2. Alexandrov ~1968, Williams 1992: **REINFORCE**
   a. What if Policy P(s, **theta**)? How do we optimize P?
   b. What is ∇E[R]?
   c. Answer: ∇E[R] = E[∇ logP(theta) * r] $\hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$
3. Trust Region Policy Optimization (TRPO)
   a.

$$\underset{\theta}{\text{maximize}}\, \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t - \beta \, \text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right]$$

GRPO: GRPO Algorithm Summary

$$\hat{\mathbb{E}}_t\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$ **REINFORCE**

$$= \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$ **TRPO**

$$\frac{1}{|o|}\sum_{t=1}^{|o|}\min\left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}A_t, \mathrm{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon\right)A_t\right] - \beta\,\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]$$ **PPO**

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \mathrm{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}, \quad (3)$$ **GRPO**

Mohamed bin Zayed University of Artificial Intelligence

GRPO: GRPO Algorithm Summary

1. Key Difference:
   a. Advantage Term (A)
   b. In GRPO we can save one model, and estimate from “Group Estimator”.

$$\left| \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})} A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right] - \beta\, \text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right.$$

PPO

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})} \hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL}\left[\pi_\theta || \pi_{ref}\right] \right\}, \quad (3)$$

GRPO

Mohamed bin Zayed University of Artificial Intelligence

GRPO: GRPO Algorithm

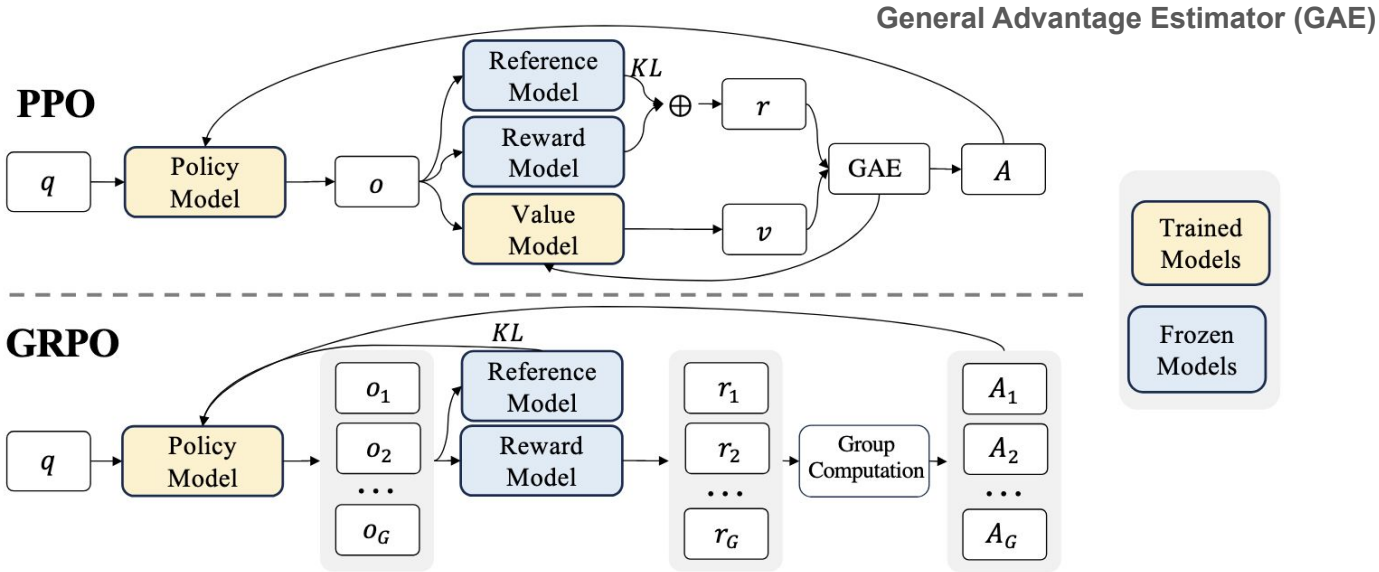
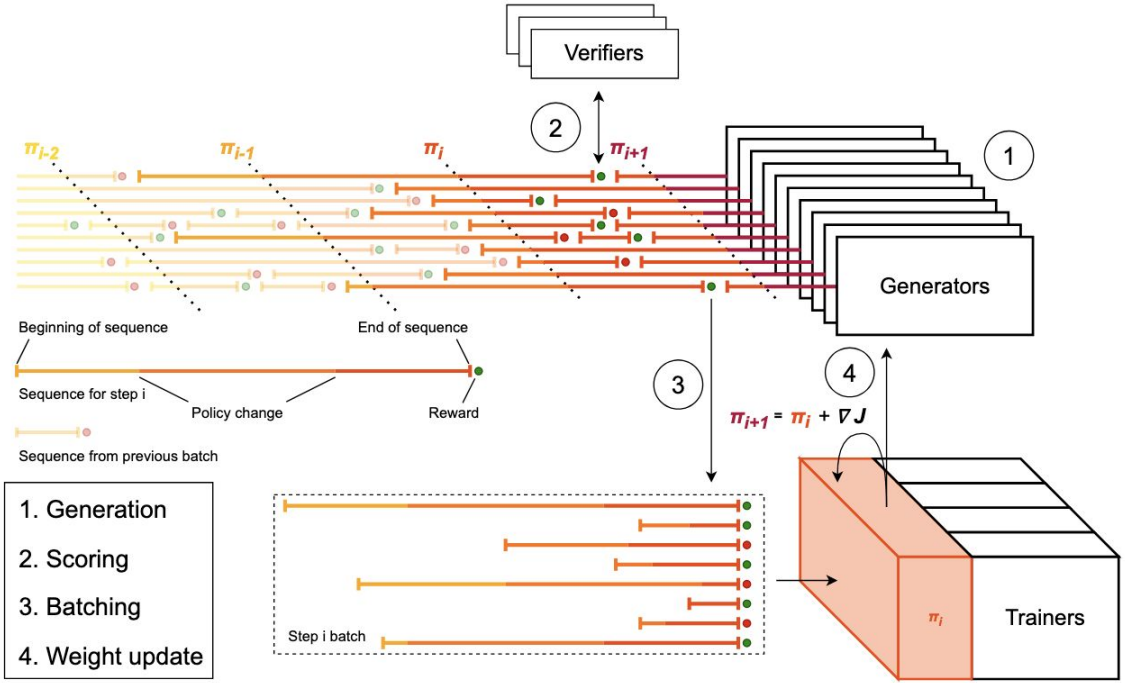


Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

GRPO: GRPO Training Pipeline



Group Sequence Policy Optimisation (**GSPO**)

GSPO: Group Sequence Policy Optimization

Group Sequence Policy Optimization

Chujie Zheng* Shixuan Liu Mingze Li Xiong-Hui Chen Bowen Yu*
Chang Gao Kai Dang Yuqiong Liu Rui Men An Yang Jingren Zhou
Junyang Lin

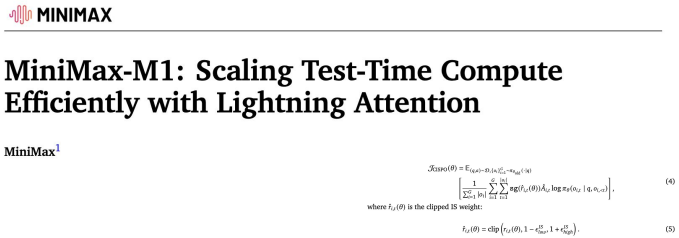
Qwen Team, Alibaba Inc.

29.07.2025

Mohamed bin Zayed University of Artificial Intelligence

GSPO: Motivation

1. Training Instability for GRPO:
   a. **CISPO** (MiniMax Paper (https://arxiv.org/pdf/2506.13585))
      i. Problem 1: Logprobs between Inference Engine (e.g. vllm) & Training Engine different
      ii. Problem 2: Clipping of Tokens that are important leads to failed learning
      iii. Solution: They Propose clipping importance ratio (not actual gradient update) & higher accuracy for log-probs
   b. GSPO:
      i. Problem 1: PPO/GRPO require clipping due to (locally) off-policy nature of training
      ii. Problem 2: **Fundamentally ill-posed objective**


MINIMAX

MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention

MiniMax[1]

$$\mathcal{J}_{\text{CISPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \text{sg}(\hat{\imath}_{i,t}(\theta))\hat{A}_{i,t} \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \right], \quad (4)$$

where $\hat{\imath}_{i,t}(\theta)$ is the clipped IS weight:

$$\hat{\imath}_{i,t}(\theta) = \text{clip}\left(r_{i,t}(\theta), 1-\epsilon_{\text{low}}^{\text{IS}}, 1+\epsilon_{\text{high}}^{\text{IS}}\right). \quad (5)$$

GSPO: Motivation (ill-posed objective)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(w_{i,t}(\theta)\widehat{A}_{i,t},\ \text{clip}\left(w_{i,t}(\theta),1-\varepsilon,1+\varepsilon\right)\widehat{A}_{i,t}\right)\right], \quad (2)$$

where G is the number of generated responses to each query x (i.e., the group size), and the importance ratio $w_{i,t}(\theta)$ and advantage $\widehat{A}_{i,t}$ of token $y_{i,t}$ are:

$$w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}, \quad \widehat{A}_{i,t} = \widehat{A}_i = \frac{r(x,y_i) - \text{mean}\left(\{r(x,y_i)\}_{i=1}^{G}\right)}{\text{std}\left(\{r(x,y_i)\}_{i=1}^{G}\right)}, \quad (3)$$

respectively, where all the tokens in y_i share the same advantage as $\widehat{A}_i$.

$$\mathbb{E}_{z\sim\pi_{\text{tar}}}[f(z)] = \mathbb{E}_{z\sim\pi_{\text{beh}}}\left[\frac{\pi_{\text{tar}}(z)}{\pi_{\text{beh}}(z)}f(z)\right]. \quad (4)$$

Crucially, this relies on averaging over multiple samples ($N \gg 1$) from the behavior distribution π_{beh} for the importance weight $\frac{\pi_{\text{tar}}(z)}{\pi_{\text{beh}}(z)}$ to effectively correct for the distributional mismatch.

In contrast, GRPO applies the importance weight $\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}$ at each token position t. Since this weight is based on a single sample $y_{i,t}$ from each next-token distribution $\pi_{\theta_{\text{old}}}(\cdot|x,y_{i,<t})$, it fails to perform the intended distribution-correction role. Instead, it introduces high-variance noise into the training

GSPO: Algorithm

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(s_i(\theta)\widehat{A}_i,\ \text{clip}\left(s_i(\theta),1-\varepsilon,1+\varepsilon\right)\widehat{A}_i\right)\right], \tag{5}$$

where we adopt the group-based advantage estimation:

$$\widehat{A}_i = \frac{r(x,y_i) - \text{mean}\left(\{r(x,y_i)\}_{i=1}^{G}\right)}{\text{std}\left(\{r(x,y_i)\}_{i=1}^{G}\right)}, \tag{6}$$

and define the importance ratio $s_i(\theta)$ based on sequence likelihood (Zheng et al., 2023):

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}\right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}\right). \tag{7}$$

GSPO: Results

Figure 1 shows that the training with GSPO proceeds stably throughout. We observe that **GSPO can deliver continuous performance improvement through increasing the training compute, regularly updating the query set, and extending the generation length**. Moreover, GSPO also demonstrates superior training efficiency over GRPO, achieving better training accuracy and benchmark performance under the same training compute and consumed queries. Finally, we have successfully applied GSPO to the RL training of the latest Qwen3 models, strongly proving the efficacy of GSPO in unleashing the power of RL scaling for large language models.

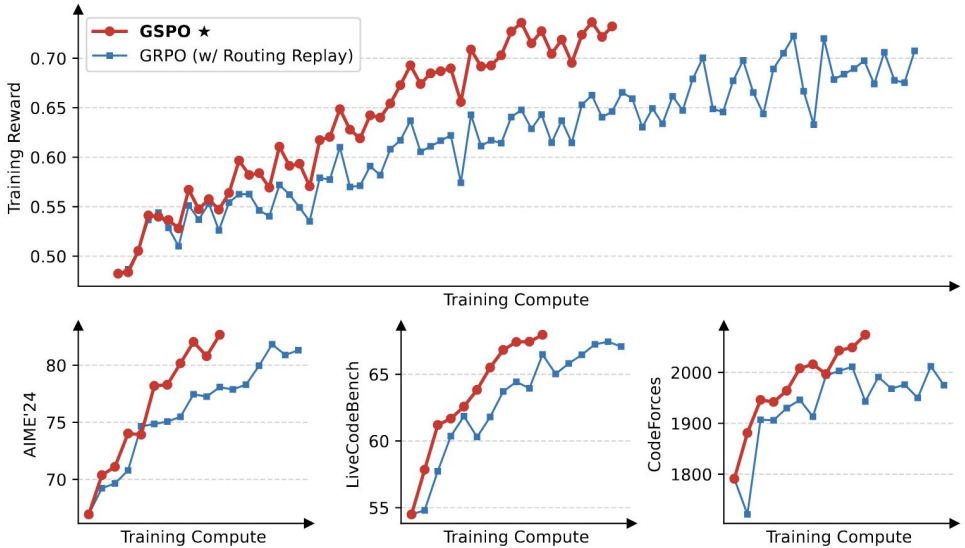


Figure 1: Training curves of a cold-start model fine-tuned from Qwen3-30B-A3B-Base. GSPO possesses remarkably higher training efficiency than GRPO.

GSPO: Results (Routing Replay)

Our Previous Approach To tackle this challenge, we previously employed the **Routing Replay** training strategy. Specifically, we cache the activated experts in $\pi_{\theta_{\text{old}}}$ and "replay" these routing modes in π_θ when computing the importance ratios $w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}$. In this way, for each token $y_{i,t}$, $\pi_\theta(y_{i,t}|x,y_{i,<t})$ and $\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})$ share the same activated network, so that we can restore the stability of the token-level importance ratios and ensure optimization of the consistent activated network across gradient updates. Figure 3 demonstrates that Routing Replay serves as an essential technique in the normal convergence of the GRPO training of MoE models.

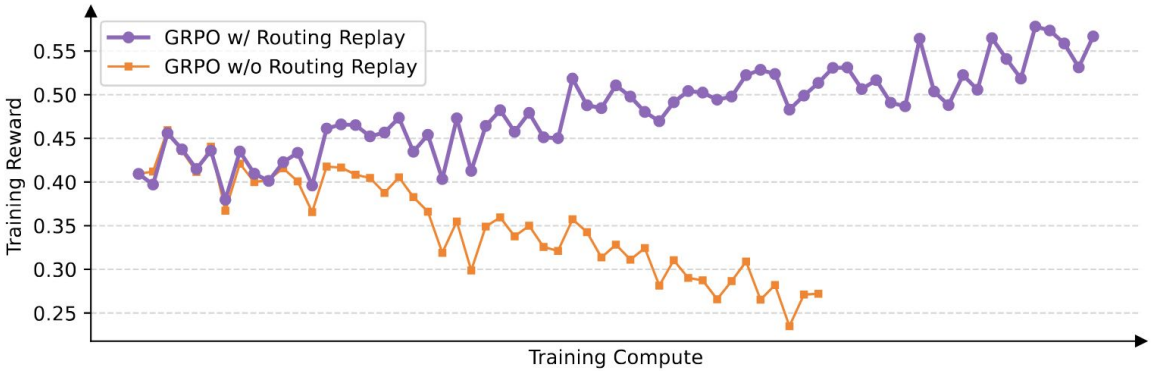


Figure 3: The Routing Replay strategy plays a critical role in the normal convergence of the GRPO training of MoE models.

GSPO: Benefits

1. Solves ill-posed problem (more stability)
2. Solves MoE sparse activation problem (full capacity of MoE)
3. Solves Floating-point problem (minimax) (faster training engine)
4. **Question:** Important tokens clipping? (needs more investigation)

Benefit of GSPO Although Routing Replay enables the GRPO training of MoE models to converge properly, its practice of reusing routing modes incurs additional memory and communication overhead and can also limit the actual capacity of the MoE model. In contrast, as shown in Figure 1, GSPO eliminates the dependency on Routing Replay and is fully capable of computing the importance ratios $s_i(\theta)$ conventionally, converging normally, and optimizing stably. The key insight is that GSPO focuses only on the sequence likelihood (i.e., $\pi_\theta(y_i|x)$) and is not sensitive to the individual token likelihood (i.e., $\pi_\theta(y_{i,t}|x, y_{i,<t})$). Since the MoE model always maintains its language modeling capability, the sequence likelihood will not fluctuate drastically. In summary, GSPO fundamentally resolves the expert-activation volatility issue in MoE models, obviating the need for complex workarounds like Routing Replay. This not only simplifies and stabilizes the training process but also allows the model to leverage its full capacity without artificial constraints.

5.4 Benefit of GSPO for RL Infrastructure

Given the precision discrepancies between training engines (e.g., Megatron) and inference engines (e.g., SGLang and vLLM), in practice, we typically use the training engine to recompute the likelihoods of sampled responses under the old policy $\pi_{\theta_{\text{old}}}$. However, GSPO uses only sequence-level, rather than token-level, likelihoods for optimization, and intuitively, the former is much more tolerant of precision discrepancies. Hence, GSPO makes it possible to directly use the likelihoods returned by the inference engine for optimization, thereby avoiding the need for recomputation with the training engine. This can be especially beneficial in scenarios like partial rollout and multi-turn RL and in the training-inference disaggregated frameworks.

Conclusion & Takeaways

Conclusions & Takeaways

1. Reading old-papers is very relevant (both for understanding & developing new methods, e.g. PPO) and for building new methods (e.g. fastText).

2. GRPO is quite a successful algorithm. However, careful analysis reveals short-comings.
3. Need to analyse carefully (CISPO, GSPO) to actually understand how to improve the algorithm:
    a. Floating Point stability,
    b. Clipping of Important Tokens,
    c. ill-posed GRPO objective,
    d. MoE different activations, etc.

Concrete Todos:

When doing **RL** training:

1. Look at token entropy
2. Look at clipping ratios (which tokens get clipped how often)
3. Look at training vs. inference (log-probs)
4. Look at noise level of gradient updates (need more normalization)
5. Look at general in-efficiencies in training setup

References

Bibliography

- GRPO: https://arxiv.org/pdf/2402.03300
- MiniMax: (CISPO): https://arxiv.org/pdf/2506.13585
- GSPO: https://arxiv.org/pdf/2507.18071
- DeepSeek-R1: https://arxiv.org/pdf/2501.12948
- PPO: https://arxiv.org/pdf/1707.06347
- MathShepard: https://arxiv.org/pdf/2312.08935
- Trust Region Policy Optimization (TRPO): https://arxiv.org/pdf/1502.05477 (that's the paper, where the p()/p_old estimate for the policy gradient comes from.
- Better Reasoning with Alignment: https://arxiv.org/pdf/2309.02144
- Weak to Strong Supervision (OpenAI): https://arxiv.org/pdf/2312.09390
- Magistral (Mistral): https://arxiv.org/abs/2506.10910