Clipping-Free Policy Optimization for Large Language Models

Anonymous ACL submission

Abstract

Reinforcement learning has become central to post-training large language models, yet dominant algorithms rely on clipping mechanisms that introduce optimization issues at scale, including zero-gradient regions, reward hacking, and training instability. We propose Clipping-Free Policy Optimization (CFPO), which replaces heuristic clipping with a convex quadratic penalty derived from Total Variation divergence constraints, yielding an everywhere-differentiable objective that enforces stable policy updates without hard boundaries. We evaluate CFPO across both reasoning and alignment settings. In reasoning, CFPO matches clipping-based methods on downstream benchmarks while extending the stable training regime. In alignment, CFPO mitigates verbosity exploitation and reduces capability degradation, while achieving competitive instruction-following performance. CFPO requires only a one-line code change and no additional hyperparameters. Our results suggest that CFPO is a promising drop-in alternative to clipping-based methods for LLM post-training.

1 Introduction

Reinforcement learning (RL) has become central to large language model (LLM) post-training, from aligning models with human preferences (Christiano et al., 2023; Stiennon et al., 2022; Bai et al., 2022; Ouyang et al., 2022) to eliciting complex reasoning behaviors (DeepSeek-AI, 2025; OpenAI, 2024; Google DeepMind, 2025). The dominant algorithms, including PPO (Schulman et al., 2017b), GRPO (Shao et al., 2024), and their variants, rely on clipping probability ratios to a narrow range, approximating trust region constraints without the computational overhead of second-order methods. However, clipping remains a heuristic, and its limitations become increasingly apparent at scale.

The core issue is a discontinuity induced by hard clipping in the optimization landscape. Within the clipping range, the objective reduces to unconstrained advantage maximization, enabling rapid entropy collapse as the policy aggressively concentrates around high-advantage actions (Liu et al., 2025; Huang et al., 2025; Yu et al., 2025; Team, 2025; MiniMax, 2025). Beyond the clipping boundary, gradients vanish entirely, eliminating corrective signals for off-policy samples and contributing to training instability (Liu et al., 2025; Yu et al., 2025). Together, these dynamics create pathological behaviors: models learn to exploit superficial correlates of reward such as verbosity (Gao et al., 2022), and rapid policy drift degrades capabilities acquired during pretraining (Ouyang et al., 2022; Casper et al., 2023). These problems are further amplified in Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017; Fedus et al., 2022), where expert activation volatility causes token-level probability ratios to fluctuate drastically after each gradient update, rendering clipping-based objectives unreliable and leading to catastrophic training collapse (Zheng et al., 2025; MiniMax, 2025). Moreover, PPO-style clipping has been shown to permanently filter out certain tokens—particularly discourse and transition words—preventing them from ever receiving gradients during training (MiniMax, 2025). These failures manifest across dense and sparse architectures alike, suggesting they are intrinsic to the clipping mechanism rather than incidental to specific implementations.

Recent work (see Appendix A for an extended discussion) has proposed targeted modifications: asymmetric clipping bounds (Yu et al., 2025), modified advantage normalization (Liu et al., 2025), dynamic clipping thresholds (Yang et al., 2025), auxiliary regularization (Wang et al., 2025b), and sequence-level clipping to stabilize MoE training (Zheng et al., 2025). While these methods mitigate specific failure modes, they treat clipping as a mechanism to be patched or relocated rather than replaced, introducing additional hyperparam-

eters or shifting the granularity of clipping while leaving the fundamental discontinuity intact.

In this paper, we propose Clipping-Free Policy Optimization (CFPO), a principled alternative that eliminates clipping entirely. CFPO builds on the quadratic penalty objective introduced by Simple Policy Optimization (SPO) (Xie et al., 2025), which replaces clipping with a convex penalty derived from Total Variation divergence constraints (Queeney et al., 2021). While SPO demonstrated this objective in simulation environments, its behavior in the LLM post-training regime—where off-policy pressure, reward hacking, and capability degradation pose distinct challenges—has not been studied. We conduct a systematic empirical investigation across both RLVR and RLHF settings, revealing failure modes of clipping that the quadratic penalty specifically addresses: verbosity exploitation, alignment tax, and training instability under off-policy pressure. We further demonstrate that the penalty integrates cleanly with modern training recipes, confirming its orthogonality to other recent improvements.

We evaluate CFPO across diverse post-training settings, substituting it for clipped objectives while holding all other training details constant. Our experiments span Qwen2.5 (Yang et al., 2024) models from 1.5B to 7B parameters for reasoning tasks and Llama3-8B (AI@Meta, 2024) for alignment, using GRPO and RLOO (Ahmadian et al., 2024) as baselines respectively. Across settings, CFPO exhibits more conservative optimization dynamics: slower initial reward growth but sustained progress, lower clipping ratios, and gradual entropy consumption rather than rapid collapse. These dynamics yield concrete improvements:

- **Robust alignment.** CFPO mitigates verbosity exploitation, improving length-controlled AlpacaEval (Dubois et al., 2024) by 4 points while achieving competitive performance on Arena-Hard (Li et al., 2024a) and MT-Bench (Zheng et al., 2023). CFPO also reduces alignment tax from 12–16% to 4–5% on OpenLLM (Beeching et al., 2023) tasks.

- **Stable reasoning training.** GRPO becomes unstable around 8 training iterations and completely collapses by 16 across most configurations. CFPO extends the stable training regime, with controlled entropy decay, policy KL, and clipping ratio throughout, while matching GRPO in reasoning accuracy on MATH500 (Hendrycks et al., 2021), AIME24 (Mathematical Association of America, 2024), GSM8K (Cobbe et al., 2021), and GPQA-Diamond (Rein et al., 2023) datasets. CFPO also integrates directly into the DAPO training pipeline (Wang et al., 2025a) on Qwen3-4B-Base, matching accuracy while preserving its characteristic entropy dynamics, confirming that the quadratic penalty is orthogonal to other recent improvements.

Overall, our results suggest that CFPO is a practical drop-in replacement for clipping-based objectives in LLM post-training, offering more stable optimization without sacrificing downstream performance and requiring only a one-line code change.

2 Background

2.1 Proximal Policy Optimization

Proximal Policy Optimization (PPO) (Schulman et al., 2017b) is a policy gradient method that stabilizes training by constraining how much the policy can change in a single update. Given a policy π_θ, PPO maximizes a clipped surrogate objective:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \Big[min\big(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big], \tag{1}$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ is the probability ratio between the current and old policy, $\hat{A}_t$ is the estimated advantage, and ϵ is a hyperparameter controlling the clipping range. The clipping mechanism removes incentives for pushing the ratio outside $[1-\epsilon, 1+\epsilon]$, approximating a trust region constraint without the computational overhead of second-order methods like TRPO (Schulman et al., 2017a). In practice, PPO also employs a learned value function to estimate advantages via Generalized Advantage Estimation (Schulman et al., 2018).

2.2 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) adapts PPO to the language model setting (Shao et al., 2024). In standard PPO, advantages are computed using a learned value function, requiring a critic network of comparable size to the policy—a substantial memory burden for large language models. GRPO eliminates this requirement by estimating advantages through comparisons within groups of sampled responses to the same prompt.

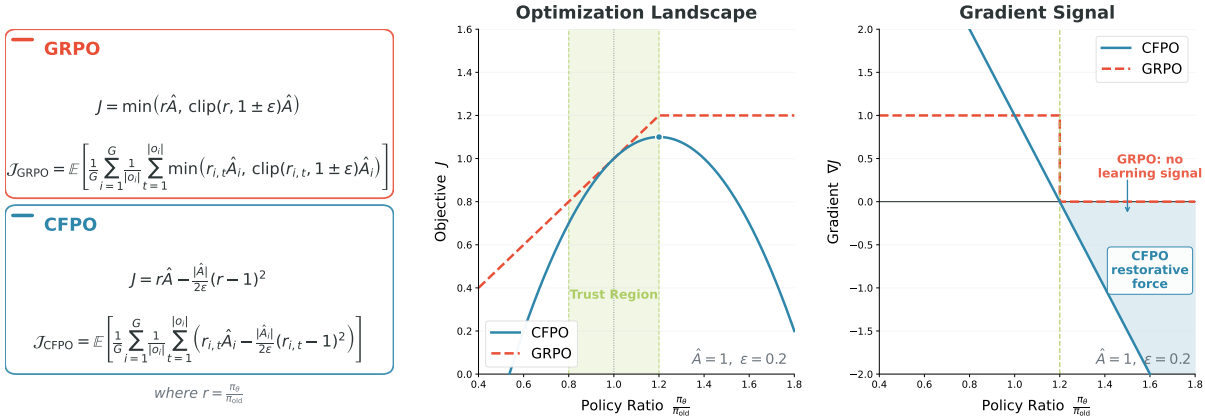

Figure 1: Optimization objective and gradient of GRPO and CFPO as functions of the policy ratio $r = \pi/\pi_{\text{old}}$, shown for advantage $A = 1$ and trust-region width $\epsilon = 0.2$. GRPO becomes flat once r exits the trust region, resulting in zero gradient beyond the clipping boundary. CFPO instead applies a convex quadratic penalty in r, yielding a continuous restoring gradient that pulls r back toward the trust region. This difference highlights why CFPO maintains stable learning signals while GRPO can stall when updates push r outside the trust region.

For each prompt q, GRPO samples a group of responses $\{o_1, o_2, \ldots, o_G\}$ from the current policy and computes advantages based on relative rewards within the group:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}, \tag{2}$$

where r_i is the reward for o_i. The GRPO objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^G}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\right. \tag{3}$$

$$\left.\sum_{t=1}^{|o_i|} L_{\text{clip}}(r_{i,t}, \hat{A}_i)\right] - \beta\, \mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}], \tag{4}$$

where L_{clip} denotes PPO's clipped objective, $r_{i,t}(\theta) = \pi_\theta(o_{i,t}|q, o_{i,<t})/\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})$ is the per-token probability ratio, and β controls KL regularization against a reference policy π_{ref}.

2.3 Simple Policy Optimization

Simple Policy Optimization (SPO) (Xie et al., 2025) provides a principled alternative to PPO's clipping mechanism. Although PPO's clipping is often viewed as heuristic, it can be interpreted as approximately enforcing a trust region defined by Total Variation (TV) divergence between successive policies (Queeney et al., 2021). Under this view, PPO implicitly optimizes the policy objective subject to a constraint on the expected deviation of probability ratios from unity.

A key motivation for SPO is that TV divergence constraints induce a strictly larger feasible policy space than the KL divergence constraints used in TRPO. Moreover, optimizing within the TV-constrained space yields a tighter policy improvement lower bound than the corresponding KL-constrained formulation. We state these results below and defer formal proofs to Appendix B.

Proposition 2.1 (TV Solution Space Contains KL Solution Space)**.** *Let Ω_{TV} and Ω_{KL} denote the policy sets satisfying per-state TV and KL divergence constraints, respectively. If $\delta_{\text{TV}} \geq \sqrt{\delta_{\text{KL}}/2}$, then $\Omega_{\text{KL}} \subset \Omega_{\text{TV}}$.*

Theorem 2.2 (TV-Constrained Policy Improvement)**.** *Let $\mathcal{L}_\pi^{\text{TV}}$ and $\mathcal{L}_\pi^{\text{KL}}$ denote the standard policy improvement lower bounds with TV and KL penalties (Xie et al., 2025). Let $\tilde{\pi}^*_{\text{TV}}$ and $\tilde{\pi}^*_{\text{KL}}$ be their respective maximizers over Ω_{TV} and Ω_{KL}. For $\delta_{\text{TV}} \geq \sqrt{\delta_{\text{KL}}/2}$, the following holds:*

$$\mathcal{L}_\pi^{\text{TV}}(\tilde{\pi}^*_{\text{TV}}) \;\geq\; \mathcal{L}_\pi^{\text{KL}}(\tilde{\pi}^*_{\text{KL}}). \tag{5}$$

Unlike PPO, which enforces the trust region via clipping and yields zero gradients for samples outside the clipping range, SPO incorporates the constraint directly using a quadratic penalty. The resulting objective is

$$\mathcal{J}_{\text{SPO}}(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\epsilon}(r_t - 1)^2\right], \tag{6}$$

where $r_t = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$. This objective is convex and differentiable in r_t, and its maximizer satisfies $r_t^* = 1 + \text{sign}(\hat{A}_t)\epsilon$, corresponding to an update at the trust region boundary while retaining nonzero gradients for all samples.

3 Methodology

3.1 Clipping-Free Policy Optimization

Motivated by the success of SPO (Xie et al., 2025) in simulation environments, we propose Clipping-Free Policy Optimization (CFPO) as an adaptation to the language model setting. CFPO retains the critic-free design common to modern LLM post-training while replacing the clipped surrogate objective with SPO's quadratic penalty. This substitution requires only a one-line code change, making CFPO a drop-in replacement in existing pipelines.

The core modification is straightforward. Where clipping-based methods enforce the trust region via:

$$L_{\text{clip}} = \min\left(r_{i,t}\hat{A}_i,\ \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\right), \tag{7}$$

CFPO instead applies a quadratic penalty:

$$L_{\text{CFPO}} = r_{i,t}\hat{A}_i - \frac{|\hat{A}_i|}{2\epsilon}(r_{i,t}-1)^2, \tag{8}$$

where $r_{i,t}(\theta) = \pi_\theta(o_{i,t}|q, o_{i,<t})/\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})$ is the per-token probability ratio and ϵ controls the trust region width. As shown in Figure 1, this yields a convex, everywhere-differentiable objective: rather than zeroing gradients when the ratio exits the clipping range, CFPO provides a continuous restoring force that pulls the policy back toward the trust region. The penalty is minimized at $r_{i,t} = 1$ and grows quadratically with deviation, while scaling by $|\hat{A}_i|$ ensures that larger advantages permit proportionally larger updates. For rare numerical outliers from precision errors, standard gradient clipping or filtering such samples provides a sufficient safeguard without introducing zero-gradient regions into the objective.

The full CFPO objective is:

$$\mathcal{J}_{\text{CFPO}}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^G}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|} \sum_{t=1}^{|o_i|}\left(r_{i,t}\hat{A}_i - \frac{|\hat{A}_i|}{2\epsilon}(r_{i,t}-1)^2\right)\Bigg] - \beta\,\mathbb{D}_{\text{KL}}[\pi_\theta\|\pi_{\text{ref}}], \tag{9}$$

where β is the KL regularization coefficient against a reference policy π_{ref}. Since CFPO modifies only the surrogate objective, it is agnostic to how advantages $\hat{A}_i$ are estimated. We exploit this modularity to evaluate CFPO with two distinct advantage estimators standard to different post-training settings.

4 Experimental Setup

4.1 Reasoning

Training Setup. We follow the training setup of Zhao et al. (2025) and train GRPO and CFPO using the Open-R1/TRL framework (Face, 2025; von Werra et al., 2020) on the training split of the MATH dataset (Hendrycks et al., 2021), which contains 7,500 problems. We use Qwen2.5-1.5B, Qwen2.5-3B, and Qwen2.5-7B (Yang et al., 2024) as backbone models, experimenting with both base and Instruct variants. All models are trained using a chat-style prompting format with 128 problems per update and 3 candidate solutions per problem (1 GPU for rollout server, remaining for training). Unless otherwise specified, we use a KL penalty coefficient of $\beta = 0.0$. For some ablations, we also use the `verl` framework (Sheng et al., 2024) with Qwen2.5-3B.

Advantage Estimation. CFPO modifies only the surrogate objective and is agnostic to how advantages are estimated. Recent progress in LLM post-training has largely focused on advantage estimation and reward shaping: GRPO (Shao et al., 2024) eliminated the critic for reasoning, RLOO (Ahmadian et al., 2024) for alignment, and DAPO (Yu et al., 2025) refined normalization and sampling strategies. We adopt the standard estimator for each setting and additionally evaluate CFPO within the full DAPO recipe. Consistent benefits across all three confirm that improvements stem from the quadratic penalty rather than interactions with specific estimators.

Off-Policy Mechanisms. Comparing GRPO and CFPO requires off-policy settings; in on-policy RL, both reduce to simple advantage maximization. Off-policy pressure arises from sample reuse (iterations) and mini-batch gradient updates. TRL supports only sample reuse, while `verl` supports both. When iteration is 1 and mini-batch size equals batch size, both frameworks are on-policy and the methods are equivalent. We vary iterations up to 16 in TRL across 3 model sizes and 2 model bases, and up to 8 in `verl` with 4 batch ratios, yielding 80 models total across both frameworks.

Evaluation. We evaluate reasoning on MATH500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AIME24 (Mathematical Association of America, 2024), and GPQA-Diamond (Rein et al., 2023) using

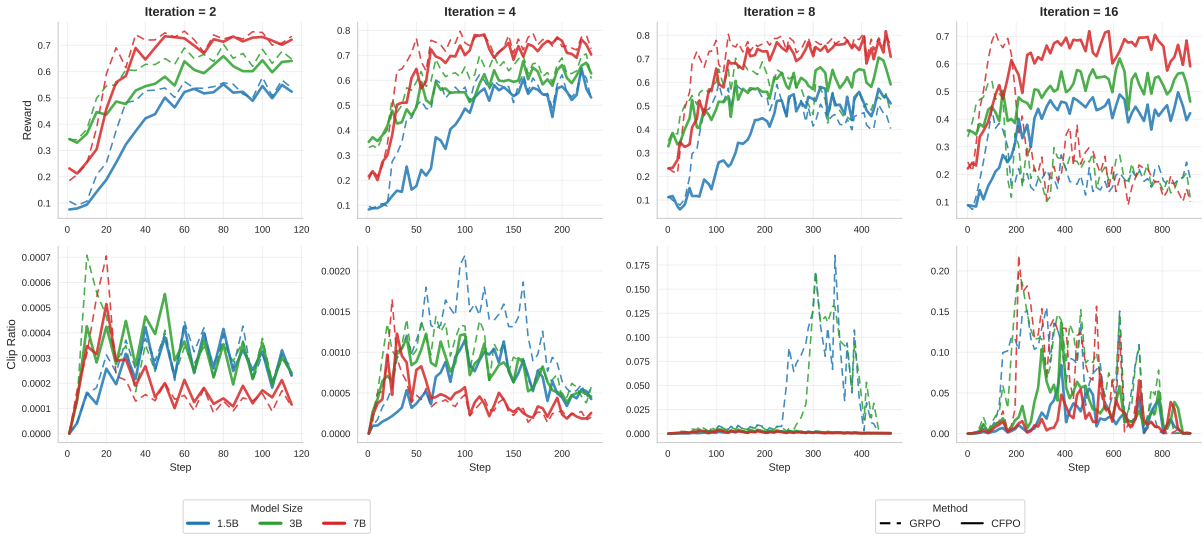

Figure 2: Training dynamics of CFPO vs. GRPO on Qwen2.5 models under increasing off-policy pressure (iterations per update). GRPO achieves faster early reward gains but exhibits rising clip ratios and training collapse at $\geq$8 iterations for most models. CFPO progresses more conservatively, maintaining low clip ratios and stable training throughout while reaching comparable reward levels.

`lighteval` (Habib et al., 2023), with temperature 0.6 and top-p 0.95.

4.2 RLHF

Training Setup. We train CFPO and RLOO using the OpenRLHF framework (Hu et al., 2024) with Llama-3-8B (AI@Meta, 2024) as the backbone, using the provided SFT and reward models. All experiments use default OpenRLHF hyperparameters with $k = 2$ rollouts per prompt. For KL-free settings, we set the KL coefficient to zero; otherwise, all parameters follow the default configuration.

Evaluation. We evaluate on AlpacaEval 2.0 (Dubois et al., 2024), Arena-Hard v0.1 (Li et al., 2024a), MT-Bench (Zheng et al., 2023), and IFEval (Zhou et al., 2023) for instruction following, and ARC Challenge (Clark et al., 2018), GSM8K (Cobbe et al., 2021), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), TruthfulQA (Lin et al., 2022), and Winogrande (Levesque et al., 2012) for capability retention.

5 Results and Analysis

Our analysis primarily examines how replacing clipping with a convex quadratic penalty affects optimization behavior and stability, and how these differences translate into final downstream performance in LLM post-training. We study this across reasoning-focused reinforcement learning (RLVR) and alignment-focused reinforcement learning (RLHF), and under multiple sources of off-policy pressure.

5.1 Reasoning

Cold-Start Training with Qwen Models. Following recent work demonstrating that RL can be applied directly to base language models without prior supervised fine-tuning (DeepSeek-AI, 2025; Marjanović et al., 2026), we perform cold-start training using both GRPO and CFPO across multiple Qwen2.5 models, scaling the number of training iterations up to 16.

In general, we expect GRPO to optimize reward more aggressively than CFPO: GRPO performs direct advantage maximization subject to a clipped constraint, whereas CFPO's quadratic penalty continues to regularize updates even when probability ratios lie within the trust region. We also expect CFPO to exhibit lower clipping ratios, since it actively discourages large updates rather than discarding clipped samples.

Figure 2 confirms these expectations. GRPO improves reward more rapidly in early training, while CFPO progresses more gradually and eventually reaches comparable levels. Consistent with its design, CFPO maintains lower clipping ratios throughout. However, GRPO becomes increasingly unstable as iterations grow: instability appears around 8 iterations, and training collapses by

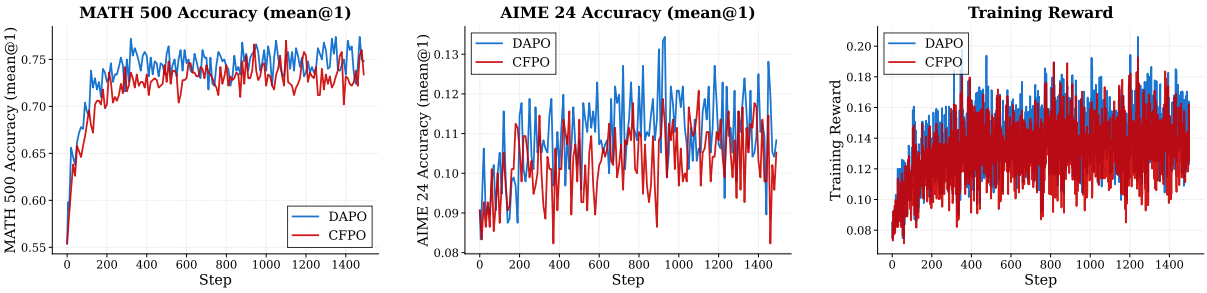

Figure 3: Downstream accuracy and training reward for DAPO and DAPO + CFPO on Qwen3-4B-Base using the DAPO recipe from Wang et al. (2025a). CFPO matches DAPO on MATH500 and AIME24 accuracy at equal compute, with comparable reward progression. DAPO exhibits faster early reward gains while CFPO converges more gradually, consistent with patterns observed across all other experimental settings.

16 across most configurations.

We evaluate downstream reasoning performance on Math500, GSM8K, AIME24, and GPQA-Diamond (Tables 8, 9, 10). On Math500, GRPO and CFPO perform comparably across model sizes. For GRPO, collapse occurs at 8 iterations for all models except Qwen2.5-7B, where it is delayed until 16. CFPO exhibits breakdown at 16 iterations for the 1.5B and 7B models, while remaining always stable for the 3B model.

Interestingly, CFPO underperforms on GSM8K compared to GRPO for the 1.5B and 7B models. Qualitative inspection suggests this gap reflects weaker instruction-following rather than degraded reasoning: CFPO-trained models more frequently produce incomplete generations or responses in unintended languages. We corroborate this using AlpacaEval, where CFPO-trained models achieve lower scores. On AIME24 and GPQA-Diamond, we find no evidence that GRPO-trained models consistently outperform CFPO, suggesting GRPO's GSM8K advantage is attributable to more aggressive instruction learning rather than better reasoning.

Overall, these results reveal a trade-off between optimization aggressiveness and stability. GRPO achieves faster early reward gains but exhibits growing instability at higher iteration counts. CFPO advances more gradually, maintaining better stability while reaching comparable downstream reasoning performance.

RLVR on Instruction-Tuned Models. To separate effects arising from instruction-following behavior from those related to reasoning, we perform RLVR on instruction-tuned versions of the same models. Consistent with cold-start results, we observe no substantial performance differences between GRPO and CFPO across model scales (Tables 5, 6, 7). The main difference remains optimization stability: GRPO exhibits instability at higher iteration counts, while CFPO remains stable across model sizes and training durations which is consistent with cold-start experiments.

Off-Policy Training with `verl`. Several RLVR recipes implemented in `verl` have non-zero clipping ratios even without explicit sample reuse, as mini-batch policy updates introduce mild policy lag despite using fresh data (Yu et al., 2025; Yang et al., 2025).

At a single iteration, GRPO optimizes reward rapidly while CFPO converges more slowly but eventually catches up, achieving comparable training reward, validation reward, clipping fractions, and KL (Figs. 6–9). Entropy dynamics differ markedly: GRPO consumes entropy substantially faster than CFPO (Fig. 10), accelerating short-horizon reward gains but likely undermining stability in longer training regimes.

Increasing the batch ratio up to 8 does not induce collapse, whereas increasing iterations leads to instability around 8 iterations (Figs. 6, 8), suggesting iteration-based sample reuse is more destabilizing than batch-ratio–induced off-policy updates. However, these effects accumulate: even at stable iteration counts, increasing the batch ratio can induce collapse (Figs. 8, 10). CFPO consistently maintains lower clipping ratios, slower entropy decay, and more stable updates across all configurations.

Compatibility with Orthogonal Techniques. Since CFPO modifies only the surrogate objective, techniques orthogonal to it such as token-level loss aggregation, dynamic sampling, and asymmetric clipping (Yu et al., 2025; Liu et al., 2025) transfer directly. We validate this by training Qwen3-4B-

Table 1: Instruction-following and preference alignment results of CFPO and RLOO on Llama3-8B. We report performance on preference-based benchmarks (Arena-Hard, AlpacaEval, MT-Bench) and instruction-following (IFEval), including unregularized variants with KL coefficient set to zero. Across benchmarks, CFPO maintains strong alignment behavior while exhibiting reduced sensitivity to verbosity compared to RLOO, as reflected by differences between raw and length-controlled evaluations.

Method	Arena-Hard	AlpacaEval-LC	AlpacaEval-WR	MT-Bench	IFEval
Llama3-8B-SFT	6.5	5.22	3.52	7.84	59.59
RLOO	21.8	7.25	18.17	7.90	47.00
RLOO (KL=0)	22.1	8.15	16.39	7.86	44.12
CFPO	20.1	11.26	17.08	7.87	55.64
CFPO (KL=0)	22.5	11.55	16.08	8.04	54.20

Table 2: Downstream capability evaluation of CFPO and RLOO after RLHF on Llama3-8B, measured on the OpenLLM Leaderboard tasks. Comparisons against the SFT baseline illustrate how different alignment objectives affect general-purpose capabilities. CFPO preserves substantially more of the base model's capabilities across tasks, whereas RLOO induces broader degradation.

Method	ARC	GSM8K	Hella	MMLU	TruthfulQA	Winogrande	Avg
Llama3-8B-SFT	0.525	0.762	0.584	0.627	0.401	0.719	0.603
RLOO	0.382	0.813	0.435	0.626	0.392	0.536	0.531
RLOO (KL=0)	0.374	0.683	0.408	0.624	0.406	0.535	0.505
CFPO	0.432	0.797	0.542	0.631	0.441	0.617	0.577
CFPO (KL=0)	0.431	0.789	0.539	0.631	0.428	0.614	0.572

Base (Team, 2025) using the DAPO recipe of Wang et al. (2025a), substituting only the clipped loss with CFPO's quadratic penalty (see Appendix C.2 for full details).

Figures 3 and 5 show results at equal compute budgets. CFPO matches DAPO on MATH500 accuracy and AIME24, with comparable training reward trajectories. Consistent with all prior experiments, DAPO optimizes reward more aggressively in early training while CFPO converges more gradually. The entropy plot reveals the same pattern observed across our Qwen2.5 and verl experiments: CFPO preserves substantially more policy entropy throughout training, reflecting less aggressive updates. These results confirm that CFPO serves as a drop-in replacement within modern RLVR recipes, maintaining compatibility with orthogonal improvements while exhibiting its characteristic conservative optimization dynamics on a newer model family.

5.2 RLHF

Optimization Dynamics. As in the reasoning setting, we first examine how the choice of objective shapes optimization behavior during training. Figure 4 shows that RLOO exhibits aggressive early optimization—rewards increase rapidly before plateauing—while CFPO demonstrates more gradual, approximately linear reward improvement throughout training. This pattern extends to other metrics: under RLOO, response length increases alongside reward, indicating the model learns to exploit verbosity as a reward-hacking strategy, while CFPO maintains stable response lengths. RLOO's clipping ratios also increase over training while CFPO keeps these values lower and more consistent.

Instruction Following. We next evaluate how these differing optimization dynamics translate into instruction-following behavior. Both methods achieve competitive scores on Arena-Hard and MT-Bench. The length exploitation observed during training becomes apparent in AlpacaEval: while raw win rates appear similar, length-controlled scores reveal a meaningful gap, with CFPO outperforming RLOO by 3–4 percentage points. This confirms that RLOO inflates its win rate through verbosity, while CFPO's consistent scores between raw and length-controlled metrics indicate genuine quality improvements.

Furthermore, RLOO substantially degrades performance on IFEval, dropping from 59.6 to 47.0 compared to the SFT baseline—a 12-point decline in exact instruction-following capability. CFPO preserves this capability much better, achieving 55.6 with only a 4-point drop. This suggests that RLOO's aggressive optimization not only exploits superficial reward correlates but limits the model's

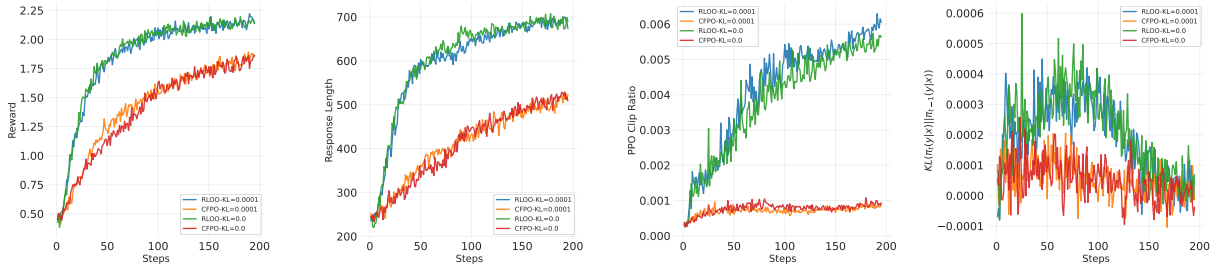

Figure 4: RLHF training dynamics on Llama3-8B under RLOO and CFPO with different KL penalty coefficients: (a) reward, (b) response length, (c) clipping ratio, and (d) KL divergence. RLOO exhibits rapid early reward gains accompanied by growing response lengths and elevated clipping, indicating more aggressive optimization. CFPO yields steadier reward improvement while maintaining stable response lengths, lower clipping ratios, and controlled KL divergence.

ability to follow precise instructions, while CFPO's conservative updates maintain this capability.

Capability Retention. Finally, we assess how these optimization differences affect retention of base model capabilities. RLHF methods often degrade base model capabilities, a phenomenon known as alignment tax (Askell et al., 2021). RLOO incurs 12–16% alignment tax depending on KL penalty settings, while CFPO variants pay only 4–5%. The degradation under RLOO spans ARC, HellaSwag, and Winogrande, whereas CFPO retains substantially more capability across all evaluations.

These results mirror our reasoning findings: across both settings, CFPO's quadratic penalty produces more stable optimization dynamics and competitive downstream performance. The conservative updates that prevent training collapse in reasoning also prevent the excessive policy drift that degrades capabilities in RLHF.

6 Limitations

Our experiments were conducted on a single node with 4×H200 GPUs, limiting evaluation to dense models up to 8B parameters. Although we motivate CFPO partly through clipping's failures in MoE architectures, we did not have sufficient compute to evaluate on MoE models. Whether CFPO's smooth gradients resolve MoE-specific instabilities such as expert activation volatility remains a promising direction for future work. More broadly, whether CFPO's conservative updates remain beneficial at frontier scales, and how they interact with sparser or noisier rewards (e.g., code generation, agentic tasks), are open questions.

7 Conclusion

We propose CFPO as a drop-in replacement for the clipped surrogate objectives used in PPO and GRPO in language model post-training. By replacing heuristic clipping with a convex quadratic penalty derived from Total Variation divergence constraints, CFPO provides smooth gradients throughout the optimization landscape while implicitly enforcing trust region bounds. Our experiments across both reasoning and alignment settings demonstrate that CFPO offers improved training stability, substantially delaying collapse at high iteration counts, reducing alignment tax, and maintaining gradual entropy consumption, while achieving competitive downstream performance, all at no additional computational cost and with only a one-line code change. Given the growing evidence that clipping introduces instabilities across both dense and MoE architectures, we believe removing the discontinuity entirely, rather than patching it, is a direction worth pursuing further.

8 Ethical considerations

This work focuses on improving the stability of reinforcement learning for large language model post-training. While our method reduces certain undesirable behaviors such as reward hacking and instability, it does not fully eliminate broader risks associated with language models, including the generation of misleading or harmful content. Additionally, our experiments are limited to relatively small-scale models and controlled benchmarks, which may not fully capture real-world deployment challenges. We encourage further evaluation at larger scales and in diverse settings, as well as continued research on alignment and safety to ensure responsible use.

9 The Use of Large Language Models

The authors initially drafted all text, after which Large Language Models were used to improve phrasing and enhance clarity.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. 2018. Maximum a posteriori policy optimisation. *Preprint*, arXiv:1806.06920.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. *Preprint*, arXiv:1705.10528.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

AI@Meta. 2024. Llama 3 model card.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, and 3 others. 2021. A general language assistant as a laboratory for alignment. *Preprint*, arXiv:2112.00861.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *Preprint*, arXiv:2310.12036.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open LLM leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Boris Belousov and Jan Peters. 2018. f-divergence constrained policy improvement. *Preprint*, arXiv:1801.00056.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, and 13 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Preprint*, arXiv:2307.15217.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation matters in deep policy gradients: A case study on ppo and trpo. *Preprint*, arXiv:2005.12729.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *Preprint*, arXiv:2402.01306.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. *Preprint*, arXiv:2210.10760.

Google DeepMind. 2025. Gemini 2.5 pro. https://deepmind.google/models/gemini/pro/. Accessed: November 2025.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

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

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. 2025. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. *Preprint*, arXiv:2510.03222.

Sham M Kakade. 2001. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Sham M. Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*.

Guanghui Lan. 2022. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Preprint*, arXiv:2102.00135.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024a. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2024b. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *Preprint*, arXiv:2310.10505.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, pages 3214–3252.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, and Siva Reddy. 2026. Deepseek-r1 thoughtology: Let's think about llm reasoning. *Preprint*, arXiv:2504.07128.

Mathematical Association of America. 2024. AIME problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

MiniMax. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *Preprint*, arXiv:2506.13585.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *Preprint*, arXiv:1910.00177.

James Queeney, Ioannis Ch. Paschalidis, and Christos G. Cassandras. 2021. Generalized proximal policy optimization with sample reuse. *Preprint*, arXiv:2111.00072.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2017a. Trust region policy optimization. *Preprint*, arXiv:1502.05477.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-dimensional continuous control using generalized advantage estimation. *Preprint*, arXiv:1506.02438.

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

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. 2021. Mirror descent policy optimization. *Preprint*, arXiv:2005.09814.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Yining Wang, Jinman Zhao, Chuangxin Zhao, Shuhao Guan, Gerald Penn, and Shinan Liu. 2025b. λ-grpo: Unifying the grpo frameworks with learnable token preferences. *Preprint*, arXiv:2510.06870.

Yuhui Wang, Hao He, Chao Wen, and Xiaoyang Tan. 2020. Truly proximal policy optimization. *Preprint*, arXiv:1903.07940.

Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. 2025. Simple policy optimization. *Preprint*, arXiv:2401.16025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Preprint*, arXiv:2203.14465.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2025. Learning to reason without external rewards. *Preprint*, arXiv:2505.19590.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Related Work

Stable Policy Gradient Methods. Trust region methods have been fundamental to stable policy optimization since the natural policy gradient (Kakade, 2001) and TRPO (Schulman et al., 2017a) established theoretical guarantees for monotonic improvement. PPO (Schulman et al., 2017b) made these methods practical through clipping, becoming dominant for both continuous control and language model training. However, Wang et al. (2020) showed that clipping fails to bound KL divergence, while Engstrom et al. (2020) demonstrated that implementation details matter more than clipping itself. These findings have motivated principled alternatives: f-divergence generalizations (Belousov and Peters, 2018), mirror descent interpretations (Tomar et al., 2021; Lan, 2022), and algorithms like MPO (Abdolmaleki et al., 2018) and AWR (Peng et al., 2019) that enforce trust regions through different mechanisms. SPO (Xie et al., 2025) replaces clipping with a convex quadratic penalty derived from TV divergence, providing non-zero gradients while implicitly enforcing trust region bounds.

Reinforcement Learning for Language Models. In RLHF, reward models trained on human preferences guide policy optimization (Christiano et al., 2023; Ziegler et al., 2020; Stiennon et al., 2022; Bai et al., 2022; Ouyang et al., 2022), though challenges like reward overoptimization (Gao et al., 2022) have motivated simpler approaches. DPO (Rafailov et al., 2024) bypasses reward models entirely, spawning variants including IPO (Azar et al., 2023), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024). Among online methods, RLOO (Ahmadian et al., 2024) and ReMax (Li et al., 2024b) eliminate the critic while retaining on-policy benefits.

For reasoning, RL with verifiable rewards has proven effective (OpenAI, 2024; DeepSeek-AI, 2025), building on work showing process supervision improves reasoning (Lightman et al., 2023; Zelikman et al., 2022). GRPO (Shao et al., 2024) is now standard, using group-relative advantages without a critic. However, scaling has revealed clipping-related instabilities, motivating variants like DAPO (Yu et al., 2025), Dr.GRPO (Liu et al., 2025), and λ-GRPO (Wang et al., 2025b) that modify clipping behavior or token weighting. Rather than patching clipping, CFPO replaces it entirely with the quadratic penalty, providing a unified approach across both RLHF and reasoning settings.

B Theoretical Results of Simple Policy Optimization

In this appendix, we restate the key theoretical results from Xie et al. (2025) that motivate Simple Policy Optimization (SPO) as an alternative to PPO's clipping mechanism.

B.1 Performance Improvement Bounds

The foundation of trust region methods lies in the policy performance difference theorem (Kakade and Langford, 2002), which expresses the performance gap between policies in terms of advantage functions. Building on this, Achiam et al. (2017) established a performance improvement lower bound using Total Variation (TV) divergence:

Theorem B.1 (Performance Improvement Lower Bound)**.** *Given any two policies π and $\tilde{\pi}$, the following bound holds:*

$$\eta(\tilde{\pi}) - \eta(\pi) \geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_\pi, a\sim\tilde{\pi}}\left[A_\pi(s,a)\right] - \frac{2\xi\gamma}{(1-\gamma)^2}\mathbb{E}_{s\sim\rho_\pi}\left[D_{\mathrm{TV}}(\pi\|\tilde{\pi})[s]\right], \tag{10}$$

where $\xi = \max_s \left|\mathbb{E}_{a\sim\tilde{\pi}(\cdot|s)}\left[A_\pi(s,a)\right]\right|$ and D_{TV} denotes the Total Variation divergence.

Using the relationship between TV divergence and probability ratios, this bound can be rewritten as:

$$\eta(\tilde{\pi}) - \eta(\pi) \geq \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim\pi}\left[\frac{\tilde{\pi}(a|s)}{\pi(a|s)}A_\pi(s,a)\right] - \frac{\xi\gamma}{(1-\gamma)^2}\mathbb{E}_{s,a\sim\pi}\left[\left|\frac{\tilde{\pi}(a|s)}{\pi(a|s)} - 1\right|\right]. \tag{11}$$

This formulation reveals why constraining the probability ratio $|r_t - 1| \leq \epsilon$ is beneficial: it directly controls the penalty term in the performance bound.

B.2 Advantages of TV Divergence over KL Divergence

A key insight from Xie et al. (2025) is that TV divergence constraints offer theoretical advantages over the KL divergence constraints used in TRPO.

Proposition B.2 (TV Solution Space Contains KL Solution Space)**.** *Given an old policy π, define the solution spaces under TV and KL divergence*

constraints as:

$$\Omega_{\mathrm{TV}} = \{\tilde{\pi} \mid D_{\mathrm{TV}}(\pi\|\tilde{\pi})[s] \le \delta_{\mathrm{TV}}, \forall s \in \mathcal{S}\}, \tag{12}$$

$$\Omega_{\mathrm{KL}} = \{\tilde{\pi} \mid D_{\mathrm{KL}}(\pi\|\tilde{\pi})[s] \le \delta_{\mathrm{KL}}, \forall s \in \mathcal{S}\}, \tag{13}$$

where $\delta_{\mathrm{KL}} > 0$ is a predefined threshold. For $\delta_{\mathrm{TV}} \ge \sqrt{\frac{1}{2}\delta_{\mathrm{KL}}}$, we have $\Omega_{\mathrm{KL}} \subset \Omega_{\mathrm{TV}}$.

Proof. For any $\tilde{\pi} \in \Omega_{\mathrm{KL}}$, by Pinsker's inequality:

$$D_{\mathrm{TV}}(\pi\|\tilde{\pi})[s] \le \sqrt{\frac{1}{2}D_{\mathrm{KL}}(\pi\|\tilde{\pi})[s]} \le \sqrt{\frac{1}{2}\delta_{\mathrm{KL}}} \le \delta_{\mathrm{TV}}. \tag{14}$$

Thus $\tilde{\pi} \in \Omega_{\mathrm{KL}} \Rightarrow \tilde{\pi} \in \Omega_{\mathrm{TV}}$, establishing $\Omega_{\mathrm{KL}} \subset \Omega_{\mathrm{TV}}$. □

Furthermore, optimizing within the larger TV-constrained space yields superior bounds:

Theorem B.3 (Superiority of TV-Constrained Optimization)**.** *Let $\mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi})$ and $\mathcal{L}_\pi^{\mathrm{KL}}(\tilde{\pi})$ denote the performance improvement lower bounds with TV and KL divergence penalties respectively:*

$$\mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi}) = \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim\pi}\left[\frac{\tilde{\pi}(a|s)}{\pi(a|s)}A_\pi(s,a)\right] - \frac{2\xi\gamma}{(1-\gamma)^2}\mathbb{E}_{s\sim\rho_\pi}\left[D_{\mathrm{TV}}(\pi\|\tilde{\pi})[s]\right]$$

$$\mathcal{L}_\pi^{\mathrm{KL}}(\tilde{\pi}) = \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim\pi}\left[\frac{\tilde{\pi}(a|s)}{\pi(a|s)}A_\pi(s,a)\right] - \frac{2\xi\gamma}{(1-\gamma)^2}\mathbb{E}_{s\sim\rho_\pi}\left[\sqrt{\frac{1}{2}D_{\mathrm{KL}}(\pi\|\tilde{\pi})[s]}\right]. \tag{15}$$

Define the optimal policies in each constrained space:

$$\tilde{\pi}^*_{\mathrm{TV}} = \arg\max_{\tilde{\pi}\in\Omega_{\mathrm{TV}}} \mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi}), \quad \tilde{\pi}^*_{\mathrm{KL}} = \arg\max_{\tilde{\pi}\in\Omega_{\mathrm{KL}}} \mathcal{L}_\pi^{\mathrm{KL}}(\tilde{\pi}). \tag{16}$$

For $\delta_{\mathrm{TV}} \ge \sqrt{\frac{1}{2}\delta_{\mathrm{KL}}}$, we have $\mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi}^*_{\mathrm{TV}}) \ge \mathcal{L}_\pi^{\mathrm{KL}}(\tilde{\pi}^*_{\mathrm{KL}})$.

Proof. Since $\Omega_{\mathrm{KL}} \subset \Omega_{\mathrm{TV}}$ by Proposition B.2:

$$\begin{aligned}
\mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi}^*_{\mathrm{TV}}) &\ge \mathcal{L}_\pi^{\mathrm{TV}}(\tilde{\pi}^*_{\mathrm{KL}}) \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim\pi}\left[\frac{\tilde{\pi}^*_{\mathrm{KL}}(a|s)}{\pi(a|s)}A_\pi(s,a)\right] - \frac{2\xi\gamma}{(1-\gamma)^2}\cdot \mathbb{E}_{s\sim\rho_\pi}[D_{\mathrm{TV}}(\pi\|\tilde{\pi}^*_{\mathrm{KL}})[s]] \\
&\ge \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim\pi}\left[\frac{\tilde{\pi}^*_{\mathrm{KL}}(a|s)}{\pi(a|s)}A_\pi(s,a)\right] - \frac{2\xi\gamma}{(1-\gamma)^2}\cdot \mathbb{E}_{s\sim\rho_\pi}\left[\sqrt{\tfrac{1}{2}D_{\mathrm{KL}}(\pi\|\tilde{\pi}^*_{\mathrm{KL}})[s]}\right] \\
&= \mathcal{L}_\pi^{\mathrm{KL}}(\tilde{\pi}^*_{\mathrm{KL}}).
\end{aligned} \tag{17}$$

where the second inequality follows from Pinsker's inequality. □

B.3 The ϵ-Aligned Objective Class

To understand why PPO's clipping fails to constrain probability ratios while SPO succeeds, Xie et al. (2025) introduce the concept of ϵ-aligned objectives.

Based on the performance bound in Equation (11), the goal is to solve the following constrained optimization problem:

$$\begin{aligned}
\max_\theta \ & \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[r_t(\theta)\cdot\hat{A}_t\right], \\
\text{s.t. } & \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[|r_t(\theta)-1|\right] \le \epsilon,
\end{aligned} \tag{18}$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ and $\hat{A}_t = \hat{A}(s_t,a_t)$.

For a single data point with advantage $A \ne 0$, this simplifies to:

$$\max_r \ rA, \quad \text{s.t. } |r-1| \le \epsilon. \tag{19}$$

Since the objective is linear in r, the optimal solution lies at the constraint boundary: $r^* = 1 + \mathrm{sign}(A)\cdot\epsilon$.

Definition B.4 (ϵ-Aligned Objective)**.** For any given $A \ne 0$ and $\epsilon > 0$, a function $f(r, A, \epsilon)$ is *ϵ-aligned* if it is differentiable and convex with respect to r, and attains its maximum at $r = 1 + \mathrm{sign}(A)\cdot\epsilon$.

An ϵ-aligned objective converts the constrained problem into an unconstrained one whose optimal solution automatically satisfies the constraint. The

PPO and SPO objectives can be expressed as:

$$f_{\text{PPO}} = \min\left[rA, \text{clip}(r, 1-\epsilon, 1+\epsilon)\cdot A\right], \quad (20)$$

$$f_{\text{SPO}} = rA - \frac{|A|}{2\epsilon}(r-1)^2. \quad (21)$$

Theorem B.5 (SPO is ϵ-Aligned)**.** *The SPO objective f_{SPO} is ϵ-aligned, while the PPO objective f_{PPO} is not.*

Proof. For f_{SPO}: The objective is a quadratic polynomial in r, hence differentiable and convex. Setting the derivative to zero:

$$\frac{\partial f_{\text{SPO}}}{\partial r} = A - \frac{|A|}{\epsilon}(r-1) = 0 \implies r = 1 + \text{sign}(A)\cdot\epsilon. \quad (22)$$

Thus f_{SPO} is ϵ-aligned.

For f_{PPO}: The clipping operation causes the gradient to vanish when $r > 1+\epsilon$ and $A > 0$, or when $r < 1-\epsilon$ and $A < 0$. This means f_{PPO} is not differentiable everywhere and does not satisfy the ϵ-aligned definition. □

The practical consequence is that PPO's clipped objective zeros gradients for data points outside the trust region, providing no corrective signal to bring them back. In contrast, SPO maintains non-zero gradients that consistently guide optimization toward the constraint boundary $r = 1 + \text{sign}(A)\cdot\epsilon$, ensuring effective probability ratio control throughout training.

C Training Details

C.1 Reasoning Hyperparameters

Training hyperparameters for reasoning experiments are listed in Table 3. The same hyperparameters are used for both base and instruct model variants.

The system prompts used during training are shown below. The same prompts are used for both base and instruct model variants.

System prompt used for Qwen2.5-1.5B models.

You are a helpful AI Assistant, designed to provided well-reasoned and detailed responses. You FIRST think about the reasoning process step by step and then provide the user with the answer. Please enclose your final answer in the box: \boxed{Your Answer}.

System prompt used for Qwen2.5-3B models.

You are a helpful AI Assistant, designed to provided well-reasoned and detailed responses. You FIRST think about the reasoning process step by step and then provide the user with the answer. Please enclose your final answer in the box: \boxed{Your Answer}. Please stop generation immediately after outputing the box.

System prompt used for Qwen2.5-7B models.

You are a helpful AI Assistant, designed to provided well-reasoned and detailed responses. Please provide a step-by-step solution to the following problem.

C.2 DAPO Experiment Details

We follow the DAPO-based training recipe of Wang et al. (2025a) and train Qwen3-4B-Base (Team, 2025) using the `verl` framework. The recipe includes token-mean loss aggregation, dynamic batching, 16 rollouts per prompt, and dual-clip PPO with $\epsilon_{\text{low}} = 0.2$ and $c = 10.0$. We train on the 54.4k-problem combined math dataset from Wang et al. (2025a), which aggregates problems from MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and additional competition-level

Table 3: Training hyperparameters for Qwen2.5 reasoning experiments. Only hyperparameters that affect the learned policy or evaluation are listed. Unspecified fields inherit the TRL_v0.8 defaults.

Parameter	1.5B/3B	7B
Learning Rate	3×10^{-6}	1×10^{-6}
Batch Size	128	128
Group Size	3	3
KL Penalty (β)	0	0
Max Prompt Length	512	512
Max Completion Length	3072	3072
Temperature	0.9	0.9
Clip Ratio	0.2	0.2
Lr Scheduler Type	Cosine	Cosine
Warmup Ratio	0.1	0.1
Optimizer	AdamW (β_1=0.9, β_2=0.999, ε=10^{-8})	

sources. Training uses a batch size of 128, mini-batch size of 8, maximum response length of 8192 tokens, learning rate of 1×10^{-6}, and temperature 1.0. Evaluation uses MATH500 and AIME24 with top-p 0.7. The only modification between runs is the policy loss: `loss_mode=vanilla` for the clipping baseline and `loss_mode=cfpo` for CFPO. Training is conducted on 4×H200 GPUs; we report results at equal compute budgets.

C.3 RLHF Hyperparameters

For RLHF experiments on Llama-3-8B, we use the OpenRLHF framework with default hyperparameters, only modifying the policy loss to use CFPO. Key hyperparameters are listed in Table 4. We use Llama-3-8B-SFT-Mixture as the base model and Llama-3-8B-RM-700K as the reward model, both from OpenRLHF. Training is conducted on the OpenRLHF prompt-collection-v0.1 dataset.

D Qwen2.5 Results and Additional Figures

Table 4: RLHF training hyperparameters for Llama-3-8B experiments.

Parameter	Value
Actor Learning Rate	5×10^{-7}
Train Batch Size	64
Rollout Batch Size	512
Samples per Prompt	2
Max Prompt Length	1024
Max Generation Length	1024
KL Penalty (β)	1×10^{-4} / 0×10^{-4}
Advantage Estimator	RLOO

Table 5: RLVR performance of Qwen2.5-1.5B-Instruct on MATH500, GSM8K, GPQA-Diamond, and AIME24 under GRPO and CFPO across increasing iteration counts. Both methods refine existing instruction-following and reasoning behavior with limited gains under small training budgets. The primary difference emerges in optimization stability: GRPO degrades at higher iteration counts, while CFPO maintains more consistent performance across tasks.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-1.5B-Instruct	0.528	0.648	0.272	0.033
GRPO Training				
+ GRPO-iteration 2	0.580	0.745	0.222	0.033
+ GRPO-iteration 4	0.544	0.682	0.277	0.033
+ GRPO-iteration 8	0.370	0.604	0.248	0.033
+ GRPO-iteration 16	0.100	0.349	0.262	0.000
CFPO Training				
+ CFPO-iteration 2	0.544	0.726	0.242	0.033
+ CFPO-iteration 4	0.534	0.692	0.333	0.033
+ CFPO-iteration 8	0.524	0.714	0.282	0.000
+ CFPO-iteration 16	0.500	0.689	0.248	0.000

Table 6: RLVR results for Qwen2.5-3B-Instruct across multiple reasoning benchmarks and iteration counts. With instruction-following already present, both GRPO and CFPO yield similar downstream behavior, indicating limited scope for qualitative improvement under short-horizon RL. Nevertheless, CFPO remains stable across all iteration counts, whereas GRPO exhibits degradation as off-policy pressure increases.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-3B-Instruct	0.645	0.764	0.328	0.03
GRPO Training				
+ GRPO-iteration 2	0.656	0.793	0.292	0.067
+ GRPO-iteration 4	0.612	0.785	0.323	0.000
+ GRPO-iteration 8	0.450	0.759	0.262	0.033
+ GRPO-iteration 16	0.080	0.205	0.258	0.067
CFPO Training				
+ CFPO-iteration 2	0.658	0.762	0.333	0.033
+ CFPO-iteration 4	0.674	0.796	0.288	0.067
+ CFPO-iteration 8	0.660	0.769	0.343	0.100
+ CFPO-iteration 16	0.558	0.752	0.328	0.100

Table 7: RLVR performance of Qwen2.5-7B-Instruct under GRPO and CFPO across iteration counts. At this scale, both methods achieve comparable refinement of reasoning and instruction-following behavior. However, GRPO becomes unstable at high iteration counts, while CFPO preserves consistent performance, highlighting its robustness under extended training.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-7B-Instruct	0.756	0.816	0.363	0.100
GRPO Training				
+ GRPO-iteration 2	0.750	0.893	0.369	0.067
+ GRPO-iteration 4	0.774	0.906	0.354	0.133
+ GRPO-iteration 8	0.764	0.879	0.394	0.133
+ GRPO-iteration 16	0.080	0.316	0.277	0.000
CFPO Training				
+ CFPO-iteration 2	0.758	0.845	0.348	0.100
+ CFPO-iteration 4	0.772	0.839	0.338	0.167
+ CFPO-iteration 8	0.768	0.835	0.318	0.100
+ CFPO-iteration 16	0.758	0.869	0.318	0.130

Table 8: Cold-start RL performance of Qwen2.5-1.5B on reasoning benchmarks under GRPO and CFPO across iteration counts. Both methods enable the emergence of reasoning behavior from a base model without prior instruction tuning. GRPO exhibits rapid early improvement followed by instability as iterations increase, whereas CFPO progresses more conservatively and maintains stable performance over longer training horizons.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-1.5B	0.002	0.090	0.171	0
GRPO Training				
GRPO-Iteration2	0.506	0.672	0.242	0.00
GRPO-Iteration4	0.558	0.729	0.293	0.00
GRPO-Iteration8	0.120	0.030	0.278	0.00
GRPO-Iteration16	0.074	0.039	0.197	0.00
CFPO Training				
CFPO-Iteration2	0.514	0.548	0.288	0.067
CFPO-Iteration4	0.574	0.570	0.253	0.033
CFPO-Iteration8	0.530	0.280	0.253	0.033
CFPO-Iteration16	0.380	0.020	0.308	0.033

Table 9: Cold-start RL results for Qwen2.5-3B across reasoning benchmarks and iteration counts. This model scale shows the clearest stability advantage for CFPO, which remains robust across all iteration settings. In contrast, GRPO degrades as iteration counts grow, consistent with the training dynamics observed in reward and clipping metrics.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-3B	0.673	0.544	0.262	0.06
GRPO Training				
GRPO-Iteration2	0.642	0.825	0.318	0.067
GRPO-Iteration4	0.648	0.796	0.303	0.033
GRPO-Iteration8	0.462	0.669	0.248	0.033
GRPO-Iteration16	0.220	0.673	0.252	0.033
CFPO Training				
CFPO-Iteration2	0.634	0.766	0.313	0.100
CFPO-Iteration4	0.616	0.767	0.328	0.067
CFPO-Iteration8	0.594	0.779	0.268	0.067
CFPO-Iteration16	0.604	0.798	0.313	0.067

Table 10: Cold-start RL performance of Qwen2.5-7B under GRPO and CFPO across increasing iteration counts. While both objectives improve reasoning behavior at moderate training lengths, GRPO collapses under aggressive iteration settings. CFPO sustains stable optimization over longer horizons, though extreme iteration counts eventually degrade performance even under quadratic regularization.

Model	**Math 500**	**GSM8K**	**GPQA-Diamond**	**AIME 24**
Qwen2.5-7B	0.553	0.636	0.308	0.100
GRPO Training				
GRPO-Iteration2	0.730	0.699	0.303	0.100
GRPO-Iteration4	0.722	0.762	0.323	0.056
GRPO-Iteration8	0.710	0.645	0.323	0.067
GRPO-Iteration16	0.002	0.004	0.298	0.000
CFPO Training				
CFPO-Iteration2	0.704	0.596	0.313	0.067
CFPO-Iteration4	0.711	0.661	0.257	0.100
CFPO-Iteration8	0.722	0.602	0.323	0.100
CFPO-Iteration16	0.206	0.059	0.237	0.000

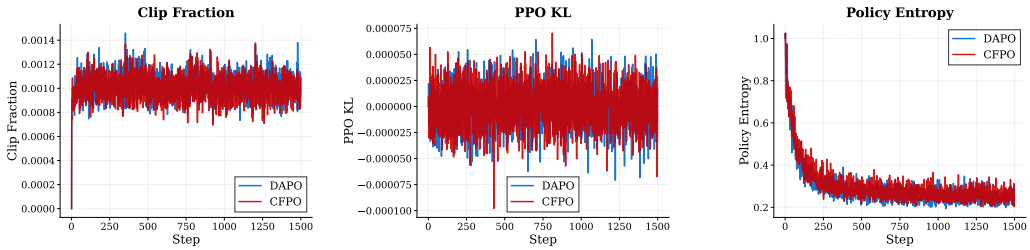

Figure 5: Training dynamics for DAPO and DAPO + CFPO on Qwen3-4B-Base. Clip fraction and KL divergence are comparable across methods, while CFPO maintains higher policy entropy throughout training—consistent with the entropy preservation pattern observed in Qwen2.5 experiments (Figure 10). This confirms that CFPO's conservative optimization behavior persists when combined with DAPO's orthogonal modifications on a different model family.

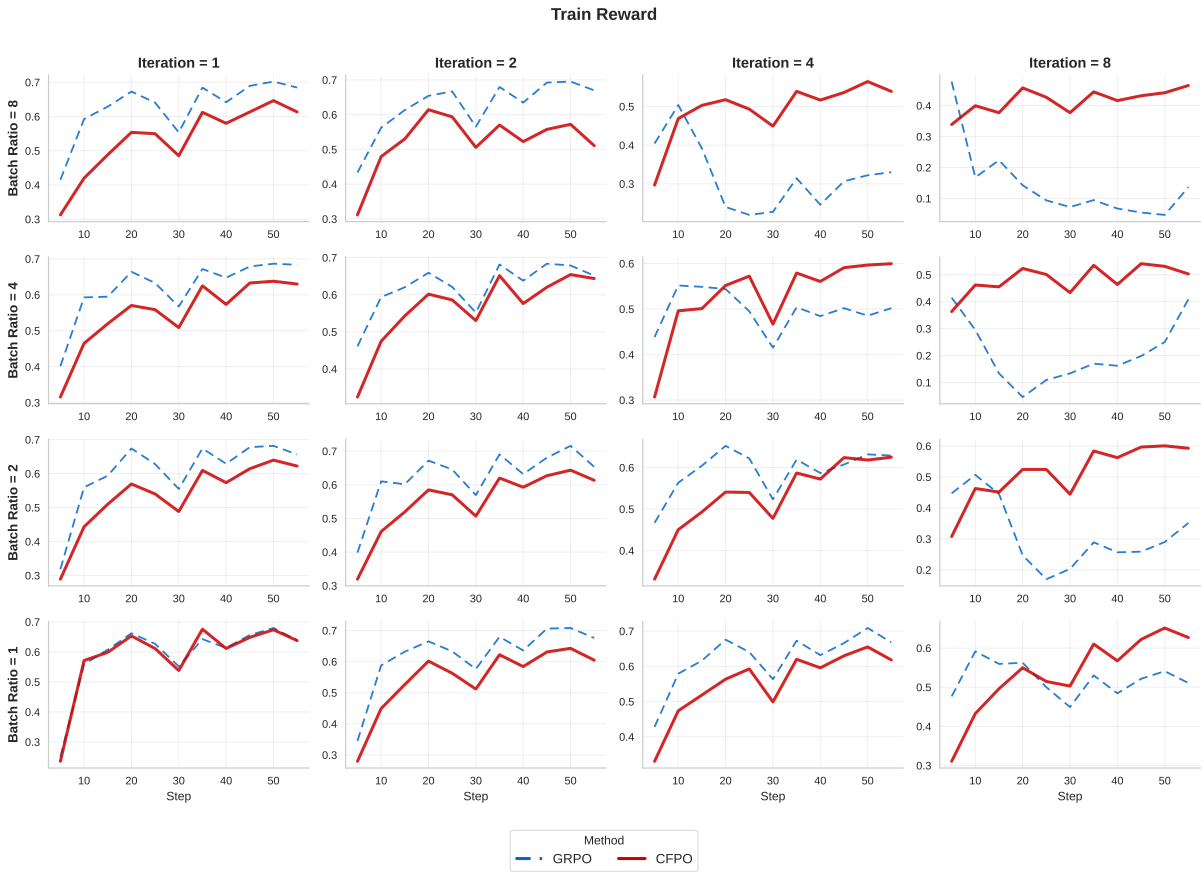

Figure 6: Training reward dynamics for cold-start RLVR training of Qwen2.5-3B using `verl` across batch ratios and iteration counts. GRPO exhibits faster early reward improvement, while CFPO progresses more gradually and converges later. Increasing iteration count leads to instability around 8 iterations, whereas increasing batch ratio alone remains comparatively stable, highlighting the stronger destabilizing effect of iteration-based sample reuse.

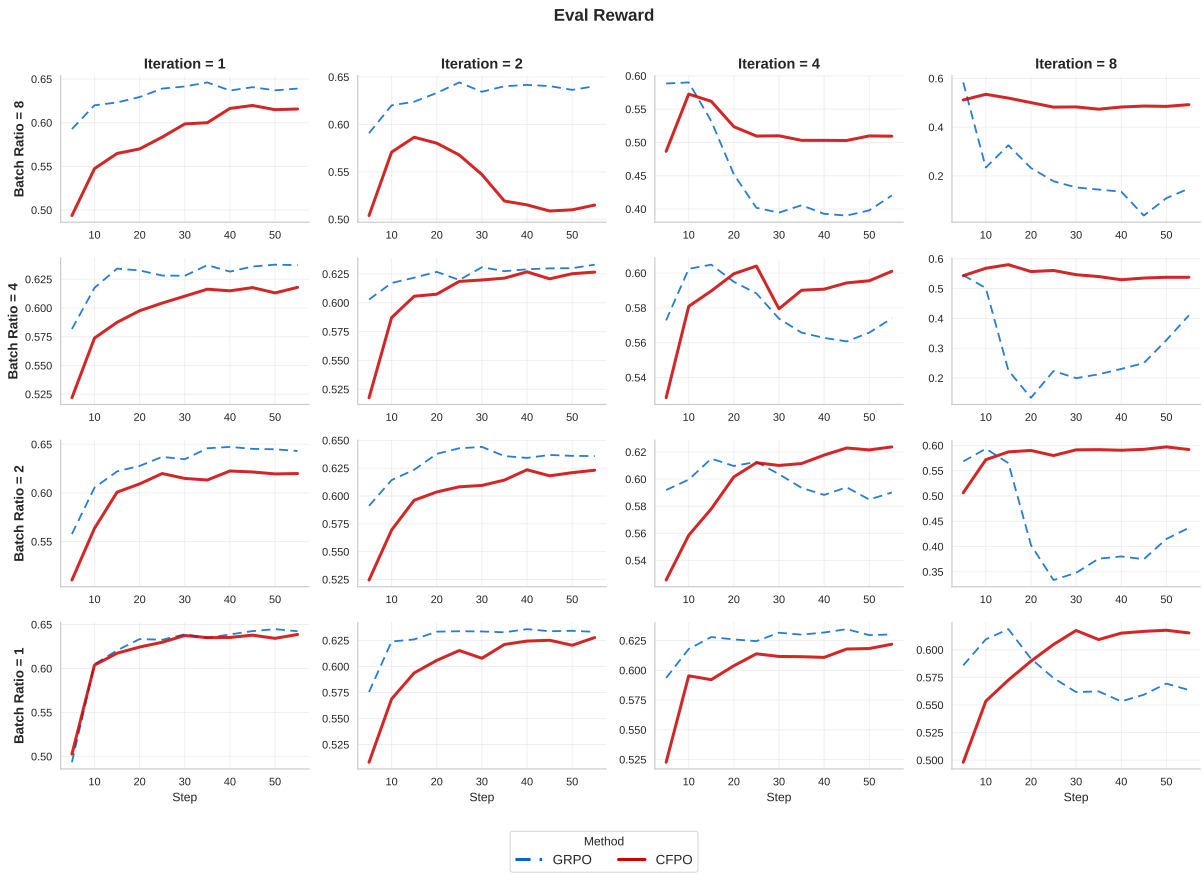

Figure 7: Validation reward during cold-start RLVR training of Qwen2.5-3B under GRPO and CFPO across batch ratios and iteration counts. Trends largely mirror training reward, with no systematic gains from increased off-policy updates. Instability emerges primarily with higher iteration counts rather than larger batch ratios, indicating limited generalization benefits from aggressive off-policy training.

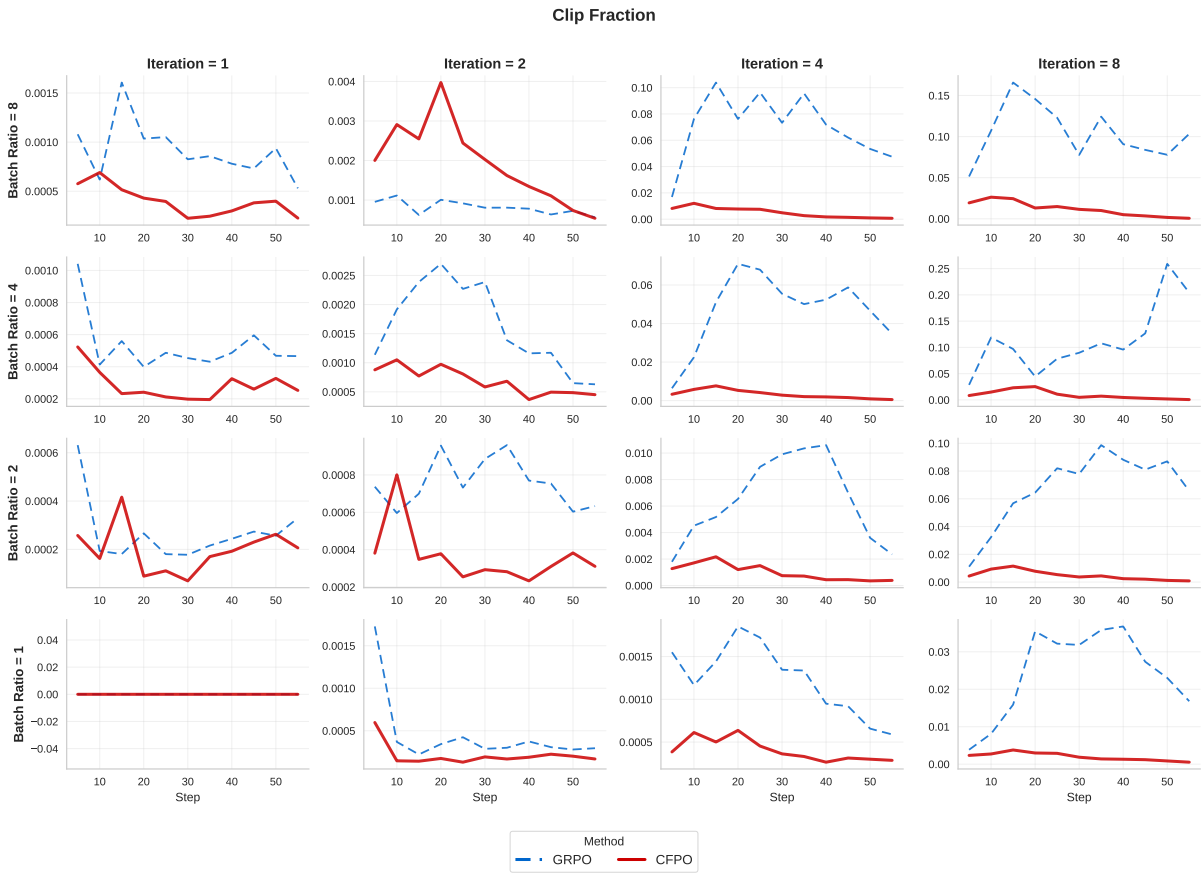

Figure 8: Policy gradient clipping ratios during cold-start RLVR training of Qwen2.5-3B using `verl`. For GRPO, clipping activity increases with both batch ratio and iteration count, and sharp rises in clipping precede training instability. CFPO consistently maintains lower and more stable clipping behavior across all settings, reflecting its smoother update geometry.

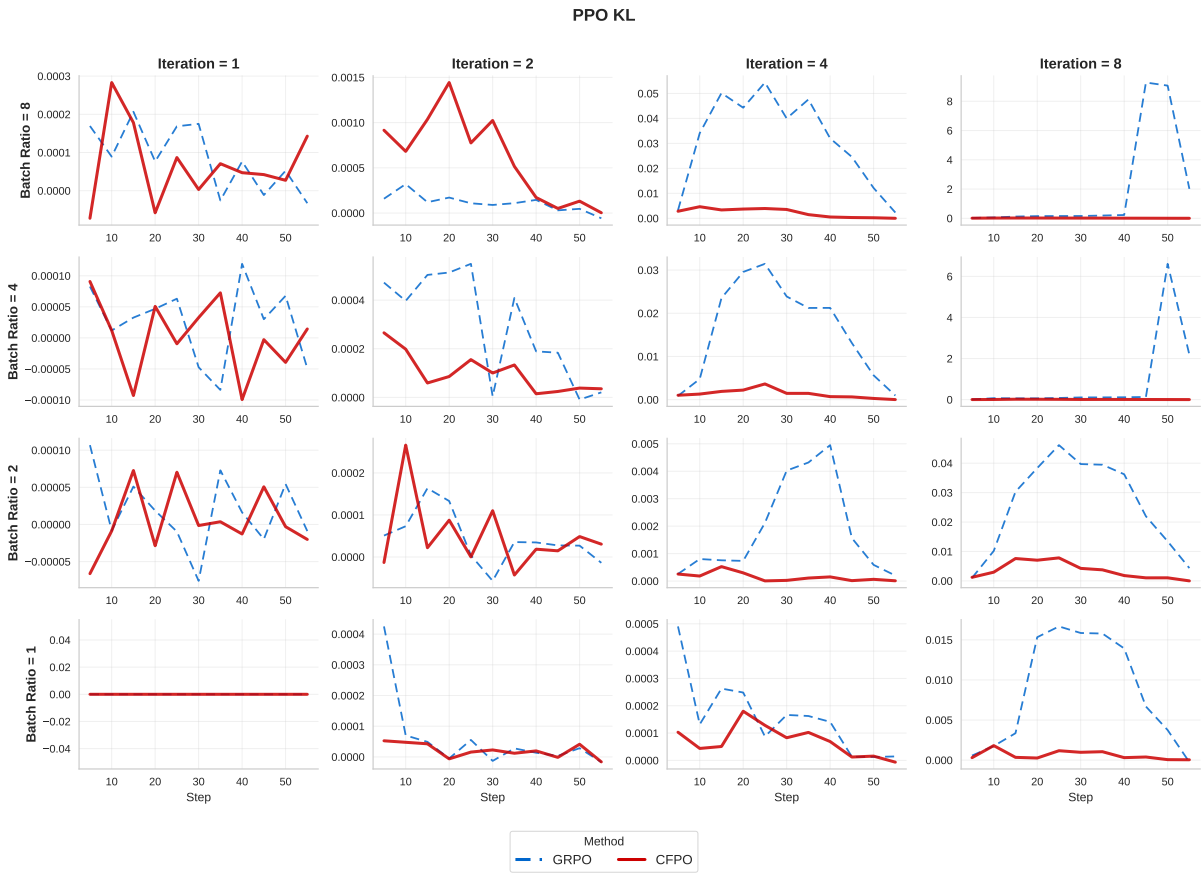

Figure 9: KL divergence between consecutive policies during cold-start RLVR training of Qwen2.5-3B across batch ratios and iteration counts. Despite differing optimization behavior, GRPO and CFPO exhibit similar KL magnitudes throughout training. This indicates that observed stability differences are not explained by large inter-policy shifts, but rather by differences in how updates are regularized within the trust region.

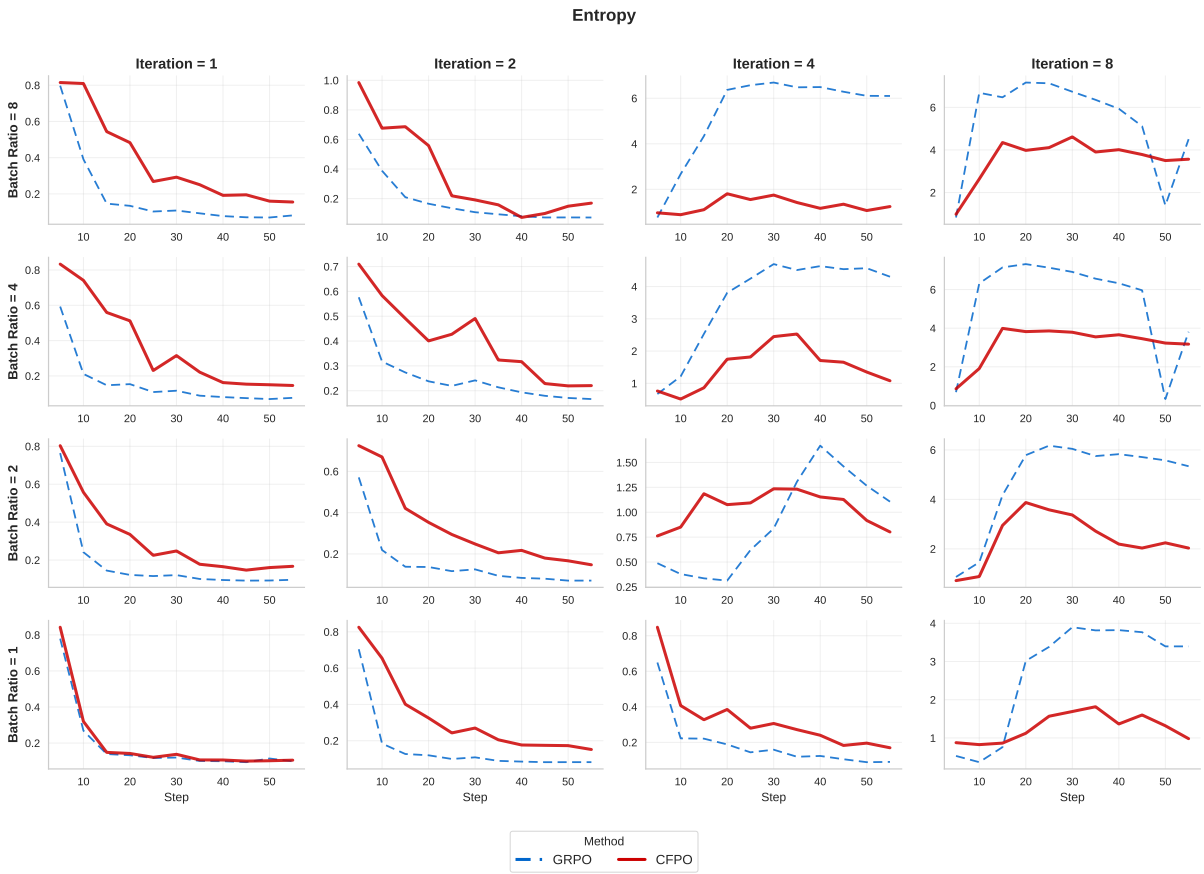

Figure 10: Policy entropy during cold-start RLVR training of Qwen2.5-3B under GRPO and CFPO across batch ratios and iteration counts. GRPO exhibits a faster reduction in entropy, particularly at higher iteration counts, consistent with its more aggressive optimization behavior. CFPO maintains higher entropy values over training, reflecting less aggressive policy updates and more gradual concentration of the policy distribution.