

# Clipping-Free Policy Optimization for Large Language Models

Ömer Veysel Çağatan<sup>1</sup> Barış Akgün<sup>1,2</sup> Gözde Gül Şahin<sup>1,2</sup> Xuandong Zhao<sup>3</sup>

## 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 a central component of large language model (LLM) post-training. Early work demonstrated that RL from human feedback (RLHF) could align models with human preferences and instructions (Christiano et al., 2023; Stiennon et al., 2022; Bai et al., 2022; Ouyang et al., 2022), and more recent efforts have shown that RL with verifiable rewards (RLVR) can elicit complex reasoning behaviors (DeepSeek-AI, 2025; OpenAI, 2024b; Google DeepMind, 2025). These successes have established RL as an essential stage in modern LLM training pipelines.

<sup>1</sup>KUIS AI Center, Koç University, İstanbul, Türkiye  
<sup>2</sup>Koç University, İstanbul, Türkiye <sup>3</sup>University of California, Berkeley, CA, USA. Correspondence to: Ömer Veysel Çağatan <ocagatan19@ku.edu.tr>, Xuandong Zhao <xuandongzhao@berkeley.edu>.

Preprint. February 2, 2026.

The dominant algorithms for LLM post-training, including PPO (Schulman et al., 2017b), GRPO (Shao et al., 2024), and their variants, rely on clipped surrogate objectives to stabilize policy updates. Clipping serves as a computationally efficient approximation to trust region constraints: probability ratios between the current and previous policy are clipped to a narrow range, removing incentives for updates that would change the policy too drastically. This mechanism has enabled scaling RL to large language models, but it 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. Beyond the clipping boundary, gradients vanish entirely. This combination creates pathological dynamics: models learn to exploit superficial correlates of reward such as verbosity (Gao et al., 2022), rapid policy drift degrades capabilities acquired during pretraining (Ouyang et al., 2022; Casper et al., 2023), and zero-gradient regions contribute to entropy collapse and training instability (Liu et al., 2025; Huang et al., 2025; Yu et al., 2025; Team, 2025; MiniMax, 2025). These failures manifest across model scales and training configurations, suggesting they are intrinsic to clipping rather than incidental to specific implementations.

Recent work 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), and auxiliary regularization (Wang et al., 2025). While these methods mitigate specific failure modes, they treat clipping as a mechanism to be patched rather than replaced, introducing additional hyperparameters 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 Simple Policy Optimization (SPO) (Xie et al., 2025), which replaces clipping with a convex quadratic penalty derived from Total Variation divergence constraints (Queeney et al., 2021). Unlike clipping, this objective is everywhere differentiable: the quadratic term applies a restoring force proportional to deviation from the old policy, rather than zeroing gradients beyond a threshold. As we show empirically, this produces more stable

optimization dynamics.

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:

- **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.
- **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 leaderboard (Beeching et al., 2023) tasks.

Overall, our results suggest that CFPO is a promising drop-in alternative to clipping-based methods, offering more stable training without sacrificing downstream performance while requiring only a one-line code change and no additional hyperparameters.

## 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  $\pi_\theta$ , PPO maximizes a clipped surrogate objective:

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

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  $\epsilon$  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.

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

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

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

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\}_{i=1}^G} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left( L_{\text{clip}}(r_{i,t}, \hat{A}_i) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}] \right) \right], \quad (3)$$

where  $L_{\text{clip}}$  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  $\beta$  controls KL regularization against a reference policy  $\pi_{\text{ref}}$ .

### 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 A.

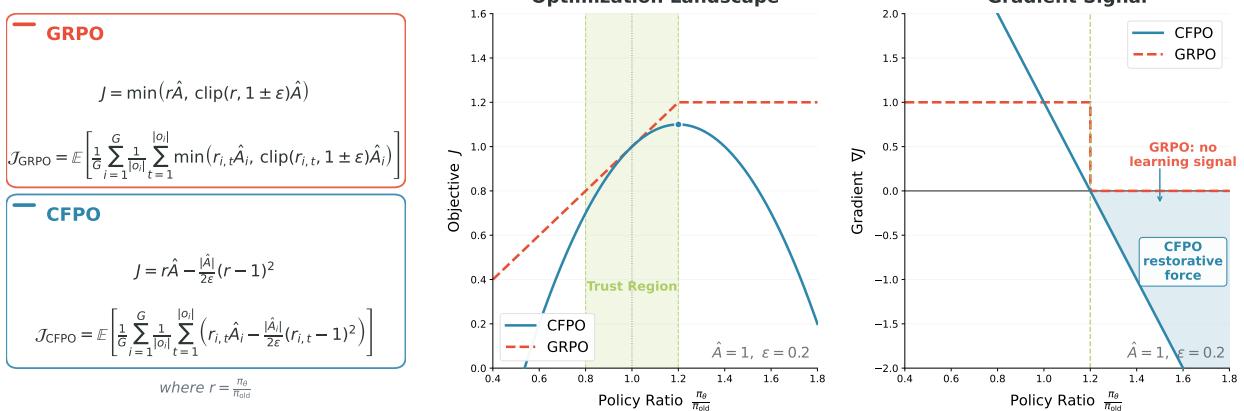


Figure 1. Optimization objective and gradient of GRPO and CFPO as functions of the policy ratio  $r = \pi / \pi_{\text{old}}$ , shown for advantage  $A = 1$  and trust-region width  $\epsilon = 0.2$ . GRPO becomes flat once  $r$  exits the trust region, resulting in zero gradient beyond the clipping 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.

**Proposition 2.1** (TV Solution Space Contains KL Solution Space). *Let  $\Omega_{\text{TV}}$  and  $\Omega_{\text{KL}}$  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  $\Omega_{\text{TV}}$  and  $\Omega_{\text{KL}}$ . For  $\delta_{\text{TV}} \geq \sqrt{\delta_{\text{KL}}/2}$ , the following holds:*

$$\mathcal{L}_{\pi}^{\text{TV}}(\tilde{\pi}_{\text{TV}}^*) \geq \mathcal{L}_{\pi}^{\text{KL}}(\tilde{\pi}_{\text{KL}}^*). \quad (4)$$

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

$$\mathcal{J}_{\text{SPO}}(\theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} \left[ r_t \hat{A}_t - \frac{|\hat{A}_t|}{2\epsilon} (r_t - 1)^2 \right], \quad (5)$$

where  $r_t = \pi_{\theta}(a_t | s_t) / \pi_{\theta_{\text{old}}}(a_t | s_t)$ . 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:

$$L_{\text{clip}} = \min \left( r_{i,t} \hat{A}_i, \text{clip}(r_{i,t}, 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right), \quad (6)$$

CFPO instead applies a quadratic penalty:

$$L_{\text{CFPO}} = r_{i,t} \hat{A}_i - \frac{|\hat{A}_i|}{2\epsilon} (r_{i,t} - 1)^2, \quad (7)$$

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  $\epsilon$  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 restorative 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.

The full CFPO objective is:

$$\mathcal{J}_{\text{CFPO}}(\theta) = \mathbb{E}_{q, \{o_i\}_{i=1}^G} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left( r_{i,t} \hat{A}_i - \frac{|\hat{A}_i|}{2\epsilon} (r_{i,t} - 1)^2 - \beta \mathbb{D}_{\text{KL}}[\pi_{\theta} \| \pi_{\text{ref}}] \right) \right], \quad (8)$$

where  $\beta$  is the KL regularization coefficient against a reference policy  $\pi_{\text{ref}}$ . 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.

### 3.2. Advantage Estimation

**Group-Relative Advantages.** For reasoning tasks, we follow GRPO (Shao et al., 2024). Given a prompt  $q$ , we sample a group of  $G$  responses  $\{o_1, o_2, \dots, o_G\}$  from the current policy and compute group-normalized advantages:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}, \quad (9)$$

where  $R_i$  is the reward for response  $o_i$ . Group-relative advantages suit reasoning tasks where rewards are verifiable (e.g., correctness of mathematical solutions), and normalization by standard deviation helps stabilize learning when reward magnitudes vary across different problems.

**Leave-One-Out Advantages.** For alignment tasks, we adopt the REINFORCE Leave-One-Out (RLOO) estimator (Ahmadian et al., 2024). Given  $K$  sampled responses per prompt, RLOO computes the advantage as:

$$\hat{A}_i = R_i - \frac{1}{K-1} \sum_{j \neq i} R_j, \quad (10)$$

where each sample uses the remaining  $K-1$  samples as an unbiased baseline estimate, functioning as a parameter-free value function. Unlike group-relative advantages, RLOO does not normalize by standard deviation, resulting in different gradient scaling behavior.

The use of two distinct advantage estimators allows us to assess whether CFPO’s improvements stem from the quadratic penalty itself or from interactions with specific estimation choices. As we show in our experiments, CFPO exhibits consistent stability benefits across both settings, suggesting that replacing clipping with a convex penalty is broadly beneficial regardless of how advantages are computed.

## 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. Since the base models exhibit limited instruction-following ability prior to training, we do not require explicit separation between intermediate reasoning and final answers. Each training update processes 128 problems, and we generate 3 candidate solutions per problem. 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.

**Off-Policy Mechanisms in `verl` and TRL.** Comparing GRPO and CFPO requires off-policy settings where the effectiveness of trust region methods can be meaningfully evaluated; in on-policy RL, both methods reduce to simple advantage maximization. In the traditional deep RL literature (Schulman et al., 2017b; Engstrom et al., 2020; Huang et al., 2021; Xie et al., 2025), off-policy behavior is typically introduced through two mechanisms: sample reuse and mini-batch gradient updates.

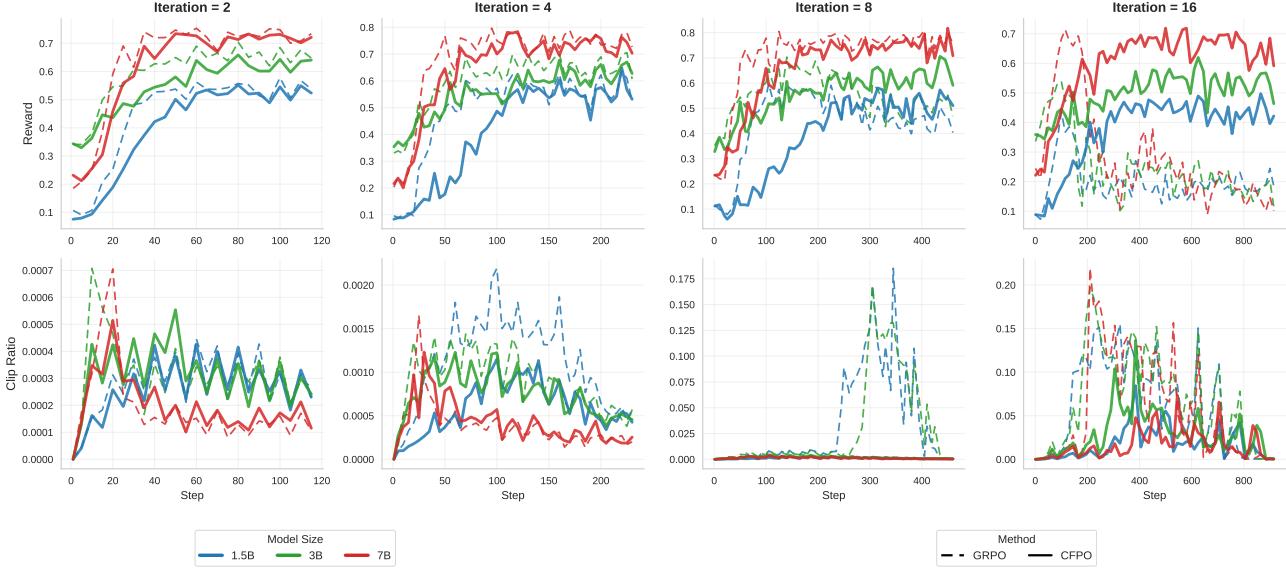
However, modern RL frameworks for LLMs differ in their implementations of policy gradients. `verl` follows the standard deep RL structure for off-policy training, while TRL, which Open-R1 is based on, deviates from this approach. TRL follows the GRPO recipe and only supports sample reuse (also referred to as iterations) without mini-batch updates. This discrepancy has practical consequences: for instance, clipping may be observed in `verl` but not in TRL for certain recipes that have different mini-batch size than batch size. Note that when iteration is set to 1 and mini-batch size equals batch size, both frameworks operate in the on-policy regime, where GRPO and CFPO are equivalent.

To fairly compare objectives across frameworks, we conduct experiments with the following configurations. For TRL-based models, we vary iterations as powers of 2 up to 16, across 3 model sizes, 2 policy loss types (GRPO and CFPO), and 2 model bases (base and instruct), yielding 48 models. For `verl`, we vary iterations up to 8, with 4 different batch ratios (batch size divided by mini-batch size, corresponding to the number of local updates), and 2 policy loss types, yielding 32 models. In total, we train 80 models across both frameworks.

**Evaluation.** We adopt the same chat-style prompting format used during training for evaluation. We use sampling-based decoding with temperature 0.6 and top- $p$  0.95 for all evaluations. We evaluate on MATH500 (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021), and AIME24 (Mathematical Association of America, 2024) for math reasoning, and GPQA-Diamond (Rein et al., 2023) for scientific reasoning, all using the `lighteval` library (Habib et al., 2023).

### 4.2. RLHF

**Training Setup.** To compare CFPO against RLOO (critic-free PPO with reinforce leave-one-out advantage estimation) in standard RLHF pipeline, we train models using the RLOO advantage estimation (Ahmadian et al., 2024). We use the supervised fine-tuning (SFT) and reward models provided by the OpenRLHF repository (Hu et al., 2024),



**Figure 2.** Training dynamics of CFPO vs. GRPO under increasing off-policy pressure. Reward (top) and clip ratio (bottom) trajectories for Qwen2.5 models trained with different numbers of iterations per update (columns). GRPO (dashed) exhibits faster early reward gains but increasingly large and unstable updates as iterations grow, reflected in rising clip ratios and eventual training collapse at higher iteration counts ( $\geq 8$  for most models). In contrast, CFPO (solid) progresses more conservatively, maintaining consistently low clip ratios and stable training across extended horizons, while ultimately reaching comparable reward levels. These dynamics illustrate the trade-off between optimization aggressiveness and stability in off-policy post-training, and highlight CFPO’s robustness to repeated sample reuse.

both based on Llama-3 backbones (AI@Meta, 2024). RLHF training additionally uses the prompt collections provided by OpenRLHF. All RLHF experiments use default OpenRLHF hyperparameters, with  $k = 2$  rollouts per prompt. For KL-free RLHF settings, we explicitly set the KL coefficient to zero; otherwise, all parameters follow the default configuration.

**Evaluation.** We evaluate RLHF-trained models on a range of instruction-following benchmarks, including 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). AlpacaEval 2.0 and Arena-Hard v0.1 are judged using GPT-4.1 (OpenAI, 2025), while MT-Bench evaluations use GPT-4 (OpenAI, 2024a). To assess whether RLHF preserves general model capabilities, we additionally evaluate on a subset of tasks from the OpenLLM Leaderboard (Beeching et al., 2023), specifically 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).

## 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

*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	HellaSwag	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

iterations, and training collapses by 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.

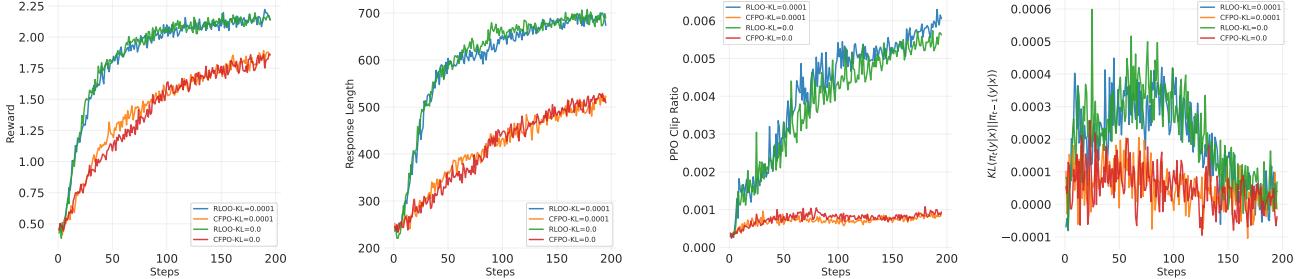
Interestingly, CFPO underperforms on GSM8K compared to GRPO for the 1.5B and 7B models. Qualitative inspection suggests this gap reflects weaker instruction-following rather than degraded reasoning: CFPO-trained models more frequently produce incomplete generations or responses in unintended languages. 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 `ver1`.** Several RLVR recipes implemented in `ver1` have non-zero clipping ratios even without explicit sample reuse. Inspection of these configurations reveals that off-policy effects arise primarily from mini-batch policy updates, which introduce mild policy lag despite using fresh data. This regime has motivated the use of relatively large clipping thresholds in prior work (Yu et al., 2025; Yang et al., 2025).

At a single iteration, observations closely mirror those from TRL. GRPO optimizes reward rapidly, while SPO converges more slowly but eventually catches up, achieving comparable training reward, validation reward, policy clipping fractions, and KL (Figs. 4, 5, 6, 7). In contrast, entropy dynamics differ markedly: GRPO consumes entropy substantially faster than SPO, reflecting more aggressive optimization behavior (Fig. 8). While this accelerates short-horizon reward gains, it likely undermines stability in longer training regimes where sustained exploration is required.



**Figure 3.** RLHF training dynamics on Llama3-8B under RLOO and CFPO with different KL penalty coefficients. We report trajectories over training steps for (a) reward, (b) generated response length, (c) policy clipping ratio, and (d) KL divergence between consecutive policy updates. RLOO exhibits rapid early reward increases accompanied by growing response lengths and elevated clipping activity, particularly when the KL penalty is weak or removed, indicating more aggressive optimization. In contrast, CFPO yields steadier reward improvement while maintaining stable response lengths, lower clipping ratios, and controlled KL divergence across settings, reflecting more conservative and stable policy updates during RLHF.

We next examine how different sources of off-policy pressure affect stability. Increasing the batch ratio up to 8 does not, by itself, induce collapse, whereas increasing the number of iterations leads to instability around 8 iterations (Figs. 4, 6). This contrast suggests that batch-ratio–induced off-policy updates with fresh data are less destabilizing than iteration-based sample reuse, which more readily amplifies policy lag.

However, these effects are not independent. Even at iteration counts where GRPO remains stable (e.g., 4 iterations), increasing the batch ratio can still induce collapse, indicating that off-policy pressure accumulates across mechanisms (Figs. 6, 8). In contrast, SPO consistently maintains lower clipping ratios, slower entropy decay, and more stable policy updates across configurations, contributing to improved robustness under off-policy regimes.

**Extensibility.** Moreover, recent RLVR work has introduced orthogonal techniques including token-level loss aggregation, dynamic sampling, and entropy regularization (Yu et al., 2025; Liu et al., 2025). Since CFPO modifies only the surrogate objective, these techniques transfer directly; practitioners can replace clipped objectives with CFPO’s penalty without further modification.

## 5.2. RLHF

We train CFPO in standard RLHF using RLOO advantage estimation on LLaMA-3-8B. All hyperparameters are taken directly from the OpenRLHF repository without tuning; the only modification is replacing the clipped surrogate with CFPO’s quadratic penalty. This isolates the effect of the objective itself. Tables 1 and 2 report results, with training dynamics shown in Figure 3.

**Optimization Dynamics.** As in the reasoning setting, we first examine how the choice of objective shapes optimiza-

tion behavior during training. Figure 3 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 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. 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 & 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. We extend SPO to language model training.

**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, 2024b; 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  $\lambda$ -GRPO (Wang et al., 2025) 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.

## 7. Discussion and Future Work

Our experiments span models from 1.5B to 8B parameters across reasoning and alignment tasks, but are limited to Qwen and LLaMA model families trained on MATH and OpenRLHF datasets. Frontier models operate at substantially larger scales with longer training horizons; we did not have sufficient compute to evaluate CFPO in these regimes, and whether its conservative updates remain beneficial or become overly restrictive at scale is an open question. Similarly, due to resource constraints, we could not explore greater diversity in model architectures, training datasets, and domains with sparser or noisier rewards such as code generation or agentic applications which may reveal different optimization dynamics. We leave these directions to future work.

## 8. 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. These findings suggest that clipping’s aggressive optimization dynamics, while effective at rapidly acquiring surface-level patterns, introduces instabilities problematic at scale, and that CFPO’s more conservative updates offer a promising alternative for language model post-training.

## Acknowledgements

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG).

## References

- Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation, 2018. URL <https://arxiv.org/abs/1806.06920>.
- Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization, 2017. URL <https://arxiv.org/abs/1705.10528>.
- Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL <https://arxiv.org/abs/2402.14740>.
- AI@Meta. Llama 3 model card. 2024. URL [https://github.com/meta-llama/llama3/blob/main/MODEL\\_CARD.md](https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md).
- Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Kernion, J., Ndousse, K., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., and Kaplan, J. A general language assistant as a laboratory for alignment, 2021. URL <https://arxiv.org/abs/2112.00861>.
- Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences, 2023. URL <https://arxiv.org/abs/2310.12036>.
- Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional ai: Harmlessness from ai feedback, 2022. URL <https://arxiv.org/abs/2212.08073>.
- Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open LLM leaderboard. [https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard), 2023.
- Belousov, B. and Peters, J. f-divergence constrained policy improvement, 2018. URL <https://arxiv.org/abs/1801.00056>.
- Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T., Marks, S., Segerie, C.-R., Carroll, M., Peng, A., Christoffersen, P., Damani, M., Slocum, S., Anwar, U., Siththanjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krasheninnikov, D., Chen, X., Langosco, L., Hase, P., Biyik, E., Dragan, A., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023. URL <https://arxiv.org/abs/2307.15217>.
- Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences, 2023. URL <https://arxiv.org/abs/1706.03741>.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.
- Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep policy gradients: A case study on ppo and trpo, 2020. URL <https://arxiv.org/abs/2005.12729>.
- Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization, 2024. URL <https://arxiv.org/abs/2402.01306>.
- Face, H. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization, 2022. URL <https://arxiv.org/abs/2210.10760>.

- Google DeepMind. Gemini 2.5 pro. <https://deepmind.google/models/gemini/pro/>, 2025. Accessed: November 2025.
- Habib, N., Fourrier, C., Kydlíček, H., Wolf, T., and Tunstall, L. Lighteval: A lightweight framework for llm evaluation, 2023. URL <https://github.com/huggingface/lighteval>.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Hu, J., Wu, X., Zhu, Z., Xianyu, Wang, W., Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.
- Huang, G., Xu, T., Wang, M., Yi, Q., Gong, X., Li, S., Xiong, R., Li, K., Jiang, Y., and Zhou, B. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward, 2025. URL <https://arxiv.org/abs/2510.03222>.
- Huang, S., Dossa, R. F. J., Ye, C., and Braga, J. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms, 2021. URL <https://arxiv.org/abs/2111.08819>.
- Kakade, S. M. A natural policy gradient. In Dietterich, T., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL [https://proceedings.neurips.cc/paper\\_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf).
- Kakade, S. M. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002. URL <https://api.semanticscholar.org/CorpusID:31442909>.
- Lan, G. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes, 2022. URL <https://arxiv.org/abs/2102.00135>.
- Levesque, H., Davis, E., and Morgenstern, L. The Winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.
- Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., and Stoica, I. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline, 2024a. URL <https://arxiv.org/abs/2406.11939>.
- Li, Z., Xu, T., Zhang, Y., Lin, Z., Yu, Y., Sun, R., and Luo, Z.-Q. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models, 2024b. URL <https://arxiv.org/abs/2310.10505>.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step, 2023. URL <https://arxiv.org/abs/2305.20050>.
- Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, pp. 3214–3252, 2022.
- Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective, 2025. URL <https://arxiv.org/abs/2503.20783>.
- Marjanović, S. V., Patel, A., Adlakha, V., Aghajohari, M., BehnamGhader, P., Bhatia, M., Khandelwal, A., Kraft, A., Krojer, B., Lü, X. H., Meade, N., Shin, D., Kazemnejad, A., Kamath, G., Mosbach, M., Stańczak, K., and Reddy, S. Deepseek-r1 thoughtology: Let's think about llm reasoning, 2026. URL <https://arxiv.org/abs/2504.07128>.
- Mathematical Association of America. AIME problems and solutions. [https://artofproblemsolving.com/wiki/index.php/AIME\\_Problems\\_and\\_Solutions](https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions), 2024.
- Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward, 2024. URL <https://arxiv.org/abs/2405.14734>.
- MiniMax. Minimax-m1: Scaling test-time compute efficiently with lightning attention, 2025. URL <https://arxiv.org/abs/2506.13585>.
- OpenAI. Gpt-4 technical report, 2024a. URL <https://arxiv.org/abs/2303.08774>.
- OpenAI. Openai o1 system card, 2024b. URL <https://arxiv.org/abs/2412.16720>.
- OpenAI. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>, April 2025. Accessed: 15 May 2025.

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL <https://arxiv.org/abs/1910.00177>.
- Queeney, J., Paschalidis, I. C., and Cassandras, C. G. Generalized proximal policy optimization with sample reuse, 2021. URL <https://arxiv.org/abs/2111.00072>.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2024. URL <https://arxiv.org/abs/2305.18290>.
- Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL <https://arxiv.org/abs/2311.12022>.
- Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. Trust region policy optimization, 2017a. URL <https://arxiv.org/abs/1502.05477>.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017b. URL <https://arxiv.org/abs/1707.06347>.
- Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation, 2018. URL <https://arxiv.org/abs/1506.02438>.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. Learning to summarize from human feedback, 2022. URL <https://arxiv.org/abs/2009.01325>.
- Team, Q. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Tomar, M., Shani, L., Efroni, Y., and Ghavamzadeh, M. Mirror descent policy optimization, 2021. URL <https://arxiv.org/abs/2005.09814>.
- von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.
- Wang, Y., He, H., Wen, C., and Tan, X. Truly proximal policy optimization, 2020. URL <https://arxiv.org/abs/1903.07940>.
- Wang, Y., Zhao, J., Zhao, C., Guan, S., Penn, G., and Liu, S.  $\lambda$ -gpro: Unifying the gpro frameworks with learnable token preferences, 2025. URL <https://arxiv.org/abs/2510.06870>.
- Xie, Z., Zhang, Q., Yang, F., Hutter, M., and Xu, R. Simple policy optimization, 2025. URL <https://arxiv.org/abs/2401.16025>.
- Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- Yang, S., Dou, C., Guo, P., Lu, K., Ju, Q., Deng, F., and Xin, R. Dcpo: Dynamic clipping policy optimization, 2025. URL <https://arxiv.org/abs/2509.02333>.
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Qiao, M., Wu, Y., and Wang, M. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL <https://arxiv.org/abs/2503.14476>.
- Zelikman, E., Wu, Y., Mu, J., and Goodman, N. D. Star: Bootstrapping reasoning with reasoning, 2022. URL <https://arxiv.org/abs/2203.14465>.
- Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of*

*the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL <https://aclanthology.org/P19-1472>.

Zhao, X., Kang, Z., Feng, A., Levine, S., and Song, D. Learning to reason without external rewards, 2025. URL <https://arxiv.org/abs/2505.19590>.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models, 2023. URL <https://arxiv.org/abs/2311.07911>.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences, 2020. URL <https://arxiv.org/abs/1909.08593>.

## A. 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.

### A.1. Performance Improvement Bounds

The foundation of trust region methods lies in the policy performance difference theorem (Kakade & 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 A.1** (Performance Improvement Lower Bound). *Given any two policies  $\pi$  and  $\tilde{\pi}$ , the following bound holds:*

$$\eta(\tilde{\pi}) - \eta(\pi) \geq \frac{1}{1-\gamma} \mathbb{E}_{s \sim \rho_\pi, a \sim \tilde{\pi}} [A_\pi(s, a)] - \frac{2\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s \sim \rho_\pi} [D_{\text{TV}}(\pi \| \tilde{\pi})[s]], \quad (11)$$

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

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

$$\eta(\tilde{\pi}) - \eta(\pi) \geq \frac{1}{1-\gamma} \mathbb{E}_{s, a \sim \pi} \left[ \frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s, a) \right] - \frac{\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s, a \sim \pi} \left[ \left| \frac{\tilde{\pi}(a|s)}{\pi(a|s)} - 1 \right| \right]. \quad (12)$$

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.

### A.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 A.2** (TV Solution Space Contains KL Solution Space). *Given an old policy  $\pi$ , define the solution spaces under TV and KL divergence constraints as:*

$$\Omega_{\text{TV}} = \{ \tilde{\pi} \mid D_{\text{TV}}(\pi \| \tilde{\pi})[s] \leq \delta_{\text{TV}}, \forall s \in \mathcal{S} \}, \quad (13)$$

$$\Omega_{\text{KL}} = \{ \tilde{\pi} \mid D_{\text{KL}}(\pi \| \tilde{\pi})[s] \leq \delta_{\text{KL}}, \forall s \in \mathcal{S} \}, \quad (14)$$

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

*Proof.* For any  $\tilde{\pi} \in \Omega_{\text{KL}}$ , by Pinsker’s inequality:

$$D_{\text{TV}}(\pi \| \tilde{\pi})[s] \leq \sqrt{\frac{1}{2} D_{\text{KL}}(\pi \| \tilde{\pi})[s]} \leq \sqrt{\frac{1}{2} \delta_{\text{KL}}} \leq \delta_{\text{TV}}. \quad (15)$$

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

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

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

$$\mathcal{L}_\pi^{\text{TV}}(\tilde{\pi}) = \frac{1}{1-\gamma} \mathbb{E}_{s, a \sim \pi} \left[ \frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s, a) \right] - \frac{2\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s \sim \rho_\pi} [D_{\text{TV}}(\pi \| \tilde{\pi})[s]], \quad (16)$$

$$\mathcal{L}_\pi^{\text{KL}}(\tilde{\pi}) = \frac{1}{1-\gamma} \mathbb{E}_{s, a \sim \pi} \left[ \frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s, a) \right] - \frac{2\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s \sim \rho_\pi} \left[ \sqrt{\frac{1}{2} D_{\text{KL}}(\pi \| \tilde{\pi})[s]} \right]. \quad (17)$$

Define the optimal policies in each constrained space:

$$\tilde{\pi}_{\text{TV}}^* = \arg \max_{\tilde{\pi} \in \Omega_{\text{TV}}} \mathcal{L}_\pi^{\text{TV}}(\tilde{\pi}), \quad \tilde{\pi}_{\text{KL}}^* = \arg \max_{\tilde{\pi} \in \Omega_{\text{KL}}} \mathcal{L}_\pi^{\text{KL}}(\tilde{\pi}). \quad (18)$$

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

*Proof.* Since  $\Omega_{\text{KL}} \subset \Omega_{\text{TV}}$  by Proposition A.2:

$$\mathcal{L}_{\pi}^{\text{TV}}(\tilde{\pi}_{\text{TV}}^*) \geq \mathcal{L}_{\pi}^{\text{TV}}(\tilde{\pi}_{\text{KL}}^*) \quad (19)$$

$$= \frac{1}{1-\gamma} \mathbb{E}_{s,a \sim \pi} \left[ \frac{\tilde{\pi}_{\text{KL}}^*(a|s)}{\pi(a|s)} A_{\pi}(s,a) \right] - \frac{2\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s \sim \rho_{\pi}} [D_{\text{TV}}(\pi \| \tilde{\pi}_{\text{KL}}^*)[s]] \quad (20)$$

$$\geq \frac{1}{1-\gamma} \mathbb{E}_{s,a \sim \pi} \left[ \frac{\tilde{\pi}_{\text{KL}}^*(a|s)}{\pi(a|s)} A_{\pi}(s,a) \right] - \frac{2\xi\gamma}{(1-\gamma)^2} \mathbb{E}_{s \sim \rho_{\pi}} \left[ \sqrt{\frac{1}{2} D_{\text{KL}}(\pi \| \tilde{\pi}_{\text{KL}}^*)[s]} \right] \quad (21)$$

$$= \mathcal{L}_{\pi}^{\text{KL}}(\tilde{\pi}_{\text{KL}}^*), \quad (22)$$

where the second inequality follows from Pinsker's inequality.  $\square$

### A.3. The $\epsilon$ -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  $\epsilon$ -aligned objectives.

Based on the performance bound in Equation (12), the goal is to solve the following constrained optimization problem:

$$\begin{aligned} & \max_{\theta} \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} [r_t(\theta) \cdot \hat{A}_t], \\ & \text{s.t. } \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} [|r_t(\theta) - 1|] \leq \epsilon, \end{aligned} \quad (23)$$

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  $A \neq 0$ , this simplifies to:

$$\max_r rA, \quad \text{s.t. } |r - 1| \leq \epsilon. \quad (24)$$

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

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

An  $\epsilon$ -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:

$$f_{\text{PPO}} = \min [rA, \text{clip}(r, 1 - \epsilon, 1 + \epsilon) \cdot A], \quad (25)$$

$$f_{\text{SPO}} = rA - \frac{|A|}{2\epsilon} (r - 1)^2. \quad (26)$$

**Theorem A.5** (SPO is  $\epsilon$ -Aligned). *The SPO objective  $f_{\text{SPO}}$  is  $\epsilon$ -aligned, while the PPO objective  $f_{\text{PPO}}$  is not.*

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

$$\frac{\partial f_{\text{SPO}}}{\partial r} = A - \frac{|A|}{\epsilon} (r - 1) = 0 \implies r = 1 + \text{sign}(A) \cdot \epsilon. \quad (27)$$

Thus  $f_{\text{SPO}}$  is  $\epsilon$ -aligned.

For  $f_{\text{PPO}}$ : 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  $f_{\text{PPO}}$  is not differentiable everywhere and does not satisfy the  $\epsilon$ -aligned definition.  $\square$

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.

## B. Training Details

### B.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.

*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	$3 \times 10^{-6}$	$1 \times 10^{-6}$
Batch Size	128	128
Group Size	3	3
KL Penalty ( $\beta$ )	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	$\text{AdamW} (\beta_1=0.9, \beta_2=0.999, \varepsilon=10^{-8})$	

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 outputting 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.

### B.2. 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.

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

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

## C. Qwen2.5 Results in TRL

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
<i>GRPO Training</i>				
+ 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
<i>CFPO Training</i>				
+ 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
<i>GRPO Training</i>				
+ 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
<i>CFPO Training</i>				
+ 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
<i>GRPO Training</i>				
+ 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
<i>CFPO Training</i>				
+ 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
<i>GRPO Training</i>				
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
<i>CFPO Training</i>				
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
<i>GRPO Training</i>				
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
<i>CFPO Training</i>				
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
<i>GRPO Training</i>				
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
<i>CFPO Training</i>				
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

## D. `ver1` Figures

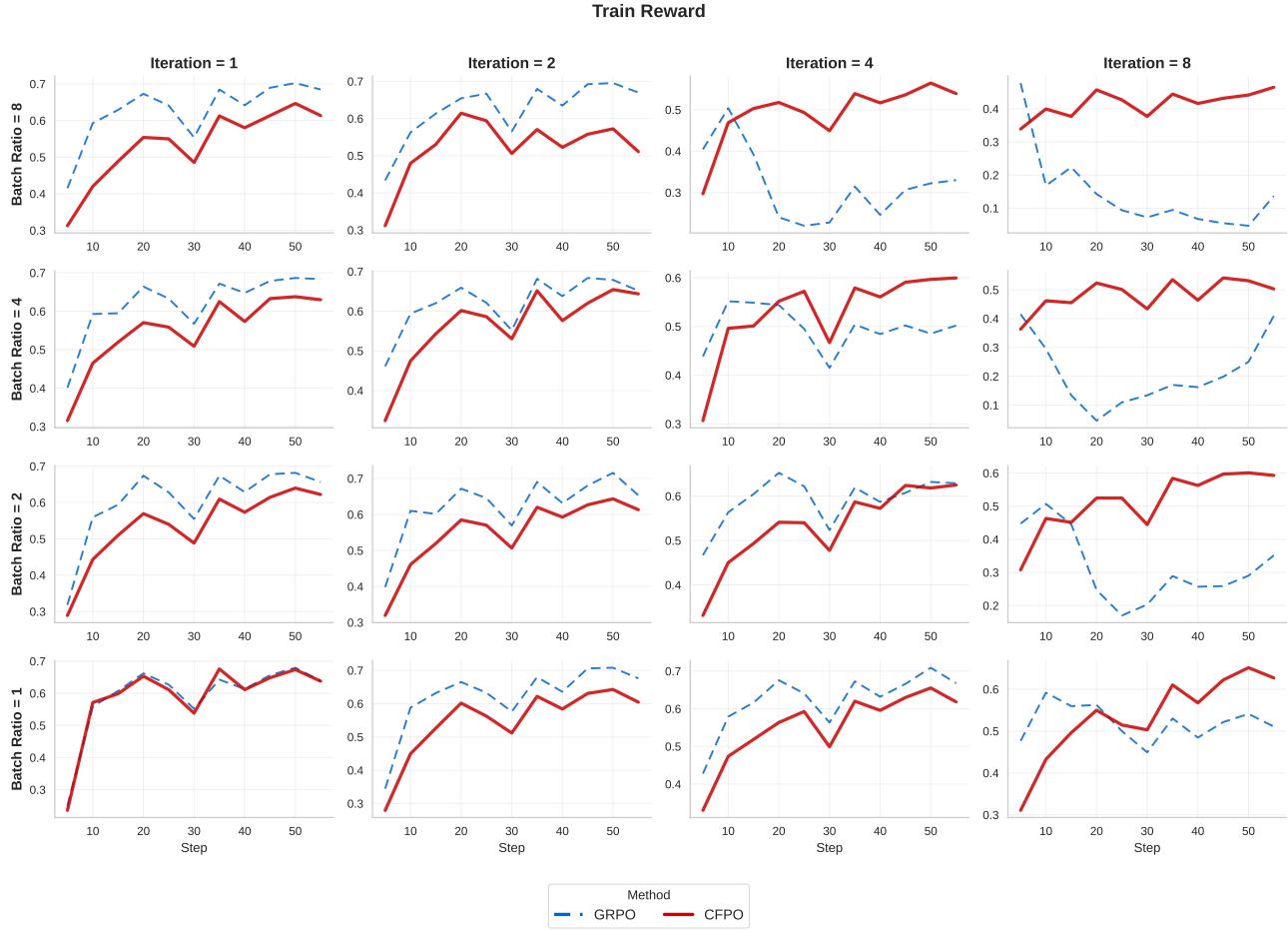
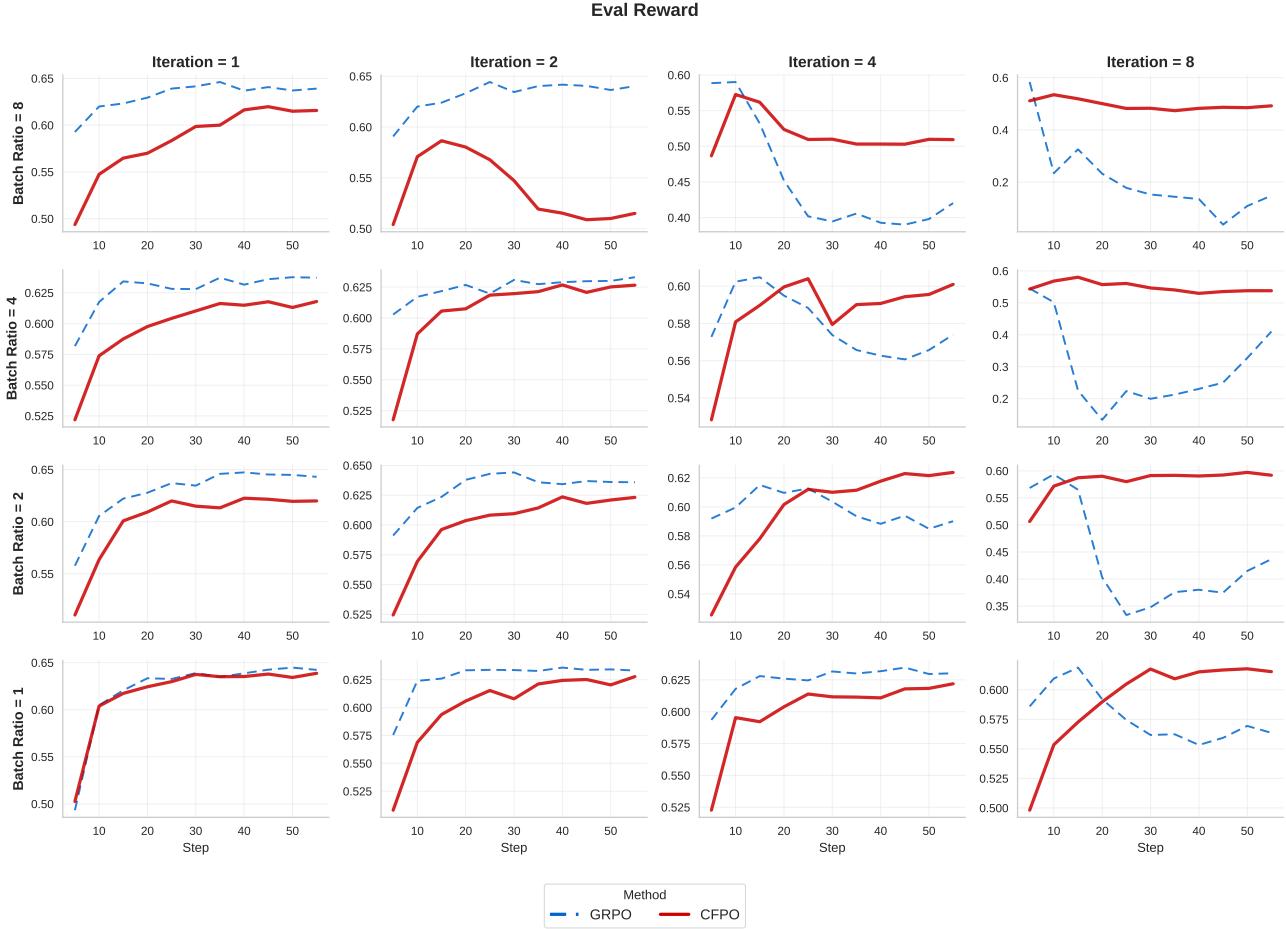


Figure 4. Training reward dynamics for cold-start RLVR training of Qwen2.5-3B using `ver1` across batch ratios and iteration counts. GRPO exhibits faster early reward improvement, while CFPO progresses more gradually and converges later. Increasing iteration count leads to instability around 8 iterations, whereas increasing batch ratio alone remains comparatively stable, highlighting the stronger destabilizing effect of iteration-based sample reuse.



*Figure 5.* 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.

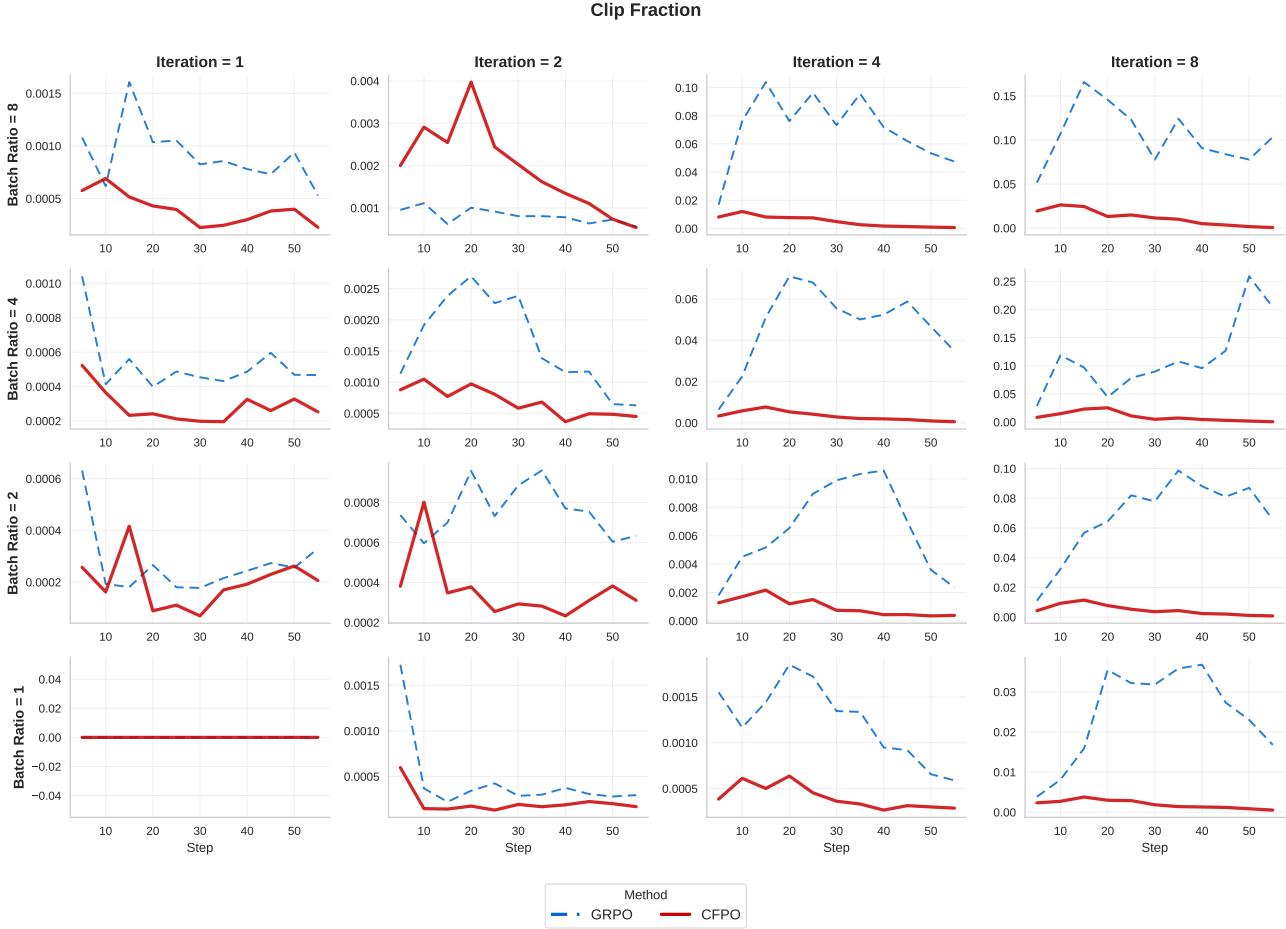
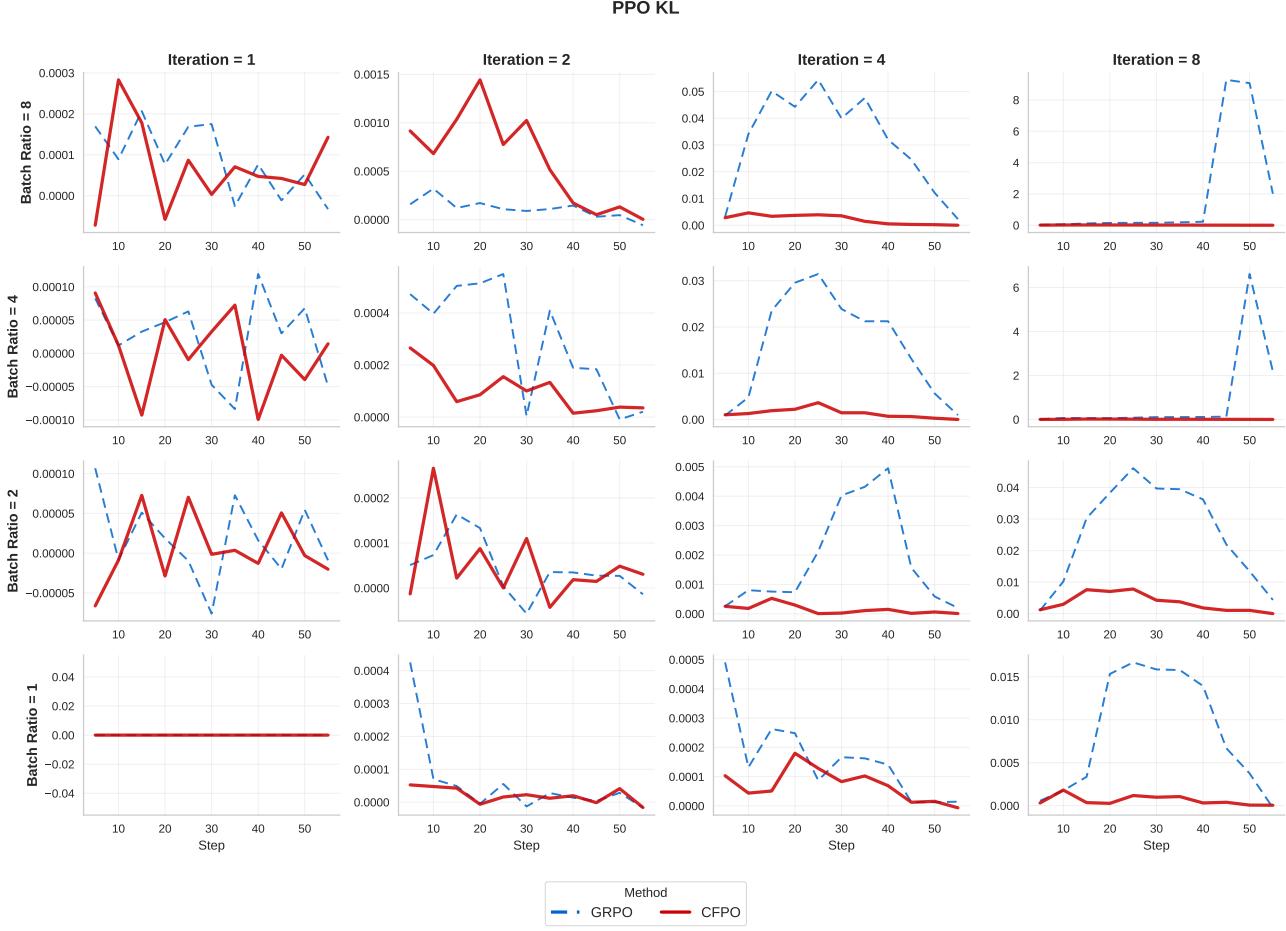
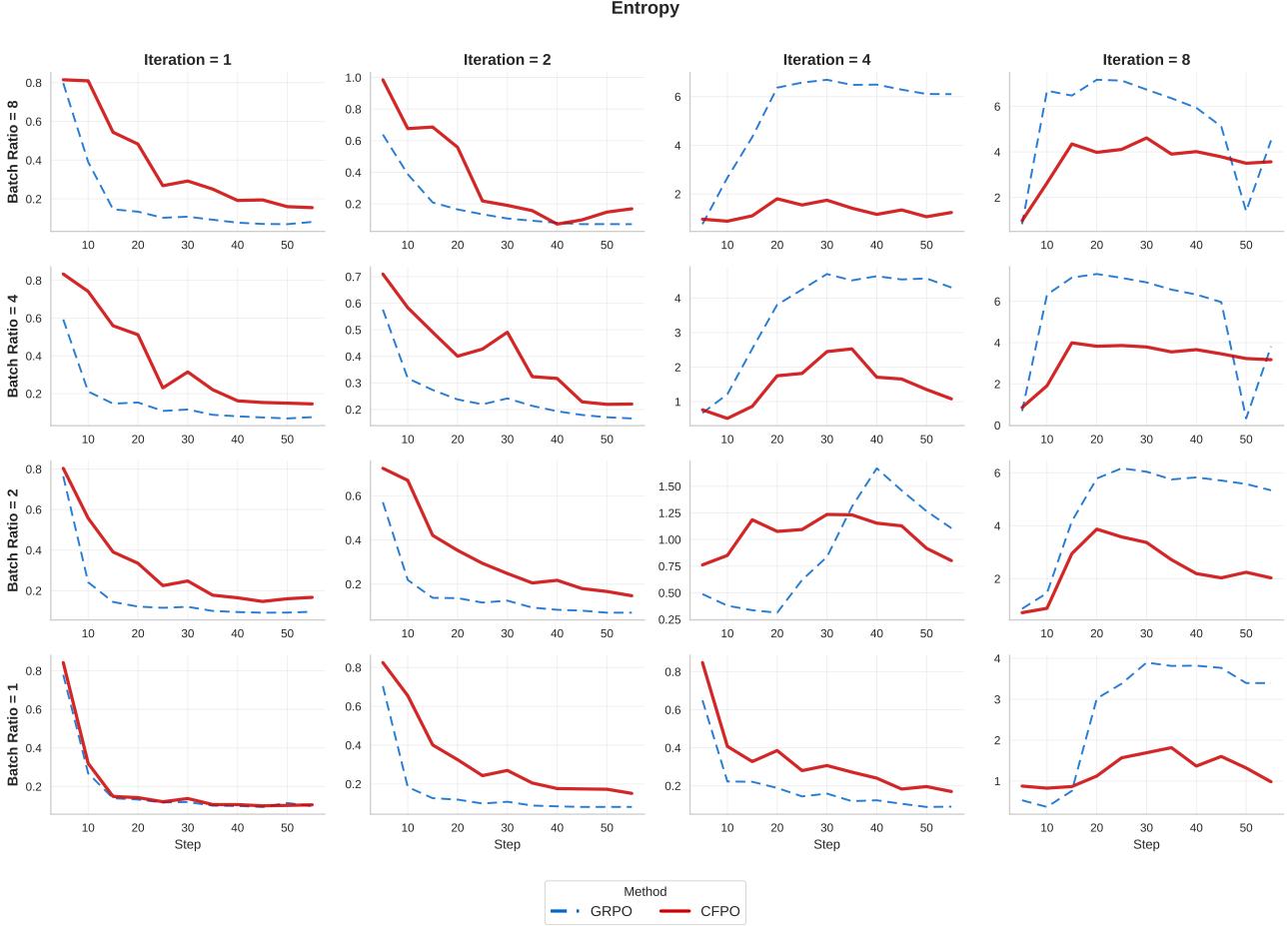


Figure 6. Policy gradient clipping ratios during cold-start RLVR training of Qwen2.5-3B using `ver1`. For GRPO, clipping activity increases with both batch ratio and iteration count, and sharp rises in clipping precede training instability. CFPO consistently maintains lower and more stable clipping behavior across all settings, reflecting its smoother update geometry.



*Figure 7.* 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 8.* Policy entropy during cold-start RLRV 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.