Token-Level Policy Optimization: Linking Group-Level Rewards to Token-Level Aggregation via Markov Likelihood

Anonymous ACL submission

Abstract

Group Relative Policy Optimization (GRPO) has significantly advanced the reasoning ability of large language models (LLMs), particularly by boosting their mathematical performance. However, GRPO and related entropy-regularization methods still face challenges rooted in the sparse token rewards inherent to chain-of-thought (CoT). Current approaches often rely on undifferentiated token-level entropy adjustments, which frequently lead to entropy collapse or model collapse. In this work, we propose `TEPO`, a novel token-level framework that incorporates Markov Likelihood (sequence likelihood) links group-level rewards with tokens via token-level aggregation. Experiments show that `TEPO` consistently outperforms existing baselines across key metrics (including @k and accuracy). It not only sets a new state of the art on mathematical reasoning tasks but also significantly enhances training stability.

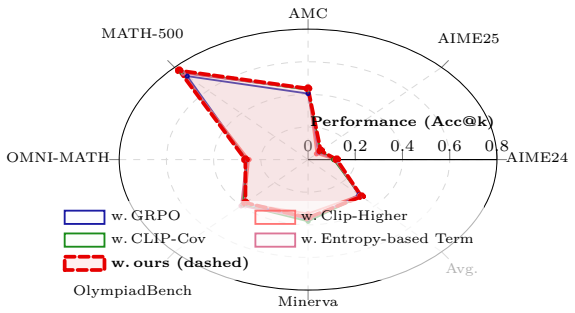


Figure 1: **Performance on math reasoning benchmarks.** Our method, shown as a dashed red line, stands out and remains best or near-best on most tasks, especially on MATH-500, and it also achieves a strong overall performance on average.

1 Introduction

LLMs have significantly advanced mathematical reasoning capabilities by leveraging state-of-the-art reinforcement learning (RL) techniques. A pivotal advancement in this domain is GRPO, which not only enhances mathematical reasoning performance, but also addresses key limitations of prior methods like Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022), such as memory efficiency, sample utilization, and training stability (Shao et al., 2024). These advantages render GRPO indispensable for the extensive application of LLMs in mathematical reasoning tasks.

Despite its advantages, applying GRPO within reinforcement learning (RL) faces challenges in balancing exploration and exploitation (the E–E trade-off). (Sutton, 1988). This challenge has driven a prevailing paradigm: existing methods either minimize entropy to yield convergent, credible outputs (Gao et al., 2025; Agarwal et al., 2025), or maximize it to boost exploration—such as DAPO's decoupled clipping (Yu et al., 2025). Yet such entropy-maximizing or -minimizing approaches are inherently fragile: on the one hand, these blunt approaches to entropy manipulation lack stability; on the other hand, their inherent flaws without laborious parameter tuning, further manifest as policy entropy collapse (Yu et al., 2025) and model collapse (Chu et al., 2025; Gao et al., 2025; Zheng et al., 2025). Specifically, a well-established relationship between model performance (R) and policy entropy (H) is defined as $R = -a \cdot \exp(H) + b$ (Cui et al., 2025a); this formulation reveals an inherent dilemma: improved performance comes at the cost of reduced entropy (Cheng et al., 2025). To address this inherent trade-off, KL-divergence has become a prevalent choice. However, KL constraints alone can trigger model collapse in CoT reasoning (Chu et al., 2025; Gao et al., 2025; Zheng et al., 2025), further

hindering stable policy learning. Moreover, other existing KL-based solutions—including ProRL's KL regularization (Liu et al., 2025) and newer Clip-Cov/KL-Cov methods (Cheng et al., 2025)—rely on intricate hyperparameters, which often require fine-tuning dynamicsly tailored to token-level.

Core Insight: We argue that these challenges, stemming from high-variance noise accumulation over extended reasoning sequences, are particularly acute in GRPO due to its critic-free nature (Zhang et al., 2025). When discovering a novel CoT structures, policies may undergo significant divergence from their initial distributions (Cheng et al., 2025). These will incur the accumulation of high-variance noise throughout long CoT sequences, a phenomenon that is further exacerbated by GRPO (Zheng et al., 2025). Notably, entropy regularization alone may cause model collapse (Chu et al., 2025; Gao et al., 2025; Zheng et al., 2025), further destabilizing policy optimization. These issues are worsened in GRPO by its critic-free design (Zhang et al., 2025), which lets high-variance noise accumulate in long reasoning sequences.

To address these issues, we propose `TEPO`, a novel framework that leverages Markov Likelihood (Chung, 1967) to adapt entropy control and connect group-level reward with token-level optimization. Our key contributions consist of three main components:

- **Markov Likelihood and Token Mean Optimization**: We use a Markov model to link group-level reward with tokens via token-level aggregation.
- **Investigating Entropy Regularization in Critic-Free Paradigms**: We provide theoretical and empirical evidence that entropy regularization is not a good fit for the critic-free GRPO setting.
- **Extensive Experiments**: Our method improves average accuracy by 2% over baseline GRPO. We also run ablations on entropy regularization and sparse rewards, and the results support the effectiveness of `TEPO`.

Our experiments show that `TEPO` significantly outperforms existing methods on key performance metrics such as @k and accuracy, setting new records on mathematical reasoning tasks while maintaining superior training stability.

2 Preliminaries

This section lays the theoretical foundation for our method: we derive the policy-entropy gradient to motivate our regularizer, review policy-gradient basics and limits in GRPO.

2.1 Policy Entropy Formulation

In LLMs, the state s corresponds to the prompt context, and an action a refers to a token from the vocabulary $\mathcal{A}$. For state is and action a, the policy is defined as follows:

$$\pi_\theta(a \mid s) = \frac{\exp(\phi_\theta(s,a))}{\sum_{a' \in \mathcal{A}} \exp(\phi_\theta(s,a'))},$$

which is a softmax function. Here, $\phi_\theta(s,a) \in \mathbb{R}$ is the score (the token logit), and $\theta = \{\phi_\theta(s,a) \mid s \in \mathcal{S}, a \in \mathcal{A}\}$ collects all scores. The entropy of policy distribution $\pi_\theta(\cdot \mid s)$ measures its uncertainty:

$$\mathcal{H}\left(\pi_\theta(\cdot \mid s)\right) = -\sum_a \pi_\theta(a \mid s) \log \pi_\theta(a \mid s).$$

Applying the chain rule yields:

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial \phi_\theta(s,a_i)} = -\sum_a \Bigg[& \frac{\partial \pi_\theta(a \mid s)}{\partial \phi_\theta(s,a_i)} \log \pi_\theta(a \mid s) \\ & + \pi_\theta(a \mid s) \frac{\partial \log \pi_\theta(a \mid s)}{\partial \phi_\theta(s,a_i)} \Bigg].\end{aligned}$$

The partial derivatives are given by:

$$\frac{\partial \pi_\theta(a \mid s)}{\partial \phi_\theta(s,a_i)} = \begin{cases} \pi_\theta(a \mid s)\left(1 - \pi_\theta(a \mid s)\right), & a = a_i \\ -\pi_\theta(a \mid s)\pi_\theta(a_i \mid s), & a \neq a_i \end{cases},$$

$$\frac{\partial \log \pi_\theta(a \mid s)}{\partial \phi_\theta(s,a_i)} = \begin{cases} 1 - \pi_\theta(a_i \mid s), & a = a_i \\ -\pi_\theta(a_i \mid s), & a \neq a_i \end{cases}.$$

Substituting these derivatives yields the following compact entropy-gradient expression:

$$\frac{\partial \mathcal{H}}{\partial \phi_\theta(s,a_i)} = \pi_\theta(a_i \mid s)\left(\log \pi_\theta(a_i \mid s) + \mathcal{H}\left(\pi_\theta(\cdot \mid s)\right)\right). \tag{1}$$

2.2 Policy Gradient for LLMs Alignment

We study RL fine-tuning for language models on verifiable tasks (RLVR) (Lambert et al., 2024), which aims to maximize a rule-based reward $A(\boldsymbol{y}) = r(y)$ (Williams, 1992) :

$$\max_\theta J(\theta) := \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}, \boldsymbol{y} \sim \pi_\theta(\boldsymbol{x})}\left[A(\boldsymbol{y})\right], \tag{2}$$

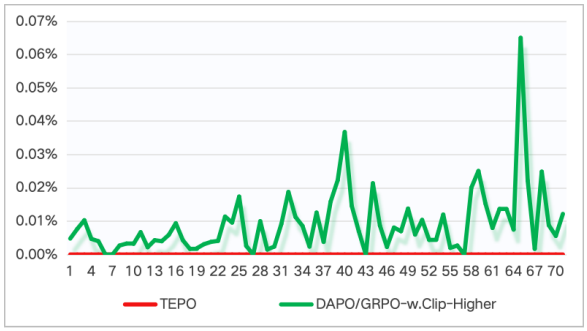


Figure 2: Clip Ratio over Steps

where $\boldsymbol{x} \sim \mathcal{D}$ is the input prompt, and $\boldsymbol{y} = \{y_1, \ldots, y_T\}$ is the generated sequence of length T. Proximal Policy Optimization (PPO) (Schulman et al., 2017) improves training stability by using a clipped important sampling:

$$L(\theta) = \mathbb{E}_t \Bigg[\min \left(\frac{\pi_\theta(y_t \mid \boldsymbol{y}_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid \boldsymbol{y}_{<t})} A_t, \right. \\ \left. \text{clip} \left(\frac{\pi_\theta(y_t \mid \boldsymbol{y}_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid \boldsymbol{y}_{<t})}, 1-\epsilon, 1+\epsilon \right) A_t \right) \Bigg], \tag{3}$$

with ϵ controls the clipping range. The policy gradient components are derived as:

$$\frac{\partial J}{\partial \phi_\theta(s, a_i)} = \mathbb{E}_{\pi_\theta} \left[\frac{\partial \log \pi_\theta(a \mid s)}{\partial \phi_\theta(s, a_i)} A(s, a) \right].$$

Plugging in the softmax gradient gives the basic policy gradient:

$$\frac{\partial J}{\partial \phi_\theta(s, a_i)} = \pi_\theta(a_i \mid s) \left(A(s, a_i) - \mathbb{E}_{\pi_\theta} \left[A(s, a) \right] \right), \tag{4}$$

where $\mathbb{E}_{\pi_\theta} \left[A(s, a) \right] = \sum_a \pi_\theta(a \mid s) A(s, a)$.

2.3 Limitations of Current Approaches

Methods like RLOO (Kool et al., 2019) and GRPO (Shao et al., 2024) apply group-level normalization to cut variance while keeping training stable. We sample K responses per prompt and compute the advantage as follows:

$$A_t = \frac{r(\boldsymbol{y}) - \text{mean}\left(r\left(\boldsymbol{y}^{1:K} \right) \right)}{\text{std}\left(r\left(\boldsymbol{y}^{1:K} \right) \right)}. \tag{5}$$

However, these methods still struggle to balance exploration and exploitation, because group-level advantages are not suitable for sentence-level and token-level scenarios.

3 Methodology

This section presents our proposed method. We first analyze the limitations of GRPO in entropy utilization, then derive our theoretical framework, and finally present the formal objective function and algorithm.

3.1 Limitations of Entropy Regularization in GRPO

3.1.1 Policy Ascent Formulation

At step k, starting from the current policy π_k, we get the next policy π_{k+1} by solving the following constrained problem for each state s:

$$\max_p \ \mathbb{E}_{a \sim p} \left[A(s, a) \right] - \eta \cdot \text{KL} \left(p \| \pi_k(\cdot \mid s) \right)$$

subject to $\sum_a p(a \mid s) = 1$. Using a Lagrangian view, these policy update methods lead to an iterative update rule written as:

$$\pi_{k+1}(a \mid s) = \frac{\pi_k(a \mid s) \exp\left(\eta A^\pi(s, a) \right)}{\mathbb{E}_{a' \sim \pi_k(\cdot \mid s)} \left[\exp\left(\eta A^\pi(s, a') \right) \right]}. \tag{6}$$

More concretely, the normalized update is:

$$\pi_{k+1}(a \mid s) \propto \pi_k(a \mid s) \exp\left(\frac{1}{\eta} A^\pi(s, a) \right).$$

The update keeps the stability benefits of KL-regularized policy optimization while maximizing expected return:

$$\theta_{s,a}^{k+1} = \theta_{s,a}^k + \frac{1}{\eta} A^\pi(s, a). \tag{7}$$

This η stabilizes the training process while counteracting policy ascent(Williams, 1992; Kakade, 2001; Schulman et al., 2015, 2017; Guo et al., 2025). Many studies show that strong policy learning needs sufficient exploration. When trying new CoT patterns, the policy can drift far from its starting distribution. In fact, using only a KL constraint can cause model collapse (Chu et al., 2025; Gao et al., 2025; Zheng et al., 2025), making stable training harder.

3.1.2 Entropy-Policy Gradient Misalignment

Lemma 3.1. *For a softmax policy $\pi_\theta(a \mid s) \propto \exp(\phi_\theta(s, a))$:*

- *When $A(s, a) < 0$ (suboptimal actions): $\langle \nabla_{\phi_\theta} \mathcal{H}, \nabla_{\phi_\theta} J \rangle > 0$*
- *When $A(s, a) > 0$ (optimal actions): $\langle \nabla_{\phi_\theta} \mathcal{H}, \nabla_{\phi_\theta} J \rangle < 0$*

Proof. We analyze the inner product between entropy gradient and policy gradient:

$$\langle \nabla_{\phi_\theta} \mathcal{H}, \nabla_{\phi_\theta} J \rangle = \sum_{a_i} \frac{\partial \mathcal{H}}{\partial \phi_\theta(s, a_i)} \cdot \frac{\partial J}{\partial \phi_\theta(s, a_i)}.$$

Plugging in the entropy gradient (Eq. 1) and the policy gradient (Eq. 4) gives:

$$\begin{aligned}&\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle \\ &= \sum_{a_i} \pi_\theta(a_i|s)^2 \left(\log \pi_\theta(a_i|s) + \mathcal{H}(\pi_\theta(\cdot|s))\right) A(s, a_i).\end{aligned}$$

Case 1: Suboptimal Actions. If the advantage is negative ($A(s,a) < 0$):

- Small probability $\pi_\theta(a \mid s) \to 0$ implies $\log \pi_\theta(a \mid s) \to -\infty$
- The term $\log \pi_\theta(a \mid s) + \mathcal{H}(\pi_\theta(\cdot \mid s))$ becomes negative
- With $\pi_\theta(a \mid s)^2 > 0$, $A(s,a) < 0$, and the logarithmic term negative, each product term is positive

Thus: $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle > 0$.

Case 2: Optimal Actions. If the advantage is positive ($A(s,a) > 0$):

- Large probability $\pi_\theta(a \mid s) \to 1^-$ implies $\log \pi_\theta(a \mid s) \to 0^-$
- The term $\log \pi_\theta(a \mid s) + \mathcal{H}(\pi_\theta(\cdot \mid s))$ remains negative
- With $\pi_\theta(a \mid s)^2 > 0$, $A(s,a) > 0$, and the logarithmic term negative, each product term is negative

Thus: $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle < 0$.

3.1.3 Entropy-Natural Policy Gradient Misalignment

Lemma 3.2. *The entropy difference $\Delta\mathcal{H}$ correlates with the gradient inner product:*

- *When $A(s,a) < 0$(suboptimal actions), $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle > 0$ and $\Delta\mathcal{H} > 0$ (entropy increases).*
- *When $A(s,a) > 0$ (optimal actions), $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle < 0$ and $\Delta\mathcal{H} < 0$ (entropy decreases).*

Proof. The policy parameter update follows:

$$\phi_\theta(s,a) \leftarrow \phi_\theta(s,a) + \alpha \cdot \frac{\partial J}{\partial \phi_\theta(s,a)},$$

where $\alpha > 0$ is the learning rate.

Using first-order Taylor expansion (Nocedal and Wright, 2006), the entropy change approximates as follows:

$$\Delta\mathcal{H} \approx \sum_a \frac{\partial \mathcal{H}}{\partial \phi_\theta(s,a)} \cdot \Delta\phi_\theta(s,a).$$

Plugging in the update rule $\Delta\phi_\theta(s,a) = \alpha \cdot \frac{\partial J}{\partial \phi_\theta(s,a)}$ gives:

$$\Delta\mathcal{H} \approx \alpha \cdot \langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle.$$

When $A(s,a) < 0$(suboptimal actions), $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle > 0$ and $\Delta\mathcal{H} > 0$.

When $A(s,a) > 0$ (optimal actions), $\langle \nabla_{\phi_\theta}\mathcal{H}, \nabla_{\phi_\theta} J\rangle < 0$ and $\Delta\mathcal{H} < 0$.

This shows that, in no-critic setups, entropy regularization conflicts with policy ascent as Eq. 4. In no-critic setups, adding entropy control can hinder the improvement of final task performance. For example, maximizing entropy requires extensive hyperparameter tuning, and KL-Divergence can cause model collapse (Zheng et al., 2025).

3.2 Theoretical Analysis of TEPO

3.2.1 E-E Trade-off Analysis

Policy Gradient (PG) (Williams, 1992) and Natural Policy Gradient (NPG) (Kakade, 2001) keep updates stable and consistent with the KL-Divergence (Schulman et al., 2015):

$$\begin{aligned}&\mathcal{H}(\pi_\theta^{k+1} \mid s) - \mathcal{H}(\pi_\theta^{k} \mid s) \\ &\approx -\frac{1}{\eta} \cdot \mathrm{Cov}_{a\sim\pi_\theta^k(\cdot|s)}\left(\log \pi_\theta^k(a \mid s), r(s,a)\right),\end{aligned} \tag{8}$$

where the nonnegative covariance term Cov tracks the policy's step-by-step change.

This links small entropy changes to the policy's covariance structure across updates and characterizes how entropy changes relate to the exploitation policy. **This relationship thus uncovers the essence of exploration and exploitation: balancing the two sides of the equation yields their optimal trade-off.**

In GRPO (Shao et al., 2024), we study how policy entropy changes from iteration k to k+1. Given the state distribution (d^{π_k}) induced by π_k, this entropy change decomposes as:

$$\begin{aligned}&\mathcal{H}(\pi_{k+1}) - \mathcal{H}(\pi_k) \\ &= \underbrace{\mathbb{E}_{s\sim d^{k+1}}\mathcal{H}(\pi_{k+1}(\cdot \mid s)) - \mathbb{E}_{s\sim d^{k}}\mathcal{H}(\pi_{k+1}(\cdot \mid s))}_{\text{State Distribution Shift}} \\ &\quad + \underbrace{\mathbb{E}_{s\sim d^{k}}\mathcal{H}(\pi_{k+1}(\cdot \mid s)) - \mathbb{E}_{s\sim d^{k}}\mathcal{H}(\pi_{k}(\cdot \mid s))}_{\text{Policy Update Effect}} \\ &\approx \mathbb{E}_{s\sim d^k}\left[\mathcal{H}(\pi_{k+1}(\cdot \mid s)) - \mathcal{H}(\pi_k(\cdot \mid s))\right],\end{aligned} \tag{9}$$

where the approximation Eq. 8 holds under the assumption $d^{\pi_{k+1}} \approx d^{\pi_k}$.

However, the required near-stationarity ($d^{\pi_{k+1}} \approx d^{\pi_k}$) is hard to achieve in CoT reasoning. GRPO uses a no-critic setup. This makes the problem worse. Policy updates can change the policy greatly. This leads to large distribution shifts. And it breaks the approximation.

Critic-free GRPO is ill-suited for entropy regularization. Mainly because KL is always non-negative, so it fails to capture small, useful reward differences when signals exist only in a few tokens. Increasing entropy leads to more wrong trials and long-term degradation, while decreasing it rapidly erodes diversity and crashes the model.

3.2.2 Markov Likelihood Resolves the E-E Trade-Off

In GRPO, the token-level importance weights $\frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}$ do not fully reflect how the whole sequence distribution changes. This leads to very noisy (high-variance) gradient estimates and can often cause model collapse that is hard to undo (Zheng et al., 2025; Chen et al., 2025; Chu et al., 2025). The main reason is that GRPO has no critic (value baseline) and uses only sparse token-level rewards.

LLMs generate text one token at a time: each token's probability depends on prior tokens (a Markov factorization) (Chung, 1967). Thus, the sequence-level (Markov Likelihood) importance ratio $\mathcal{IS}_i(\theta)$ factors into a product of token ratios:

$$\begin{aligned}\mathcal{IS}_i(\theta) &= \left(\frac{\pi_\theta(y_i \mid x)}{\pi_{\theta_{\text{old}}}(y_i \mid x)}\right)^{\frac{1}{|y_i|}} \\ &= \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}\right),\end{aligned}$$

which represents a geometric mean.

3.3 Proposed Method

3.3.1 Objective Function

Putting the update rule (Eq. 7) and the entropy–covariance link (Eq. 8) into the PPO loss gives the following joint objective:

$$\begin{aligned}L(\theta) = \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Bigg(&\mathcal{IS}_i(\theta)\hat{A}_{i,t}, \\ &\text{clip}\left(\mathcal{IS}_i(\theta),\ 1-\varepsilon,\ 1+\varepsilon\right)\hat{A}_{i,t}\Bigg), \\ \text{s.t. } 0 < |\{o_i \mid \text{is_equivalent}(a, o_i)\}| < G,\end{aligned} \tag{10}$$

where the $\mathcal{IS}_i(\theta)$ operates in the backward pass as: $\frac{\partial L}{\partial \pi_\theta} = \frac{\partial L}{\partial \mathcal{IS}_i} \cdot \mathcal{IS}_i \cdot \frac{\text{mask}_{i,t}}{|o_i| \cdot \pi_\theta}$.

3.3.2 Computation Graph for the Token Level

We use a careful backward pass to keep training stable and consistent with our theory, mainly by handling importance-sampling ratios at both the sequence and token levels.

Algorithm 1 Simplified Backward Iteration for Policy Gradient

Require: Loss function: $L(\theta) = \frac{1}{\text{total_mask}}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\mathcal{IS}_i(\theta)\hat{A}_{i,t} \cdot \text{mask}_{i,t}$
Ensure: Policy parameter gradient $\nabla_\theta L(\theta)$
1: Compute gradient of L w.r.t. total loss sum ($\text{sum_loss} = \sum_{i,t}\mathcal{IS}_i\hat{A}_{i,t} \cdot \text{mask}_{i,t}$):
2: $\frac{\partial L}{\partial \text{sum_loss}} = \frac{1}{\text{total_mask}}$
3: Propagate gradient to token-level term $\mathcal{IS}_i\hat{A}_{i,t}$:
4: $\frac{\partial L}{\partial(\mathcal{IS}_i\hat{A}_{i,t})} = \frac{\text{mask}_{i,t}}{\text{total_mask}}$
5: Calculate gradient of importance sampling ratio $\mathcal{IS}_i$:
6: $\frac{\partial L}{\partial \mathcal{IS}_i} = \sum_{t=1}^{|o_i|}\hat{A}_{i,t} \cdot \frac{\text{mask}_{i,t}}{\text{total_mask}}$
7: Backpropagate to current policy probability π_θ:
8: $\frac{\partial L}{\partial \pi_\theta} = \frac{\partial L}{\partial \mathcal{IS}_i} \cdot \mathcal{IS}_i \cdot \frac{\text{mask}_{i,t}}{|o_i| \cdot \pi_\theta}$
9: Obtain $\nabla_\theta L(\theta)$ by backpropagating through the policy network.

The algorithm follows the gradient in Eq. 10. Here, $\frac{\partial L}{\partial \mathcal{IS}_i} \cdot \mathcal{IS}_i$ gives a batch-level signal for each sequence, while $\frac{\text{mask}_{i,t}}{|o_i| \cdot \pi_\theta}$ gives a sequence-normalized signal for each token. Using the geometric mean in $\mathcal{IS}_i(\theta)$ keeps gradients stable across different sequence lengths and reduces the variance spikes noted in Sec. 3.

3.3.3 TEPO with Empirical Proof

Our method addresses these limitations through the following steps:

- Using a Markov Likelihood to connect group-level reward with tokens, which reduces gradient bias (see Fig. 2).
- Adopting a token-leveln calculation to enable token-level policy optimization.
- Maintaining training stability while preserving the ability to explore (see Fig. 3).

These results match the known link between performance and policy entropy: ($R = -a \cdot \exp(H) + b$) (Cui et al., 2025a). As it notes, optimizing downstream tasks tends to lower entropy; pushing entropy higher rarely helps,

and adding a KL term often hurts. The reason is that inCoT, token distributions shift across steps, so entropy and KL affect tokens unevenly and lead to collapse.

4 Experiment

4.1 Experimental Setup

Implementation Details. We ran all experiments on eight NVIDIA A100 GPUs. For each rollout step, we processed 64 prompts per batch and sampled 8 responses per prompt with temperature 1.0. The policy was updated 8 times using these responses. To keep training effective, we removed prompts whose sampled responses were all correct or all wrong, following (Yu et al., 2025). Key hyperparameters were: learning rate $lr = 5 \times 10^{-7}$, maximum prompt length $max_prompt_length = 2024$, maximum response length $max_response_length = 8192$, and training prompt mini-batch size $train_prompt_mini_bsz = 16$.

Datasets. We trained Qwen2.5 models (Yang et al., 2025) on DAPO-MATH (Yu et al., 2025) and evaluated on seven math benchmarks: MATH-500, AIME24/25 (Li et al., 2024), AMC, OMNI-MATH, OlympiadBench, and Minerva (Lewkowycz et al., 2022). For AIME and AMC we used temperature 0.6; for the others we used greedy decoding. The maximum generation length was 8192 tokens. AIME, AIME25, and AMC results were reported with 32 samples per problem (@32) following prior work (Guo et al., 2025; DeepSeek-AI et al., 2025; Shao et al., 2024).

Baseline Methods. We compared `TEPO` with strong baselines. Original GRPO/DAPO (Shao et al., 2024) used the standard formulation without extra changes. GRPO/DAPO with Clip-Higher applied an upper threshold $\epsilon = 0.28$ in the PPO loss (Yu et al., 2025). Clip-Cov (Cui et al., 2025a) clipped tokens with high covariance (clip ratio $r = 2 \times 10^{-4}$). An entropy-based term (Cheng et al., 2025) added entropy regularization to advantage estimation with scale $\alpha = 4 \times 10^{-4}$ and clipping parameter $\kappa = 2$. For `TEPO`, we set $\epsilon = 0.2$.

4.2 Main Results

Table 1 reports results on seven math benchmarks. `TEPO` attains the best average accuracy at 32.04%, beating all baselines. Compared with GRPO, the average gain is 1.74 points (32.04% vs. 30.30%). The largest gain appears on MATH-500: `TEPO` reaches 77.20%, a +4.8 point jump over GRPO. This suggests `TEPO` handles sparse rewards better on hard math tasks. On AIME, it stays competitive, scoring 12.18% on AIME24 and 7.60% on AIME25. Our method performs relatively poorly on the Minerva and OlympiadBench datasets. The core reason is the strict output format rules of these two datasets, especially the LaTeX requirement. Therefore, Minerva and OlympiadBench affect the performance of methods that tend to explore, and the same is true for the DAPO/GRPO w.Clip-Higher method.

Figure 3 presents the training curves. In Fig. 3a, `TEPO` achieves steadier and higher rewards over steps, demonstrating more effective learning. In Fig. 3b, `TEPO` 's gradient norms are consistently higher, reflecting more active parameter updates. These trends validate our assertion that token-level explicit policy optimization delivers finer-grained learning signals.

4.3 Ablation Studies

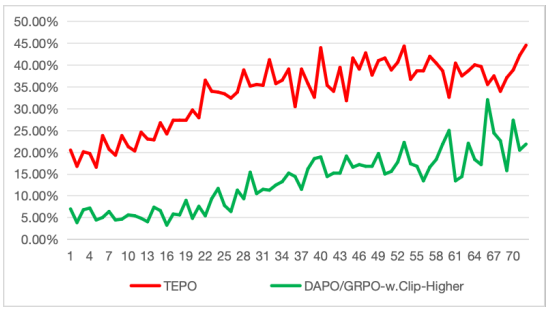
4.3.1 Analysis of Entropy Regularization Components

We ran ablations to study the entropy parts of our method. Table 2 shows the numbers, and Figure 4 shows the training curves. Both maximum-entropy and KL-based regularization hurt performance in our GRPO setup. The maximum-entropy variant lowers the average accuracy by about 0.39 points (31.65% vs. 32.04%). The KL variant is worse, dropping accuracy by 5.79 points (26.25% vs. 32.04%).

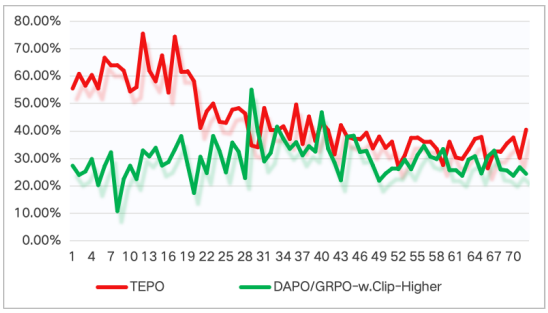
We think this happens because these regularizers work best when rewards are dense, but math reasoning gives sparse feedback. In GRPO, only a small set of token actions get useful signals. Adding entropy terms spreads probability mass and weakens those scarce signals. The stronger drop with KL suggests that forcing the policy to stay close to the initial model makes it hard to move toward the few actions that earn reward, so the model adapts less to the sparse feedback.

Table 1: Performance Comparison of Qwen2.5-7B on Mathematical Reasoning Benchmarks (Accuracy Percentage %). Best results are in **bold**, second best are underlined.

Method	AIME24	AIME25	AMC	MATH-500	OMNI-MATH	OlympiadBench	Minerva	Avg.
Qwen2.5-7B	0.94	0.94	14.34	43.20	13.73	16.74	21.69	13.30
w. GRPO	11.56	6.25	40.47	72.40	26.00	37.77	36.76	30.30
w. Clip-Higher	11.25	5.00	42.80	76.60	25.00	37.92	33.08	30.85
w. CLIP-Cov	10.83	7.40	42.84	75.60	26.11	39.40	**37.86**	31.63
w. Entropy-based Term	11.35	6.15	43.18	74.80	26.14	**40.14**	36.02	31.62
TEPO	**12.18**	**7.60**	**43.56**	**77.20**	**26.57**	37.48	35.66	**32.04**



(a) Reward progression across training steps: TEPO shows more stable and consistently higher rewards than GRPO, which indicates better optimization.



(b) Gradient Norm over training steps: TEPO maintains sustain higher gradient norms, suggesting more active policy updates and deeper reasoning.

Figure 3: Comparative analysis of TEPO versus DAPO/GRPO w. Clip-Higher training dynamics. The left panel shows reward progression, while the right panel displays gradient norm throughout training.

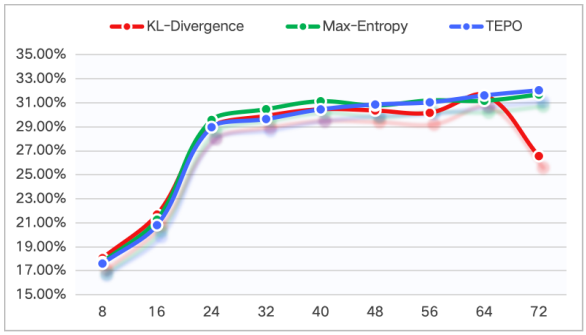


Figure 4: Training dynamics with different entropy regularization strategies. The complete TEPO demonstrates superior stability and performance compared to variants with maximum entropy or KL-divergence regularization.

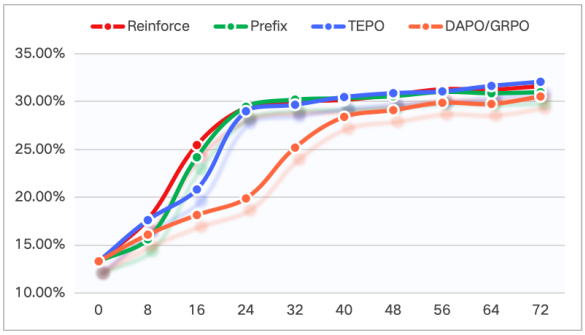


Figure 5: Comparison of importance sampling strategies in sparse reward environments. Our approach demonstrates more effective utilization of sparse learning signals compared to REINFORCE and prefix-based importance sampling.

4.3.2 Analysis of Sparse Reward in GRPO

We further evaluated different importance sampling designs for sparse rewards. Table 2 presents the scores, and Figure 5 illustrates performance across steps.

The REINFORCE importance sampling employs $sg[\mathcal{IS}_i(\theta)] \cdot \log \pi$, implementing an online single-step policy similar to bandit sentence sampling. While this approach shows some improvement over basic GRPO, it still underperforms our method. The prefix importance sampling strategy computes:

$$\mathcal{IS}_{i,t}(\theta) = \left(\frac{\pi_\theta(y_{i,j\leq t} \mid x)}{\pi_{\theta_{\text{old}}}(y_{i,j\leq t} \mid x)} \right)^{\frac{1}{|y_{i,j\leq t}|}},$$

where $y_{i,j\leq t}$ represents the prefix of the i-th sentence up to the t-th token.

Our method is best among the three. It is +1.09 points over prefix importance sampling (32.04% vs. 30.95%) and +0.46 points over REINFORCE-style sampling (32.04% vs.

Table 2: Ablation on entropy regularization and importance sampling (Accuracy, %). Higher is better.

Method	AIME24	AIME25	AMC	MATH-500	OMNI-MATH	OlympiadBench	Minerva	Avg.
Panel A: Entropy Regularization								
w. Max-Entropy	13.02	5.94	42.80	76.60	26.85	38.37	37.13	31.65
w. KL-Divergence	5.52	3.96	42.84	75.60	22.41	32.74	30.88	26.25
TEPO	**12.18**	**7.60**	**43.56**	**77.20**	**26.57**	**37.48**	**35.66**	**32.04**
Panel B: Importance Sampling								
w.GRPO	11.56	6.25	40.47	72.40	26.00	37.77	36.76	30.30
w. Reinforce	11.04	6.77	42.80	74.80	26.96	38.51	32.72	31.58
w. Prefix IS	11.08	6.67	42.44	73.40	25.11	39.11	35.66	30.95
TEPO	**12.18**	**7.60**	**43.56**	**77.20**	**26.57**	**37.48**	**35.66**	**32.04**

31.58%). Relative to plain GRPO, it improves the average by 1.74 points.

Prefix importance sampling shows the critic-free GRPO is a sparse rewards method. Importance sampling in REINFORCE reveals bias in sentence-level rewards. Ours balances group-level credit with token-level updates via a sentence likelihood, which reduces variance and performs better under sparse rewards.

5 Related work

5.1 Exploration-Exploitation Trade-off in Traditional RL

Balancing exploration and exploitation is a key problem in RL (Sutton et al., 1998). Under the maximum-entropy view, an entropy term gives a clear way to handle this trade-off (Ziebart et al., 2008; Toussaint, 2009; Chao et al., 2024). This idea shows up in common algorithms: DQN usually relies on ϵ-greedy exploration instead of an explicit entropy term (Mnih et al., 2015), PPO often adds an entropy bonus to keep exploration active (Schulman et al., 2017), and SAC directly optimizes a maximum-entropy objective (Haarnoja et al., 2017, 2018). In RL with LLMs, the role of entropy is still unclear. RLHF training typically uses a KL penalty to a reference policy (Ouyang et al., 2022; Hu et al., 2024). GRPO-style objectives and recent studies often find little or unclear benefit from standard entropy bonuses, so the effect on generation quality and training stability remains an open question (Shao et al., 2024; DeepSeek-AI et al., 2025; He et al., 2025; Cui et al., 2025b; Shen, 2025; Zhang et al., 2024; Rafailov et al., 2023; Hong et al., 2024; Xiao et al., 2024).

5.2 RL Advancements in LLMs

RL is a key part of LLM post-training (Ouyang et al., 2022; Grattafiori et al., 2024), and combining it with verifiable rewards further improves reasoning (Jaech et al., 2024; Guo et al., 2025). Recent work shows three clear trends. First, simple policy-gradient methods still work well: GPG keeps training straightforward by using REINFORCE (Chu et al., 2025), and CISPO improves efficiency by clipping importance weights and detaching them from the update; both use a standard policy-gradient loss (Chen et al., 2025). Second, GSPO shifts to sequence-level learning: it defines importance ratios with whole-sequence likelihoods and applies clipping, rewards, and updates at the sequence level (Zheng et al., 2025). Third, entropy matters: (Cui et al., 2025a) find performance (R) fits $R = -a\exp(\mathcal{H}) + b$, meaning lower entropy usually gives better results. This matches RLVR findings (Jaech et al., 2024; Guo et al., 2025; Lambert et al., 2024) and aligns with LLM scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022; Gao et al., 2023).

6 Conclusion

GRPO has improved LLM mathematical reasoning but shows fragile convergence in CoT: uniform token-level entropy often triggers collapse or early stop. We introduce TEPO, which links group loss to sentence representations via a Markov likelihood and then to tokens through token-mean aggregation, stabilizing updates. On math benchmarks, TEPO surpasses baselines (e.g., Acc@k), sets a new state of the art, and markedly improves training stability.

Limitations

Although our method proposes a novel token-level framework that incorporates Markov Likelihood to link group-level rewards with tokens via token-mean aggregation, it only connects tokens to group-level rewards. It leaves token optimization to backward and fails to further distinguish how different tokens affect performance. Future research should focus on exploring tokens' roles in sentences and how to link with group-level reward via a general paradigm.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Chen-Hao Chao, Chien Feng, Wei-Fang Sun, Cheng-Kuang Lee, Simon See, and Chun-Yi Lee. 2024. Maximum entropy reinforcement learning via energy-based normalizing flow. In *Advances in Neural Information Processing Systems*.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, and 1 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Kai Lai Chung. 1967. Markov chains. *Springer-Verlag, New York*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025a. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025b. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

DeepSeek-AI, Daya Guo, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Zitian Gao, Lynx Chen, Joey Zhou, and Bryan Dai. 2025. One-shot entropy minimization. *arXiv preprint arXiv:2505.20282*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Sham M Kakade. 2001. A natural policy gradient. *Advances in neural information processing systems*, 14.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 reinforce samples, get a baseline for free!

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and 1 others. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

J Nocedal and S Wright. 2006. Numerical optimization. 2nd edn springer. *New York*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

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

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

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

Han Shen. 2025. On entropy control in llm-rl algorithms. *arXiv preprint arXiv:2509.03493*.

Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44.

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Marc Toussaint. 2009. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, pages 1049–1056.

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

Wenyi Xiao, Zechuan Wang, Leilei Gan, Shuai Zhao, Zongrui Li, Ruirui Lei, Wanggui He, Luu Anh Tuan, Long Chen, Hao Jiang, Zhou Zhao, and Fei Wu. 2024. A comprehensive survey of direct preference optimization: Datasets, theories, variants, and applications. *arXiv preprint arXiv:2410.15595*.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, and 1 others. 2025. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale, 2025.

Hanning Zhang, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. 2024. Entropy-regularized process reward model. *arXiv preprint arXiv:2412.11006*.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, and 1 others. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA.

A Policy Optimization with KL-Divergence

The objective of GRPO or RL in LLMs is to identify a new policy, π_θ, which is as closely aligned as possible with the ideal target distribution π^*(Williams, 1992; Chu et al., 2025; Schulman et al., 2017, 2015; Kakade, 2001; Shao et al., 2024). This task is equivalent to minimizing the KL- divergence, expressed as:

$$\min_\theta \mathrm{KL}(\pi^*(a|s)\|\pi_\theta(a|s)) \tag{11}$$

Substitute Eq. 6, we get :

$$\begin{aligned}
&\mathrm{KL}(\pi_\theta\|\pi^*) \\
&\propto \mathbb{E}_{a\sim\pi_\theta}\left[(\log\pi_\theta(a|s) - \log\pi_{\theta_{\text{old}}}(a|s)) - \frac{\hat{A}(a,s)}{\beta}\right] \\
&= \frac{1}{\beta}\mathbb{E}_{a\sim\pi_\theta}\left[\beta\mathrm{KL}(\pi_\theta\|\pi_{\theta_{\text{old}}}) - \hat{A}(a,s)\right]
\end{aligned}$$

Therefore, the PPO objective is essentially solving a distribution matching problem toward a target distribution shaped by the advantage function.

B Policy Entropy in LLMs

To extend the state-specific entropy difference to the global entropy change (i.e., entropy averaged over all states, which better reflects the overall exploration-exploitation balance of the policy), we first rely on a first-order Taylor expansion of entropy around the policy parameters θ^k. For any state s, the entropy of the updated policy $\mathcal{H}(\theta^{k+1} \mid s)$ can be approximated as: $\mathcal{H}(\theta^k \mid s) + \left\langle \nabla_\theta \mathcal{H}(\theta^k \mid s), \theta^{k+1} - \theta^k \right\rangle$, where $\nabla_\theta \mathcal{H}(\theta^k \mid s)$ is the gradient of entropy with respect to parameters θ^k, and $\langle \cdot, \cdot \rangle$ denotes the inner product. This approximation holds when the parameter update step $\theta^{k+1} - \theta^k$ is small (a common assumption in policy optimization to ensure stability), as it ignores higher-order terms in the entropy's parameter dependence. Additionally, we define $d^{\pi_{k+1}} := \langle \nabla_\theta \mathcal{H}(\theta^k \mid s), \theta^{k+1} - \theta^k \rangle$, which quantifies the "contribution" of the entropy gradient to the state distribution update.

The entropy difference relies on a critical

approximation:

$$
\begin{aligned}
&\mathcal{H}(\pi_{k+1}) - \mathcal{H}(\pi_k) \\
&= \mathbb{E}_{s\sim d^{\pi_{k+1}}}\left[\mathcal{H}(\pi_{k+1}(\cdot \mid s))\right] - \mathbb{E}_{s\sim d^{\pi_k}}\left[\mathcal{H}(\pi_k(\cdot \mid s))\right] \\
&\approx \mathbb{E}_{s\sim d^{\pi_k}}\left[\mathcal{H}(\pi_{k+1}(\cdot \mid s))\right] - \mathbb{E}_{s\sim d^{\pi_k}}\left[\mathcal{H}(\pi_k(\cdot \mid s))\right] \\
&= \mathbb{E}_{s\sim d^{\pi_k}}\left[\mathcal{H}(\pi_{k+1}(\cdot \mid s)) - \mathcal{H}(\pi_k(\cdot \mid s))\right] \\
&= -1/\eta \cdot \mathrm{Cov}_{a\sim\pi_\theta^k(\cdot|s)}\left(\log \pi_\theta^k(a \mid s), A(s,a)\right)
\end{aligned}
$$

Here, d^{π_t} (for $t = k, k+1$) denotes the state distribution of the actor network at iteration t—specifically, the distribution of states encountered by the policy during interaction with the environment (e.g., $d^{\pi_{k+1}}$ aligns with the updated policy π_{k+1}). This approximation is justified by three key considerations:

- The policy update step is small, inducing high structural similarity and distributional overlap between $d^{\pi_{k+1}}$ and d^{π_k}—a natural consequence of the step-wise policy update process (Schulman et al., 2015).

- Replacing $d^{\pi_{k+1}}$ with d^{π_k} (from the first to the second line) incurs negligible error under first-order approximation: the visitation probability difference $d^{\pi_{k+1}} - d^{\pi_k}$ is a second-order term relative to the policy vector error $\pi_{k+1} - \pi_k$ by Taylor Equation (Nocedal and Wright, 2006).

- Residual bias from distribution shifts is corrected via importance sampling (Schulman et al., 2017), preserving the approximation's unbiasedness (i.e., the approximation error has zero expectation).

C Covariance Term

To refine the covariance term (for scenarios like LLM sequence generation), we substitute two key components enhancing update stability and interpretability:

1. Normalized Advantage Function Unlike immediate reward $r(s,a)$, $A(s,a)$ quantifies action a's "relative value" under s (e.g., $A(s,a) = r(s,a) - V(s)$, with $V(s)$ as the state value function). This isolates a's incremental benefit over other actions, making the covariance term a better policy update guide. Normalized via K reward samples ($\boldsymbol{y}^{1:K}$ from π_k) like RLOO (Kool et al., 2019) and GRPO (Shao et al., 2024) as: $A(s,a) = \frac{r(\boldsymbol{y}) - \mathrm{mean}(r(\boldsymbol{y}^{1:K}))}{\mathrm{std}(r(\boldsymbol{y}^{1:K}))}$, this scales $A(s,a)$ to zero mean/unit variance, preventing large reward magnitudes from dominating updates and stabilizing optimization.

2. Token-Level Log-Probability Difference For sequence-generation tasks , we define the token-level log-probability difference to approximate the current policy's expected log-probability over the state distribution d^{π_k}:

$$
\begin{aligned}
&\mathbb{E}_{s\sim d^{\pi_k}}\left[\log \pi_\theta^k(a \mid s)\right] \\
&= \log \pi_\theta^k(a \mid s) - \mathbb{E}_{s\sim d^{\pi_k}}\left[\log \pi_\theta^k(a \mid s)\right]
\end{aligned}
$$

as the parameter update $\theta^{k+1} - \theta^k$ induces only linear changes in log-probabilities. The right-hand side matches the form of the Kullback-Leibler (KL) divergence between successive policies.

Substituting these components yields the step-wise covariance Cov:

$$
\begin{aligned}
\mathrm{Cov}(y_i) = &\left(A(y_i) - \frac{1}{N}\sum_{j=1}^{N} A(y_j)\right) \\
&\cdot \left(\log \pi_\theta(y_i) - \frac{1}{N}\sum_{j=1}^{N} \log \pi_\theta(y_j)\right)
\end{aligned}
$$

This links token-level preference changes and normalized relative values. Optimizing it balances reward maximization and entropy (exploration)—critical for LLM generation, where excessive entropy causes incoherence and insufficient entropy leads to rigidity.

D Use of LLMs

Large language models (LLMs), specifically GPT-5 and DeepSeek-R1, were used solely as a supplementary tool during the preparation of this work for tasks such as polishing the writing. The authors are solely responsible for the entire research conception, technical direction, scientific content, and interpretation of results. The LLMs were employed only to assist in the presentation and clarity of the manuscript.