Revisiting Entropy in Reinforcement Learning for Large Reasoning Models

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has emerged as a predominant approach for enhancing the reasoning capabilities of large language models (LLMs). However, the entropy of LLMs usually collapses during RLVR training, causing premature convergence to suboptimal local minima and hinder further performance improvement. Although various approaches have been proposed to mitigate entropy collapse, a systematic study of entropy in RLVR remains lacking. To address this gap, we conduct extensive experiments to investigate the entropy dynamics of LLMs trained with RLVR and analyze how model entropy relates to response diversity, calibration, and performance across various benchmarks. Our findings reveal that the diversity of training data plays a critical role in shaping the entropy of LLMs trained with RLVR. Moreover, we empirically demonstrate that tokens with positive advantages are the primary contributors to entropy collapse, and that model entropy can be effectively regulated by adjusting the relative loss weights of tokens with positive and negative advantages during training.

1 Introduction

Pioneered by OpenAI o1 (Jaech et al., 2024), DeepSeek-R1 (DeepSeek-AI et al., 2025), and Kimi k1.5 (DeepSeek-AI et al., 2025), reinforcement learning with verifiable rewards (RLVR) has been widely employed to extend the boundaries of reasoning capacity in large language models (LLMs). LLMs trained with RLVR have demonstrated remarkable performance on tasks requiring sophisticated reasoning processes, such as mathematics and coding (Ye et al., 2025; Liu et al., 2025b; Luo et al., 2025; He et al., 2025).

However, although RLVR enhances the reasoning ability of LLMs, a growing body of research has shown that it can also drive LLMs toward entropy collapse, wherein the entropy of the model decreases substantially during training, ultimately reaching a markedly low level (Yu et al., 2025; Li et al., 2025a). Entropy collapse indicates that the probability mass over the model's vocabulary becomes concentrated on a limited subset of tokens. This phenomenon further implies that LLMs increasingly prioritize exploitation over exploration. As a result, they fail to effectively explore novel reasoning paths during training, thereby potentially causing premature convergence to a local optimum.

To address entropy collapse in LLMs, numerous methods have been proposed. Shen (2025) and He et al. (2025) incorporated an entropy maximization term into the RLVR objective. DAPO (Yu et al., 2025) raised the upper clipping bound of the importance ratio to avoid clipping low-probability tokens. Cui et al. (2025) restricted parameter updates for tokens with high covariance between log-probability and advantage, alongside other approaches (Wang et al., 2025b; Chen et al., 2025; Deng et al., 2025; Li et al., 2025a; Zhu et al., 2025). Despite these advances, systematic investigations of entropy in RLVR remain scarce. Specifically, three critical questions are still underexplored: (1) How does the entropy of LLMs trained with RLVR correlate with their performance? (2) What factors govern entropy dynamics, both theoretically and empirically? and (3) How can entropy be effectively regulated to improve the performance of LLMs?

To investigate the above research questions, we conduct extensive experiments on RLVR. The results show that the entropy of LLMs trained with RLVR is strongly correlated with response diversity, as models with lower entropy tend to produce less diverse outputs. During training, both response and prompt entropy consistently decline, with in-domain prompts decreasing more rapidly than out-of-domain ones. However, prompt entropy shows only a weak correlation with response accuracy. We further find that training with off-policy data exacerbates entropy collapse and leads to model

miscalibration, whereas LLMs can still achieve performance improvements without sacrificing entropy. In addition, training data diversity is a critical factor influencing model entropy, which decreases as data diversity diminishes. Interestingly, an LLM trained on only 616 samples performs comparably to one trained on the full dataset, suggesting that dataset size is not the sole determinant of RLVR performance. Finally, we show that entropy collapse primarily arises from tokens with positive advantages, and that adjusting the relative loss weights assigned to tokens with positive and negative advantages can effectively control entropy.

2 Related Work

Entropy has long served as a regularization mechanism to encourage exploration in reinforcement learning (Ziebart et al., 2008; Ziebart, 2010). In the era of LLMs, several studies have similarly incorporated entropy maximization objectives into RLVR to promote exploration and prevent premature convergence (Shen, 2025; He et al., 2025).Aiming to mitigate entropy collapse, Clip-Higher (Yu et al., 2025) raises the upper clipping bound of the importance ratio to prevent low-probability tokens with positive advantages from being clipped. Liu (2025) and Cui et al. (2025) provide theoretical insights into entropy dynamics, showing that tokens with strong positive covariance between their probabilities and corresponding advantages primarily drive entropy collapse. Building on these insights, Clip-Cov and KL-Cov (Cui et al., 2025) limit updates on tokens exhibiting such covariance. Similarly, CE-GPPO (Su et al., 2025) mitigates entropy collapse by preserving the gradients of clipped tokens through a stop-gradient operation. Furthermore, Wang et al. (2025b) train LLMs using only high-entropy tokens to enhance model performance, while Cheng et al. (2025) incorporate entropy terms into the advantage to encourage exploration and improve reasoning capabilities. Numerous other studies have also explored entropy-based mechanisms to further enhance the performance of LLMs trained with RLVR (Li et al., 2025a; Wang et al., 2025a; Liu et al., 2025a).

3 Preliminaries

3.1 Group Relative Policy Optimization (GRPO)

Let the prompt be $\boldsymbol{x}$ and the response generated by the LLM π_θ (parameterized by θ) be $\boldsymbol{y}$. The reward for response $\boldsymbol{y}$ is denoted as $R(\boldsymbol{y})$. The RLVR objective is to maximize the expected reward of responses generated by π_θ, formulated as:

$$J(\theta) = \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})} R(\boldsymbol{y}). \tag{1}$$

To optimize π_θ for maximizing the expected reward, GRPO (Shao et al., 2024) adopts the surrogate objective of PPO (Schulman et al., 2017), replacing critic-based advantage estimation with group-based reward normalization to reduce the memory and computational cost of training a critic. Instead of learning a state-value function, GRPO samples multiple responses per prompt and computes each response's advantage by normalizing its reward with the group's mean and standard deviation. Following DAPO, we employ a token-level loss to ensure equal token contribution across responses of varying lengths and remove the KL penalty term from the original GRPO objective. Formally, let π_θ sample G responses $\{\boldsymbol{y}^j\}_{j=1}^G$ for each prompt $\boldsymbol{x}$, and let $\mathcal{D}$ denote the prompt dataset. The optimization objective of GRPO with token-level loss and without the KL penalty term is:

$$\begin{aligned}\mathcal{J}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\{\boldsymbol{y}^i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})} \\ \Bigg[\frac{1}{\sum_{i=1}^G|\boldsymbol{y}^i|}\sum_{i=1}^G\sum_{t=1}^{|\boldsymbol{y}^i|}\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \\ \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}\right)\hat{A}_{i,t}\Big)\Bigg],\end{aligned} \tag{2}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(\boldsymbol{y}_t^i|\boldsymbol{x},\boldsymbol{y}_{<t}^i)}{\pi_{\theta_{\text{old}}}(\boldsymbol{y}_t^i|\boldsymbol{x},\boldsymbol{y}_{<t}^i)}$, the advantage is given by $\hat{A}_{i,t} = \frac{R(\boldsymbol{y}^i)-\text{mean}\left(\{R(\boldsymbol{y}^j)\}_{j=1}^G\right)}{\text{std}\left(\{R(\boldsymbol{y}^j)\}_{j=1}^G\right)}$, and ε_{low} and $\varepsilon_{\text{high}}$ are clipping hyperparameters controlling the lower and upper bounds, respectively.

3.2 Entropy Regularization

Let the vocabulary of π_θ be denoted by $\mathcal{V}$. For a prompt $\boldsymbol{x}$ and response $\boldsymbol{y}$, the token-level average entropy is defined as follows:

$$\mathcal{H}(\pi_\theta) = -\frac{1}{|\boldsymbol{y}|}\sum_{t=1}^{|\boldsymbol{y}|}\sum_{v\in\mathcal{V}}\pi_\theta(v|\boldsymbol{x},\boldsymbol{y}_{<t})\log\pi_\theta(v|\boldsymbol{x},\boldsymbol{y}_{<t}). \tag{3}$$

Entropy regularization augments the RLVR objective with the term $\alpha\mathcal{H}(\pi_\theta)$, where α is the entropy regularization coefficient.

3.2.1 Adaptive Entropy Regularization

Despite the effectiveness of vanilla entropy regularization in controlling LLM entropy, prior studies

have shown that selecting an appropriate regularization coefficient is challenging (He et al., 2025): a large coefficient causes entropy to rise rapidly, while a small one renders the regularization ineffective. To address this, He et al. (2025) propose adaptive entropy regularization, which dynamically adjusts the coefficient according to the model's entropy rather than keeping it fixed during training. Let the entropy of the LLM π_θ at training step k be denoted as $\mathcal{H}_k(\pi_\theta)$. The objective is to prevent the entropy from falling substantially below a predefined threshold δ, and the entropy regularization coefficient at step k, α_k, is defined as follows:

$$\alpha_k = c_k \cdot \mathbb{I}\left\{\mathcal{H}_k(\pi_\theta) < \delta\right\}, \tag{4}$$

where the adaptive coefficient c_k is updated according to the following rule:

$$c_{k+1} = c_k + \beta\mathbb{I}\left\{\mathcal{H}_k(\pi_\theta) < \delta\right\} - \beta\mathbb{I}\left\{\mathcal{H}_k(\pi_\theta) \geq \delta\right\}. \tag{5}$$

Equation (4) indicates that entropy regularization is applied only when the entropy drops below the threshold δ, setting the coefficient to c_k. Equation (5) defines the update rule: if the entropy is below δ, c_k increases by β; otherwise, it decreases by β.

4 Experimental Setup

We train `Qwen2.5-Math-7B` (Yang et al., 2024) with RLVR using the veRL framework (Sheng et al., 2025) on the `DAPO-Math-17K` dataset (Yu et al., 2025). Training employs a rollout batch size of 256, generating 16 responses per prompt. The AdamW optimizer is applied with a cosine learning rate schedule and a peak rate of 1×10^{-6}. During rollouts, decoding parameters are fixed as: top-$p = 1.0$, temperature $= 1.0$, and a maximum generation length of 4096 tokens.

We evaluate both in-domain and out-of-domain performance. The in-domain benchmarks include AIME 2024/2025 (MAA, 2024, 2025), MATH500 (Hendrycks et al., 2021), AMC 2023 (MAA, 2023), and Minerva Math (Lewkowycz et al., 2022), while out-of-domain evaluation covers LiveCodeBench (Jain et al., 2025) for coding and IF-Eval (Zhou et al., 2023) for instruction following. For each question, we sample 64 responses and report `Avg@64` and `Pass@64`. Evaluation is performed under two decoding settings: (1) temperature 1.0, top-p 1.0, and (2) temperature 0.6, top-p 0.95, both with a maximum length of 4096 tokens. To mitigate potential data contamination in the Qwen2.5 series of LLMs (Wu et al., 2025), AIME 2025 serves as the validation set, and the best checkpoint is evaluated on all other benchmarks.

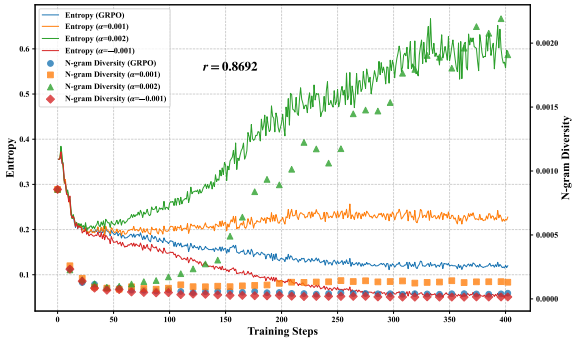

Figure 1: Evolution of entropy (solid lines) and N-gram diversity (markers) over training steps under different entropy regularization settings.

5 How Does the Entropy of LLMs Trained with RLVR Correlate with Their Performance?

> The diversity of responses generated by LLMs is strongly positively correlated with both the entropy of these responses and the entropy of the LLMs during training.

To examine the relationship between response diversity and model entropy, we control LLM entropy using vanilla entropy regularization by varying the regularization coefficient. Experiments are conducted with coefficients $\{0.001, 0.002\}$, without entropy regularization, and with a negative coefficient -0.001 that explicitly promotes entropy minimization. Response diversity is measured using the `N-gram Diversity` metric.

Figure 1 shows the entropy dynamics during RLVR training alongside the `N-gram Diversity` of responses for AIME 2024 prompts. As shown in Figure 1, LLMs trained with GRPO exhibit entropy collapse during training, which is alleviated by vanilla entropy regularization with $\alpha = 0.001$, while a larger coefficient ($\alpha = 0.002$) causes entropy to increase continuously in later stages. Conversely, applying a negative coefficient ($\alpha = -0.001$) further exacerbates entropy collapse. Notably, `N-gram Diversity` follows a similar trajectory to entropy, with an average Pearson correlation of 0.8692 across the four training settings, indicating a strong positive correlation between response diversity and model entropy during training.

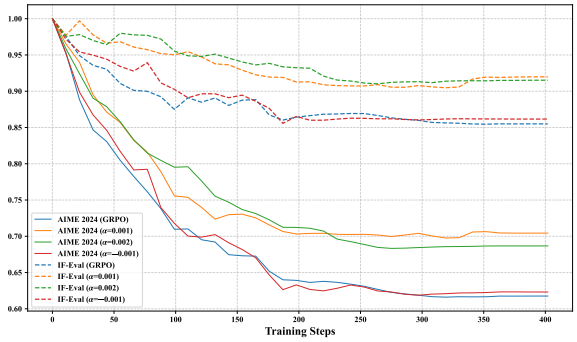

Figure 2: The ratio between the entropy of the LLMs on prompts during training and their entropy on the same prompts before training, plotted against training steps. Solid lines represent results on the AIME 2024, while dashed lines correspond to results on IF-Eval.

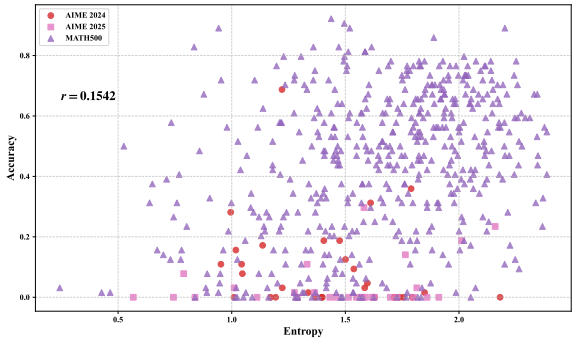

Figure 3: Scatter plot of accuracy versus entropy across AIME 2024, AIME 2025, and MATH500.

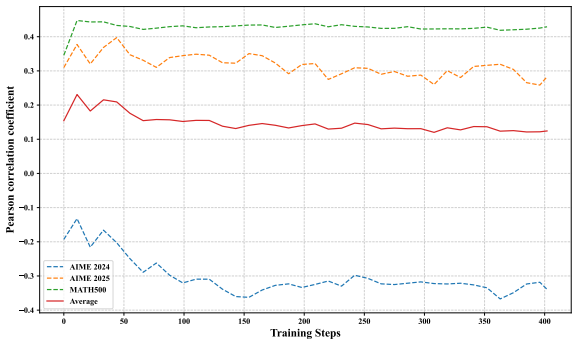

Figure 4: Pearson correlation coefficients between the entropy of LLMs on prompts and their corresponding accuracy across different training steps. "Average" indicates the mean Pearson correlation coefficient computed over AIME 2024, AIME 2025, and MATH500.

During RLVR training, the entropy of LLMs decreases for both in-domain and out-of-domain prompts, with a more substantial reduction observed for in-domain prompts.

In addition to examining the entropy dynamics of responses generated by LLMs trained with RLVR, we also analyze the entropy dynamics of LLMs on prompts during RLVR training. Specifically, we compute the ratio between the entropy of LLMs on prompts at different training steps and their entropy on the same prompts before training, as shown in Figure 2. The entropy of LLMs on prompts decreases over training and eventually stabilizes. This reduction is more pronounced for prompts from the in-domain AIME 2024 benchmark than for those from the out-of-domain IF-Eval benchmark. Moreover, similar to preventing entropy collapse in generated responses, applying a positive entropy regularization coefficient helps maintain the entropy of LLMs on prompts.

The entropy of LLMs on prompts shows only a weak correlation with their accuracy.

To examine the correlation between the entropy of LLMs on prompts and their corresponding accuracy, we measure the accuracy of each prompt as the proportion of correct responses among 64 generated responses. Figure 3 presents the scatter plot of entropy versus accuracy across AIME 2024, AIME 2025, and MATH500 for `Qwen2.5-Math-7B`. The figure shows that the entropy of LLMs on prompts is only weakly correlated with accuracy, as evidenced by the small average Pearson correlation coefficient (0.1542) across the three benchmarks. We further compute the Pearson correlation coefficient between entropy and accuracy at different training steps. As shown in Figure 4, although the coefficient fluctuates throughout training, it remains consistently small, confirming the weak correlation between entropy and accuracy.

More off-policy updates exacerbate entropy collapse, leading to higher training rewards and stronger coding capacity, which improve Avg@64 but lower Pass@64.

In GRPO, a batch of prompts is first sampled and used for rollout, advantage estimation, and the computation of log-probabilities with the LLM employed for rollout. This batch is then divided into several mini-batches, and the model parameters are updated once per mini-batch. Since the parameters change after the first mini-batch update, the data in the remaining mini-batches can be regarded as off-policy data with respect to the updated policy.

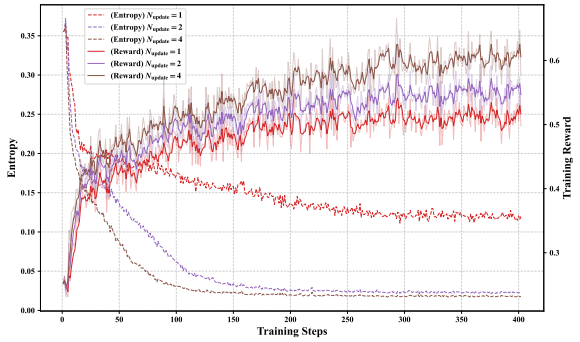

Figure 5: Evolution of the entropy and reward of LLMs during training.

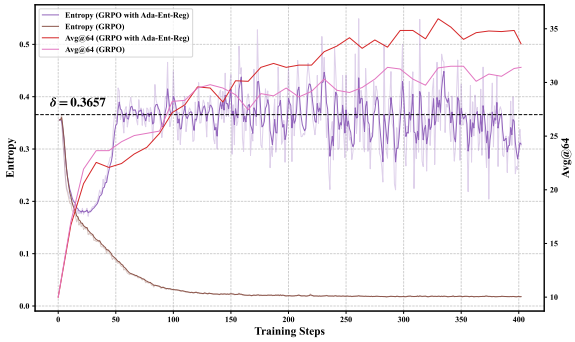

Figure 6: Evolution of the entropy of LLMs during training and their Avg@64 performance on AIME 2024 for models trained with RLVR, with and without adaptive entropy regularization. The predefined entropy threshold δ is set to 0.3657, corresponding to the entropy of the LLM on responses to 1,000 randomly sampled prompts from the training dataset prior to training. "Ada-Ent-Reg" denotes adaptive entropy regularization.

To examine the impact of off-policy data on the entropy and reward of LLMs during training, we varied the number of mini-batches per batch, which corresponds to the number of parameter updates per batch and is denoted as N_{update}, while keeping the rollout batch size and all other hyperparameters fixed. Specifically, N_{update} was set to 1, 2, and 4.

Figure 5 presents the evolution of entropy and reward under these configurations. As shown in the figure, increasing N_{update} leads to a more pronounced entropy collapse, consistent with the findings of He et al. (2025), while yielding higher training rewards. Moreover, Table 2 further shows that lower entropy is generally associated with stronger coding capabilities. Consequently, performing more off-policy updates tends to enhance the model's coding capacity. However, as entropy decreases, we observe that the model's average Avg@64 performance across benchmarks improves, whereas Pass@64 declines.

> The performance of LLMs continues to improve during training, even as their entropy fluctuates around the value observed before training, indicating that performance gains are not solely achieved by trading off entropy.

To maintain the entropy of LLMs at a level comparable to that observed before training, we apply adaptive entropy regularization during RLVR training. Specifically, we randomly sample 1,000 prompts from the training dataset and compute the entropy of the LLMs on the corresponding responses prior to training. The computed entropy value is then set as the predefined threshold δ, ensuring that the entropy of the LLMs remains above this threshold throughout training.

Figure 6 illustrates the evolution of entropy and accuracy on AIME 2024. As shown in Figure 6, under adaptive entropy regularization, the entropy of LLMs decreases sharply at the beginning of training, then increases, and subsequently fluctuates around the specified threshold. Meanwhile, accuracy on AIME 2024 exhibits a upward trend, even surpassing that of the model trained without adaptive entropy regularization. This observation indicates that performance improvement of LLMs is not merely achieved by trading off entropy. In contrast, LLMs trained without entropy regularization experience rapid entropy collapse, wherein entropy declines to a low value. Correspondingly, the Avg@64 metric rises sharply in the early stages of training but soon plateaus, remaining inferior to that of the model trained with entropy regularization. These results suggest that entropy collapse may lead to performance degradation.

> While RLVR enhances the performance of LLMs, it can lead to miscalibration, causing the models to become increasingly overconfident in their predictions. This miscalibration typically intensifies as entropy collapse becomes more severe. However, incorporating entropy regularization can mitigate this problem.

We further investigate the relationship between entropy and the calibration of LLMs. For a well-calibrated model, correct responses should receive higher probabilities, whereas incorrect ones should be assigned lower probabilities. To assess this, we compute the distributions of average per-token log

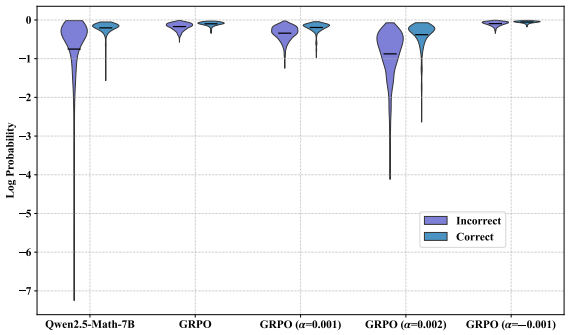

Figure 7: Distribution of log probabilities for correct and incorrect responses across different LLMs. "GRPO" denotes the LLM trained using GRPO, while "GRPO" followed by parentheses indicates LLMs trained with entropy regularization under varying coefficients. In each violin plot, the black line represents the mean.

probabilities for correct and incorrect responses generated by LLMs, as illustrated in Figure 7.

As shown in Figure 7, prior to training, `Qwen2.5-Math-7B` generally assigns higher probabilities to correct responses than to incorrect ones. However, after GRPO training, the probabilities of both correct and incorrect responses increase, indicating that the model becomes more overconfident. Meanwhile, the probability gap between correct and incorrect responses narrows, suggesting reduced discriminability and poorer calibration, which aligns with the observations of Bereket and Leskovec (2025). Furthermore, when the entropy regularization coefficient is set to a negative value, thereby promoting entropy collapse, both overconfidence and miscalibration become more pronounced. In contrast, employing a positive entropy regularization coefficient mitigates these effects. Considering Figures 7 and 1 jointly, we observe a consistent trend in overconfidence and miscalibration: GRPO ($\alpha = -0.001$) > GRPO > GRPO ($\alpha = 0.001$) > GRPO ($\alpha = 0.002$). A similar ordering is observed for entropy collapse, suggesting a potential correlation between miscalibration and entropy collapse.

6 What Factors Govern Entropy Dynamics, Both Theoretically and Empirically?

Liu (2025) and Cui et al. (2025) provided theoretical analyses identifying the factors that govern entropy dynamics during RLVR training. In brief, they showed that tokens exhibiting strong positive covariance between their probabilities and corresponding advantages drive the substantial entropy decrease observed in RLVR. Formally, letting η denote the learning rate at training step k, the entropy change of the LLM π_θ between steps $k+1$ and k can be approximated as:

$$\mathcal{H}\left(\pi_\theta^{k+1}\left(\boldsymbol{y_t}|\boldsymbol{x},\boldsymbol{y}_{<t}\right)\right) - \mathcal{H}\left(\pi_\theta^{k}\left(\boldsymbol{y_t}|\boldsymbol{x},\boldsymbol{y}_{<t}\right)\right) \approx$$
$$-\eta \cdot \mathrm{Cov}_{\boldsymbol{y}\sim\pi_\theta^k(\cdot|\boldsymbol{x})}\left(\log \pi_\theta^k\left(\boldsymbol{y}_t|\boldsymbol{x},\boldsymbol{y}_{<t}\right), \pi_\theta^k\left(\boldsymbol{y}_t|\boldsymbol{x},\boldsymbol{y}_{<t}\right)\hat{A}_t\right). \tag{6}$$

While Liu (2025) and Cui et al. (2025) focus on theoretical explanations of entropy dynamics, we complement their findings with an empirical investigation emphasizing the influence of data diversity. Specifically, we trained LLMs on subsets of the original training data obtained via either K-means clustering or random sampling, ensuring that both methods selected the same number of samples. Intuitively, data selected through K-means clustering are expected to exhibit lower diversity than data obtained through random sampling. Furthermore, as the dataset size decreases, the overall data diversity diminishes correspondingly.

For K-means clustering, each prompt was represented by the mean embedding of its tokens, extracted from the final layer of `Qwen2.5-Math-7B`. K-means clustering was then applied to these embeddings with K set to 1,000. The resulting clusters were sorted in descending order of size, and samples were drawn from the top M clusters. Specifically, M was set to 281, 112, 49, 21, and 9, corresponding to subsets containing 10,001, 5,031, 2,538, 1,246, and 616 samples, respectively.

For random sampling, we selected the same numbers of samples (10,001, 5,031, 2,538, 1,246, and 616) as in the corresponding K-means configurations to ensure a fair comparison. Finally, we trained `Qwen2.5-Math-7B` with RLVR using the subsets constructed by both K-means clustering and random sampling, while keeping all other training hyperparameters identical.

Since the entropy of LLMs evolves dynamically during training, we employ an Exponential Moving Average (EMA) to mitigate short-term fluctuations and report the smoothed entropy at the final training step. Formally, at step k, the EMA-smoothed entropy $\mathcal{H}_k^{\text{EMA}}$ is defined as:

$$\mathcal{H}_k^{\text{EMA}} = \mathcal{H}_k(\pi_\theta^k)(1-\varphi) + \varphi\mathcal{H}_{k-1}^{\text{EMA}}, \tag{7}$$

where the smoothing coefficient φ is set to 0.6. The experimental results are summarized in Table 1.

Interestingly, Table 1 reveals that, despite the substantial reduction in training data, LLMs trained

Model	AIME 2024	AIME 2025	MATH500	AMC 2023	Minerva	LiveCodeBench	IF-Eval	Average	Entropy
Qwen2.5-Math-7B	10.00 / 60.00	3.80 / 33.33	43.76 / 95.60	30.04 / 92.50	14.41 / 60.29	3.62 / 30.15	22.67 / 80.46	18.33 / 64.62	N/A
+ GRPO (Full-data)	28.75 / 63.33	14.69 / 50.00	78.14 / 96.80	64.38 / 97.50	34.64 / 64.34	7.85 / 33.46	30.17 / 72.90	36.94 / 68.33	0.11838
+ GRPO (10,001$_{\text{K-means}}$)	31.41 / 63.33	14.22 / 46.67	76.69 / 95.60	64.69 / 95.00	34.38 / 65.81	7.67 / 33.82	29.71 / 73.74	36.97 / 67.71	0.11419
+ GRPO (10,001$_{\text{random}}$)	30.00 / 63.33	15.10 / 50.00	77.29 / 96.00	63.48 / 92.50	34.48 / 65.81	8.08 / 35.29	30.56 / 71.46	37.00 / 67.77	0.11777
+ GRPO (5,031$_{\text{K-means}}$)	30.16 / 66.67	13.75 / 46.67	74.86 / 95.40	62.85 / 95.00	33.99 / 65.81	8.40 / 34.19	30.90 / 71.58	36.42 / 67.90	0.10722
+ GRPO (5,031$_{\text{random}}$)	30.47 / 70.00	14.53 / 50.00	74.8 / 94.80	63.36 / 95.00	33.32 / 64.34	7.95 / 34.93	30.29 / 72.78	36.39 / 68.84	0.09953
+ GRPO (2,538$_{\text{K-means}}$)	30.94 / 70.00	13.85 / 46.67	75.29 / 95.40	62.23 / 100.00	34.09 / 63.24	8.77 / 35.66	30.41 / 73.14	36.51 / 69.16	0.08674
+ GRPO (2,538$_{\text{random}}$)	30.00 / 70.00	14.32 / 53.33	74.58 / 95.00	63.55 / 95.00	33.67 / 63.24	8.13 / 31.62	29.49 / 72.90	36.25 / 68.73	0.10369
+ GRPO (1,246$_{\text{K-means}}$)	30.31 / 70.00	14.01 / 36.67	76.83 / 95.80	63.98 / 95.00	34.71 / 62.50	8.65 / 33.09	30.45 / 71.70	36.99 / 66.39	0.08588
+ GRPO (1,246$_{\text{random}}$)	30.47 / 63.33	14.69 / 56.67	75.99 / 95.20	65.27 / 95.00	34.29 / 65.44	8.87 / 34.56	30.47 / 70.50	37.15 / 68.67	0.09030
+ GRPO (616$_{\text{K-means}}$)	29.32 / 60.00	17.60 / 46.67	78.57 / 94.80	64.10 / 87.50	36.05 / 59.19	9.06 / 34.93	30.70 / 67.75	37.91 / 64.40	0.02398
+ GRPO (616$_{\text{random}}$)	27.55 / 63.33	15.00 / 53.33	68.54 / 89.40	65.27 / 95.00	29.62 / 58.09	9.99 / 36.03	31.23 / 70.26	35.32 / 66.49	0.02926

Table 1: Performance of `Qwen2.5-Math-7B` trained with GRPO under varying data scales across multiple benchmarks. Entropy denotes the EMA-smoothed entropy at the final training step. The numbers in parentheses indicate the number of training samples, where subscripts "k-means" and "random" refer to datasets selected via K-means clustering and random sampling, respectively. Results are reported as A / B, corresponding to `Avg@64` and `Pass@64`, evaluated with temperature = 1.0, top-p = 1.0, and a maximum generation length of 4096 tokens.

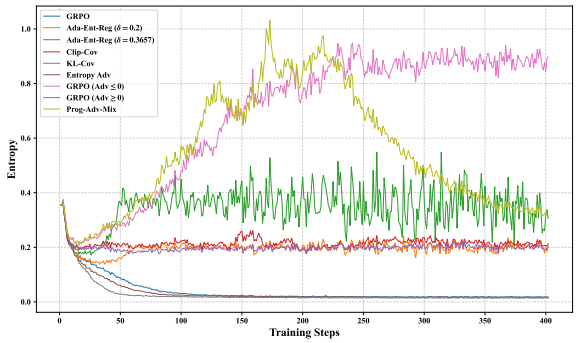

Figure 8: Evolution of the entropy of LLMs during RLVR training under different methods. "Ada-Ent-Reg" denotes Adaptive Entropy Regularization, and "Prog-Adv-Mix" refers to Progressive Advantage Mixing.

on significantly smaller subsets can still achieve performance comparable to those trained on the full dataset. This finding suggests that data scale is not the sole determinant of model performance, aligning with prior studies (Li et al., 2025b; Ye et al., 2025; Muennighoff et al., 2025). Furthermore, we observe that entropy decreases as the amount of training data is reduced. Notably, LLMs trained on data constructed using K-means clustering consistently exhibit lower entropy than those trained on randomly sampled data, except for the case of LLMs trained on 5,031 samples. These results further demonstrate that the diversity of training data influences entropy dynamics, as entropy tends to decrease when data diversity diminishes.

7 How Can Entropy Be Effectively Regulated to Improve the Performance of LLMs?

To effectively regulate entropy, we first examine which types of tokens that drive entropy changes during training. Liu (2025) and Cui et al. (2025) mathematically proved that the entropy change of an LLM between two consecutive training steps can be approximated by the product of the learning rate and the covariance between $\log \pi_\theta^k(\boldsymbol{y}_t|\boldsymbol{x}, \boldsymbol{y}_{<t})$ and $\pi_\theta^k(\boldsymbol{y}_t|\boldsymbol{x}, \boldsymbol{y}_{<t})\,\hat{A}_t$. This implies that tokens with both high probabilities and high advantages, or low probabilities and low advantages, primarily contribute to the entropy collapse.

Since tokens with high probabilities are more likely to be sampled during decoding, most tokens appearing in model-generated responses tend to have high probabilities. Based on this intuition, we hypothesize that the entropy collapse of LLMs mainly stems from tokens with high advantages. To empirically validate this hypothesis, we conducted two comparative experiments: one in which LLMs were trained exclusively on tokens with advantages ≥ 0, and another in which training was restricted to tokens with advantages ≤ 0. We further compared these settings against the following baselines:

- **Adaptive Entropy Regularization** dynamically adjusts the entropy regularization coefficient to keep model entropy above a predefined threshold δ. We consider two settings for δ: (1) $\delta = 0.2$, following He et al. (2025), and (2) δ set to the entropy of the LLM on responses to 1,000 randomly sampled training prompts before training, which equals 0.3657.
- **Clip-Cov** (Cui et al., 2025) mitigates entropy collapse by zeroing the gradients of a small subset of tokens with high covariance between log probabilities and advantages.
- **KL-Cov** (Cui et al., 2025) adopts a similar approach to Clip-Cov but applies a KL penalty

Model	AIME 2024	AIME 2025	MATH500	AMC 2023	Minerva	LiveCodeBench	IF-Eval	Average	Entropy
Qwen2.5-Math-7B	10.00 / 60.00	3.80 / 33.33	43.76 / 95.60	30.04 / 92.50	14.41 / 60.29	3.62 / 30.15	22.67 / 80.46	18.33 / 64.62	N/A
+ GRPO ($N_{\text{update}} = 1$)	28.75 / 63.33	14.69 / 50.00	78.14 / 96.80	64.38 / 97.50	34.64 / 64.34	7.85 / 33.46	30.17 / 72.90	36.94 / 68.33	0.11838
+ GRPO ($N_{\text{update}} = 2$)	29.58 / 70.00	16.98 / 46.67	76.56 / 94.40	67.42 / 92.50	33.28 / 62.50	10.66 / 34.56	28.72 / 71.10	37.60 / 67.39	0.02286
+ GRPO ($N_{\text{update}} = 4$)	31.41 / 63.33	14.90 / 56.67	72.09 / 90.80	75.43 / 90.00	31.14 / 55.51	12.37 / 34.56	29.83 / 70.38	38.17 / 65.89	0.01789
+ Ada-Ent-Reg ($\delta = 0.2$)	32.92 / 66.67	16.30 / 50.00	69.05 / 90.40	69.34 / 92.50	27.63 / 61.03	7.16 / 33.09	31.26 / 68.94	36.24 / 66.09	0.19692
+ Ada-Ent-Reg ($\delta = 0.3657$)	33.96 / 66.67	18.65 / 50.00	73.98 / 92.80	68.52 / 97.50	31.66 / 61.76	6.31 / 32.35	29.66 / 69.78	37.53 / 67.27	0.30941
+ Clip-Cov	31.98 / 70.00	18.18 / 53.33	74.27 / 95.80	68.13 / 97.50	32.23 / 62.50	7.85 / 34.56	31.05 / 69.06	37.67 / 68.97	0.20899
+ KL-Cov	33.96 / 66.67	16.20 / 53.33	70.10 / 93.60	68.79 / 97.50	28.63 / 61.03	7.61 / 34.93	30.82 / 68.35	36.59 / 67.91	0.19695
+ Entropy Adv	31.09 / 66.67	15.52 / 46.67	76.65 / 93.60	70.63 / 87.50	33.80 / 60.29	11.75 / 31.25	29.21 / 70.14	38.38 / 65.16	0.01669
+ Adv ≤ 0	29.79 / 63.33	11.04 / 46.67	76.55 / 96.00	63.40 / 97.50	32.73 / 63.97	3.64 / 26.47	33.70 / 62.11	35.84 / 65.15	0.88384
+ Adv ≥ 0	27.55 / 53.33	13.23 / 43.33	72.20 / 91.00	64.49 / 92.50	34.05 / 58.09	10.92 / 33.46	28.17 / 71.58	35.80 / 63.33	0.01460
+ Prog-Adv-Mix	31.72 / 56.67	15.21 / 46.67	78.79 / 96.00	64.84 / 95.00	33.67 / 65.81	4.92 / 31.99	33.07 / 60.91	37.46 / 64.72	0.32983

Table 2: Performance of Qwen2.5-Math-7B trained with GRPO and its entropy-regularized variants. "Ada-Ent-Reg" refers to Adaptive Entropy Regularization, while "Prog-Adv-Mix" denotes the Progressive Advantage Mixing approach. Results are reported as A / B, corresponding to `Avg@64` and `Pass@64`, evaluated with temperature = 1.0, top-p = 1.0, and a maximum generation length of 4096 tokens.

to high-covariance tokens.

- **Entropy Adv** (Cheng et al., 2025) augments the original advantage with an entropy term to encourage exploratory reasoning tokens and enhance LLM performance.

As shown in Section 5, increasing the number of parameter updates per batch (N_{update}) intensifies entropy collapse. Accordingly, we primarily conduct experiments under the setting where N_{update} is fixed at 4 to evaluate the effectiveness of different approaches in controlling the entropy of LLMs.

Figure 8 illustrates the evolution of entropy during training under different methods, while Table 2 summarizes their performance across various benchmarks. As shown in Figure 8, Clip-Cov, KL-Cov, and Adaptive Entropy Regularization effectively alleviate entropy collapse, whereas Ent-Adv intensifies it. Training LLMs exclusively on tokens with advantages ≥ 0 leads to the most severe entropy collapse, while training on tokens with advantages ≤ 0 drives entropy to a high level. These findings empirically validate our hypothesis that entropy collapse primarily stems from tokens with high advantages, implying that adjusting the relative loss weights of tokens with advantages ≥ 0 and ≤ 0 can regulate model entropy.

Building on this insight, we propose Progressive Advantage Mixing (Prog-Adv-Mix), which controls entropy by dynamically adjusting the contributions of tokens with different advantages during training. Specifically, we introduce a hyperparameter λ that determines the loss weight of tokens with advantages ≥ 0. During the first half of training, λ is fixed at 0, meaning that the model is trained primarily on tokens with non-positive advantages. In the second half, λ increases linearly from 0 to 1, progressively amplifying the influence of positive-advantage tokens as training proceeds. This gradual inclusion of high-advantage tokens prevents early-stage entropy collapse while facilitating performance improvements in later stages.

As shown in Figure 8, Prog-Adv-Mix induces a early rise in entropy followed by a stable decline, confirming that adjusting the relative loss weights of tokens with advantages ≥ 0 and ≤ 0 effectively controls entropy. Moreover, Table 2 shows that Prog-Adv-Mix achieves performance comparable to other entropy-regularization methods, demonstrating its ability to mitigate entropy collapse while maintaining competitive performance.

8 Conclusion

In this paper, we systematically investigate the entropy dynamics of LLMs trained with RLVR. Through extensive empirical analyses, we reveal that the entropy of LLMs is correlated with response diversity, model calibration, and overall performance across multiple benchmarks. Our findings indicate that entropy collapse primarily stems from tokens with positive advantages and is further exacerbated by off-policy updates and limited training data diversity. Moreover, we demonstrate that entropy can be effectively regulated by adjusting the relative loss weights assigned to tokens with positive and negative advantages. Building on this insight, we introduce Progressive Advantage Mixing (Prog-Adv-Mix), a simple yet effective approach that dynamically balances the influence of these tokens during training, thereby mitigating entropy collapse while maintaining competitive performance compared with other entropy regularization methods across benchmarks.

Limitations

Due to computational constraints, our experiments were conducted solely on `Qwen2.5-Math-7B`. As LLMs vary in their pre-training data, training strategies, and architectures, some of our findings may not generalize directly to other models. Nevertheless, we hope that our analysis of entropy dynamics in RLVR training provides valuable empirical insights for the research community and motivates the development of more effective entropy regularization methods in future work.

References

Michael Bereket and Jure Leskovec. 2025. Uncalibrated reasoning: GRPO induces overconfidence for stochastic outcomes. *CoRR*, abs/2508.11800.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. 2025. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *CoRR*, abs/2508.10751.

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

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Jia Deng, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Decomposing the entropy-performance exchange: The missing keys to unlocking effective reinforcement learning. *CoRR*, abs/2508.02260.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *CoRR*, abs/2505.22312.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, and Jing Yang. 2025a. CURE: critical-token-guided re-concatenation for entropy-collapse prevention. *CoRR*, abs/2508.11016.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025b. LIMR: less is more for RL scaling. *CoRR*, abs/2502.11886.

Jiacai Liu. 2025. How does rl policy entropy converge during iteration? `https://zhuanlan.zhihu.com/p/28476703733`.

Zheng Liu, Mengjie Liu, Siwei Wen, Mengzhang Cai, Bin Cui, Conghui He, and Wentao Zhang. 2025a. From uniform to heterogeneous: Tailoring policy optimization to every token's nature. *Preprint*, arXiv:2509.16591.

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

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepcoder: A fully open-source 14b coder at o3-mini level. Notion Blog.

MAA. 2023. American Mathematics Competitions - AMC 2023.

MAA. 2024. American Invitational Mathematics Examination - AIME 2024.

MAA. 2025. American Invitational Mathematics Examination - AIME 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *CoRR*, abs/2501.19393.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. 2025. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *Preprint*, arXiv:2509.20712.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. 2025a. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents. *Preprint*, arXiv:2509.09265.

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

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. 2025. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. *CoRR*, abs/2507.10532.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

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

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in LLM reasoning. *CoRR*, abs/2506.01347.

Brian D. Ziebart. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. thesis, Carnegie Mellon University, USA.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1433–1438. AAAI Press.