Shorter Answers Are All You Need

Kyriakos Ftellehas
University College London

November 13, 2025

Abstract

We propose a novel curriculum training method for mathematical reasoning using GRPO. The purpose is to reduce training time while maintaining performance by training the model to produce shorter reasoning traces for most of the training period, thereby significantly cutting the training time. We then elongate the model responses and observe similar accuracies to experiments that were trained with the fully verbose reasoning traces for the whole training.

1 Introduction

Large Language Models (LLMs) have struggled with logical reasoning since their inception, which was remedied early on with chain of thought patterns [Wei et al., 2023]. Logical reasoning, particularly in Math and code, was revolutionized by the popularization of RLVR (Reinforcement Learning with Verifiable Rewards) by the DeepSeek-R1 model and its related papers [DeepSeek-AI et al., 2025,Shao et al., 2024], where the authors used a variant of PPO (Proximal Policy Optimization) [Schulman et al., 2017] called GRPO (Group Relative Policy Optimization), in which the value estimation for each state was replaced with the average reward of the model on a group of answers to the same question. This significantly reduced computational overhead and allowed larger scale training, which proved crucial in training larger LLMs with hundreds of billions of parameters. In this paper we propose a curriculum training method that further speeds up the GRPO training, by incentivizing the model to answer questions in a concise manner for the bulk of training and then incentivizing to recover its verbosity just before training finishes. During the 'brevity phase' of the training we observe a 2 to 3 times decrease in response length and a similar magnitude speedup in training, as time taken per epoch is approximately proportional to tokens generated during it.

With correct tuning of the method, we also observe negligible drop in performance, something further justified by the larger concentration of high entropy tokens in the short answers, something that is evidenced to improve reasoning gains in works such as [Wang et al., 2025]

2 Methodology

2.1 Group-Relative Policy Optimization for reasoning with KL-Penalty

GRPO adapts PPO for outcome rewards. Let π_θ be the policy and π_{ref} be the reference policy. For each problem x in batch D

1. **Sampling:** Generate k chains $\{y_1, \ldots, y_k\} \sim \pi_\theta(\cdot|x)$
2. **Reward:** Use the reward function $R(x, y_i)$ to score each response
3. **Advantage:** Compute the advantage $\hat{A}(x, y_i) = \frac{R(x,y_i) - \frac{1}{k}\sum_{j=1}^{k} R(x,y_j)}{std(R(x,y_1),\ldots,R(x,y_k))}$

4. **Loss:** For each x

$$L_x = \frac{1}{\sum_{i=1}^{k} |y_i|} \sum_{i=1}^{k} \left[-\sum_{t=1}^{|y_i|} \log(\pi_\theta(y_{i,t}|x, y_{i,<t}) \cdot \hat{A}(x, y_i) \right] + \beta \cdot \mathrm{KL}(\pi_\theta||\pi_{\text{ref}}) \tag{1}$$

with Monte Carlo Estimator for the KL divergence. Let $r_t = \log \pi_{\text{ref}}(y_{i,t}|x, y_{i,<t}) - \log \pi_\theta(y_{i,t}|x, y_{i,<t})$

$$\hat{KL}_i(\pi_\theta||\pi_{\text{ref}}) = \sum_{t=1}^{|y_i|} (\exp r_t - r_t - 1) \text{ and } \mathrm{KL}(\pi_\theta||\pi_{\text{ref}}) \sim \hat{KL}(\pi_\theta||\pi_{\text{ref}}) = \frac{1}{k} \sum_{i=1}^{k} \hat{KL}_i(\pi_\theta||\pi_{\text{ref}}) \tag{2}$$

2.2 Noteable choices

1. **Normalization:** A notable choice we made is the $\sum_{i=1}^{k} |y_i|$ normalization in (1). Which is to avoid the length bias present in the original GRPO paper as pointed out in the DrGRPO framework developed in [Liu et al., 2025]. Look at the aforementioned paper for a discussion on the topic. Another potential choice for normalization is to use a constant factor $N \cdot k$ ($N = 3000$ in Liu et al., 2025, chosen as the maximum reasoning length]). We have however observed that for our purposes, since the reasoning length $|y_i|$ is variable over time using a constant normalization term $N \cdot k$ makes the relative magnitude of the policy objective $\frac{1}{N \cdot k} \sum_{i=1}^{k} \left[-\sum_{t=1}^{|y_i|} \log(\pi_\theta(y_{i,t}|x, y_{i,<t}) \cdot \hat{A}(x, y_i) \right]$ change compared to the KL part $\beta \cdot KL(\pi_\theta||\pi_{\text{ref}})$ for constant β causing the training to be more unstable. So we have used a modification of the DrGRPO objective that still does not have any length bias. (As it still avoids scaling each particular answer by its *own* length and instead uses the length of the whole group)

2. **KL estimator:** We have observed that while the estimator: $\sum_{t=1}^{|y_i|} r_t$ as defined in (2) is the obvious choice for an unbiased KL estimator, the choice $\sum_{t=1}^{|y_i|} (\exp(r_t) - r_t - 1)$ apart from lower variance, provides much more stable gradients. This was empirically observed from the stability of our experiments, and can be intuitively validated as well by considering the derivative of these gradients wrt θ. This provides further justification for the KL estimator choice in [Shao et al., 2024] which is identical to ours.

2.3 Verbosity curriculum

While the usual $R(x, y_i)$ is 1 when y_i is correct and 0 otherwise for our algorithm we define the following. Let o be the order of reasoning traces in the group of answers. (e.g. if y_i has the shortest reasoning trace $o(y_i) = 1$)

1. $R_{long}(x, y_i) = \frac{o(y_i)}{k}$ if y_i is correct 0 otherwise
2. $R_{short}(x, y_i) = \frac{k - o(y_i) + 1}{k}$ if y_i is correct 0 otherwise
3. $R_{\text{neutral}}(x, y_i) = 1$ if y_i is correct 0 otherwise

We initially choose $s = 0.7$, $l = 5$. For the first (total epochs) $\cdot$ s epochs we use the R_{short} reward, for the next l epochs we use R_{long} reward and for the rest of the epochs we use R_{neutral}. We define these phases as:

1. **Short Phase:** If (total epochs) $= N$ it lasts for $N \cdot s$ and makes the reasoning traces shorter (s is a number close to 1)

2. **Long Phase:** Lasts for l epochs (l is a small number like 5 or 10) and gives a kickstart for the model to elongate its answers

3. **Neutral Phase:** Lasts for the remaining epochs and here is where the model is observed to elongate its answers and regain its full accuracy

3 Experiments

3.1 Setup

All experiments were ran on 1 **A100** GPU. The training consisted of **1 pass over the training set of GSM8K dataset (7.5K problems)**. For each problem **8 rollouts** were created, **k = 8**. **AdamW optimizer** was used, with **0 weight decay**, $3 \cdot 10^{-6}$ **learning rate** and **batch size 64**. The evaluation consisted of a **Pass@8** test over the test set of **1.3k problems**

3.2 Results and Analysis

We will refer to our **Verbosity Curriculum augmented GRPO** as **VC-GRPO** for brevity in the rest of the paper

1. **Model:** Qwen0.5B-Instruct

Table 1: GSM8K pass@8 accuracy for different experiments

Method	**Pass@8 Accuracy**
GRPO Baseline	0.675 ± 0.005
VC-GRPO ($\beta = 0$,$l = 5$,$s = 0.7$)	0.46 ± 0.01
VC-GRPO ($\beta = 0.001$,$l = 5$, $s = 0.7$)	0.67 ± 0.0025

2. **Model:** Gemma-3-1b-it

Table 2: GSM8K pass@8 accuracy for different experiments

Method	**Pass@8 Accuracy**
GRPO Baseline	0.81 ± 0.003
VC-GRPO ($\beta = 0$,$l = 5$,s $= 0.7$)	0.535 ± 0.015
VC-GRPO ($\beta = 0.001$,$l = 5$,$s = 0.7$)	0.7 ± 0.03
VC-GRPO ($\beta = 0.001$, $l = 10$,$s = 0.7$)	0.74 ± 0.04
VC-GRPO ($\beta = 0.001$, $l = 15$,$s = 0.6$)	0.78 ± 0.02

3.3 Training Graphs

Figure 1: **Representative Graph for the Response Length throughout training**

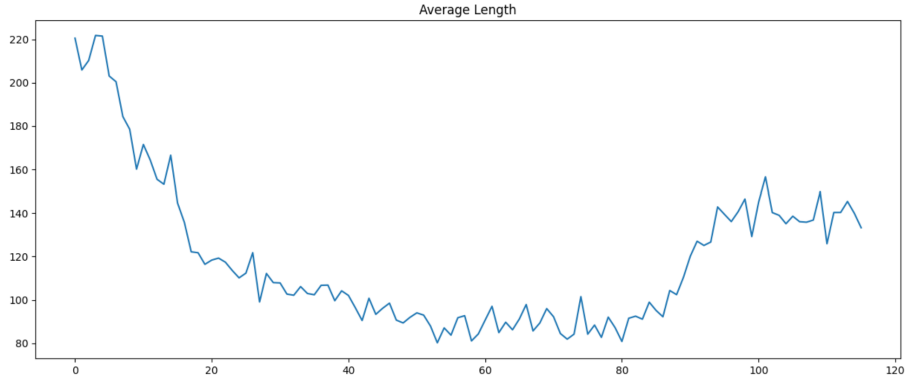


Figure 2: **Representative Graph for the Training Accuracy throughout training**

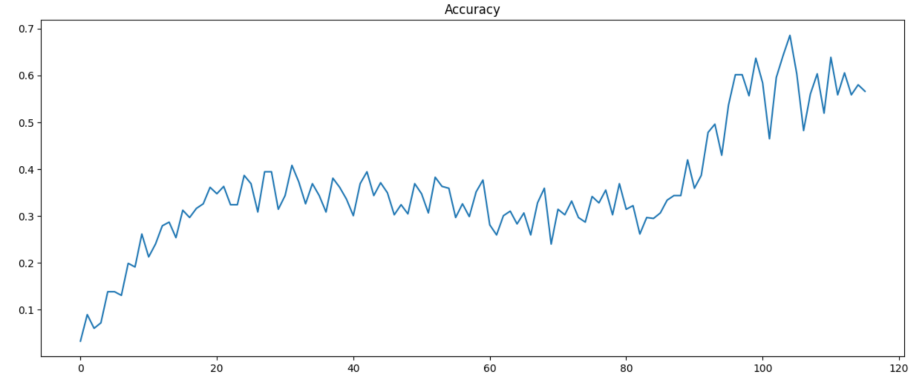
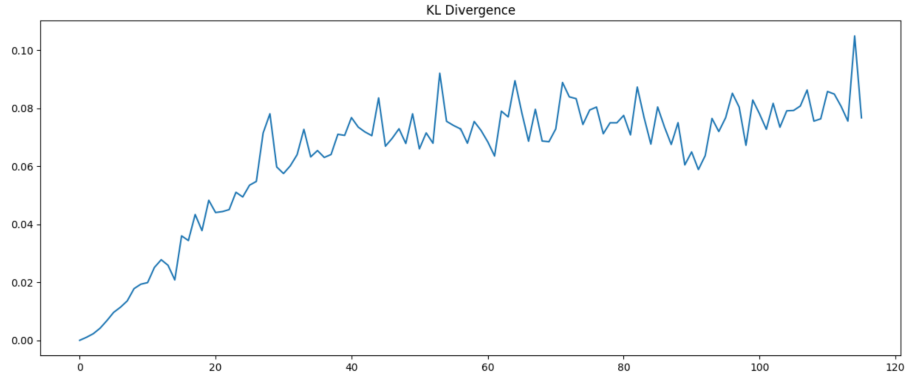


Figure 3: **Representative Graph for the KL Divergence throughout training**



3.4 Analysis

The representative graphs come from one of the runs on the **Qwen0.5B-Instruct** model with $\beta = 0.001$ and $l = 5$. The rate at which training progresses is $\sim$ **average length** at any point. For the **Qwen** model the short phase average length was around 80 and for the **Gemma** was around 100. Without the length penalty the models preferred to write answers of approximately 150 and 300 words respectively, therefore our method effectively achieves approximately a 1.9x and 3x speedup.

Role of β: Without the $\beta = 0.001$ regularization, the model struggled to increase its average length and recover its accuracy. This is in line with the findings of Shenfeld et al., 2025 where it is found that a low KL divergence on the **new task** signals **lower** catastrophic forgetting of the old tasks (verbalizing longer answers in our case). In fact without the regularization β was around 5x higher at ~ 0.5, which clearly resulted in failure of the **long** and **neutral** phase in improving the accuracy and average length as can be seen in figures 3.3 and 3.3.

Role of l: It was observed that higher l was necessary to get the **Gemma** model to fully recover its verbosity, as the gap between the verbosity of the short phase and the preferred length of the model was larger (100 to 300).

4 Entropy justification

The success of the method could be explained by the higher concentration of high entropy tokens in the answers of the short phase in the training.

Consider an answer y to question x comprised of tokens $\{y_1, \ldots, y_T\}$, the current policy π_θ and vocabulary $\mathcal{V} := \{v_1, \ldots, v_{|\mathcal{V}|}\}$. The entropy at token t is defined as:

$$H_t = -\sum_{v \in \mathcal{V}} \pi_\theta(y_t | x, y_{j<t}) \cdot \log \pi_\theta(y_t | x, y_{j<t}) \tag{3}$$

and it is a measure of how unsure the model is about the generation of that token. It is documented in works such as Wang et al., 2025 that the all reasoning gains come from training on the 20% of highest entropy tokens, which also aligns with our intuition of the fact that learning happens when the model learns to take the correct path when it is unsure of which path to take.

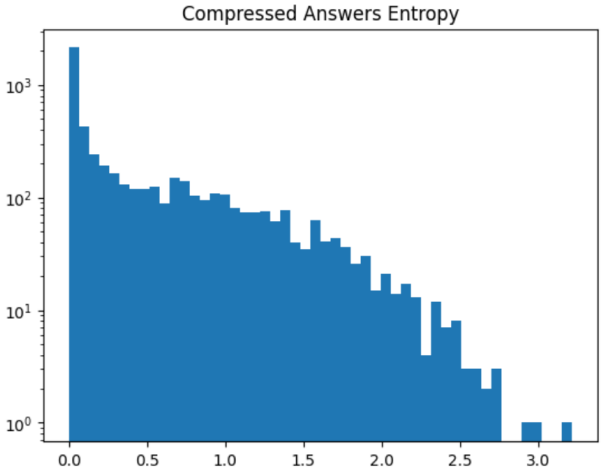


Figure 4: Distribution of token entropies in the compressed answers (log scale).Horizontal axis is entropy, vertical axis is frequency

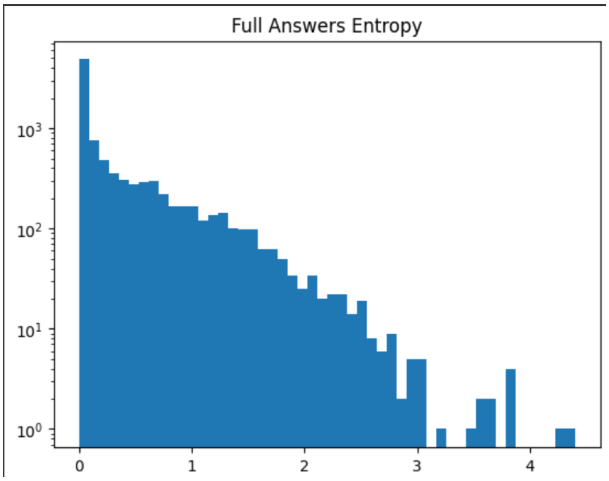


Figure 5: Distribution of token entropies in the long answers (log scale). Horizontal axis is entropy, vertical axis is frequency

Observing the distributions of token entropies (log scale), it is clear that the high entropy tokens remain in the shortened answers of the short phase, supporting the hypothesis that training on those shorter reasoning traces is just as impactful to the models internal understanding, since the model trains on the same amount of high entropy tokens, just in a shorter time span.

5 Notes on Experiments

1. Higher learning rate such as $\mathbf{LR} = 10^{-5}$ was proven to be effective for learning, but contributed to training instability and model collapse in a few cases and so we decided that a lower learning rate was more appropriate

2. Interestingly, the higher the learning rate the higher the β had to be to mainain the KL-divergence within the same levels. We recommend lowering the β if you also lower the learning rate for similar results as the paper. Further research

3. If l is too high the models starts to pointlessly elongate its responses, often leading to model collapse.

6 Conclusion

Our curriculum training method uses much fewer tokens to train with marginal performance loss. This can contribute to significant cost cutting in the training of reasoning large language models. The effect that reasoning gains occur in significantly shorter reasoning traces is supported by the previously researched and phenomenon that the higher entropy tokens drive the majority of reasoning gains and our observation that the shorter reasoning traces retain the high entropy tokens that would have occurred in their corresponding longer version, dropping mostly the lower entropy tokens.

Limitations: Only tested on a single dataset (GSM8K) Tested for shorter periods of

References

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., ... Zhang, Z. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. https://arxiv.org/abs/2501.12948

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., & Lin, M. (2025). Understanding r1-zero-like training: A critical perspective. https://arxiv.org/abs/2503.20783

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. https://arxiv.org/abs/1707.06347

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. https://arxiv.org/abs/2402.03300

Shenfeld, I., Pari, J., & Agrawal, P. (2025). Rl's razor: Why online reinforcement learning forgets less. https://arxiv.org/abs/2509.04259

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., Liu, Y., Yang, A., Zhao, A., Yue, Y., Song, S., Yu, B., Huang, G., & Lin, J. (2025). Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. https://arxiv.org/abs/2506.01939

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models. https://arxiv.org/abs/2201.11903