ENTROPY SCHEDULING IN REINFORCEMENT LEARNING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We observe that *entropy* in reinforcement learning functions analogously to the *learning rate* in LLMs. Maintaining stable entropy, as demonstrated in DAPO (Yu et al., 2025), helps stabilize RL training, while rapid entropy annealing (i.e., so-called entropy collapse) accelerates local performance improvement and enables faster convergence. We argue that these two processes are not antithetical, but can be effectively controlled and scheduled within a single training run, similar to learning rate scheduling. We propose ***E**ntropy **S**chduling* (ES), which optimizes different pre-set goals (e.g. k in optimizing Pass@k) by controlling and scheduling entropy at each step of the RL process. We find that maintaining stable entropy early in training followed by entropy annealing achieves superior performance. Moreover, since stable-state entropy and annealed entropy exhibit distinctly different learning dynamics, curriculum learning can be seamlessly integrated to maximize model performance based on different entropy phases. We show that entropy scheduling is straightforward to implement and intuitive in design. Extensive experiments suggest that it delivers consistent and stable performance improvements across diverse models and algorithms.

1 INTRODUCTION

The field of reinforcement learning with verifiable rewards (RLVR) for large language models (LLMs) has witnessed rapid advancements, as illustrated by strong reasoning models such as DeepSeek-R1(DeepSeek-AI et al., 2025), Kimi-K1.5(Team et al., 2025), and the OpenAI o-series (OpenAI, 2024; 2025), which demonstrate significant improvements in reasoning capabilities. A critical aspect of RLVR is the *policy entropy*, which quantifies the unpredictability of the policy distribution. High entropy denotes that the policy model samples actions more uniformly with greater randomness, while low entropy suggests the policy is concentrating on a narrower set of actions. Entropy collapse (or entropy annealing), a phenomenon commonly observed during RL for LLMs, occurs when the policy entropy diminishes too rapidly, limiting exploration. For example, vanilla GRPO (Shao et al., 2024), a widely used RL algorithm for large reasoning models, has been shown to cause entropy collapse during RLVR training (Yu et al., 2025). To mitigate this, several strategies have been proposed (Yu et al., 2025; Cui et al., 2025) to maintain entropy at relatively high levels (referred to as stable entropy). Elevated and stable entropy levels are known to encourage extensive exploration, enabling the model to generate diverse reasoning outputs and stabilizing the training process.

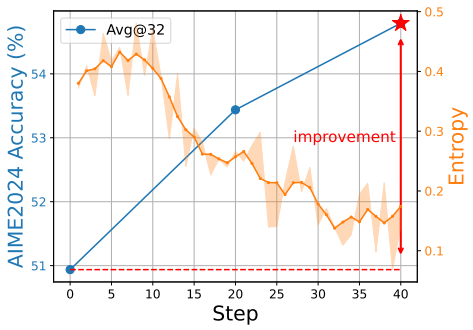


Figure 1: We train the DAPO-32B (Yu et al., 2025) with a lower clip value and the same dataset to achieve entropy annealing, and improve AIME2024 from 50.9 to 54.9 within 40 training steps.

In contrast to traditional perspectives, very recent studies demonstrate that unsupervised learning in RLVR, driven solely by entropy minimization, can significantly enhance model performance within a short, localized training step (Wang et al., 2025c; Agarwal et al., 2025; Gao et al., 2025). As a pilot

study, we continually train and anneal the entropy of `DAPO-Qwen-32B` [1], which is a well trained model ending with high entropy (Yu et al., 2025). As illustrated in Fig. 1, this approach achieved a 4.0% absolute performance improvement in merely 40 training steps.

We reconcile these two seemingly opposing perspectives and optimization methods. We find that entropy resembles the learning rate in LLM training. Maintaining stable entropy contributes to the long-term sustained development of RL training, while subsequent entropy decay or annealing helps the model rapidly improve performance and converge in the short term. As a summary, entropy that stabilizes first and then decays, compared to entropy that decays first and then stabilizes, exhibits weaker performance initially but stronger performance in the end. We refer to this empirical law as *the Parallelogram Law of Entropy*.

On the other hand, we also find that what makes entropy more compelling than the learning rate is that many important hyperparameters in RL (e.g., clip-higher coefficient (Yu et al., 2025), entropy loss coefficient, and KL penalty coefficient) ultimately control the model's performance by managing entropy. In other words, entropy is the (almost only) fundamental factor in altering model performance (like exploration and explanation).

Inspired by learning rate schedule (LRS) techniques, we propose *Entropy Scheduling* (ES) to combine entropy stable stage and entropy annealing stage. Specifically, as a demonstration, we control the entropy with constant, cosine and the stable-annealing schedules. Moreover, we train the model with different stable constant values and different annealing ratios in stable-annealing schedule. We find that there exists an optimal entropy settings (e.g., constant entropy value, annealing ratio, etc.) for different pre-set goals (e.g. k in optimizing Pass@k).

A major challenge in implementing ES is controlling entropy within expected and predefined ranges at each training step. Essentially, entropy is merely an indicator of the model's computation on the current batch of data and cannot be directly regulated. It can only be controlled indirectly by adjusting hyper-parameters (e.g., clip-higher, entropy loss coefficient) several steps to hundreds of steps in advance to keep entropy within a preset range at each training step. As such, we proposed a simple PID (Proportional-Integral-Derivative)-based delay control algorithm (Ang et al., 2005) as a baseline, which effectively achieves our entropy scheduling objectives.

Furthermore, due to the different learning dynamics between the entropy stable and entropy annealing, curriculum learning can be seamlessly integrated to maximize model performance during the entropy annealing phase. We can integrate all factors that contribute to performance improvement into the annealing phase. Specifically, high-quality data can be concentrated in the entropy annealing phase to amplify its advantages. Additionally, increasing the rollout number and extending the maximum response length during the annealing phase proves to be an effective method for boosting model performance.

2 PRELIMINARY

The LLMs act as the policy model π_θ and autoregressively generate output sequence $y = \{y_1, \cdots, y_t, \cdots, y_T\}$ given the prompt x. Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a widely utilized algorithm in RL, known for its stability and efficiency. It employs a clipped surrogate objective function to constrain policy updates within a proximal region relative to the previous policy.

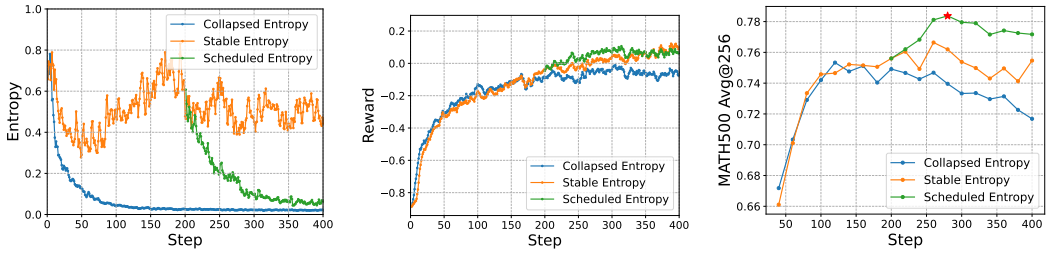
$$\mathcal{J}(\theta) = \mathbb{E}\left[\min\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}\hat{A}_t, \text{clip}\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right)\right] \tag{1}$$

where $\hat{A}_t$ is the advantage at time step t. In this work, we primarily focus on GRPO (Shao et al., 2024), a variation of PPO. GRPO estimates the advantage in a group-relative manner. For each prompt, the policy model samples a group of n responses and estimate the advantage as follows:

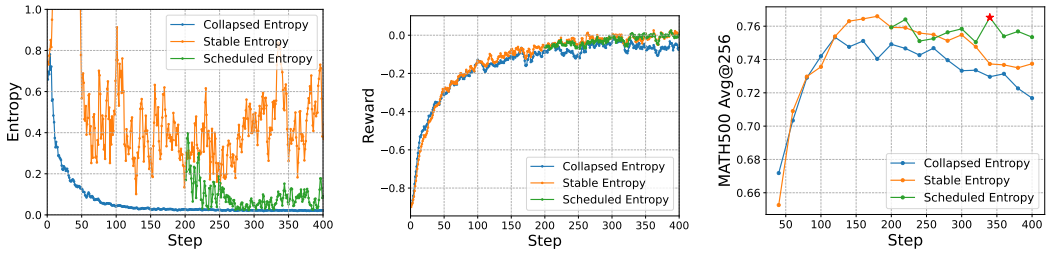
$$\hat{A}_{i,t} = \frac{r_i - \text{mean}\left(\{R_i\}_{i=1}^n\right)}{\text{std}\left(\{R_i\}_{i=1}^n\right)} \tag{2}$$

where $\{R_i\}_{i=1}^n$ is the reward of each generated response.

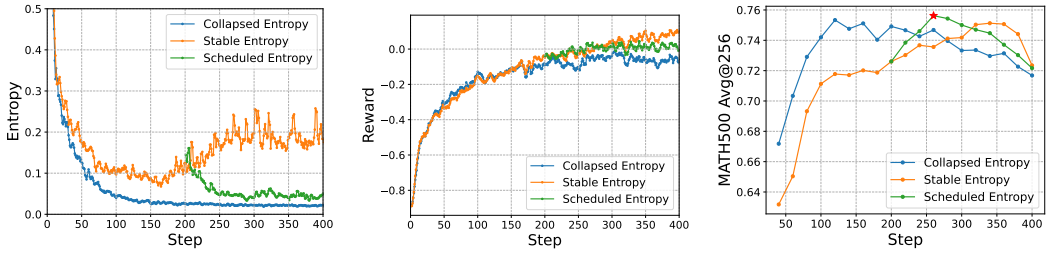
[1] `https://huggingface.co/BytedTsinghua-SIA/DAPO-Qwen-32B`



(a) Entropy, reward and MATH500 avg@256 of stable and scheduled entropy using **clip-higher**.



(b) Entropy, reward and MATH500 avg@256 of stable and scheduled entropy using **entropy bonus**.



(c) Entropy, reward and MATH500 avg@256 of stable and scheduled entropy using **KL penalty**.

Figure 2: We leverage the clip-higher, entropy bonus and KL penalty method to control different stable and scheduled entropy. All these methods or hyperparameters enable control model's reward and performance by managing entropy. The scheduled entropy that stabilizes first and then decays (orange lines) exhibits higher reward and benchmark performance compared with other scheduled entropy, which is referred as the parallelogram law of entropy.

KL penalty is used to penalize the divergence between the online policy and the reference policy (Abdolmaleki et al., 2018; Kappen et al., 2012). The objective function with KL is:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\min\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}\hat{A}_t, \text{clip}\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right) - \beta D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right)\right] \tag{3}$$

The entropy bonus term is used to encourage stochasticity in the optimal policy model and could be incorporated into the objective function:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\min\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}\hat{A}_t, \text{clip}\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right) + \alpha\mathcal{H}\left(\pi_\theta, \mathcal{D}\right)\right] \tag{4}$$

where

$$\mathcal{H}\left(\pi_\theta, \mathcal{D}\right) = -\mathbb{E}_{\mathcal{D},\pi_\theta}\left[\log \pi_\theta\left(y_t \mid \boldsymbol{y}_{<t}\right)\right] \tag{5}$$

Several factors influence the entropy dynamics of policy models during training. DAPO (Yu et al., 2025) claims that clip-higher is an effective method to enhance the policy model entropy and generate more diverse samples. Similarly, entropy bonus directly encourages the policy model to maintain a high entropy value, making it one of the most straightforward methods to promote exploration. Meanwhile, the KL penalty is also an effective method to avoid the entropy collapse which is equiv-

alent to the entropy bonus to some extent (Jaques et al., 2019).

$$\mathcal{J}(\theta) = \mathbb{E}\Bigg[\min\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)} \hat{A}_t, \operatorname{clip}\left(\frac{\pi_\theta\left(y_t \mid y_{<t}\right)}{\pi_{\theta_{\text{old}}}\left(y_t \mid y_{<t}\right)}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_t \right) + \alpha \mathcal{H}\left(\pi_\theta, \mathcal{D}\right) - \beta D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right) \Bigg], \tag{6}$$

3 Pilot Study: The Parallelogram Law of Entropy

3.1 Experiment Setup

We train the models with three kinds of scheduled entropy: (1) entropy annealing from the beginning of training, (2) maintaining entropy stable throughout the entire training process, and (3) introducing entropy annealing only during the final phase of training. For entropy annealing from scratch, we adopt the GRPO method without applying any KL penalty. To achieve entropy stable, we separately employ techniques such as clip-higher, entropy bonus, and KL penalty to maintain the entropy at a relatively high and stable value. Following the stable phase, we revert to the lower clip value or remove the entropy bonus and KL penalty, allowing entropy to anneal naturally from this stable state. We utilize the Qwen-2.5-7B-Base(Qwen et al., 2025) as the initial model and train it on the DAPO-17k(Yu et al., 2025) dataset. For the clip-higher method, we set the clip value to 0.28, consistent with the configuration reported in DAPO. Additionally, we use coefficients of 0.0015 and 0.005 for entropy bonus and KL penalty, respectively. We test the model performance in the MATH500 (Lightman et al., 2023), AIME2024 and AIME2025.

3.2 Parallelogram Law of Entropy

DAPO (Yu et al., 2025) shows that entropy collapse induces performance degrade in the end of training. However, as shown in their paper and our experiments, the entropy collapse also leads to a better performance in the initial training steps, though much worse in the end. As illustrated in Fig. 2, entropy annealing from the beginning of training (blue line) achieves higher rewards and better performance than maintaining entropy stable (orange line) during the initial training steps. It seems to show that, lower entropy exchanges for better performance within limited training steps.

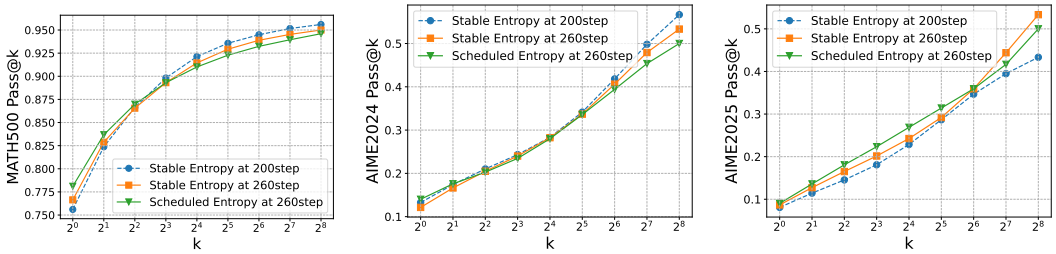
However, as entropy drops to a very low level, the model's performance eventually saturates, with only few improvements observed in later training stages. In contrast, maintaining entropy stable prevents saturation, allowing for steady improvements over time. Consequently, entropy stabilization gradually surpasses entropy annealing from scratch at longer training steps.

Now, it comes with a natural question, if we put the entropy reduction into the last stage of training, would it be beneficial to the final performance? We actually make this experiment that introducing further entropy annealing (green line) from an entropy stable phase in the final training stage. As shown in Fig. 2, this entropy annealing yields rapid improvement in both rewards and accuracy, outperforming models trained with stable entropy in same training steps. This finding aligns with concepts in learning rate annealing, where results in the rapid validation loss drop in the final learning rate annealing training stage.

Therefore, synthesizing these phenomena, here comes a conclusion: the same entropy reduction, is far more effective near the end of training than in the early stages. Figuratively speaking, the entropy annealing from the beginning line (blue line), stable entropy line (orange line) and entropy annealing in the final phase line (green line) form a parallelogram. The performance at the top line of parallelogram is initially worse but ultimately better than the performance of the bottom line of parallelogram. It is worth noting that, we not only adopt the clip-higher strategy to control entropy but also use a more direct entropy bonus and KL penalty term to control entropy, achieving quite similar effects.

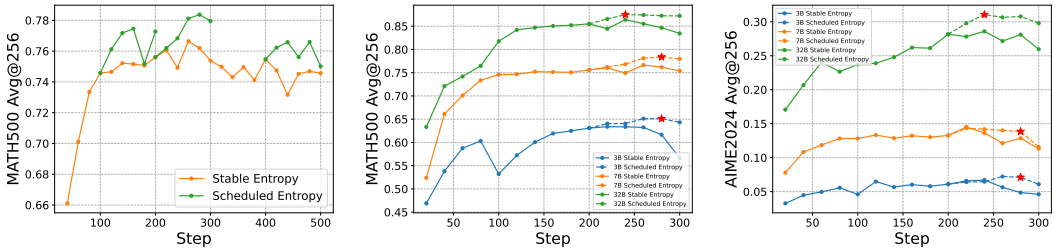
3.3 What happens in entropy annealing

The trade-off between exploration and exploitation is a fundamental concept in reinforcement learning, where entropy serves as a key mechanism to regulate the transition between these two stages.



(a) Pass@k curve of MATH500. (b) Pass@k curve of AIME2024. (c) Pass@k curve of AIME2025.

Figure 3: The Pass@k curves reveal a distinct trade-off between annealing entropy and stable entropy. Specifically, by comparing the stable entropy (orange line) and annealing entropy (green line) with same training steps, we find that entropy annealing sacrifices the Pass@k values for larger k (exploration) to improve the Pass@k values for smaller k (exploitation).



(a) Entropy annealing across training steps in MATH500. (b) Entropy annealing across model sizes in MATH500. (c) Entropy annealing across model sizes in AIME2024.

Figure 4: The benchmark metric of scheduled entropy across longer training steps and larger model sizes. We show that all training steps and model sizes could benefit from entropy annealing.

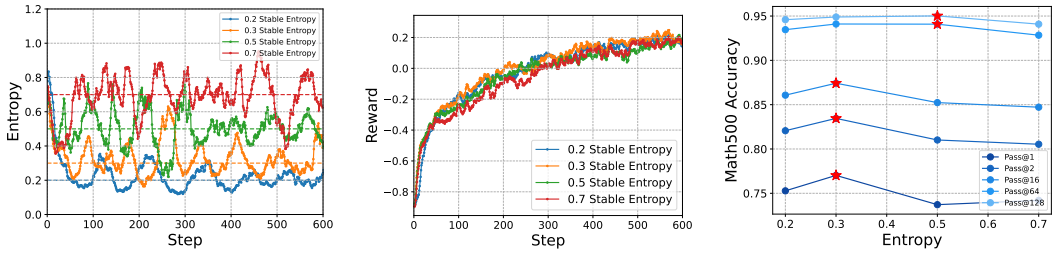
We evaluate the Pass@k curve to quantitatively describe the exploration and exploitation of the policy models. Firstly, as depicted in Fig. 3, the RLVR process will gradually reduce the model's Pass@k score of large k (referred to exploration ability) but increase the model's Pass@k score of small k (referred to exploitation ability) by comparing the stable entropy at different training steps (blue and orange lines). Secondly, we compare the Pass@k curves for entropy stable and entropy annealing under identical training steps and datasets. The entropy annealing achieves higher Pass@k values compared to entropy stable for smaller values of k, but stable entropy exhibits higher Pass@k values for larger values of k. This phenomenon suggests that entropy annealing sacrifices performance on larger k (exploration) to enhance performance on smaller k (exploitation), which explains that the annealing entropy phase achieve a higher reward or Pass@1 score.

3.4 Entropy Annealing from DAPO-32B

DAPO-32B (Yu et al., 2025) model is trained using the clip-higher method, maintaining a high entropy value throughout the entire training process. To further investigate the impact of entropy annealing, we introduce an entropy annealing phase for DAPO-32B by reducing the clipping value from 0.28 to 0.2. As depicted in Fig. 1, applying entropy annealing to DAPO-32B results in a rapid performance improvement on the AIME2024 benchmark, with performance increasing from 50.9 to 54.9 within just 40 training steps using the same training dataset. This result also highlights the effectiveness of putting the entropy reduction or annealing into the last stage of training to achieve a rapid performance improvement over relatively short training durations.

3.5 Scaling on Training Steps and Model Sizes

We validate the effectiveness of entropy annealing across extended training steps and varying model sizes. Specifically, we apply entropy annealing over longer training durations and observe consistent performance improvements as shown in Fig. 4a. Furthermore, we evaluate the impact of entropy



(a) Entropy of constant ES with different values.

(b) Reward of constant ES with different values.

(c) MATH500 of constant ES with different values.

Figure 5: The entropy, reward and performance of different models trained with different constant entropy values. There exist an optimal constant entropy value for different Pass@k objective.

annealing on models of different sizes (3B, 7B, and 32B), maintaining consistent training steps and using the same training dataset across experiments. As illustrated in Fig. 4b and Fig. 4c, all model sizes benefit from entropy annealing, achieving performance improvement. Even larger models, which typically exhibit higher baseline performance, demonstrate a certain degree of improvement, highlighting the generalization of entropy annealing.

4 Entropy Scheduling

Similar to learning rate scheduling, balancing between the entropy stable phase and the annealing phase is essential for achieving optimal performance (Tissue et al., 2024; Wang et al., 2025b). Entropy stable facilitates steady reward improvements, analogous to the behavior of a constant learning rate, while entropy annealing enables rapid performance improvement, akin to learning rate annealing. Based on this analogy, we propose *Entropy Scheduling (ES)* to strategically plan the entropy dynamics in the training. However, there is actually a significant difference between learning rate and entropy. Essentially, different with learning rate as a hyper-parameter, entropy is merely an indicator of the model's computation on the current batch of data and cannot be directly regulated. It can only be controlled indirectly by adjusting hyper-parameters (e.g., clip-higher, entropy loss coefficient), and this impact on entropy does not really take effect immediately until the actor model is considerably changed after some training steps. Inspired by the PID (Proportional-Integral-Derivative) algorithm (Ang et al., 2005) in the field of automatic control, we propose an adaptive clip-higher method as a simple PID algorithm to control the entropy at preset scheduling values. We further conduct experiments using various ES methods, including constant ES, cosine ES, stable with annealing ES and cyclic ES which are all common in the LRS.

4.1 Adaptive Clip-Higher

As shown in Fig. 2, although these there methods (clip-higher, entropy bonus and KL penalty) could effectively control the policy entropy, the entropy bonus and KL penalty are extremely sensitive to the value of the coefficients. Clip-higher is a more proper method to control the policy entropy. However, existing implementations of clip typically apply a fixed clip value throughout the entire training process. During the stable entropy phase, a fixed higher clip value may fail to consistently maintain entropy at a stable level and cannot precisely determine the specific constant entropy value. Similarly, during the entropy annealing phase, using a fixed lower clip will lead to entropy decrease at a fixed rate, limiting flexibility and adaptability.

We propose the *adaptive clip-higher* approach, which is adjusted real-time in order to control entropy as scheduled value at each step. The details of the *adaptive clip-higher* are provided in Algorithm 1. Our method is a simple PID controller that adjusts the control variable (clip value) by computing the target entropy error. Firstly, the policy entropy for each training step is pre-specified according to the different ES strategy similar with LRS. Then, a sliding window is employed to compute the average actual entropy over recent training steps, which is then compared with the averaged pre-specified entropy. Based on the deviation between the actual and pre-specified entropy values, the clip value is adjusted dynamically by adding or subtracting a fixed clip offset.

Algorithm 1 Adaptive Clip-Higher Algorithm

Require: Warm-up steps $T \in \mathbb{N}$, sliding window size $w \in \mathbb{N}$, adjustment step $\delta \in \mathbb{R}^+$, initial clip value $c_0 \in \mathbb{R}^+$, feasible range $[c_{\min}, c_{\max}] \subset \mathbb{R}^+$

Initialize $c_t \leftarrow c_0$, $t \leftarrow 0$
while training continues **do**
  $t \leftarrow t + 1$
  if $t \leq T$ **then**
    $c_t \leftarrow c_0$ ▷ Warm-up phase
  else
    $\mathcal{W}_t \leftarrow \{t - w + 1, t - w + 2, \ldots, t\}$ ▷ Sliding window
    $H_{\text{sched}}(t) \leftarrow \text{EntropyScheduler}(\mathcal{W}_t)$
    $H_{\text{curr}}(t) \leftarrow \frac{1}{w} \sum_{\tau \in \mathcal{W}_t} H(\tau)$
    $\Delta c \leftarrow \begin{cases} +\delta & \text{if } H_{\text{sched}}(t) > H_{\text{curr}}(t) \\ -\delta & \text{if } H_{\text{sched}}(t) < H_{\text{curr}}(t) \\ 0 & \text{otherwise} \end{cases}$
    $c_t \leftarrow \text{clamp}(c_t + \Delta c, c_{\min}, c_{\max})$
  end if
  Execute training step with clip parameter c_t
end while

4.2 Different Entropy Scheduling

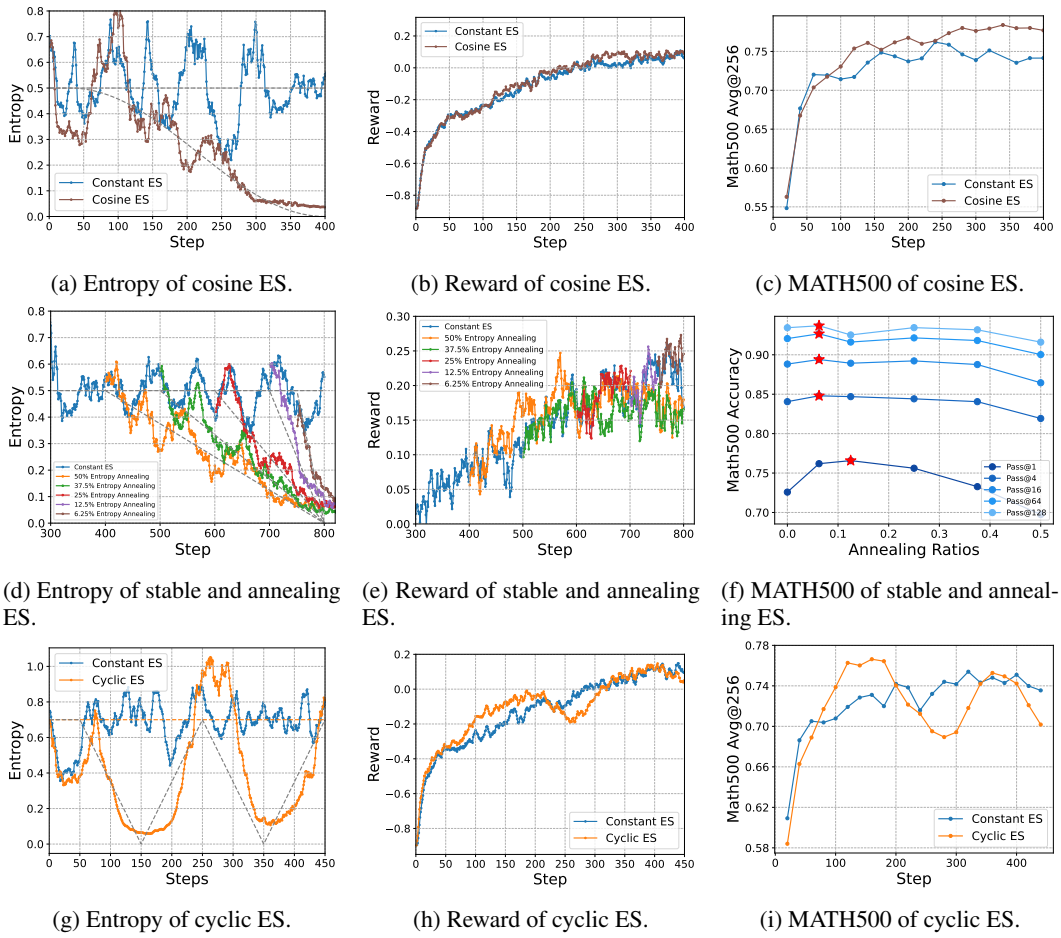
After we have figured out how to control entropy, the next step is find the most appropriate scheduling. We conduct some ES experiments, like constant, cosine, and first stable then annealing schedules which are all most popular LRS in LLM training.

Constant Entropy Scheduling We train the models using constant ES with varying constant entropy values. As shown in Fig. 5a, the adaptive clip-higher method efficiently controls entropy to oscillate around the pre-specified constant values. We observe that different constant entropy values lead to varying levels of performance under the same training steps. Moreover, there exists an optimal constant entropy value for for different pre-set goals (e.g. k in optimizing Pass@k) as shown in Fig. 5c. For the optimization objective of Pass@k with a smaller k, the optimal constant value is small. For the Pass@k with larger k, the optimal constant value gradually increases. Higher constant entropy value encourage the LLMs to generate more diverse actions. This may lead to a lower reward or Pass@1 score, but increases the probability of generating correct answers when more samples are generated. This phenomenon is analogous to the concept of the optimal maximum learning rate encountered in the pre-training of LLMs (Bjorck et al., 2024).

Cosine Entropy Scheduling Cosine LRS (Loshchilov & Hutter, 2016) is one of the most widely used strategies in the pre-training of LLMs. This scheduling approach effectively balances the stable and annealing phases, thereby enhancing overall model performance. Inspired by this, we extend the cosine scheduling concept to ES and evaluate its effectiveness, as shown in Fig. 6. Our results show that the reward and MATH500 avg@256 (both are Pass@1 metric) of cosine ES is initially lower than that of constant ES during the early stages of training. However, in the later training stages, cosine ES surpasses constant ES. This behavior can be attributed to the entropy annealing component inherent in the cosine scheduling approach. Notably, this behavior is similar to relationship between cosine and constant LRS in terms of their impact on validation loss.

Stable and Annealing Scheduling WSD (Hu et al., 2024) is an effective LRS strategy that stabilizes training by maintaining a constant learning rate and then accelerates convergence through a final learning rate annealing phase. We control the entropy similar with WSD, that the entropy remain a stable during most of the training phase and then annealing in the final stage.

As shown in Fig. 6d, our method effectively maintains a high constant entropy during the stable phase and subsequently anneals with varying annealing steps or ratios. By comparing the final performance across different annealing ratios, we observe that there also exists an optimal annealing ratio for different Pass@k optimization objective. As shown in Fig. 6f, the Pass@k for large k tends



(a) Entropy of cosine ES. (b) Reward of cosine ES. (c) MATH500 of cosine ES.

(d) Entropy of stable and annealing ES. (e) Reward of stable and annealing ES. (f) MATH500 of stable and annealing ES.

(g) Entropy of cyclic ES. (h) Reward of cyclic ES. (i) MATH500 of cyclic ES.

Figure 6: The model entropy, reward and MATH500 avg@256 using cosine ES, stable with annealing ES and cyclic ES. Different ES have different training dynamics in reward and performance.

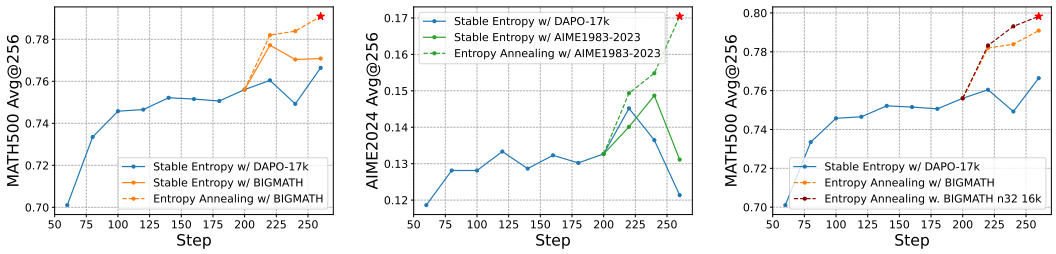
to adapt a smaller annealing ratio. The smaller annealing ratio means maintaining the high entropy for a longer training process to make the model in a state of generating diversity responses, and fewer training steps to trade high Pass@k performance for low Pass@k performance.

Cyclic Scheduling We also train the model using cyclic ES, which involves an initial entropy annealing phase followed by an increase phase. This cyclic approach aims to capture the training dynamics by alternating between low and high entropy levels.

As illustrated in Fig. 6h and Fig. 6i, during the entropy increase phase, model reward or performance improvement slows down and may even experience a decline. This behavior occurs because restoring a high-entropy state increases the model's generation diversity at the expense of exploitation ability. This is consistent with the scenario where an increasing learning rate leads to a rise in validation loss. Finally, the subsequent decrease in entropy resumes model exploitation ability and gets a nearly same performance with constant ES in the end.

5 Entropy Staged Learning

The entropy stable and entropy annealing exhibit the different learning dynamics. The entropy annealing phase could solidify the model's potential for diverse reasoning generation into more stable correct reasoning generation, marking a phase of rapid performance improvement. We could maximize the advantage of entropy annealing through integrating curriculum learning, like introducing more high-quality training data and other useful RL training settings in the annealing phase.



(a) MATH500 w. / wo. entropy annealing using high-quality data.

(b) AIME2024 w. / wo. entropy annealing using high-quality data.

(c) MATH500 w. lager rollout samples and context length.

Figure 7: The model performance of introducing the high-quality data and other useful RL training settings in entropy annealing phase. The entropy annealing could further maximize the advantages of high-quality data, large rollout numbers and long context length.

5.1 High-Quality Training Data

In pre-training and continual pre-training of LLMs, some methods attempt to allocate high-quality data to the learning rate annealing phase to achieve better performance (Grattafiori et al., 2024; Parmar et al., 2024). This strategy is particularly effective because the quantity of high-quality data is often limited and cannot fully support the entire training process. Concentrating such data in the annealing phase can maximize its impact on model performance.

Similarly, we explore leveraging high-quality data more effectively during the entropy annealing phase to amplify its advantages. Specifically, we use AIEM1983-2023 and BigMATH (Albalak et al., 2025) datasets with difficulty levels 0-2 as the high-quality training data. As shown in Fig. 7, we compare the performance of training with high-quality data both with and without entropy annealing. Our results demonstrate that high-quality data alone (without entropy annealing) could lead to better performance compared to the baseline dataset DAPO-17k. Furthermore, incorporating entropy annealing amplifies the effect of the high-quality data, resulting in even higher accuracy.

5.2 Increasing the Rollout number and Context Length

The sample number of per prompt in GRPO and maximum response length are also the important factors in RLVR training to affect the model performance. However, under limited computational resources, consistently maintaining a large rollout samples and long response length is challenging. In the continual pre-training in LLMs, the model will enlarge context length after adequate pre-training with short context. Meanwhile, gradually increasing response length is also a common method in the RL training (Luo et al., 2025).

Inspired by the better learning dynamics of entropy annealing, we could only adapt larger rollout samples and long response length in the short annealing stage. We enlarge the rollout samples from 8 to 32 and response length from 4k to 16k. As shown in Fig. 7c, we compare the original and larger rollout samples and response length in the entropy annealing stage and observe the further performance improvement.

6 Conclusion

In this work, we discovered the parallelogram law of entropy, revealing that entropy reduction is more effective at the end of RL training than in early stages, leading us to propose entropy scheduling (ES) as a simple yet powerful technique analogous to learning rate scheduling. By drawing parallels between entropy and learning rate—where high entropy/learning rate encourages global exploration while low entropy/learning rate promotes local convergence—we developed adaptive control mechanisms to maintain scheduled entropy throughout training. Our experiments demonstrate that strategic entropy scheduling, particularly with late-stage entropy reduction, can significantly improve model performance (e.g., from 50.9% to 54.9% on AIME2024 in just 40 steps), and different entropy schedules can be tailored for specific optimization goals like Pass@k, establishing entropy scheduling as an intuitive and effective approach for enhancing RL training in large language models.

Reproducibility Statement

Our experiments are conducted using publicly available datasets and open-source models. We provide detailed experimental setups, including hyperparameter configurations, training framework, and implementation details, in Appendix A.

References

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

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

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025. URL https://arxiv.org/abs/2502.17387.

Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal lr across token horizons. *arXiv preprint arXiv:2409.19913*, 2024.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, et al. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025.

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

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. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/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, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,

Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao

Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024. URL https://arxiv.org/abs/2404.06395.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87:159–182, 2012.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

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

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

OpenAI. Introducing openai o1. https://openai.com/o1/, 2024.

OpenAI. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, 2025.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models, 2024. URL https://arxiv.org/abs/2407.07263.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/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. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

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

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang,

Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Howe Tissue, Venus Wang, and Lu Wang. Scaling law with learning rate annealing, 2024. URL https://arxiv.org/abs/2408.11029.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025a.

Xingjin Wang, Howe Tissue, Lu Wang, Linjing Li, and Daniel Dajun Zeng. Learning dynamics in continual pre-training for large language models, 2025b. URL https://arxiv.org/abs/2505.07796.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv: 2504.20571*, 2025c.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

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

A Experimental Setup

We use the Qwen-2.5-Base (Qwen et al., 2025) as the initial model and use DAPO-17k (Yu et al., 2025) as training data. The training batch size is 512 and we samples 8 responses per prompt with a temperature 1. The mini batch size is 32 and we performance 16 policy updates in each training batch size. The max response length in the training is 4096. We use verl RL framework (Sheng et al., 2024) for all the training. We use the rule-based reward same with the DAPO (Yu et al., 2025):

$$R(\hat{y}, y) = \begin{cases} 1, & \text{equivalent}(\hat{y}, y) \\ -1, & \text{otherwise} \end{cases} \tag{7}$$

where y is the ground-truth answer and $\hat{y}$ is the predicted answer.

In the evaluation, we also use temperature of 1 to evaluate the test accuracy. Specifically, for each problem x_i, we generate $n = 256$ samples and count the number of correct samples as c_i. We compute low-variance and unbiased Pass@k estimation:

$$\text{Pass}\,@k := \mathbb{E}_{x_i \sim \mathcal{D}} \left[1 - \frac{\binom{n-c_i}{k}}{\binom{n}{k}} \right] \tag{8}$$

B Related Work

Reinforcement Learning for LLM Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for enhancing the reasoning capabilities of large language models (Chen et al., 2025; Hu et al., 2025; He et al., 2025). DeepSeek directly uses GRPO with verifiable rewards to obtain the DeepSeek-R1 (DeepSeek-AI et al., 2025). Some recent works point out the shortcomings in the PPO(Schulman et al., 2017) or GRPO (Shao et al., 2024) algorithms, like training instability (Yu et al., 2025), reward noise (Liu et al.) and model collapse in Mixture-of-Experts (MoE) RL training (Zheng et al., 2025). We mainly focus on the entropy collapse in the RLVR training and propose the entropy scheduling to combine the stable entropy and annealing entropy to achieve better performance.

Entropy in Reinforcement Learning Policy entropy is an important metric in the RLVR which could quantify the unpredictability of the policy distribution. DAPO (Yu et al., 2025) points out that the entropy collapse happened in the GRPO (Shao et al., 2024) and proposes to use a higher clip value to avoid the collapse. Some recent works (Cui et al., 2025; Wang et al., 2025a) also explain the mechanism behind entropy dynamics and propose some token-level methods to avoid the entropy collapse. Cheng et al. (2025) proposes the correlations between entropy and exploration and introduces a novel entropy related advantage function. We also uncover that entropy could control the exploration and explanation of policy models through measuring Pass@k metric. What's more, we point out that the best entropy dynamics should combine stable and annealing entropy and propose the entropy scheduling which is similar with learning rate scheduling in the RL training process.

C Response Length

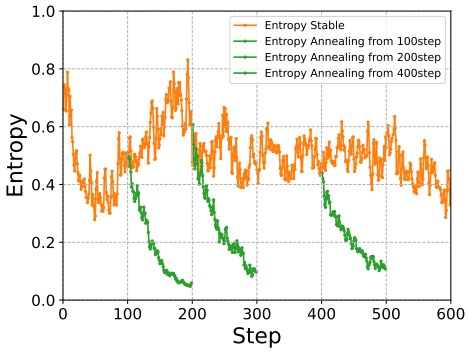
We also compare the response length of stable entropy and annealing entropy phase. As shown in Fig. 8, the response length will also increase in the entropy annealing phase, which explains the performance improvement to a certain extent. What's more, the different constant ES corresponds to different response length.

D Additional Experiments

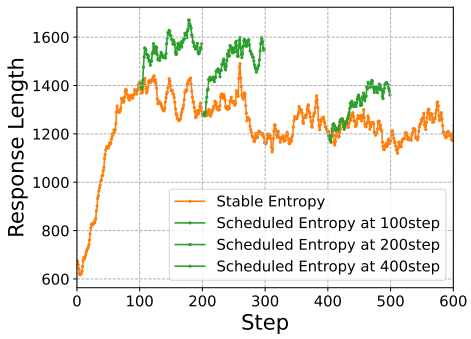
We also use other training datasets and other training features to validate the generality of ES. We use the `OpenR1-Math` [2] as the training dataset and leverage all other training settings or features

[2] `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`

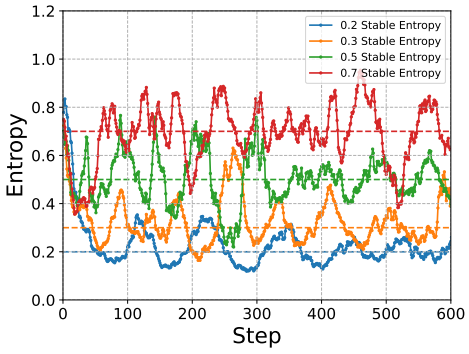
in the DAPO (Yu et al., 2025) including token-level loss, dynamic samples and overlong reward to train the model. As shown in Fig. 9, the ES could effectively apply to other training dataset and training features.



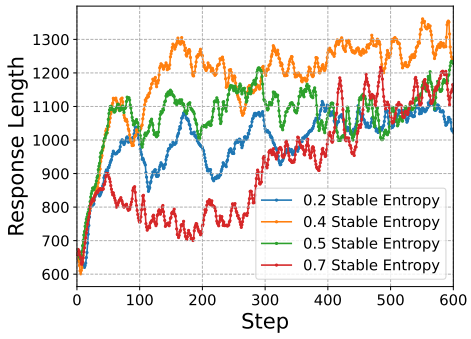
(a) Entropy of annealing from different training steps.



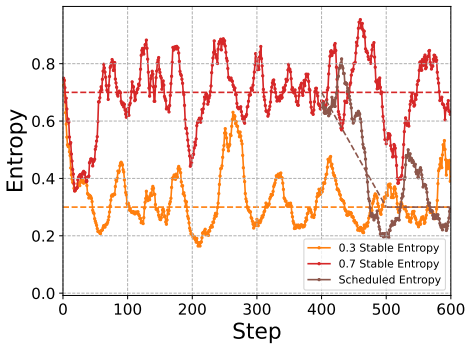
(b) Response length of annealing from different training steps.



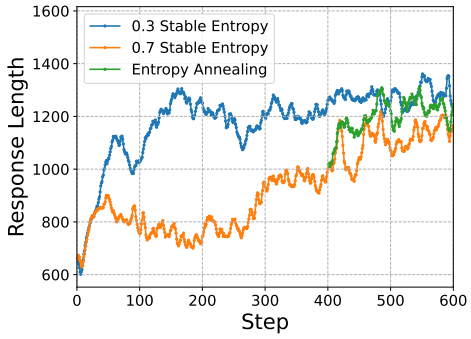
(c) Entropy of different constant ES.



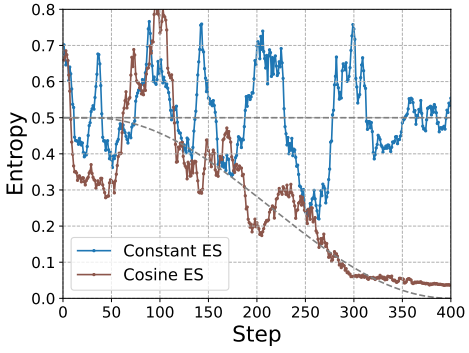
(d) Response length of different constant ES.



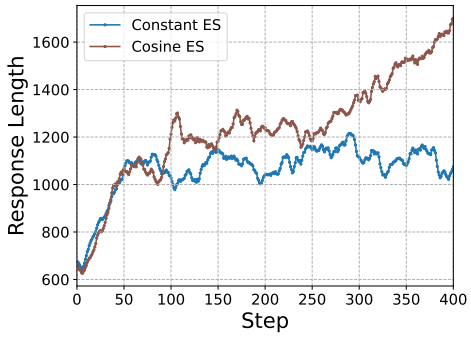
(e) Entropy of annealing from large constant to small constant.



(f) Response length of annealing from large constant to small constant.

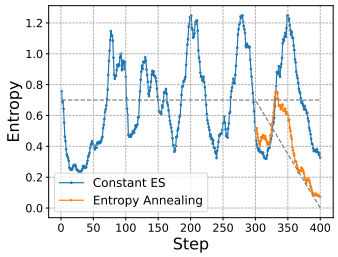


(g) Entropy of constant and cosine ES.

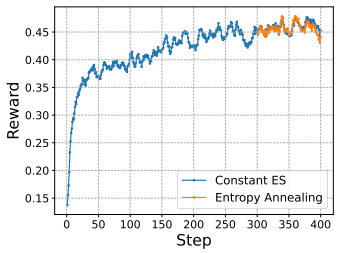


(h) Response length of constant and cosine ES.

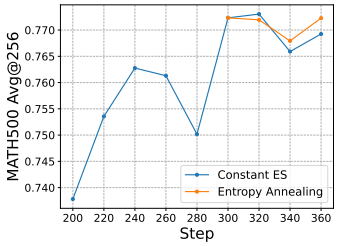
Figure 8: The dynamics of response length in the entropy annealing and different ES.



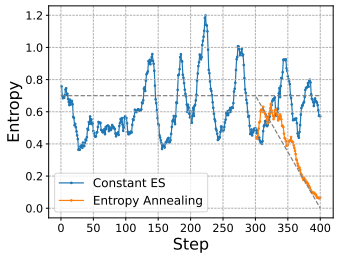
(a) Entropy of training with OpenR1-Math.



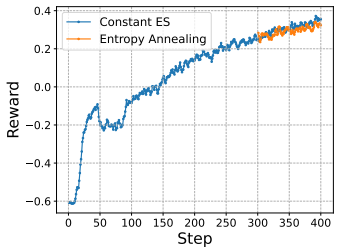
(b) Reward of training with OpenR1-Math.



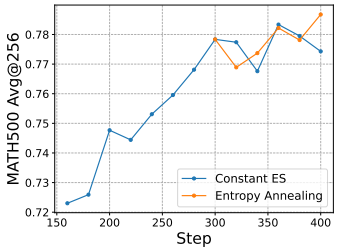
(c) MATH500 avg@256 of training with OpenR1-Math.



(d) Entropy of training with DAPO features.



(e) Reward of training with DAPO features.



(f) MATH500 avg@256 of training with DAPO features.

Figure 9: The entropy, reward and math500 avg@256 of entropy scheduling training with OpenR1-Math dataset and other DAPO RL training features except for clip-higher.