DLER: Doing Length pEnalty Right - Incentivizing More Intelligence per Token via Reinforcement Learning

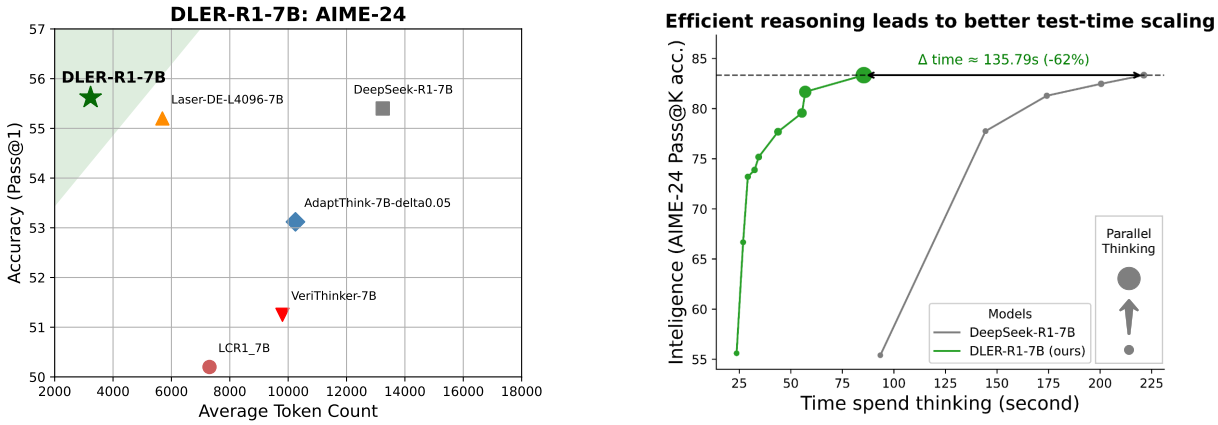
Shih-Yang Liu[1], Xin Dong[*], Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng[1], Yejin Choi, Jan Kautz, Pavlo Molchanov

NVIDIA

Reasoning language models such as OpenAI-o1, DeepSeek-R1, and Qwen achieve strong performance via extended chains of thought but often generate unnecessarily long outputs. Maximizing intelligence per token—accuracy relative to response length—remains an open problem. We revisit reinforcement learning (RL) with the simplest length penalty—truncation—and show that accuracy degradation arises not from the lack of sophisticated penalties but from inadequate RL optimization. We identify three key challenges: (i) large bias in advantage estimation, (ii) entropy collapse, (iii) sparse reward signal. We address them with **D**oing **L**ength p**E**nalty **R**ight (**DLER**), a training recipe combining batch-wise reward normalization, higher clipping, dynamic sampling, and simple truncation length penalty. DLER achieves state-of-the-art accuracy–efficiency trade-offs, cutting output length by over 70% while surpassing all previous baseline accuracy. It also improves test-time scaling: compared to DeepSeek-R1-7B, DLER-7B generates multiple concise responses in parallel with 28% higher accuracy and lower latency. We further introduce Difficulty-Aware DLER, which adaptively tightens truncation on easier questions for additional efficiency gains. We also propose an update-selective merging method that preserves baseline accuracy while retaining the concise reasoning ability of the DLER model which is useful for scenarios where RL training data is scarce.

Models on Hugging Face: Reasoning Token Efficiency

1. Introduction



(a) DLER Training on DeepSeek-R1-7B

(b) DLER-R1-7B enable better test-time scaling

Figure 1 | (a) DLER achieves state-of-the-art accuracy/length trade-offs, shortening CoT by up to 70% without losing accuracy. (b) On AIME-24, DLER-R1 models enable better test-time scaling. Results for 1.5B models are shown in Fig. 10 and Fig. 11 in the appendix.

Reasoning models such as OpenAI-o1 [1], DeepSeek-R1 [2], and Qwen [3] achieve strong performance through long chains of thought (CoT) [4], but this comes at the cost of heavy token usage, higher latency, and redundant outputs for questions solvable with shorter responses. Therefore, how to maximize intelligence per token remains an open research question. Recent work has addressed the inefficiency of extended

[*] project lead.
[1] affiliated with HKUST. Work done during Shih-Yang's internship at NVIDIA Research.

reasoning by developing methods to reduce output length. These approaches fall into three categories: prompt engineering [5], supervised fine-tuning [6, 7, 8, 9, 10], and reinforcement learning (RL) [11, 12, 13, 14, 15]. Among these, RL-based methods have emerged as the most principled approach for achieving optimal accuracy-efficiency trade-offs. These methods typically incorporate length penalties into the reward function to incentivize reasoning within predefined token budgets. However, despite demonstrating substantial reductions in reasoning length, existing approaches often suffer from accuracy degradation that varies significantly across tasks of different complexity levels—a limitation we hypothesize stems from suboptimal optimization techniques.

In this work, **we revisit reinforcement learning (RL) for reasoning efficiency by re-examining the simplest length penalty—truncation**, which assigns zero reward to responses exceeding a fixed limit. Prior RL methods using truncation often fail to recover accuracy; we find this stems not from the length penalty itself but from sub-optimal RL optimization techniques. Three issues drive this degradation: (i) biased advantage estimation under Group Relative Policy Optimization (GRPO) [16] due to substantial reward noise, especially in early state of training, as many responses are abruptly cut off and assigned zero reward, (ii) persistent entropy collapse that hampers exploration of diverse reasoning paths, and (iii) sparse reward signals arising from a large portion of prompts in each batch where all rollouts are truncated and thus assigned zero reward. We address these by adopting batch-wise normalization [17], higher clipping thresholds [18], to promote exploration via low-probability, high-entropy tokens, and curriculumized filtering to gradually introduce harder prompts [18].

By combining all essential elements, our final training recipe, ***D**oing **L**ength p**E**nalty **R**ight (**DLER**)*, achieves state-of-the-art accuracy-to-token efficiency. As illustrated in Fig. 1a, DLER fully recovers the accuracy drop while reducing the average response length by over 70%. This underscores key insights:

Key Insight 1: It is not the sophisticated design of the length penalty that determines performance, but rather the choice of RL optimization algorithm. Even the simplest length truncation can achieve state-of-the-art accuracy-to-token efficiency when combined with our DLER recipe. See Sec. 4.2 for more details.

Key Insight 2: We additionally apply DLER recipe to a variety of length penalties and find that they no longer push the frontier of accuracy-efficiency frontier, but instead serve as tools for fine-grained adjustment of the trade-offs. See Sec. 4.5 for more details.

We also benchmark test-time scaling by generating multiple responses in parallel. Fig. 1b shows that our DLER-R1-7B delivers significant improvements over the original DeepSeek-R1-7B, achieving a 27% accuracy gain (AIME-24) within the same wall-clock "thinking time." This marks an important shift in perspective: whereas recent efforts [2, 19, 18] to enhance reasoning ability have largely pursued accuracy through increasingly long reasoning traces, our findings demonstrate that:

Key Insight 3: Improving reasoning efficiency not only lowers the cost of single response but also enables superior test-time parallel scaling. See Sec. 4.4 for more details.

We further propose a difficulty-aware variant of DLER (**DA-DLER**), where the truncation target length is dynamically adjusted according to an estimate of the model's ability to solve the question. For questions that the model can already reliably answer within the target length, the target length is further shortened to encourage even shorter reasoning, while more challenging questions are allowed more tokens. DA-DLER can achieve an additional 15% and 11% reduction in response length on DeepSeek-R1-1.5B and 7B, respectively, further advancing the efficiency frontier.

We also provide a solution for practical scenarios where accessing the original RL training dataset is not feasible. Often, applying small-scale academic RL training datasets to proprietary models leads to accuracy degradation, and length penalties exacerbate this issue [12] while remaining effective at reducing output length. To completely mitigate accuracy degradation without access to the original dataset, we adopt an update-selective weight merging strategy that combines the original baseline model with the DLER-trained model. This method recovers all lost accuracy while still reducing output tokens by 47%.

Key Insight 4: Weight merging allows for a better trade-off between accuracy and length reduction especially when accessing the original high-quality proprietary datasets is not an option. See Sec. 4.6 for more details.

In summary, our contributions are as follows:

- We revisit the simplest length penalty—truncation—and identify the fundamental factors behind the poor accuracy reported in previous studies - the inadequate RL optimization techniques.
- Building on these insights, we carefully integrate effective RL optimization techniques to address these issues. This allows us to achieve state-of-the-art accuracy-to-length ratios using only the simplest truncation penalty, significantly outperforming prior methods that rely on more complex length penalties for enhancing the reasoning efficiency of DeepSeek-R1-1.5B and 7B. Our findings reveal a key takeaway: improvements in the accuracy–efficiency trade-off are influenced less by the design of the penalty function and more by the choice of optimization algorithm.
- We introduce a difficulty-aware extension of DLER that adaptively adjusts truncation length based on model capability, shortening truncation length for easy questions and relaxing them for harder ones, achieving an additional 15% and 11% reduction in response length on DeepSeek-R1-1.5B and 7B.
- RL training with small-scale datasets cuts response length by 55% but causes an accuracy drop on some tasks due to insufficient data. We address this with a update-selective weight merging strategy that recovers the accuracy while still being able to reduce the average output length by 47%, offering a training-free path to accurate and efficient reasoning models without high-quality proprietary data.

2. Preliminary

Reinforcement learning is widely applied to enhance the reasoning ability of modern LMs [20, 21], with GRPO [16] becoming popular for its efficiency in removing the critic model and using group-relative advantage estimation. This approach maintains token-level advantage estimation accuracy while significantly reducing the overhead. Specifically, for each question-answer pair (q, a), the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G responses $\{o_i\}_{i=1}^{G}$. The advantage for the i-th response is then computed by normalizing the group-level rewards $\{R_i\}_{i=1}^{G}$ as:

$$A_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})} \tag{1}$$

and the optimization objective is formulated as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim D,\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\left(s_{i,t}(\theta)\, A_{i,t},\ \text{clip}(s_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, A_{i,t}\right) \right] \tag{2}$$

where $s_t(\theta) = \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})}$ and ϵ is the clipping threshold. We omit the KL Loss for simplicity. Recent studies [12, 11, 14, 15], aiming to enhance reasoning efficiency, commonly adopt GRPO as the optimization algorithm, coupled with custom-designed length penalty rewards. Under this setup, the new reward R_i' is generally formulated as: $R_i' = R_i + L_i$ where R_i denotes the correctness reward, computed using rule-based heuristics, and L_i represents the length penalty. Depending on the specific length penalty, L_i may either impose a larger penalty on longer outputs or, in the case of the simple truncation penalty, assign a reward of zero to any response that exceeds a predefined length limit.

3. Re-examining the Simplest Length Penalty - Truncation

Prior studies that aim to enhance training efficiency tend to treat the underlying policy optimization algorithm as a fixed, reliable component, often attributing improvements in accuracy-to-length ratio primarily to the design of length penalties. However, this overlooks the possibility that the optimization algorithm itself may

introduce performance bottlenecks. In this work, we re-examine the structure of the policy optimization objective by adopting the simplest possible length penalty—truncation—which is simple enough to alleviate reward hacking and enables a focused analysis of how the optimization algorithm alone affects accuracy degradation.

3.1. More Aggressive Truncation Leads to Higher Reward Variance

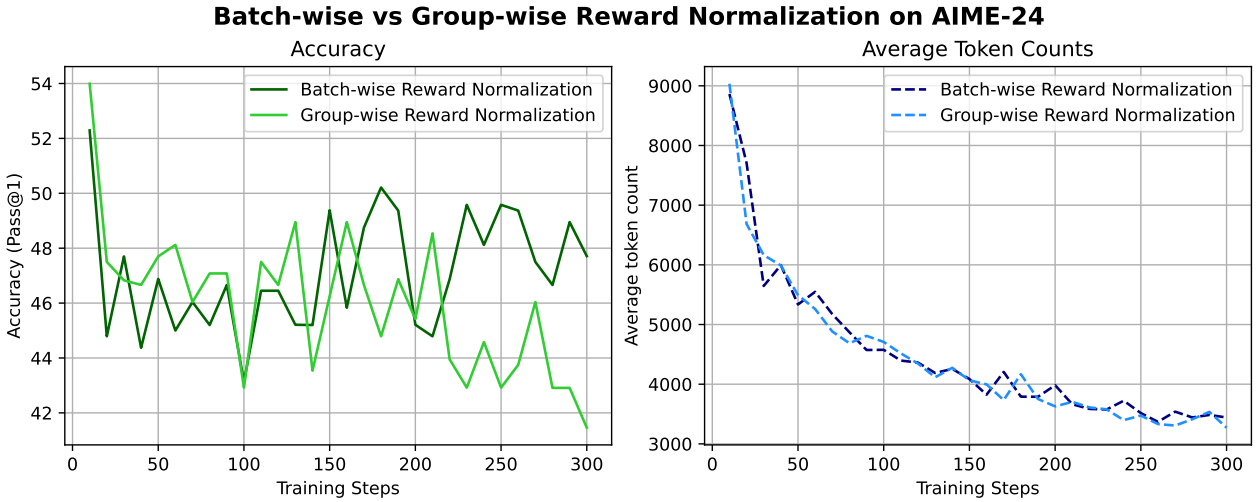


Figure 2 | Accuracy and average response length of DeepSeek-R1-7B on the AIME-24 test set, evaluated every 10 training steps across two RL training runs: group-wise reward normalization (GRPO) and batch-wise normalization. GRPO shows declining accuracy while batch-wise reward normalization remains stable despite reduced token counts.

First, we find that truncation greatly increases reward variance. To assess this, we calculate the advantage variance of DeepSeek-R1-7B at step 0 using 16 rollouts per question, and average the results over a batch of 512 questions drawn from the DeepScaleR-Preview-Dataset [22]. Testing truncation lengths of {4000, 8000, 12000, 16000}, we observe a clear pattern: more aggressive truncation lengths lead to higher per-prompt variance on average, with corresponding values of {0.4, 0.32, 0.3, 0.29}. These results suggest that truncation introduces greater training instability by increasing advantage variance, which in turn leads to more biased advantage estimates according to Equ. 1—a topic we elaborate on further in the Appendix.B.

To address the increased bias introduced by truncation, we propose replacing GRPO's prompt-wise advantage normalization with global batch-wise advantage normalization, which is also used in [17]. The idea is that by normalizing the advantages across the entire batch, we can mitigate the impact of outliers and ensure a more stable estimation of advantage variance. The advantage calculation for the i-th response after adopting batch-wise reward normalization becomes:

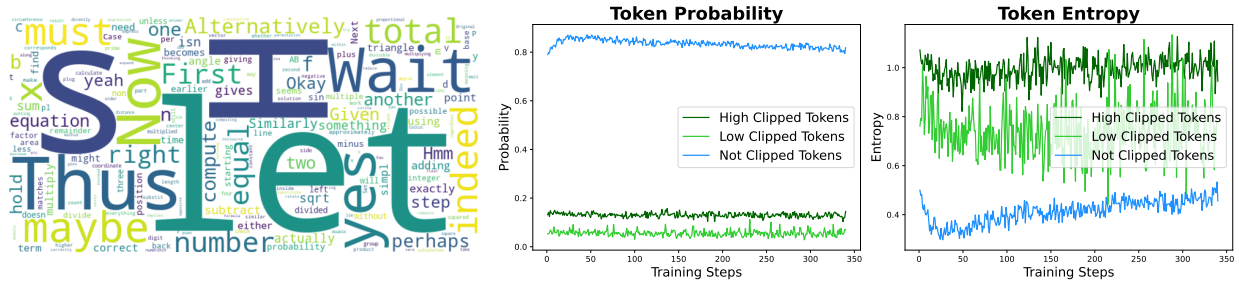
$$A_{i,t}^{\text{norm}} = \frac{A_{i,t} - \text{mean}_{\text{batch}}(A_{i,t})}{\text{std}_{\text{batch}}(A_{i,t})} \tag{3}$$

where $A_{i,t} = R'_i - \text{mean}(\{R'_i\}_{i=1}^{G})$ and we can see that the normalization is now done on batch level rather than local group level.

We compare GRPO and batch-wise reward normalization by training DeepSeek-R1-7B on the DeepScaleR-Preview-Dataset and evaluate every 10 steps on AIME-24; see Sec. 4.1 for full experimental details. As illustrated in Fig. 2, both GRPO and batch-wise reward normalization progressively shorten the average output length over training. However, GRPO shows a continuous drop in accuracy, whereas batch-wise reward normalization begins to recover accuracy after approximately 100 steps and can improve the accuracy of GRPO by around 3% using approximately the same number of tokens. This supports our earlier observation that GRPO's instability under length truncation leads to degraded performance, while switching to batch-wise reward normalization helps stabilize training and restore accuracy.

Key Finding 1: Length truncation increases reward variance, leading to biased advantage estimates and degraded performance when using GRPO. Switching to batch-wise reward variance estimation mitigates this issue and improves performance.

3.2. Entropy Collapse Limits Exploration of Reasoning Paths



(a) (b)

Figure 3 | **(a)** Word clouds of the most frequent tokens clipped by the high-threshold $(1+\epsilon)$ before applying higher clipping threshold,showing that many are transitional words important for reasoning, and clipping them limits exploration during RL training.. **(b)** Average probability and entropy of tokens not clipped, clipped by the higher threshold, and clipped by the lower threshold during RL training. Clipped tokens have much lower probabilities than unclipped ones, and those clipped by the higher threshold consistently show higher entropy, supporting Fig. 3a that these are often high-entropy transitional tokens triggering reasoning steps.

Although switching from group-wise reward normalization to the batch-wise reward normalization improves accuracy, it still falls short of fully restoring the reasoning model's performance. We observe that the current policy optimization still suffers from entropy collapse—a phenomenon identified in recent studies [18, 19] as detrimental to learning. When entropy collapses, the model's output distribution becomes overly concentrated, causing the policy to prematurely focus on a narrow set of responses. This limits exploration, introduces bias in policy updates, and ultimately stalls training progress.

We suspect that the entropy collapse issue may be caused by the clipping on the importance sampling ratio in Eq. 2. If one token is clipped, its gradient is effectively zeroed out, preventing it from contributing to the policy update. So, we are interested in understanding which tokens are most frequently clipped during training, and whether these tokens play a crucial role in exploration of either correct reasoning paths or controlling lengths.

We visualize word clouds of the clipped tokens, as shown in Fig. 3a. Even they are only about 1% of the total tokens, these tokens are largely composed of words such as "Wait," "Hmm," "Alternatively" (signaling contrast or shifts), and "thus" or "also" (indicating progression or causality), which often function as transitional cues in the model's reasoning paths [23]. Moreover, these tokens play an important role in determining response length, since a higher frequency of such tokens generally corresponds to longer reasoning sequences.

More interestingly, we find there is a big overlap between low-probability tokens, high-entropy tokens, and clipped tokens. In Fig. 3b, we plot the average token probability and entropy for three groups: 1) tokens that are not clipped, 2) tokens clipped by the high-threshold $(1+\epsilon)$, and 3) those clipped by the low-threshold $(1-\epsilon)$. We find that clipped tokens—especially those clipped by the upper threshold—tend to have lower probabilities and higher entropy simultaneously. This finding also connects studies on low-probability tokens [24] and high-entropy tokens [23] and provides an unified perspective to understand previous different findings on token importance for optimization.

By decoupling the lower and upper thresholds—previously set to the same value—and assigning a larger value to the upper threshold (ϵ_{high}), the gradient update on those high entropy exploratory tokens are retained, allowing their gradients to propagate and fostering more diverse reasoning behaviors during training.

We conduct experiments on DeepSeek-R1-7B comparing batch-wise reward normalization with and without the higher clipping strategy, under a truncation target length of 4000. We visualize the corresponding average

response length and average entropy in the training batch throughout training in Fig. 6a and Fig. 6b, where we observe that enabling higher clipping threshold clearly alleviates entropy collapse: the entropy of the model's output distribution not only avoids vanishing, but even increases after an initial drop—contrasting with the behavior observed in the original DAPO [18] and ProRL [19] paper without the truncation length penalty.

Key Finding 2: The clipped tokens are often low-probability, high-entropy tokens that play a crucial role in exploration of reasoning paths and length control. Adopting a higher clipping threshold helps retain these tokens in gradient updates, thereby mitigating entropy collapse.

3.3. Length Penalty Over-sparsify Training Signal

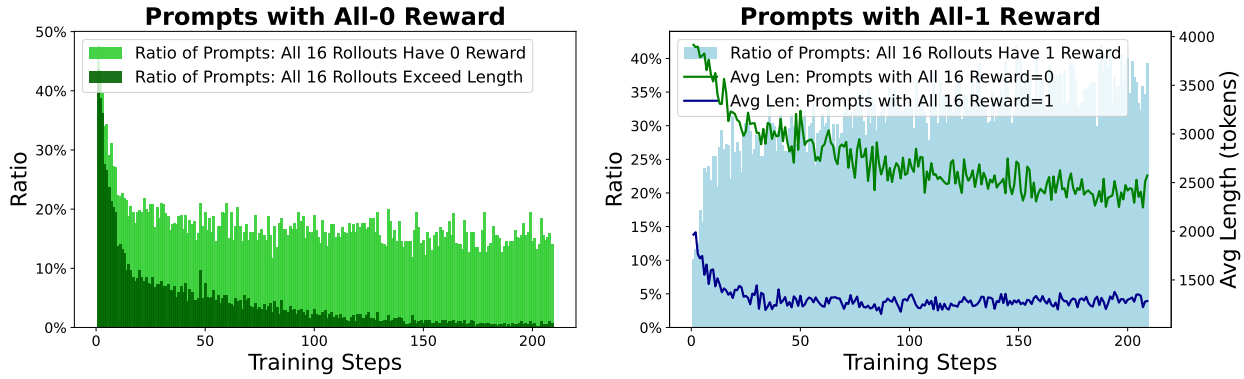


Figure 4 | **Left**: Ratio of training prompts with all 16 rollouts receiving zero reward, including those caused by exceeding the truncation length. Around half of the prompts fall into this category early in training, weakening the signal and biasing the model toward easier prompts that model already know how to solve within the target length. **Right**: Ratio of training prompts with all 16 rollouts receiving reward score of one steadily increases, while average response length declines and remains markedly shorter than that for prompts whose rollouts all receive a reward of zero.

Another phenomenon we identify with the application of length penalty is the prevalence of zero-reward signals across training rollouts. Specifically, a substantial fraction of the prompts receive zero reward for all 16 rollouts, primarily because all 16 responses exceed the target length.

As illustrated in Fig. 4, at the start of training nearly half of all prompts are affected by this issue. Such sparse and noisy feedback biases the model toward shorter, easier prompts it can already solve, thereby reducing exploration and constraining effective learning on improving accuracy.

On the other hand, in the middle and later phases of training, the proportion of prompts with all rollouts receiving positive reward increases their domination, consisting of near 40% of the batch as shown in Fig. 4. These prompts are typically easier, allowing the model to consistently generate quite short responses to answer them correctly. However, when such prompts dominate, the model overfits to over-shorter responses and fails to fully utilize the target length budget. This explains why a suboptimal RL training run with this issue (Fig. 6b) plateaus at a very short response length of 2k tokens despite the maximum length being set to 4k tokens, as the model has prematurely overfitted to these easy prompts.

In summary, both challenging prompts that receive all zero rewards and easy prompts that receive all positive rewards can skew the training distribution, leading to suboptimal learning outcomes. To mitigate this issue, we discard prompts whose rollouts all yield zero reward or all yield positive reward and resample until the target batch size is reached [18]. This dynamic sampling strategy implicitly induces a curriculum, as it progressively incorporates harder examples that initially demand longer reasoning chains to solve. As evident from the training dynamics shown in Fig. 6b and 6c, the model autonomously learns to rapidly reduce token usage in the early training stages, then gradually increases length to fully exploit the target token budget.

This behavior is driven purely by the simple truncation length penalty. Without dynamic sampling, the model tends to plateau at a shorter length, as shown in Fig. 6b, indicating that it has prematurely overfitted into a suboptimal local minimum.

Key Finding 3: Dynamic sampling filters out prompts either too easy or too hard and responses either too long or too short, either of which can over-dominate the training batch and skew the reward shape. This leads to a more balanced training signal and enables the model to better utilize the target length budget.

3.4. Combining All Ingredients: Do Length pEnalty Right

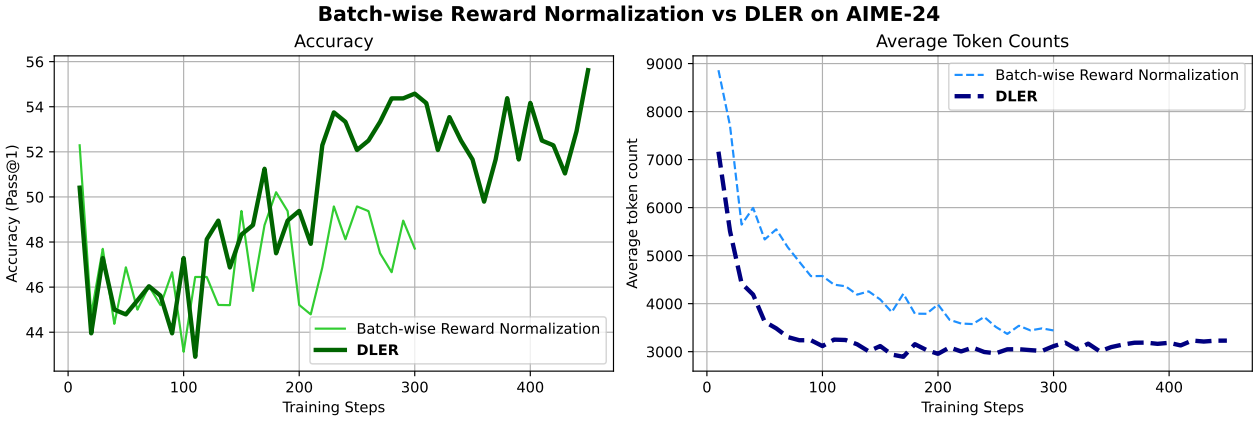
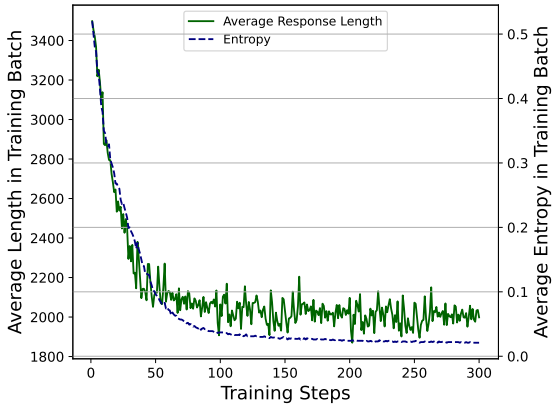


Figure 5 | Accuracy and average response length of DeepSeek-R1-7B on the AIME-24 test set, evaluated every 10 training steps across two RL training runs: one with batch-wise reward normalization and the other with our DLER recipe. DLER enhances the accuracy–token efficiency by reducing token usage while fully recovering the accuracy loss of applying plain batch-wise reward normalization.

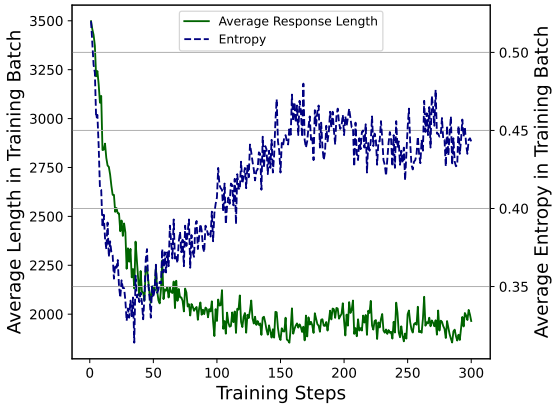
Building on our findings, we unify batch-wise reward normalization, a higher policy update clipping threshold, dynamic sampling to remove instances lacking balanced training signals, and a simple length truncation penalty into a comprehensive training recipe, which we term **DLER** (**D**oing **L**ength p**E**nalty **R**ight). We apply DLER to train DeepSeek-R1-7B for 450 steps and compare its trajectory of accuracy and response length on AIME-24 (Fig. 5) and training dynamics (Fig. 6). All ingredients work complementary to each other. Together, these components systematically address the optimization challenges identified in previous sections and synergistically contribute to DLER's superior performance.

3.5. Difficulty-Aware DLER

We present a difficulty-aware extension of DLER (DA-DLER) that improves efficiency by adaptively assigning truncation lengths according to question difficulty, yielding greater redundancy reduction than a fixed truncation scheme. Question difficulty is estimated from the correctness ratio of model responses, and truncation targets are dynamically adjusted across difficulty tiers. Concretely, for a question q with a sampled response set G, the correctness ratio is defined as the fraction of correct responses in G. Each question is then categorized into one of n difficulty levels $\{d_i\}_{i=1}^{n}$, each associated with a truncation length $\{l_i\}_{i=1}^{n+1}$. For instance, if the correctness ratio lies between d_i and d_{i+1}, the truncation length l_i is applied. DA-DLER pushes the boundary of reasoning efficiency by encouraging the model to solve questions it already handles reliably with even fewer reasoning steps, reducing unnecessary token usage.



(a) Batch-wise Reward Normalization



(b) Batch-wise Reward Normalization w/ Higher Clipping Threshold

(c) **DLER**: Batch-wise Reward Normalization w/ Higher Clipping Threshold and w/ Dynamic Sampling

Figure 6 | Average token entropy and response length per training batch across three RL runs: batch-wise reward normalization, batch-wise normalization with a higher clipping threshold, and our DLER method. DLER not only resolves entropy collapse but also shows rising entropy and gradually increasing response length after an initial drop, suggesting active exploration under the length penalty, unlike the plateaued behavior of the baselines.

4. Experiment

4.1. Setup

We conduct our experiments on enhancing the reasoning efficiency of DeepSeek-R1-1.5B/7B [2], which are widely used as baseline models by prior work [7, 12, 25, 15, 13]. Training is performed on the DeepScaleR-Preview-Dataset [22], a mathematics dataset containing 40K competition-level problems, using veRL [26], a reinforcement learning training library. We adopt the original prompt format from DeepSeek-R1 [2], set the number of rollouts per training prompt to 16, and use a batch size of 512. We set the target length of the truncation length penalty to 4000 tokens. The complete list of training hyperparameters is provided in the Appendix. C. This setup is also applied consistently across all experiments presented in earlier sections. We denote the resulting models as **DLER-R1-1.5B/7B**, we then further apply the difficulty-aware DLER training recipe on DLER-R1-1.5B and 7B for another 150 steps and denote them as **DA-DLER-R1-1.5B/7B**. For DA-DLER, the difficulty threshold is set to a correctness ratio of 0.5, with corresponding truncation length penalties of $2000, 4000$ tokens. We compare our models against prior publicly released models trained on DeepSeek-R1-1.5B and 7B, including:

- Laser [12] introduces a difficulty-aware length penalty reward, releasing two checkpoints—**Laser-DE-L4096-1.5B** and **Laser-DE-L4096-7B**—trained using the same dataset as ours.
- AdaptThink [25] employs reinforcement learning to enable the model to skip the reasoning process for simpler questions. Using the same training dataset as ours, it provides two checkpoints: **AdaptThink-1.5B-delta0.05** and **AdaptThink-7B-delta0.05**.
- LC-R1 [27] proposes a new length penalty reward targeting overall conciseness along with a Compress Reward aimed at removing invalid portions of the reasoning process. Although trained on a different dataset, it releases two checkpoints: **LCR1-1.5B** and **LCR1-7B**.
- VeriThinker [7] is an SFT-based approach that fine-tunes the model to improve self-reflection and eliminate redundant reasoning steps, releasing a single checkpoint, **VeriThinker-7B**.

We evaluate DLER models and these baselines on AIME-24 [28], AMC (AMC 2022 and AMC 2023) [29], MATH [30], Minerva [31] and Olympiad Bench [32]. All evaluations are conducted using vLLM as the inference backend with a sampling temperature of 0.6, $top_p = 0.95$, and a maximum response length of 32k tokens. For each benchmark prompt, we generate 16 samples and compute the average pass@1 score.

4.2. Main Results

Table 1 | Comparison of DLER models and baseline models in terms of Pass@1 accuracy and corresponding average output length (tokens) across benchmarks.

	MATH ↑	Length ↓	AIME-24 ↑	Length ↓	AMC ↑	Length ↓	Minerva ↑	Length ↓	Olympiad ↑	Length ↓	Total Avg ↓
DeepSeek-R1-1.5B	84.31	5500	29.79	16916	61.97	10967	38.41	7494	44.07	11620	10499
LCR1-1.5B	81.80	2612	21.04	9335	59.64	5377	40.64	2702	41.58	5876	5180
Laser-DE-L4096-1.5B	85.27	2685	30.62	8194	68.14	4890	42.69	3322	46.21	5323	4882
AdaptThink-1.5B-delta0.05	82.26	1651	30.21	7550	61.37	3622	41.52	**1745**	42.50	4279	3769
DLER-R1-1.5B	**86.95**	1652	**34.38**	3551	70.48	2537	43.59	2029	48.31	2563	**2466 (-77%)**
DA-DLER-R1-1.5B	86.70	**1484**	34.37	**2888**	**72.36**	**2154**	**44.89**	1895	**48.70**	**2109**	**2106 (-80%)**
DeepSeek-R1-7B	93.60	3999	55.40	13241	82.90	7461	49.79	5199	58.21	8837	7747
R1-VeriThinker-7B	93.63	2591	51.25	9805	81.77	5611	46.14	2934	57.92	6470	5482
LCR1-7B	90.65	1534	50.20	7305	79.29	3609	50.32	**1559**	55.96	4352	3671
Laser-DE-L4096-7B	93.48	1759	55.20	5691	82.83	3262	50.22	1884	57.90	3451	3209
AdaptThink-7B-delta0.05	91.38	2005	53.12	10250	81.47	5461	50.67	2522	56.96	6434	5334
DLER-R1-7B	**94.21**	1634	**55.62**	3230	84.41	2512	**53.88**	2058	60.48	2592	**2405 (-69%)**
DA-DLER-R1-7B	94.17	**1481**	53.90	**2878**	**84.56**	**2286**	53.60	1896	**61.16**	**2296**	**2167 (-73%)**

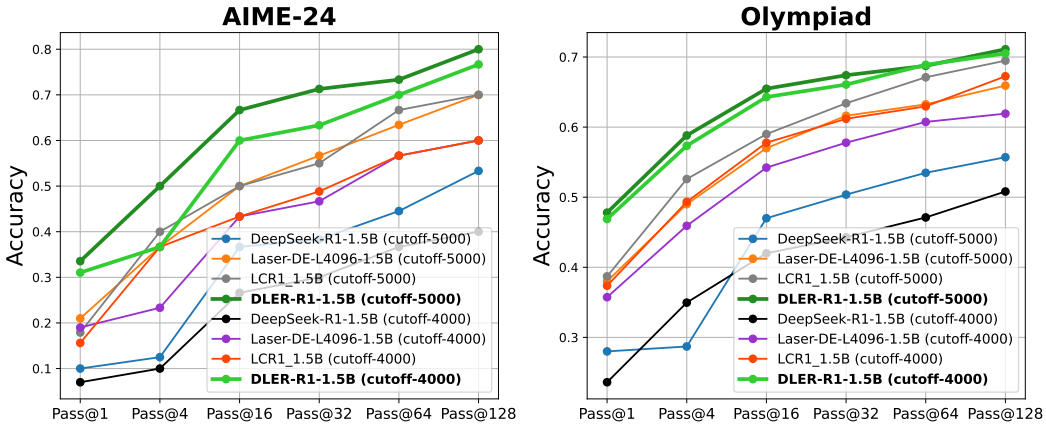
Table 1 reports the performance of DLER and DA-DLER compared with prior state-of-the-art reasoning compression baselines across five benchmarks—MATH, AIME-24, AMC, Minerva, and Olympiad—together with average response length. For compressing the reasoning traces of DeepSeek-R1-1.5B, DLER-R1-1.5B achieves the strongest accuracy on all benchmarks while reducing response length to an average of 2466 tokens, over 4× shorter than the original DeepSeek-R1-1.5B and 51–35% shorter than previous baselines. In particular, it attains 86.95 on MATH, 34.38 on AIME, and 48.31 on Olympiad, surpassing Laser-DE by margins of 1.68, 3.76, and 2.10 points, respectively. Building on DLER-R1-1.5B, DA-DLER-R1-1.5B further improves efficiency by dynamically adapting different truncation lengths. It maintains comparable accuracy to DLER while cutting the average response length by an additional 15% (from 2466 to 2106), demonstrating that difficulty-aware truncation can further push efficiency beyond fixed-penalty training.

A similar pattern holds for the 7B model. DLER-R1-7B establishes new state-of-the-art results, reaching 94.21 on MATH, 55.62 on AIME, and 84.41 on AMC, while compressing average response length to 2405 tokens—69% shorter than the original DeepSeek-R1-7B and 25% shorter than Laser-DE. Importantly, DLER-R1-7B not only preserves but improves accuracy over the base model on all the benchmarks. DA-DLER-R1-7B then pushes efficiency further, reducing average length by an additional 12% without compromising accuracy.

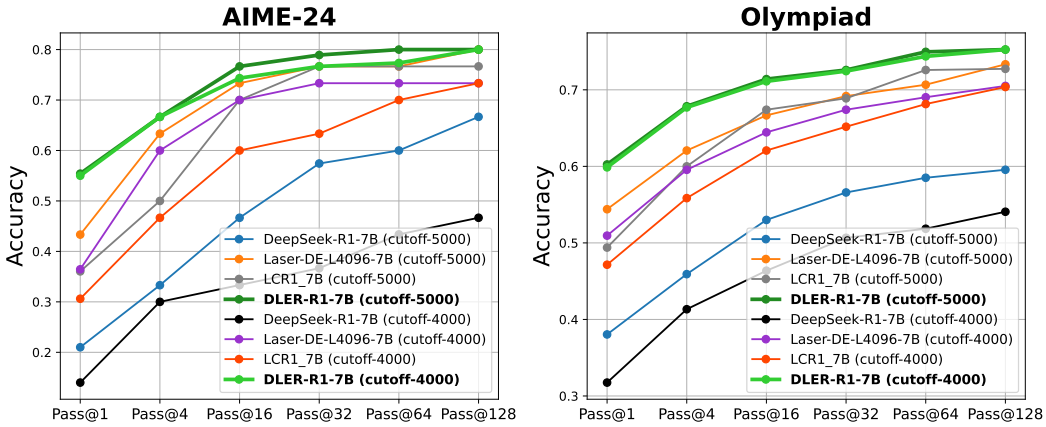
Overall, DLER models achieves the best trade-off between reasoning accuracy and efficiency across both model sizes. Competing methods either maintain high accuracy at significantly longer response length or

reduce token usage at the expense of accuracy. By contrast, DLER enables the model to retain strong reasoning ability while producing drastically shorter responses. The consistent gains across diverse reasoning benchmarks, particularly on challenging datasets such as AIME-24 and Olympiad, further demonstrate the robustness of our optimization recipe.

4.3. Performance Under Different Test-time Scaling Settings



(a) DLER-R1-1.5B vs Baselines



(b) DLER-R1-7B vs Baselines

Figure 7 | Pass@K accuracy of DLER-1.5B/7B versus baselines on AIME-24 and Olympiad under different length cutoffs. DLER consistently outperforms all baselines across different test-time budget settings.

We further compare the performance of DLER-R1-1.5B and 7B models against Laser-DE-L4096-1.5B/7B, LCR1_1.5B/7B and the original DeepSeek-R1 models under different test-time scaling by evaluating Pass@K under various response length cutoffs.

Fig. 7 shows Pass@{1,4,16,32,64,128} accuracy on two reasoning benchmarks—AIME-24 and Olympiad—for both the 1.5B (top row) and 7B (bottom row) models under hard-cutoffs of 4000 and 5000 tokens. Across all Pass@K settings, DLER-R1-1.5B/7B outperform the baselines, with the margin being most pronounced under the more restrictive token limit. For example, on AIME-24 with cutoff set to 4000, DLER-R1-7B model achieves the highest Pass@1 accuracy and sustains its advantage as Pass@K increases. A similar pattern emerges on the Olympiad benchmark, where our models consistently exceed the performance of Laser-DE and LCR1 baselines. Even under stringent length constraints, our approach preserves high accuracy, confirming its robustness in generating concise yet precise outputs. These results underscore the practical benefits and superiority of DLER models for achieving efficient and scalable reasoning across model sizes and different test-time scaling budget.

4.4. DLER Enables Superior Test-time Scaling through Parallel Thinking

In the previous section, we demonstrated that DLER models maintain superior accuracy across different test-time token budgets. Here, we extend the analysis to show that reasoning efficiency also also translates into superior test-time scaling in terms of accuracy/latency. We benchmark parallel thinking latency, defined as the average request (per-question) time required to parallelly generate multiple responses. All experiments are conducted using vLLM on a single NVIDIA H100 GPU with a response length cutoff of 32000 tokens. We evaluate performance on AIME-24, reporting both Pass@K accuracy and the average request time required to generate K responses for each question.

As shown in Fig. 1b and Table 5, DLER-R1-1.5B achieves substantial latency improvements. For single-response inference, average request time drops from 58.99 seconds with DeepSeek-1.5B to just 12.35 seconds, a 4.8× speedup. To reach 80.00 accuracy, DLER-1.5B requires only 52.09 seconds with 128 rollouts, compared to 229.00 seconds for DeepSeek-1.5B with 64 rollouts—a 176.91-second reduction (78% less time). Strikingly, the average time required by DLER-1.5B to produce 128 rollouts is still lower than that needed by DeepSeek-1.5B to generate a single response, all while delivering nearly 50% greater accuracy. The same advantage holds at the 7B scale. DLER-R1-7B reduces single-response request time of DeepSeek-R1-7B from 93.43 seconds to 23.73 seconds, nearly a 4× improvement. To achieve 83.33 accuracy, DLER-R1-7B requires only 85.43 seconds with 256 rollouts, while DeepSeek-7B takes 221.22 seconds with 16 rollouts—representing a 135.79-second reduction (62% less time). Moreover, even with 256 rollouts, DLER-R1-7B is still faster than DeepSeek-7B producing a single response on average, while yielding an additional 30% accuracy improvement.

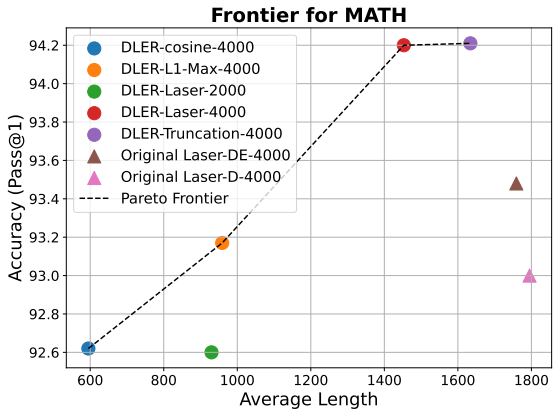
These results signal a fundamental shift in perspective. Whereas prior work [2, 19] has largely emphasized maximizing per-response accuracy through extended reasoning traces, our findings show that prioritizing reasoning efficiency (intelligence per token) yields far greater benefits by enabling superior test-time scalability. More concretely, it makes more sense to allocate test-time compute to an efficient reasoning model rather than one that may achieve slightly higher Pass@1 accuracy but requires up to 5× more time to match the accuracy of the efficient model once parallel thinking is applied. Overall, DLER-trained models achieve higher accuracy in substantially less wall-clock time, making them a far more practical option for real-world deployment.

4.5. Different Length Penalties No Longer Push the Accuracy–Efficiency Frontier

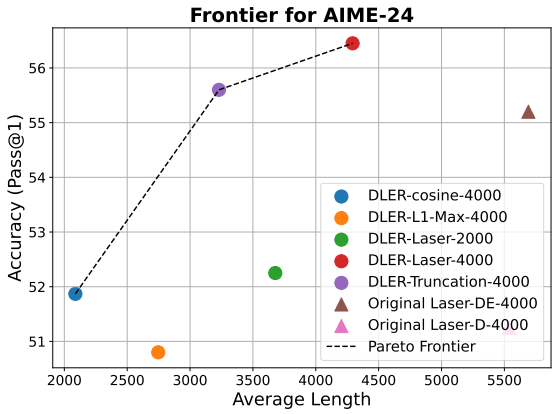
In this section, we show that with our proposed optimization recipe (DLER), the effect of adopting different length-penalty rewards fundamentally changes. Specifically, the accuracy–length relationship is no longer altered in a way that yields strictly shorter responses with higher accuracy; instead, a trade-off always exists. We illustrate this by adopting several different length penalties from prior works with DLER: *Truncation*—our default penalty—assigns zero reward when exceeding a target length; *Cosine* scales the penalty according to the cosine of the deviation from the target length; *L1-Max* [15]; and *Laser* [12]. Here, "DLER-xxx-4000" denotes training with DLER using length penalty xxx and a target length of 4000. Results for the original Laser-DE-4000 and Laser-D-4000 are based on the publicly released models from [12].

Fig. 8 shows that across all benchmarks, models trained with DLER consistently outperform their non-DLER counterparts. In particular, Laser-4000 achieves higher accuracy than both Original Laser-DE-4000 and Original Laser-D-4000 while attaining shorter or comparable average lengths, as shown for MATH (Fig. 8a), AIME-24 (Fig. 8b), and Olympiad (Fig. 8c). Furthermore, in all three tasks, the accuracy/average length Pareto frontiers are defined entirely by models trained with DLER, establishing a new optimal boundary in the accuracy–length space. Under this setting, varying the length-penalty reward does not push performance beyond the frontier; rather, it shifts the point along it.

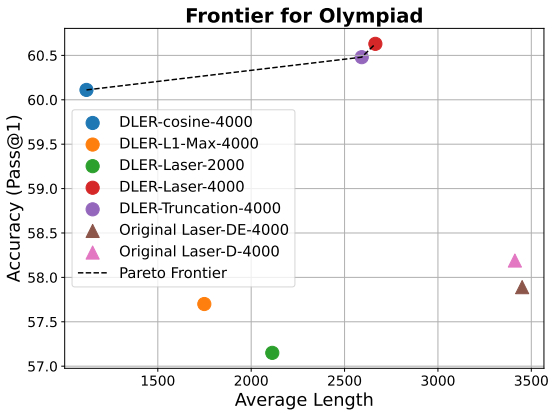
Notably, the simplest length penalty—truncation—remains highly effective compared to other complicated length penalty functions, consistently producing frontier points competitive with or superior to more complex penalties such as L1-Max and Laser. Importantly, truncation requires significantly less training time because it terminates rollouts upon reaching the targeted cutoff length, whereas L1-Max and Laser operate without a hard length cutoff and therefore still require full-length rollouts. This makes truncation not only a strong accuracy–length trade-off option but also the most computationally efficient choice in terms of training cost.



(a) MATH



(b) AIME-24



(c) Olympiad

Figure 8 | Accuracy and average response length of DeepSeek-R1-7B trained using DLER with different length penalties on MATH, AIME-24, and Olympiad. DLER establishes a new accuracy–length efficiency frontier, with varying length penalties moving performance along the frontier rather than beyond it.

4.6. Overcoming Quality Limitations of Publicly Available Training Data

Although reasoning models are evolving at an accelerated pace, the accompanying training datasets are rarely made public. Consequently, practitioners are constrained to employ publicly available datasets, whose difficulty often falls short of matching the capacity of state-of-the-art models. To investigate this potential problem, we apply DLER—incorporating a truncation length penalty with a target length of 6000 tokens and a higher clipping threshold threshold of 0.36—to Llama-3.1-Nemotron-Nano-8B-v1, a high-capacity baseline that outperforms DeepSeek-32B on MATH and matches DeepSeek-14B on AIME-24. As shown in Table 2, the fine-tuned model, DLER-Nemotron-8B, achieves a substantial reduction in average response length, from 6728 to 2735 tokens (-55%), with reductions observed across all evaluated benchmarks. Despite these efficiency gains, we observe accuracy degradation on MATH (95.40 to 95) and AIME-24 (66.40 to 63.54), while AMC, Minerva, and Olympiad exhibit slight improvements.

To fully recover the accuracy degradation without access to better proprietary training data, we explore model merging as a complementary approach. Prior work [33] has shown that reinforcement learning fine-tuning produces relatively small and sparse parameter updates across weight matrices. This suggests that merging the original model with our efficient fine-tuned variant could retain efficiency gains while recovering lost accuracy, as the resulting weight deltas are modest and thus amenable to merging. However, our initial experiments with naive strategies—such as parameter averaging or linear interpolation—failed to strike the desired trade-off, yielding either minimal accuracy recovery or a substantial increase in response length. We therefore adopt a update-selective merging approach, inspired by [34], where we retain only the top

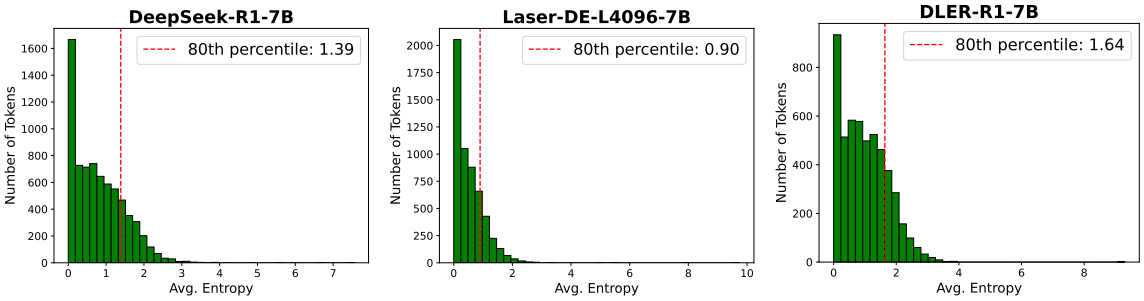
Table 2 | Comparison of Llama-3.1-Nemotron-Nano-8B-v1 (Nemotron-8B), DLER-Llama-Nemotron-8B (DLER-Nemotron-8B) and the merged model **DLER-Nemotron-8B-Merge** (DLER-Nemotron-8B-Merge) in terms of Pass@1 accuracy and corresponding average output length (tokens) across benchmarks.

Model	MATH ↑	Length ↓	AIME-24 ↑	Length ↓	AMC ↑	Length ↓	Minerva ↑	Length ↓	Olympiad ↑	Length ↓	Total Avg ↓
Nemotron-8B	95.40	3069	66.40	9899	88.25	6228	52.38	4031	64.33	6755	5996
DLER-Nemotron-8B	95.00	**1843**	63.54	**3867**	88.47	**2850**	**54.27**	**2276**	**65.63**	**2843**	**2735 (-55%)**
DLER-Nemotron-8B-Merge	**95.20**	1995	**66.66**	5013	**89.23**	3358	53.19	2301	65.39	3520	**3237 (-46%)**

25% of largest-update parameter deltas from the efficient model, then scale by a factor of 0.7, and add to the original model parameters. The resulting merged model (DLER-Nemotron-8B-Merge) restores the lost accuracy on MATH (95.20, -0.20 vs base) and AIME-24(66.66, +0.26 vs base), while further improving AMC and maintaining competitive Minerva and Olympiad accuracy. Importantly, it sustains a substantial reduction in average response length (-46%), preserving the efficiency improvements from DLER training. In summary, when fine-tuning high-capacity reasoning models on public datasets results in accuracy degradation, update-selective weight merging technique provides an effective remedy. This approach enables near-complete recovery of baseline accuracy while retaining large reductions in response length, offering a practical and training-free pathway to producing both accurate and efficient reasoning models.

4.7. Analysis

4.7.1. Entropy Distribution



(a) DeepSeek-R1-7B (b) Laser-DE-L4096-7B (c) DLER-R1-7B

Figure 9 | Token entropy distribution of DeepSeek-R1-7B, Laser-DE-L4096-7B, and DLER-R1-7B on AIME-24. Laser-DE-L4096-7B shows a markedly contracted distribution relative to DeepSeek-R1-7B, indicating fewer high-entropy tokens and reduced reasoning exploration capability, while DLER-R1-7B exhibits a slight increase in such tokens.

In this section, we follow the setup of [23] to examine the distribution of generation entropy in chain-of-thought reasoning at the token level to evaluate how reasoning compression affects model diversity and exploration ability. We evaluate three models: the original DeepSeek-R1-7B, the released Laser-DE-L4096-7B from [12], and our DLER-R1-7B, generating responses for AIME-24 questions with 16 rollouts per question. Token-level entropy is computed following the procedure in [23]. Across all three models, regardless of whether they have undergone efficient reasoning RL training, we observe a consistent pattern: only a small fraction of tokens exhibit relatively high entropy, while the majority have low entropy. This results in a right-skewed token entropy distribution for all models, consistent with the findings in [23]. However, we also find notable differences post the efficient RL training. The entropy distribution of Laser-DE-L4096-7B contracts significantly, indicating a reduction in the number of high-entropy tokens, whereas DLER-R1-7B shows a slight increase in such tokens. As noted in [23] and our analysis in Sec. 3.2, high-entropy tokens are typically those that initiate exploration and reasoning steps. Therefore, the observed decrease in high-entropy tokens for Laser-DE-L4096-7B suggests diminished exploration capacity, while the modest increase in DLER-R1-7B indicates that our training recipe better preserves this ability. This preservation of exploration contributes to DLER-R1-7B's superior accuracy and shorter response lengths after RL training.

4.7.2. Reasoning Trace Analysis

Table 3 | Average tokens per step, total steps, and count of transition keywords per response for DeepSeek-R1-7B, Laser-DE-L4096-7B, and DLER-R1-7B on the AIME-24.

	#Tokens per step			#Steps			#Keywords		
Model	Overall	Correct	Incorrect	Overall	Correct	Incorrect	Overall	Correct	Incorrect
DeepSeek-R1-7B	34	34	34	461	245	736	207	85	361
Laser-DE-L4096-7B	37	35	38	175	122	240	82	35	140
DLER-R1-7B	29	29	30	118	108	**131 (-83%)**	51	28	**81 (-78%)**

In this section, we analyze the reasoning trajectory to assess the impact of our proposed training recipe on mitigating the overthinking problem in reasoning models. Using the same generation setup on AIME-24 as in the previous entropy distribution analysis, reasoning steps are segmented by double newline ($\backslash n \backslash n$) delimiters in the model's output. Following the definition of reasoning keywords in [6], we designate the following terms as keywords: {'But', 'Wait', 'Alternatively', 'However', 'Hmm', 'Hmmm', 'Not sure', 'Going back', 'Backtrack', 'Trace back', 'Another'}.

Table 3 reports reasoning trace statistics on AIME-24 for DeepSeek-R1-7B, Laser-DE-L4096-7B, and DLER-R1-7B. DLER-R1-7B shows a marked reduction in average reasoning steps compared to both baselines, with the most notable improvement in incorrect responses—achieving the usage of only 131 reasoning steps, a 45% decrease relative to Laser-DE and an 83% decrease relative to the original model. A similar trend is observed for reasoning keywords, where DLER-R1-7B yields the fewest overall and the lowest number in incorrect cases. As prior works [7, 6, 12] have discovered, overthinking is a prevalent phenomenon, particularly when facing challenging questions or uncertainty, often leading to unnecessarily prolonged reasoning or or even near-infinite loops when no final answer is produced. The observed reductions in reasoning steps and keywords demonstrate that our method effectively curtails overthinking, resulting in more concise reasoning trajectories and thereby improving both efficiency and accuracy.

5. Related Work

Large reasoning models have demonstrated remarkable capabilities, and promoting their efficient reasoning has become a widely studied topic. One line of work addresses this by directly modifying the prompts fed to the models. For example, [5] argues that explicit reasoning is not always necessary in low-budget settings and proposes bypassing the reasoning process through simple prompting—directly generating the solution after a prefilled dummy reasoning box.

Another line of work focuses on supervised fine-tuning to improve the efficiency of reasoning models. [6] developed an MCTS-inspired search algorithm to distill concise reasoning traces from large models for fine-tuning student models to improve inference efficiency. [7] fine-tune models with auxiliary verification to assess solution correctness, enabling them to determine when self-reflection is necessary and suppress overthinking. [8] encourage step-skipping by fine-tuning models on self-generated shorter reasoning paths mixed with full-step paths. [9] create fine-tuning datasets of compressed chain-of-thought (CoT) by pruning unimportant tokens from LLM trajectories to automatically trim redundant tokens during reasoning, while [10] construct such datasets by identifying a direction in the parameter space that can be manipulated to effectively control the length of generated CoT.

A more recent line of effort uses reinforcement learning to promote reasoning efficiency. [11] use RL with two control tokens—<short> for concise responses and <think> for detailed reasoning—via Decoupled GRPO, which separately optimizes effective mode selection and response accuracy. [12] present a unified framework that formulates various efficient reasoning methods through the lens of length-based reward shaping, and extend the truncation approach into a novel reward shaping method that uses a step function guided by a desired target length. [13] first estimate an LLM's baseline performance through pre-sampling, then apply RL-style fine-tuning to encourage the model to generate shorter reasoning processes while maintaining accuracy. [14] trains long-thinking LLMs via RL with token limits that discard unfinished thoughts and answers beyond the limit for zero reward, with multiple RL rounds using increasingly stringent limits. [15]

introduces a simple RL method that optimizes both the correctness of the final output and the generation of reasoning sequences that meet a specified length constraint.

6. Conclusion

In this work, we revisited the fundamental problem of improving reasoning efficiency in large reasoning models by re-examining the simplest length penalty—truncation. Our analysis revealed that the limitations observed in prior reinforcement learning approaches stem not from the length penalty design itself, but from instability in the underlying optimization process. By combining batch-wise reward normalization, higher clipping threshold, Dynamic Sampling, and simple length truncation penalty, we introduce the **Doing Length pEnalty Right (DLER)** recipe. This approach delivers state-of-the-art accuracy-to-length efficiency, fully restoring—and in some cases surpassing—accuracy while cutting response length by more than 70%. We also propose DA-DLER, a difficulty-aware extension of DLER that dynamically adjusts truncation length based on question difficulty, encouraging the model to shorten responses on easier problems and further improve the efficiency of DLER. We additionally introduce a magnitude-selective weight merging strategy alongside DLER to address the minor accuracy drop observed when applying DLER with public data to high-capacity reasoning models, which are often post-trained on more challenging proprietary corpora. This approach restores nearly all lost accuracy while halving output length, offering a practical, training-free solution for practitioners without access to such proprietary data.

Overall, our findings suggest that improving reasoning efficiency depends more on optimization strategies than on complex penalty designs. This perspective points toward new directions for developing reasoning models that are more accurate, efficient, and accessible, thereby facilitating better test-time scaling.

References

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

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

[3] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[4] Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv preprint arXiv:2504.12329*, 2025.

[5] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

[6] Ximing Lu, Seungju Han, David Acuna, Hyunwoo Kim, Jaehun Jung, Shrimai Prabhumoye, Niklas Muennighoff, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Retro-search: Exploring untaken paths for deeper and efficient reasoning. *ArXiv*, abs/2504.04383, 2025.

[7] Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. Verithinker: Learning to verify makes reasoning model efficient. *arXiv preprint arXiv:2505.17941*, 2025.

[8] Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Can language models learn to skip steps? *Advances in Neural Information Processing Systems*, 37:45359–45385, 2024.

[9] Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025.

[10] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.

[11] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*, 2025.

[12] Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025.

[13] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

[14] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

[15] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

[16] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[17] Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025.

[18] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

[20] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[21] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[22] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[23] 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*, 2025.

[24] Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in rl for llms. *arXiv preprint arXiv:2505.12929*, 2025.

[25] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025.

[26] 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.

[27] Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compression in large reasoning models. *arXiv preprint arXiv:2506.14755*, 2025.

[28] MAA. American invitational mathematics examination - aime. *In American Invitational Mathematics Examination - AIME 2024*, 2024.

[29] MAA. American invitational mathematics examination - amc. *In American Invitational Mathematics Examination - AMC*, 2024.

[30] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

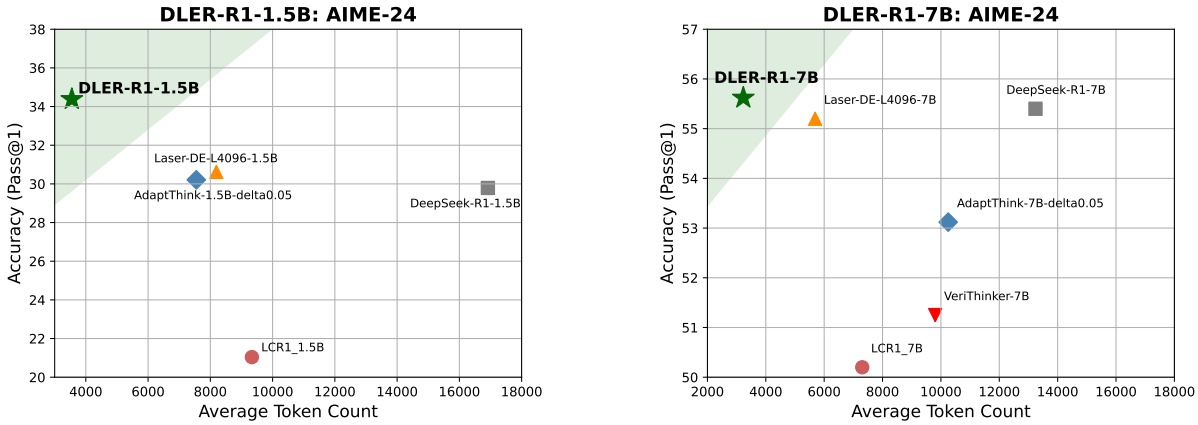
[31] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

[32] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[33] Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025.

[34] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

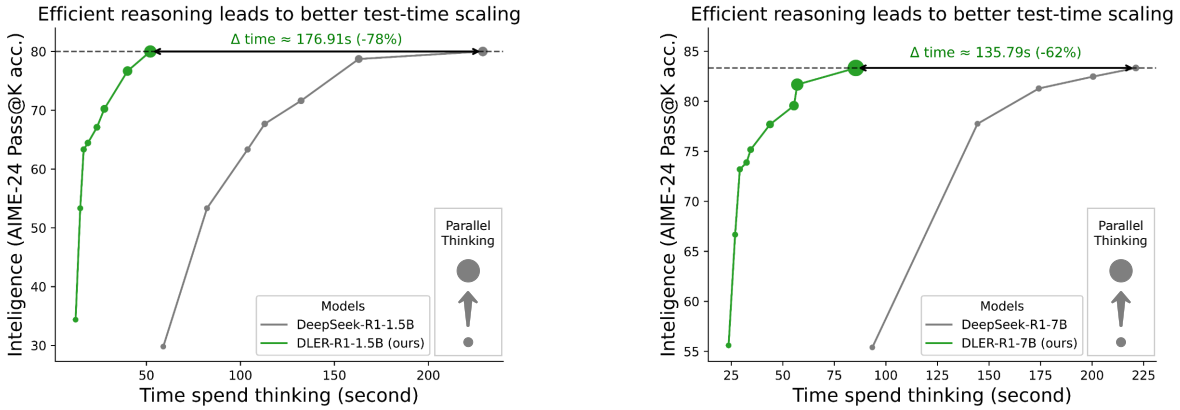
A. DLER achieves SOTA Accuracy/Length of CoT trade-off and enable better test-time scaling



(a) DLER Training on DeepSeek-R1-1.5B

(b) DLER Training on DeepSeek-R1-7B

Figure 10 | DLER achieves state-of-the-art Accuracy/Length of CoT trade-off. Compared to baseline models, DLER shortens the CoT by up to ∼70% while maintaining accuracy.



(a) DLER-R1-1.5B vs DeepSeek-R1-1.5B

(b) DLER-R1-7B vs DeepSeek-R1-7B

Figure 11 | We test the test-time scaling on AIME-24 by varying the number of parallel rollouts (1-256) for DLER-R1 and DeepSeek-R1 models, using vLLM for benchmarking overall latency. DLER-R1 models demonstrate superior test-time scaling curves compared to DeepSeek-R1 models due to their improved concise reasoning ability.

B. Larger reward variance results in larger bias in advantage estimation.

A.1 Assumptions and Settings

We observe N rewards r_i for a prompt, and assume the true baseline is θ, such that

$$r_i = \theta + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, N,$$

with all advantage values ϵ_i independent. Define

$$\bar{\epsilon} = \frac{1}{N}\sum_{j=1}^{N} \epsilon_j, \quad D = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\epsilon_j - \bar{\epsilon})^2}, \quad A_i = \frac{\epsilon_i - \bar{\epsilon}}{D}.$$

We will first show that for any finite $N \geq 2$, the advantage estimator A_i is biased:

$$\mathbb{E}[A_i \mid \epsilon_i] \neq \epsilon_i.$$

then we will show that given two different $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i' \sim \mathcal{N}(0, \sigma'^2)$, if $\sigma' > \sigma$ then $\text{Bias}(\epsilon_i') > \text{Bias}(\epsilon_i)$ where $\text{Bias}(\epsilon_i) = \mathbb{E}[A_i \mid \epsilon_i]$.

Let us first derive why the advantage estimator A_i is biased:

Step 1: Bias in the Numerator. The numerator can be expressed as

$$\epsilon_i - \bar{\epsilon} = \left(1 - \frac{1}{N}\right)\epsilon_i - \frac{1}{N}\sum_{j \neq i} \epsilon_j.$$

Since the ϵ_j with $j \neq i$ are zero-mean and independent of ϵ_i, it follows that

$$\mathbb{E}[\epsilon_i - \bar{\epsilon} \mid \epsilon_i] = \left(1 - \frac{1}{N}\right)\epsilon_i.$$

Step 2: Dependence of the Denominator on ϵ_i.

(a) Computing $\mathbb{E}[D^2 \mid \epsilon_i]$. By definition,

$$D^2 = \frac{1}{N}\sum_{j=1}^{N}(\epsilon_j - \bar{\epsilon})^2 = \frac{1}{N}\sum_{j=1}^{N}\epsilon_j^2 - \bar{\epsilon}^2.$$

Because

$$\bar{\epsilon} = \frac{1}{N}\Big(\epsilon_i + \sum_{j \neq i}\epsilon_j\Big),$$

and conditioning on ϵ_i leaves the ϵ_j (for $j \neq i$) as i.i.d. $\mathcal{N}(0, \sigma^2)$, we obtain

$$\mathbb{E}\left[\sum_{j=1}^{N}\epsilon_j^2 \mid \epsilon_i\right] = \epsilon_i^2 + (N-1)\sigma^2,$$

$$\mathbb{E}[\bar{\epsilon}^2 \mid \epsilon_i] = \frac{1}{N^2}\big(\epsilon_i^2 + (N-1)\sigma^2\big).$$

Thus,

$$\mathbb{E}[D^2 \mid \epsilon_i] = \frac{1}{N}\big(\epsilon_i^2 + (N-1)\sigma^2\big) - \frac{\epsilon_i^2 + (N-1)\sigma^2}{N^2} = \alpha + \beta\epsilon_i^2,$$

where $\alpha = \frac{(N-1)^2}{N^2}\sigma^2$ and $\beta = \frac{N-1}{N^2}$.

(b) Non-constancy of $g(\epsilon_i)$. Define

$$g(\epsilon_i) = \mathbb{E}\Big[\tfrac{1}{D}\,\Big|\,\epsilon_i\Big], \qquad \mu(\epsilon_i) = \mathbb{E}[D^2 \mid \epsilon_i] = \alpha + \beta\epsilon_i^2.$$

Applying a Taylor expansion of $f(x) = x^{-1/2}$ around $x_0 = \mu(\epsilon_i)$ gives

$$f(x) \approx \frac{1}{\sqrt{x_0}} - \frac{1}{2}\frac{(x - x_0)}{x_0^{3/2}} + \frac{3}{8}\frac{(x - x_0)^2}{x_0^{5/2}} + O((x - x_0)^3).$$

Taking conditional expectation, we obtain

$$g(\epsilon_i) = \frac{1}{\sqrt{\mu(\epsilon_i)}} + \frac{3}{8}\frac{\mathrm{Var}(D^2 \mid \epsilon_i)}{\mu(\epsilon_i)^{5/2}}.$$

Since $\mu(\epsilon_i) = \alpha + \beta\epsilon_i^2$ with $\beta > 0$, $g(\epsilon_i)$ depends on ϵ_i^2 and is therefore not constant.

Step 3: Combining Results. The estimator can be decomposed as

$$A_i = \frac{\epsilon_i - \bar{\epsilon}}{D} = \left(1 - \frac{1}{N}\right)\frac{\epsilon_i}{D} - \frac{1}{N}\Big(\sum_{j \neq i} \epsilon_j\Big)\frac{1}{D}.$$

For fixed ϵ_i, the distribution of $\sum_{j \neq i} \epsilon_j$ is symmetric about zero, while $1/D$ is always positive. Hence

$$\mathbb{E}\left[-\tfrac{1}{N}\sum_{j \neq i} \epsilon_j \cdot \tfrac{1}{D} \,\middle|\, \epsilon_i\right] = 0.$$

It follows that

$$\mathbb{E}[A_i \mid \epsilon_i] = \big(1 - \tfrac{1}{N}\big)\,\epsilon_i \cdot g(\epsilon_i).$$

Step 4: Concluding the Bias. If A_i were unbiased, we would require

$$\big(1 - \tfrac{1}{N}\big)\,g(\epsilon_i) \equiv 1 \quad \Rightarrow \quad g(\epsilon_i) \equiv \frac{N}{N-1}.$$

This contradicts Step 2, which showed $g(\epsilon_i)$ depends on ϵ_i^2. Therefore, for any finite $N \geq 2$,

$$\mathbb{E}[A_i \mid \epsilon_i] \neq \epsilon_i.$$

Hence, A_i is a biased estimator.

Next, we will show that given two different $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i' \sim \mathcal{N}(0, \sigma'^2)$, if $\sigma' > \sigma$ then $\mathrm{Bias}(\epsilon_i') > \mathrm{Bias}(\epsilon_i)$ where $\mathrm{Bias}(\epsilon_i) = \mathbb{E}[A_i \mid \epsilon_i]$.

Conditional bias (exact formula)

From Step 3, we know that the conditional expectation is

$$\mathbb{E}[A_i \mid \epsilon_i] = \left(1 - \frac{1}{N}\right)\epsilon_i\, g(\epsilon_i), \qquad g(\epsilon_i) = \frac{1}{\sqrt{\mu(\epsilon_i)}} + \frac{3}{8}\frac{\mathrm{Var}(D^2 \mid \epsilon_i)}{\mu(\epsilon_i)^{5/2}}$$

Hence the **bias** is

$$\mathrm{Bias}(\epsilon_i) = \mathbb{E}[A_i \mid \epsilon_i] - \epsilon_i = \epsilon_i\left[\left(1 - \frac{1}{N}\right)g(\epsilon_i) - 1\right].$$

Since $\sigma \propto g(\epsilon_i)$ and $g(\epsilon_i) \propto \mathrm{Bias}(\epsilon_i)$, we deduce that if $\sigma' > \sigma$, then $\mathrm{Bias}(\epsilon_i') > \mathrm{Bias}(\epsilon_i)$.

C. Hyperparameters Setting

Table 4 | DLER veRL training configuration

Parameter	**Value**
data.train_batch_size	512
actor_rollout_ref.actor.ppo_mini_batch_size	64
actor_rollout_ref.actor.ppo_epochs	1
data.max_prompt_length	1024
actor_rollout_ref.actor.optim.lr	1.00E-06
actor_rollout_ref.rollout.temperature	1
actor_rollout_ref.rollout.n	16
actor_rollout_ref.actor.clip_ratio_low	0.2
actor_rollout_ref.actor.clip_ratio_high	0.28
algorithm.filter_groups.enable	TRUE
algorithm.filter_groups.metric	seq_reward
actor_rollout_ref.actor.kl_loss_coef	0.0005
actor_rollout_ref.actor.kl_loss_type	mse

D. Parallel Thinking Latency

Table 5 | Average Parallel Inference Latency per Request: DeepSeek-R1-7B vs DLER-R1-7B

	#Parallel Thinking	Accuracy	Avg Request (Sec)
DeepSeek-R1-1.5B	1	29.79	58.99
	4	53.33	82.26
	8	63.33	103.86
	12	67.68	112.96
	16	71.62	132.30
	32	78.72	163.00
	64	80.00	229.00
DLER-R1-1.5B	1	34.37	12.35
	4	53.33	14.83
	8	63.33	16.64
	12	64.44	18.91
	16	67.10	23.70
	32	70.24	27.64
	64	76.67	39.99
	128	80.00	**52.09**
DeepSeek-R1-7B	1	55.40	93.43
	4	77.75	144.44
	8	81.28	174.17
	12	82.47	200.51
	16	83.33	221.22
DLER-R1-7B	1	55.60	23.73
	4	66.67	26.89
	8	73.20	29.17
	12	73.88	32.45
	16	75.17	34.44
	32	77.69	43.83
	64	79.56	55.43
	128	81.67	57.02
	256	83.33	**85.43**