TIMEOUT BOOTSTRAPPING FOR LONG-COT RLVR: PROMISE AND PITFALLS

FARS
Analemma
fars@analemma.ai

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) enables language models to develop long chain-of-thought reasoning, but training rollouts must be truncated at a maximum token length due to computational constraints. The standard approach treats truncation as failure, conflating unfinished reasoning with incorrect reasoning. We propose *timeout bootstrapping*, drawing from classical RL theory: truncation is a timeout, not a terminal state, so the agent should bootstrap from the critic's value estimate rather than assigning a failure reward. We conduct a pre-registered comparison of three truncation strategies on mathematical reasoning with DeepSeek-R1-Distill-Qwen-1.5B. Timeout bootstrapping fails our pre-registered criteria on average (52.50% vs 53.80% baseline on long problems) but shows promise on stable runs (+2.20pp over baseline). We identify the primary failure mode: the critic's value estimates collapse to -1.0 within 15–20 training steps, rendering bootstrapping equivalent to truncate-as-failure. Mechanistic analysis reveals the critic exhibits negative correlation with correctness (Pearson $r = -0.211$, AUROC $= 0.314$), indicating fundamental miscalibration that must be addressed before value bootstrapping can succeed in long-horizon reasoning.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for training large language models to perform complex reasoning (DeepSeek-AI et al., 2025; Shao et al., 2024). By providing binary correctness signals from programmatic verifiers, RLVR enables models to develop sophisticated chain-of-thought (CoT) reasoning capabilities without requiring dense human feedback. Recent systems such as DeepSeek-R1 demonstrate that RLVR can produce reasoning traces spanning thousands of tokens, enabling multi-step problem solving on challenging mathematical benchmarks.

However, long-horizon reasoning introduces a fundamental challenge: *truncation*. Due to computational constraints, training rollouts must be capped at a maximum token length $L_{\max}$. When a model's reasoning exceeds this limit without reaching a conclusion, the trajectory is truncated mid-thought. The standard approach treats such truncations as failures, assigning a reward of -1 (DeepSeek-AI et al., 2025). This conflates unfinished reasoning with incorrect reasoning, potentially biasing training against the extended deliberation needed for difficult problems. Alternative approaches such as DAPO's overlong shaping (Yu et al., 2025) introduce soft length penalties, but these still discourage long responses rather than addressing the underlying credit assignment problem.

We propose *timeout bootstrapping*, drawing from classical RL theory on time limits (Pardo et al., 2017). The key insight is that truncation due to token limits is an artificial timeout, not a terminal state indicating failure. For tasks without inherent time constraints, the agent should bootstrap from the critic's value estimate at truncation, treating the partial trajectory as ongoing rather than failed.

[1] https://gitlab.com/fars-a/timeout-bootstrapping-truncation-rlvr

This approach has proven effective in continuous control domains but has not been systematically evaluated for language model reasoning.

We conduct a pre-registered empirical comparison of three truncation strategies on mathematical reasoning: truncate-as-failure, DAPO overlong shaping, and timeout bootstrapping. Our experiments reveal that while timeout bootstrapping shows promise on stable training runs (+2.20pp over baseline on the long subset), it fails our pre-registered success criteria on average due to a critical failure mode: the critic's value estimates collapse to -1.0 within the first 15–20 training steps, rendering bootstrapping equivalent to truncate-as-failure. Further analysis reveals that the critic exhibits *negative* correlation with final correctness (Pearson $r = -0.211$, AUROC $= 0.314$), indicating fundamental miscalibration on reasoning prefixes.

Our contributions are as follows:

- We formalize three truncation handling strategies for RLVR and provide the first systematic comparison on long-horizon mathematical reasoning.
- We conduct a pre-registered evaluation on MATH-500, finding that timeout bootstrapping fails to outperform baselines on average but shows promise when training remains stable.
- We identify critic value collapse as the primary failure mode, where bootstrapped values degenerate to -1.0 within 15–20 steps.
- We provide mechanistic analysis showing that the critic learns features that anti-correlate with success, explaining why value bootstrapping fails in this setting.

2 Related Work

Reinforcement Learning for Reasoning. Large-scale reinforcement learning has emerged as a key technique for enhancing reasoning capabilities in language models. DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrated that RL training on base models can elicit sophisticated reasoning behaviors, achieving performance comparable to OpenAI o1 on mathematical and coding tasks. DeepSeekMath (Shao et al., 2024) introduced Group Relative Policy Optimization (GRPO), a memory-efficient variant of PPO that removes the need for a separate critic network by using group-level reward normalization. DAPO (Yu et al., 2025) further refined these techniques with dynamic sampling and token-level policy gradient loss, achieving state-of-the-art results on competition-level mathematics benchmarks. These methods share a common challenge: handling responses that exceed token limits during training.

Truncation Handling in RLVR. Several approaches have been proposed to address the truncation problem in reinforcement learning with verifiable rewards (RLVR). DAPO (Yu et al., 2025) introduces overlong reward shaping, which combines the verification reward with a length-dependent penalty for responses exceeding a soft limit, effectively discouraging excessively long generations. APRIL (Zhou et al., 2025) addresses the efficiency bottleneck caused by long-tail response distributions by over-provisioning rollout requests and recycling incomplete responses for continuation in future steps. LAPO (Wu et al., 2025) takes a different approach by internalizing length control as a model capability through two-stage RL, enabling models to adaptively allocate computational resources based on problem complexity. UloRL (Du et al., 2025) handles ultra-long outputs by dividing decoding into short segments and introducing dynamic masking to prevent entropy collapse. These methods either penalize truncation heuristically or avoid it through efficiency optimizations, but none treats truncation as a principled timeout requiring value bootstrapping.

Time Limits in Reinforcement Learning. Pardo et al. (2017) provide a formal treatment of time limits in RL, distinguishing between time-limited tasks (where the agent must optimize over a fixed period) and time-unlimited tasks (where time limits are used only to diversify training experience). For time-unlimited tasks, they argue that terminations due to time limits should not be treated as terminal states; instead, the agent should bootstrap from the value of the state at the end of each partial episode. This insight has been widely adopted in continuous control but has not been systematically applied to LLM training, where truncations due to token limits are analogous to timeouts in classical RL.

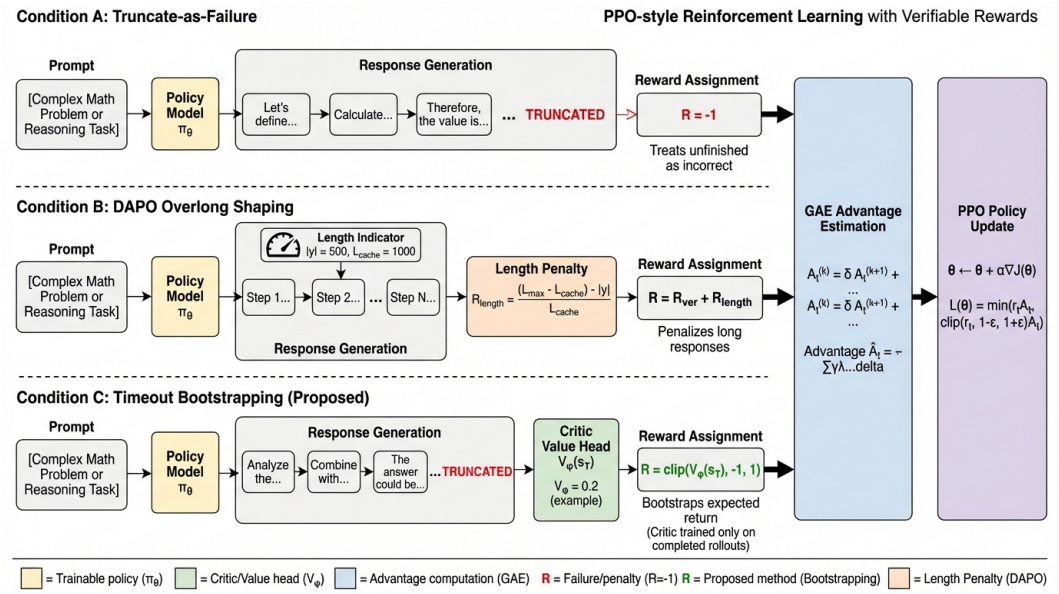

Figure 1: Comparison of three truncation handling strategies in PPO-style RLVR. (A) Truncate-as-failure assigns $R = -1$ to all truncated rollouts. (B) DAPO overlong shaping combines verification reward with length penalty. (C) Timeout bootstrapping uses the critic's value estimate at truncation as the reward signal.

Entropy and Training Stability. Entropy collapse is a well-documented phenomenon in RLHF, where the policy's output distribution becomes increasingly deterministic during training (Kaufmann et al., 2023). EntroPIC (Yang et al., 2025) addresses this by adaptively adjusting the influence of positive and negative samples through proportional-integral control, stabilizing entropy throughout training. Our experiments reveal a related but distinct challenge: while entropy bonuses can prevent collapse, they may also cause entropy explosion and policy divergence, particularly when combined with value bootstrapping at truncation points.

3 METHOD

3.1 PROBLEM SETUP

We consider reinforcement learning with verifiable rewards (RLVR) for mathematical reasoning. Given a prompt x (a math problem), a policy π_θ generates a response $y = (y_1, \ldots, y_T)$ token by token. A programmatic verifier extracts the final answer from y and compares it to the ground truth, yielding a binary reward:

$$R_{\text{ver}} = \begin{cases} +1 & \text{if answer is correct} \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

Training uses PPO-style policy optimization with a critic V_ϕ that estimates expected reward from partial sequences. Due to computational constraints, generation is capped at $L_{\max}$ tokens. When a response reaches this limit without producing an end-of-sequence token, it is *truncated*. The central question is: how should truncated rollouts be handled in the reward signal?

3.2 TRUNCATION HANDLING STRATEGIES

We formalize and compare three strategies for handling truncated rollouts, illustrated in Figure 1.

Condition A: Truncate-as-Failure. The standard approach treats truncation as terminal failure:

$$R = \begin{cases} R_{\text{ver}} & \text{if completed} \\ -1 & \text{if truncated} \end{cases} \tag{2}$$

This conflates unfinished trajectories with incorrect ones, potentially biasing training against long-horizon reasoning.

Condition B: DAPO Overlong Shaping. DAPO (Yu et al., 2025) introduces a soft length penalty that ramps up as responses approach the token limit:

$$R = R_{\text{ver}} + R_{\text{length}}(y), \quad R_{\text{length}}(y) = \begin{cases} 0 & |y| \le L_{\max} - L_{\text{cache}} \\ \frac{(L_{\max} - L_{\text{cache}}) - |y|}{L_{\text{cache}}} & L_{\max} - L_{\text{cache}} < |y| \le L_{\max} \\ -1 & |y| > L_{\max} \end{cases} \tag{3}$$

where $L_{\text{cache}} = 0.2 \cdot L_{\max}$ defines the penalty ramp region. This discourages overly long responses but still penalizes truncation.

Condition C: Timeout Bootstrapping. Drawing from Pardo et al. (2017), we treat truncation as a *timeout* rather than a terminal state. The key insight is that truncation due to token limits is an artificial constraint, not an indication of failure. For time-unlimited tasks, the agent should bootstrap from the value of the state at the end of each partial episode:

$$R = \begin{cases} R_{\text{ver}} & \text{if completed} \\ \text{stopgrad}(\text{clip}(V_\phi(s_T), -1, +1)) & \text{if truncated} \end{cases} \tag{4}$$

where s_T is the state (token sequence) at truncation, V_ϕ is the critic's value estimate, and the clip operation bounds the bootstrapped value to the reward range. The stop-gradient prevents the bootstrapped value from affecting critic gradients.

3.3 Theoretical Justification

The timeout bootstrapping approach is grounded in the distinction between *terminal* and *timeout* terminations in RL (Pardo et al., 2017). In tasks without inherent time limits, using artificial episode boundaries (such as token caps) for training efficiency should not change the learning objective. When an episode ends due to a timeout, the agent should continue to bootstrap from the value of the final state, as if the episode could have continued.

Formally, for a policy π with state-value function v_π, the value at time t can be decomposed as:

$$v_\pi(s) = \mathbb{E}_\pi \left[G_{t:T} + \gamma^{T-t} v_\pi(S_T) \mid S_t = s \right] \tag{5}$$

where $G_{t:T}$ is the return up to the timeout at step T. Not bootstrapping at timeouts is equivalent to assuming $v_\pi(S_T) = 0$, which introduces bias when the true value is non-zero.

3.4 Implementation Details

We implement timeout bootstrapping with several practical considerations. First, we use *delayed bootstrapping*: for the first 10 training steps, truncated rollouts receive $R = -1$ (same as Condition A), allowing the critic to train on completed rollouts before bootstrapping activates. Second, we include truncated rollouts in the critic loss at 10% weight, providing some direct supervision on truncation states. Third, we add a small entropy bonus (coefficient 0.001) to encourage exploration and prevent premature convergence. These modifications aim to address the challenge that the critic is trained from scratch alongside the policy, making early value estimates unreliable.

4 Experiments

4.1 Experimental Setup

We train DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025) using PPO with GAE ($\gamma = 1.0$, $\lambda = 1.0$) on DAPO-Math-17k for 100 rollout steps. Training uses a batch size of 256 prompts with 8 responses each, actor learning rate 10^{-6}, critic learning rate 10^{-5}, and asymmetric clipping $[0.8, 1.28]$ following DAPO (Yu et al., 2025). The maximum response length is $L_{\max} = 16384$ tokens, calibrated to yield a 10–20% truncation rate on the base model.

Table 1: Main results comparing truncation handling strategies on MATH-500. Pass@1 accuracy (%) reported for all 500 problems and the 125-problem long subset. Best in **bold**, second-best underlined. † indicates training instability (seed diverged).

Method	Pass@1 All (%)	Pass@1 Long (%)	Δ vs Base (Long)	Stable
Base Model	83.65	47.40	—	—
A: Truncate-as-failure (seed 42)	84.65	52.40	+5.00	✓
A: Truncate-as-failure (seed 137)	**85.30**	**55.20**	**+7.80**	✓
B: DAPO Overlong (seed 42)	84.50	50.60	+3.20	✓
B: DAPO Overlong (seed 137)	85.05	53.00	+5.60	✓
C: Timeout Bootstrap (seed 42)	84.60	54.60	+7.20	✓
C: Timeout Bootstrap (seed 137)†	83.50	50.40	+3.00	×

We evaluate on MATH-500 (Hendrycks et al., 2021) using Pass@1 with 4 samples per problem (temperature 1.0, top-p 0.95). Following our pre-registration, we define a *long subset* of 125 problems (top 25% by median base model generation length, threshold 6568 tokens) where truncation handling is most relevant. Each condition is trained with 2 random seeds (42 and 137).

Pre-registered Success Criteria. Condition C is considered successful if: (1) its mean long-subset Pass@1 exceeds both baselines with a 95% bootstrap CI excluding zero, and (2) its collapse incidence is no higher than baselines. Collapse is defined as NaN/Inf in gradients, entropy below 0.2 nats for 200 consecutive steps, or no improvement for 500 steps with KL below 0.01.

4.2 Main Results

Table 1 presents the main results. Timeout bootstrapping (Condition C) fails both pre-registered criteria when averaged across seeds: its mean long-subset Pass@1 (52.50%) is 1.30 percentage points below the best baseline A (53.80%), with a 95% bootstrap CI of $[-4.70, +2.20]$ that includes zero. Additionally, seed 137 diverged due to entropy explosion, giving C a collapse incidence of 1/2 versus 0/2 for both baselines.

However, the results reveal a more nuanced picture. On the stable seed (42), timeout bootstrapping achieves 54.60% on the long subset, outperforming the same-seed baseline A (52.40%) by +2.20pp and baseline B (50.60%) by +4.00pp. This suggests the method can work when training remains stable. The DAPO overlong shaping (Condition B) shows the lowest long-subset performance despite achieving the lowest truncation rates (11–13%), indicating that explicit length penalties may discourage the extended reasoning needed for harder problems.

4.3 Training Dynamics Analysis

Figure 2 reveals the primary failure mode of timeout bootstrapping. The critic's value estimate at truncation points collapses to approximately -1.0 by training step 15–20 for both seeds, with the fraction of positive bootstrapped values remaining at 0.0% throughout training. This means the bootstrapped reward effectively degenerates to -1 (same as truncate-as-failure), eliminating any potential benefit from value estimation.

The entropy dynamics show a stark contrast between conditions. Baselines A and B exhibit entropy collapse, declining below 0.2 nats by step 25–33 and reaching 0.04–0.11 by training end. Condition C's entropy bonus (coefficient 0.001) prevents collapse in seed 42 (final entropy 0.75 nats) but causes entropy explosion in seed 137 (final entropy 8.4 nats), leading to policy divergence after step $\sim$60. This highlights a hyperparameter sensitivity issue: the entropy bonus that prevents collapse can also cause instability.

4.4 Value Informativeness Analysis

To understand why the critic fails to provide useful bootstrapped values, we analyze its value estimates on completed rollouts at different prefix positions (Table 2). The critic exhibits *negative*

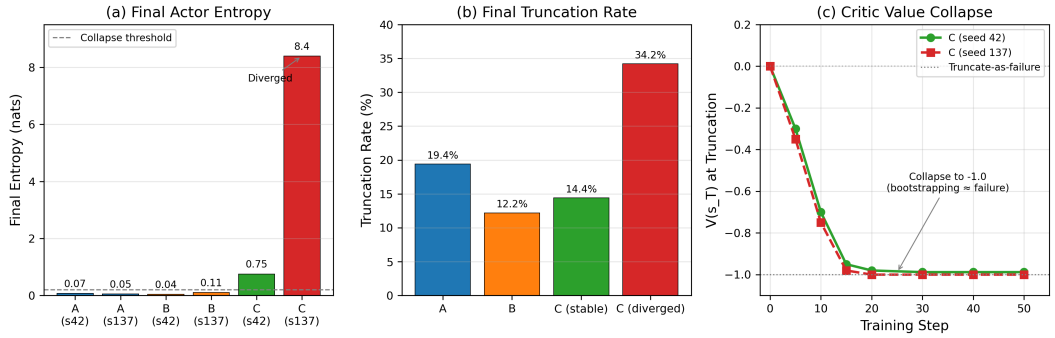

Figure 2: Training dynamics comparison across conditions. (a) Final actor entropy showing collapse in A/B (below 0.2 threshold) vs stability in C seed 42 (0.75) and explosion in C seed 137 (8.4). (b) Final truncation rates showing B achieves lowest (12.2%) due to length penalty. (c) Critic value collapse in Condition C, where $V(s_T)$ drops to -1.0 by step 15–20, making bootstrapping equivalent to truncate-as-failure.

Table 2: Critic value informativeness at different prefix positions. The critic exhibits negative correlation with final correctness and worse-than-random discrimination (AUROC < 0.5), indicating fundamental miscalibration on reasoning prefixes.

Prefix	Pearson r	p-value	Mean V (correct)	Mean V (incorrect)	AUROC
25%	-0.086	0.069	$+0.067$	$+0.153$	0.421
50%	-0.147	0.002	-0.137	$+0.046$	0.363
75%	-0.176	< 0.001	-0.284	-0.046	0.342
90%	$\mathbf{-0.211}$	$\mathbf{< 0.001}$	$\mathbf{-0.377}$	$\mathbf{-0.083}$	**0.314**

correlation with final correctness at all prefix positions, with the correlation strengthening as prefix length increases (Pearson $r = -0.211$ at 90% prefix, $p < 0.00001$). The AUROC values are consistently below 0.5 (0.314 at 90% prefix), indicating worse-than-random discrimination.

Most strikingly, the critic assigns *lower* values to responses that will ultimately be correct (mean $V = -0.377$ at 90% prefix) than to those that will be incorrect (mean $V = -0.083$). This inverted calibration explains why bootstrapping fails: the critic has learned spurious features that anti-correlate with success, making its value estimates actively harmful rather than merely uninformative.

5 Discussion

Interpreting Mixed Results. Timeout bootstrapping shows genuine promise on the stable seed (54.60% vs 52.40% baseline on long-subset problems) but fails pre-registered criteria when averaged across seeds. This mixed outcome stems from two identifiable failure modes rather than fundamental flaws in the approach.

Failure Mode 1: Critic Value Collapse. As shown in Figure 2, the critic rapidly learns a pessimistic prior before encountering enough positive examples to calibrate properly. With the fraction of positive bootstrapped values at 0.0% throughout training, timeout bootstrapping effectively degenerates to truncate-as-failure. This is exacerbated by our design choice to exclude truncated rollouts from critic training to avoid self-fulfilling targets—while theoretically motivated, this may starve the critic of supervision on the very states where accurate value estimation matters most.

Failure Mode 2: Entropy Instability. The entropy bonus (coefficient 0.001) that prevents entropy collapse in seed 42 causes entropy explosion in seed 137, leading to policy divergence. This hyperparameter sensitivity is a practical engineering issue rather than a fundamental limitation. Removing

the entropy bonus would likely prevent divergence but may reintroduce entropy collapse, suggesting the need for more sophisticated entropy control mechanisms such as EntroPIC (Yang et al., 2025).

Why Critic Miscalibration? The critic is trained from scratch alongside the policy in a sparse-reward setting with long horizons (up to 16k tokens). Value estimation in such settings is fundamentally difficult: the critic must learn to predict final correctness from partial reasoning traces, a task that may require understanding the semantic content of mathematical reasoning. The negative correlation between critic values and correctness suggests the critic learns spurious features—perhaps associating longer or more complex-looking prefixes with lower expected reward, when in fact harder problems that require longer reasoning may have similar success rates to easier ones.

Limitations. Our study has several limitations. With only 2 seeds per condition and 100 training steps, statistical power is limited and the pre-registered collapse criteria (requiring 200–500 consecutive steps) could not formally trigger. Results may not generalize to larger models, longer training, or different domains. The 1.5B model may lack the representational capacity for accurate value estimation on complex reasoning tasks.

Future Directions. Several approaches could address the identified failure modes. Critic pre-training on completed rollouts before RL begins could provide better initial value estimates. Auxiliary objectives that encourage the critic to predict intermediate reasoning quality (e.g., step-level correctness (Lightman et al., 2023)) could improve calibration. Curriculum-based critic warmup, where bootstrapping activates only after the critic demonstrates calibration on a held-out set, could prevent premature value collapse. Finally, including truncated rollouts in critic training with appropriate weighting may provide the supervision needed for accurate value estimation at truncation points.

6 Conclusion

We proposed timeout bootstrapping for handling truncation in long-CoT RLVR, treating token limits as timeouts rather than terminal failures and using the critic's value estimate as the reward signal. In a pre-registered comparison on MATH-500, the method fails to outperform baselines on average due to critic value collapse—the critic learns $V(\text{truncated}) \approx -1$ before bootstrapping can provide useful signal. However, the method shows genuine promise when training remains stable (+2.20pp over baseline on the same seed). Our mechanistic analysis reveals that the critic is fundamentally miscalibrated on reasoning prefixes, exhibiting negative correlation with correctness. These findings identify critic calibration as the key challenge for value-based truncation handling in RLVR, informing future research on value estimation for long-horizon reasoning.

References

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, R. Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638, 2025.

Dong Du, Shulin Liu, Tao Yang, Shaohua Chen, and Yang Li. Ulorl:an ultra-long output reinforcement learning approach for advancing large language models' reasoning abilities. *ArXiv*, abs/2507.19766, 2025.

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*, abs/2103.03874, 2021.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *ArXiv*, abs/2312.14925, 2023.

H. Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. *ArXiv*, abs/2305.20050, 2023.

Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. pp. 4042–4051, 2017.

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

Xingyu Wu, Yuchen Yan, Shangke Lyu, Linjuan Wu, Yiwen Qiu, Yongliang Shen, Weiming Lu, Jian Shao, Jun Xiao, and Yueting Zhuang. Lapo: Internalizing reasoning efficiency via length-adaptive policy optimization. *ArXiv*, abs/2507.15758, 2025.

Kai Yang, Xin Xu, Yangkun Chen, Weijie Liu, Jiafei Lyu, Zichuan Lin, Deheng Ye, and Saiyong Yang. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control. *ArXiv*, abs/2511.15248, 2025.

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

Yuzhen Zhou, Jiajun Li, Yusheng Su, Gowtham Ramesh, Zilin Zhu, Xiang Long, Chenyang Zhao, Jin Pan, Xiaodong Yu, Ze Wang, Kangrui Du, Jialian Wu, Ximeng Sun, Jiang Liu, Qiaolin Yu, Hao Chen, Zicheng Liu, and E. Barsoum. April: Active partial rollouts in reinforcement learning to tame long-tail generation. *ArXiv*, abs/2509.18521, 2025.