# Time-Frequency Token Advantage Clipping for Training Efficient Large Reasoning Model

**Rong Bao[1], Bo Wang[1], Xiao Wang[1], Hongyu Li[2], Rui Zheng[1], Leszek Rutkowski[3], Qi Zhang[1], Liang Ding[4*], Dacheng Tao[5*]**

[1] College of Computer Science and Artificial Intelligence, Fudan University, China
[2] Independent Researcher, Hangzhou, China
[3] Systems Research Institute of the Polish Academy of Sciences, AGH University of Krakow, 30-059 Kraków, and the SAN University, 90-113, Łódź, Poland
[4] The University of Sydney, Sydney, Australia
[5] Generative AI Lab, College of Computing and Data Science, Nanyang Technological University, Singapore
rbao22@m.fudan.edu.cn, leszek.rutkowski@ibspan.waw.pl, dacheng.tao@gmail.com

## Abstract

Long Chain-of-Thought (CoT) reasoning enhances large reasoning models' performance but suffers from severe inefficiencies, as models often overthink simple problems or underthink complex ones. Current sequence-level optimizations, like length penalties, are too coarse-grained to distinguish core logic from verbose language, precluding the necessary token-level control for efficient reasoning CoT. To overcome these limitations, we introduce Time-Frequency token Advantage Clipping (TFAC), a novel training framework designed to build efficient large reasoning models via token-level interventions. Specifically, TFAC functions along two dimensions: 1) The Frequency Dimension: It discourages inefficient loops and encourages deeper exploration by dynamically reducing the advantage scores of high-entropy tokens that are repeatedly generated within a single reasoning path. 2) The Time Dimension: It reduces excessive overthinking of the system by establishing a historical baseline for the occurrence count of each critical token in previously successful trajectories, and clipping the advantages of tokens that exceed this baseline during training. Crucially, to preserve the model's exploratory capabilities on novel problems, this suppression mechanism is automatically disabled when no historical record of success is available. Experiments conducted on the Deepseek-Distill-32B and Qwen3-8B models show that TFAC outperforms leading baseline methods, improving performance by 2.3 and 3.1 percentage points, respectively, while simultaneously reducing inference costs by 35% and 28% in scenarios where correct answers are generated. These results validate the significant efficacy of TFAC in training large reasoning models that are both powerful and highly efficient.

## 1 Introduction

The development of Large Language Models (LLMs) (OpenAI 2023; Yang et al. 2024; Bai et al. 2022; DeepSeek-AI et al. 2024) has driven significant breakthroughs in the field of artificial intelligence, particularly demonstrating strong capabilities in complex reasoning tasks that require multi-step logic. The evolution of LLMs' reasoning abilities has progressed from generating short CoT (Wei et al. 2022) with simple prompts to training models to produce long thinking processes through reinforcement learning (DeepSeek-AI et al. 2025; Jaech et al. 2024). These so-called large reasoning models (LRMs) are no longer confined to simple natural language generation tasks. Instead, they can engage in reflective thinking (Saunders et al. 2022; McAleese et al. 2024), advanced planning (Valmeekam et al. 2023; Pallagani et al. 2024), and hypothesis-deduction behaviors (Peng et al. 2024), achieving superior reasoning capabilities at the expense of greater computational demands (Snell et al. 2024).

However, a critical challenge is the dichotomy between two undesirable reasoning patterns: overthinking on simple problems and underthinking on complex ones. Specifically, when faced with simple problems, some LRMs tend to generate excessively long and redundant reasoning, a phenomenon referred to as overthinking (Chen et al. 2024, 2025). Conversely, when confronted with difficult problems, they often generate superficial and scattered thoughts, a behavior known as underthinking (Wang et al. 2025c,a). Striking the right balance between reasoning quality and computational cost is crucial for the development of LRMs (Bowman et al. 2022).

To tackle the inefficiency of long CoT reasoning, existing research has primarily focused on sequence-level optimizations. One common strategy is the use of length penalties within the reinforcement learning objective, which aims to improve efficiency by penalizing overly long outputs (Luo et al. 2025; Aggarwal and Welleck 2025). However, this coarse-grained approach does not distinguish between essential reasoning steps and verbose, auxiliary language, leading to the indiscriminate penalization of all tokens (Liu et al. 2025a; Ma et al. 2025). Another prevalent tactic is confidence-based adaptive reasoning, which terminates generation once the model's confidence exceeds a predetermined threshold (Jiang et al. 2025a). The effectiveness of this method, nevertheless, is compromised by the often unreliable calibration of model confidence, which can lead to premature convergence on incorrect answers (Jiang et al. 2025b).

The fundamental limitation of these sequence-level methods is their treatment of the CoT as an indivisible whole, overlooking the varied contributions of individual tokens. In

*Corresponding authors.

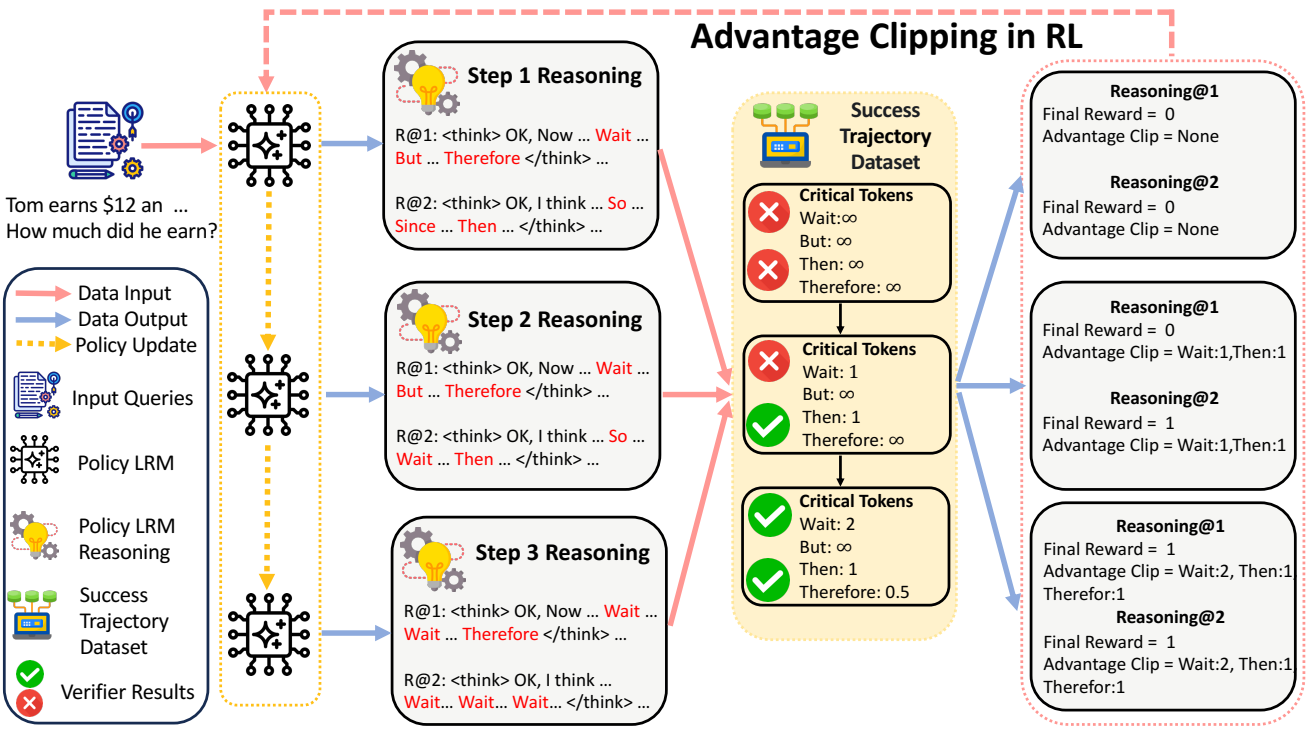



Figure 1: A schematic illustration of the Time-Frequency token Advantage Clipping (TFAC) framework for training efficient LRMs. Upon the success of a trajectory, the usage count of its critical tokens is recorded into the successful trajectory dataset, establishing a historical token consumption budget. If the model's generation frequency for a critical token exceeds its allocated budget, TFAC clips the advantage associated with that token. This mechanism penalizes model's both underthinking and overthinking, thereby steering the policy towards more efficient and concise reasoning paths.

stark contrast, recent studies reveal that the generation of long CoTs is largely governed by a small number of high-entropy tokens (Wang et al. 2025b; Cui et al. 2025b). These tokens act as critical fork points in the reasoning path. Macro-level issues, such as a model overthinking and underthinking often arise from inefficient decisions made at these specific token-level nodes (Bogdan et al. 2025).

Motivated by these insights, we propose the **Time-Frequency token Advantage Clipping (TFAC)** framework, a novel approach that enhances training efficiency through fine-grained interventions at these critical reasoning nodes. TFAC operates on two dimensions:

1. **Frequency Dimension (Intra-Path Control):** To prevent redundant forking thoughts, TFAC monitors the generation frequency of high-entropy tokens within a single reasoning path. If a specific critical token is generated repeatedly, its advantage higher clip is dynamically reduced, thereby suppressing inefficient loops and encouraging deeper exploration of promising paths.
2. **Temporal Dimension (Historical Control):** To curb systemic overthinking, TFAC computes a historical average occurrence for each critical token using previously successful reasoning trajectories. During inference, if a token's count exceeds this established historical benchmark, its advantage estimate is explicitly clipped. This action terminates potentially redundant reasoning paths.

Crucially, to avoid underthinking on novel or complex problems, this advantage clipping mechanism is automatically disabled for tasks that have no history of successful trajectories, preserving the model's exploratory capability. Comprehensive experiments on the Deepseek-Distill-32B and Qwen3-8B models demonstrate that TFAC significantly outperforms existing baseline methods. Across a range of reasoning benchmarks, TFAC achieves average accuracy improvements of **2.3 and 3.1 percentage points** over the strongest baselines, while simultaneously reducing inference costs by **35% and 28%**, respectively.

We summarize our contributions as follows:

- **Novel RL Framework:** We propose TFAC, a reinforcement learning framework that trains highly efficient LRMs by precisely controlling the advantage clipping functions of critical (high-entropy) tokens.
- **Time-Frequency Advantage Clipping:** We design a novel advantage clipping function that integrates a token's real-time occurrence frequency (frequency dimension) with its historical success data (temporal dimension), enabling fine-grained control over each reasoning step.
- **Empirical Validation and Insights:** Through extensive experiments, we validate TFAC's significant superiority and provide in-depth analyses that offer crucial insights for designing future efficient LRMs.

## 2 Related Work

**Training Efficient LRMs** The Long CoT technique extends and deepens conventional CoT methods, enabling LLMs to address complex multi-step reasoning tasks by generating longer and more intricate reasoning sequences. Although Long CoT aligns with the paradigm of expanding computation at test-time, the research community continues to pursue efficiency improvements (Qu et al. 2025; Bao et al. 2025) to find a Pareto optimum between computational cost and model performance. However, while augmenting ground-truth verification rewards with heuristic length-based penalties can partially mitigate CoT inefficiency (Aggarwal and Welleck 2025; Luo et al. 2025), this strategy is vulnerable to reward manipulation through excessive chain elongation (Yuan et al. 2025). Some employ heuristic rules to dynamically activate or deactivate Long CoT modes based on an assessment of problem difficulty (Fang, Ma, and Wang 2025; Jiang et al. 2025a; Liang et al. 2025). While this method successfully reduces deployment costs in practice, it does not resolve the core inefficiency inherent in the CoT process itself (Sui et al. 2025). Another line of research proposes early-termination strategies for Long CoT generation that are predicated on intrinsic confidence metrics (Jiang et al. 2025a; Zhang et al. 2025). The significant drawback to this approach, however, is that prematurely halting reasoning chains on challenging problems can substantially compromise model performance (Liu et al. 2025b).

**Critical Token in LRMs** Recent studies have revealed that Long CoT processes within LLMs are often driven by critical anchor tokens (Gandhi et al. 2025; Vassoyan, Beau, and Plaud 2025; Li et al. 2025). Various techniques have been developed to identify these critical tokens, including entropy-based threshold filtering (Wang et al. 2025b), mutual information estimation (Qian et al. 2025), and contrastive learning (Lin et al. 2025). In natural language, these tokens typically manifest as logical connectives or discourse markers. It has been empirically shown that imposing sampling penalties directly on critical tokens during CoT generation can mitigate model under-thinking (Wang et al. 2025c). This approach, however, risks altering the token probability distribution, which can potentially compromise the final task performance (An et al. 2025). Further studies show that high-entropy tokens trigger switches in reasoning paths, and that focusing reinforcement learning updates exclusively on these anchor points yields superior gains in reasoning capability (Yu et al. 2025).

## 3 Methodology

**Method Overview** We propose the Time-Frequency Advantage Clipping (TFAC) framework. Its core innovation lies in the dynamic reshaping of the advantage clipping mechanism within the Clipped Proximal Policy Optimization (PPO-Clip) (Schulman et al. 2017) algorithm used in reinforcement learning. As illustrated in Fig.1, we replace the fixed clipping threshold in advantage estimation with a training algorithm designed to adaptively adjust the token-level advantage clipping upper bound. This adjustment is based on the historical and current performance of critical tokens.

We delineate the TFAC framework step-by-step. In §3.1 we revisit the PPO algorithm and the limitations of its static advantage clipping mechanism, establishing the motivation for our improvement. Then we detail our methodology for identifying critical tokens based on token entropy and establishing an empirically grounded reasoning budget derived from temporal and frequency dimensions (§3.2). Finally, §3.3 comprehensively presents the concrete implementation within the TFAC framework for performing dynamic advantage clipping guided by token-level time-frequency characteristics.

### 3.1 Clipped Proximal Policy Optimization

We formalize the model's autoregressive generation process as a Markov Decision Process (MDP). In this framework, the state $s_t$ is the sequence of tokens generated up to timestep $t$, the action $a_t$ is the selection of the next token from the vocabulary, and the model itself acts as the policy $\pi_\theta$. Our training data consists of question-answer pairs $(X, Y)$, where $X$ is the input question and $Y$ is a binary label indicating the correctness of the final answer. In each training episode, the current reasoning language model $\pi_\theta$ takes the question $X$ as its initial state and autoregressively generates a sequence of actions (tokens). Upon termination of generation, an external validator grants a sparse terminal reward based on the correctness of the final answer.

This policy is typically optimized using Proximal Policy Optimization (PPO), with the following objective function:

$$L^{PPO}(\theta) = \hat{\mathbb{E}}_t \left[ \min \left( r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{1}$$

where $r_t(\theta)$ is the probability ratio of the new and old policies calculated via importance sampling, and $\hat{A}_t$ is the estimated advantage function at timestep $t$. The stability of PPO hinges on the clip function, which limits the magnitude of the policy update to $(1+\epsilon)\hat{A}_t$ for actions with a positive advantage ($\hat{A}_t > 0$). However, it treats all tokens equally, failing to distinguish between essential thinking steps for the final answer and redundant text that serves only an auxiliary role. This directly leads to a tendency for overthinking on simple problems and potential failure on complex problems due to insufficient exploration. Although existing work attempts to adjust rewards or clipping boundaries for different tokens to aid exploration (Yu et al. 2025), it often lacks a systematic framework for dynamically deciding when and to what extent to intervene.

### 3.2 Critical Token Budgeting

We first define the critical decision points in a reasoning path as the positions where the model's policy exhibits high uncertainty. For a given reasoning trajectory $\tau = \{a_1, a_2, ..., a_T\}$, we calculate the Shannon entropy of the policy network's output distribution at each timestep $t$. When generating token $a_t$, the corresponding state is $s_t = (X, a_1, ..., a_{t-1})$, and the policy entropy is calculated as:

$$H_t = H(P(\cdot|s_t)) = -\sum_{v \in V} P(v|s_t) \log_2 P(v|s_t) \tag{2}$$

It is worth emphasizing that, as pointed out in related work (Wang et al. 2025b), a high entropy value $H_t$ at timestep $t$

does not directly imply that the token $a_t$ ultimately generated at that step is itself highly uncertain. More accurately, it indicates that after reaching state $s_t$ (i.e., having generated the sequence $a_1, ..., a_{t-1}$), the model faces a high degree of uncertainty in deciding which token to generate next. Nevertheless, we still consider the token $a_t$ that is ultimately selected at this critical juncture as the representative of that decision. Specifically, after calculating the entropy values $(H_1, H_2, ..., H_T)$ for all positions in the trajectory, we select the $k$ positions with the highest entropy.

After identifying the key tokens, we quantify them from a temporal dimension to establish an empirical thinking budget. For each input question $X$, we maintain a dataset $D_{\text{success}}(X)$ containing only past successful reasoning trajectories. Crucially, this historical dataset is highly specific to the question. We then calculate a historical success budget $\mu(X, a_*)$ for each question-trajectory-key-token pair $(X, \tau, a_t \in K_\tau)$:

$$\mu(X, a_*) = \begin{cases} \frac{\sum_{\tau' \in D_{\text{success}}(X)} c(\tau', a_*)}{|D_{\text{success}}(X)|} & \text{if } |D_{\text{success}}(X)| > 0 \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

If the dataset contains successful trajectories for question $X$, the budget $\mu(X, a_t)$ is the average number of times the token has been used in past successful experiences. If question $X$ is entirely new (i.e., $D_{\text{success}}(X)$ is empty), its budget is set to positive infinity, which forms the mathematical basis of our exploration-preservation mechanism.

### 3.3 TFAC: Time-Frequency-Aware Dynamic Advantage Clipping

The TFAC framework replaces the fixed, global hyperparameter $\epsilon$ in the traditional PPO algorithm with a dynamic, context-aware clipping parameter $\epsilon_t$. Our method does not distinguish between positive and negative advantages; as long as an action $a_t$ is identified as a member of the critical tokens set $K_\tau$ of the current trajectory $\tau$ (generated from question $X$), i.e., $a_t \in K_\tau$, we apply a dynamic adjustment to its clipping parameter.

We define the dynamic clipping parameter $\epsilon_t$ for each timestep $t$ as follows:

$$\epsilon_t = \begin{cases} \epsilon_{\text{high}} \times \gamma^{\max(0, c(\tau, a_t) - \mu(X, a_t))} & \text{if } a_t \in K_\tau \\ \epsilon_{\text{high}} & \text{otherwise} \end{cases} \tag{4}$$

Here, $\epsilon_{\text{high}}$ is a high clipping value, and $\gamma \in (0, 1]$ is a decay hyperparameter that controls the penalty strength. The exponential term $\gamma^{(\cdot)}$ in this formula is our Time-Frequency Clipping factor, which implements a historically-guided intra-path accumulative decay mechanism:

- **For new problems or within-budget operations:** When encountering a new problem, $\mu(X, a_t) = \infty$, causing the exponent to be consistently 0 and the clipping factor to be $\gamma^0 = 1$. Similarly, for a known problem, as long as the cumulative count of the current token, $c(\tau, a_t)$, does not exceed its historical budget $\mu(X, a_t)$, the exponent is also 0, and the clipping factor remains 1. In both scenarios, $\epsilon_t = \epsilon_{\text{high}}$, meaning TFAC imposes no additional suppression, thereby safeguarding necessary exploration and reasoning within reasonable bounds.
- **For over-budget operations:** Once the real-time count $c(\tau, a_t)$ of a key token exceeds its historical success budget $\mu(X, a_t)$, the exponent becomes a positive integer (representing the number of times it is over budget). At this point, the clipping factor $\gamma^{(\cdot)}$ becomes less than 1. Furthermore, the more this token is repeatedly generated in the path, the larger the exponent becomes, causing $\epsilon_t$ to be exponentially and progressively decayed. This smooth suppression mechanism effectively curbs "overthinking" while avoiding the training instability that can arise from hard gating.

Finally, we integrate this dynamic $\epsilon_t$ into the PPO objective function to constrain both the upper and lower bounds of the policy update:

$$L^{TFAC}(\theta) = \hat{\mathbb{E}}_t \left[ \text{clip}(r_t(\theta), 1 - \epsilon_{low}, 1 + \epsilon_t) \hat{A}_t \right] \tag{5}$$

Through this design, TFAC deeply integrates historical success experience with current reasoning behavior, achieving an adaptive, smooth, and robust adjustment of the advantage at critical reasoning nodes.

## 4 Experiments

### 4.1 Experimental Setup

We conduct reinforcement learning using two models: Deepseek-R1-Distill-32B, abbreviated as R1-Distill-32B, and Qwen3-8B (Yang et al. 2025). Specifically, the R1-Distill-32B model is derived through supervised fine-tuning of Qwen2.5-32B-Base (Yang et al. 2024) with reasoning data generated by Deepseek-R1 (DeepSeek-AI et al. 2025). For the Qwen3-8B model, the thinking instruction template is adopted by default during training to guide the model in generating extended thought chains.

**Datasets** We perform cleaning on the Eurus-2-RL-Data dataset (Cui et al. 2025a). Any problem where the Qwen2.5-7B-Instruct (Yang et al. 2024) can provide the correct answer with a probability exceeding 50% is filtered out. Following this processing, a final dataset comprising 212k mathematical problems and 15k coding problems is obtained for reinforcement learning fine-tuning of the policy model. The correctness of the model's final generated answers is jointly determined by math verify[1] and Qwen3-30B-A3B (Yang et al. 2025) serving as a generative judge. In reinforcement learning, whether the answer is correct or not serves as the outcome reward.

**Baseline Methods** The baseline methods are reinforcement learning algorithms which are improved for training efficient reasoning models based on outcome rewards. Firstly, an improved DAPO-based algorithm (Wang et al. 2025b; Yu et al. 2025) that only updates gradients for 20% of high-entropy tokens (DAPO w. critical token). Additionally, there are LCPO (Aggarwal and Welleck 2025) which based on fixed-length

[1] https://github.com/huggingface/Math-Verify

| METHOD | MATH-500 | | AIME 2024 | | AIME 2025 | | LiveCodeBench | |
|---|---|---|---|---|---|---|---|---|
| | Avg@32 ↑ | Len@32 ↓ | Avg@32 ↑ | Len@32 ↓ | Avg@32 ↑ | Len@32 ↓ | Avg@32 ↑ | Len@32 ↓ |
| *Advanced Large Reasoning Models* | | | | | | | | |
| O3-2025-0416 | 99.40 | 3134 | 91.61 | 10901 | 88.95 | 12276 | 77.31 | 10371 |
| Deepseek-R1-0528 | 98.80 | 4134 | 91.43 | 12846 | 87.50 | 18235 | 73.31 | 15265 |
| Qwen3-235B-A30B | 98.60 | 5134 | 85.72 | 12592 | 81.55 | 17288 | 70.59 | 17358 |
| *RLVR from Qwen3-8B* | | | | | | | | |
| Base LRM | 94.25 | 7718 | 75.80 | 15226 | 67.30 | 17322 | 57.52 | 13128 |
| LCPO | 92.81 | 6231 | 71.60 | 10353 | 61.90 | 12164 | 51.83 | 9845 |
| HAPO | 92.45 | 6141 | 70.30 | 9328 | 58.00 | 11353 | 52.31 | 9793 |
| O1-Pruner | 91.53 | 5815 | 68.00 | 8315 | 59.40 | 10136 | 51.46 | 8238 |
| DAPO w. forking token | 95.32 | 8092 | 77.60 | 17428 | 69.50 | 20276 | 58.52 | 16259 |
| TFAC (Ours) | 94.96 | 5662 | 77.80 | 14826 | 69.50 | 15390 | 58.15 | 12382 |
| + dynamic sampling | **95.45** | **5535** | **78.20** | 13245 | **69.80** | 14976 | **58.59** | 10382 |
| *RLVR from R1-Distill-32B* | | | | | | | | |
| Base LRM | 94.27 | 7134 | 72.60 | 13326 | 49.30 | 18276 | 54.35 | 15382 |
| LCPO | 92.18 | 5192 | 68.40 | 10304 | 44.10 | 14748 | 47.23 | 11690 |
| HAPO | 91.26 | 4265 | 67.50 | 9357 | 43.60 | 13269 | 45.55 | 10345 |
| O1-Pruner | 91.45 | 4439 | 67.00 | 9135 | 42.50 | 12738 | 44.48 | 10523 |
| DAPO w. forking token | 96.90 | 8885 | 75.40 | 15979 | 53.20 | 21332 | 57.09 | 18997 |
| TFAC (Ours) | 97.13 | 6546 | 76.50 | 11892 | 55.50 | 16138 | 58.58 | 13293 |
| + dynamic sampling | **97.57** | 6326 | **76.60** | 11377 | **55.60** | 16092 | **58.71** | 13653 |

Table 1: Main results on mathematical and coding reasoning benchmarks. We compare our method against strong baselines using two different base models. For each benchmark, we report the average pass rate (Avg@32) and average solution length (Len@32). The highest performance achieved by our proposed approach on each metric is highlighted in **bold**.

rewards and penalties, O1-Pruner (Luo et al. 2025) that incorporates the length of generated trajectories as a correction to the reward objective, and HAPO (Huang, Zhang, and Cardie 2025) that imposes a length penalty by considering the shortest length of historically correct sampled trajectories.

**Evaluation** The policy model's performance is primarily evaluated using the Avg@K and Len@K. Avg@K denotes the frequency with which the policy model generates the correct answer across K generations. Len@K represents the average length of the policy model in K generations. The default generation temperature $T$ is 0.6.

The evaluation benchmarks include: AIME 2024, AIME 2025[2], MATH-500 (Lightman et al. 2023), LiveCodeBench (v5, 2024.10–2025.02) (Jain et al. 2025), and an out-of-distribution dataset GPQA (Rein et al. 2024).

## 4.2 Main Experimental Results

**Benchmark Performance** As shown in Table 1, our method demonstrates superior performance across both models and all benchmark datasets, although the generated chain-of-thought length is not always optimal. Key findings are summarized as follows: 1) Superior Performance on Avg@32: The proposed approach outperforms all baseline methods on the Avg@32 metric. Compared to DAPO w. forking token, it achieves absolute improvements of 2.3 points and 3.1 points in average Avg@32, while simultaneously reducing Len@32 by 35% and 28%, respectively. 2) Trade-off with Length Control: Methods employing explicit length control exhibit a significant advantage in regulating chain-of-thought length. However, this control substantially impairs model performance. While some advanced reasoning models (notably O3-2025-0416) achieve a better balance between performance and length, limited information is publicly available regarding their training specifics. 3) Dataset-Specific Effectiveness: The performance gains observed with our method on the MATH-500 dataset are less pronounced compared to its advantages on more challenging datasets like AIME. This suggests our approach may be more effective for training models to explore complex problems while simultaneously enhancing the efficiency of long reasoning chains.

**Training Dynamic** We analyze the training dynamics of our proposed model, TFAC, in comparison to several baseline methods, as illustrated in four distinct charts. First, regarding generation entropy, Fig.2a illustrates that our method maintains a moderate entropy level throughout the training process. In contrast, methods that employ length control, such as LCPO and HAPO, exhibit excessively low entropy, potentially limiting the model's generation exploration. Conversely, the DAPO method shows excessively high entropy, thus risking training instability. The performance of TFAC suggests that it strikes an effective balance, preserving the model's generative diversity and exploratory capabilities while avoiding the issues that arise from either excessively low or high entropy. Second, in terms of generation length, the results

[2]https://maa.org/maa-invitational-competitions

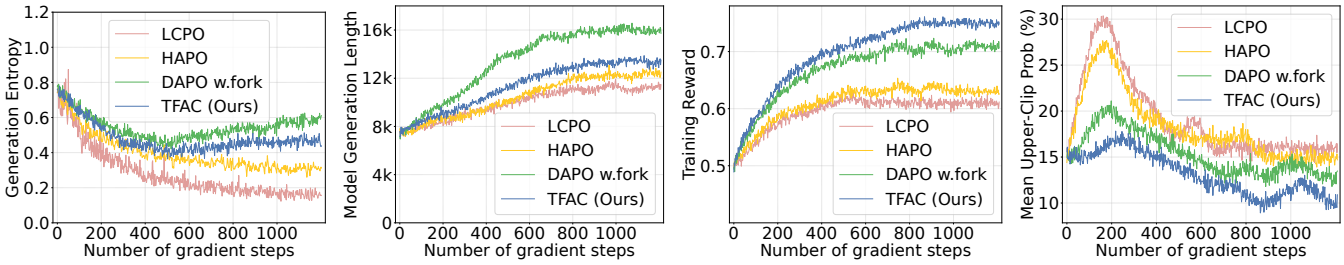
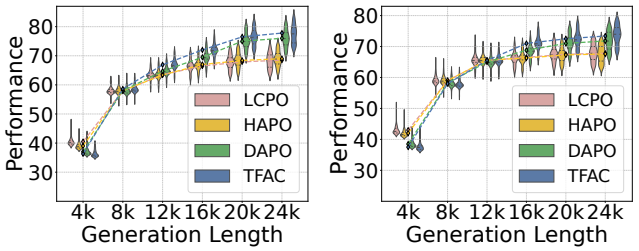



Figure 2: Analysis of Qwen3-8B training dynamics in reinforcement learning. From left to right, they are the model's generation entropy, generated token sequence length, reward curve, and average upper clipping probability during the training process.

in Fig.2b show that the text generated by TFAC is of an intermediate length. This balanced characteristic allows the model to avoid producing the overly short content typical of LCPO and HAPO and the excessively long sequences that characterize the DAPO method. Finally, our analysis of the training reward curve and mean upper-clip probability further highlights the superiority of our approach. As Fig.2c depicts, the training reward curve for TFAC converges to the highest value among all compared methods, demonstrating its leading performance on the task. Concurrently, Fig.2d reveals that TFAC has the lowest advantage upper-clip probability. This indicates a more stable advantage estimation and a smoother policy update process, validating that our method effectively enhances overall model performance while ensuring robust exploration capabilities.

**Token Budget Scaling Trend** We benchmark two baseline models on mathematical reasoning datasets under progressively expanding generation length budgets, referencing prior work. As shown in Fig.3, the performance of all methods exhibits an approximate logarithmic growth pattern. At shorter generation lengths (approximately 8k tokens), our model underperforms compared to length-controlled models. This can be attributed to the strong learning algorithms employed by the length-control methods, which explicitly emphasize length reduction as a primary training objective. Conversely, our approach demonstrates superior performance under more generous sequence length budgets.




(a) R1-Distill-32B (b) Qwen3-8B

Figure 3: Model performance curves across methods under scaling token budget. Performance values represent the average of Avg@32 scores across the MATH-500, AIME 2024, and AIME 2025 datasets.

# 5 Analysis and Discussion

## 5.1 Advantage Clipping or Reward Shaping

| Method | Avg@32 ↑ | Len@32 ↓ | $A/L$ ↑ |
|---|---|---|---|
| Qwen3-8B | 71.6 | 16274 | 4.40 |
| + TFRS$_{\lambda=0.5}$ | 72.3 | 16018 | 4.51 |
| + TFRS$_{\lambda=0.8}$ | 71.9 | 15945 | 4.50 |
| + TFAC$_{\gamma=0.5}$ | 72.4 | **15090** | 4.80 |
| + TFAC$_{\gamma=0.8}$ | **73.9** | 15108 | **4.90** |
| R1-Distill-32B | 60.3 | 15801 | 3.82 |
| + TFRS$_{\lambda=0.5}$ | 62.9 | 15552 | 4.04 |
| + TFRS$_{\lambda=0.8}$ | 62.2 | 15435 | 4.02 |
| + TFAC$_{\gamma=0.5}$ | 64.6 | **13831** | 4.67 |
| + TFAC$_{\gamma=0.8}$ | **66.0** | 14015 | **4.70** |

Table 2: Average performance of TFAC and TFRS on two AIME datasets. We conducted experiments on key hyperparameter variations for the two methods, where the A/L metric is calculated as Avg@32 divided by Len@32.

To validate the efficacy of our core design choice of intervening at the advantage function level, we compare TFAC against a strong alternative baseline: Temporal-Frequency Reward Shaping (TFRS). TFRS applies the identical temporal-frequency penalty logic directly to the terminal reward signal instead of the advantage function. In this approach, the final reward $R$ is modified after trajectory $\tau$ completion using the following shaping function:

$$R' = R - \lambda \sum_{a_* \in K_\tau} \max\Big(0, c(\tau, a_*) - \mu(X, a_*)\Big) \quad (6)$$

where $R'$ is the shaped reward, $\lambda$ is a penalty scalar.

As demonstrated in Table 2, TFAC consistently and significantly outperforms the TFRS baseline. Across both the Qwen3-8B and R1-Distill-32B models, TFAC$_{\gamma=0.8}$ achieve the highest accuracy while simultaneously producing the shortest sequences, leading to a substantially better accuracy/length efficiency ratio. The immediate, localized feedback of TFAC mitigates the credit assignment problem inherent in TFRS's delayed, trajectory-wide penalties, enabling the training of superior and more efficient policies.

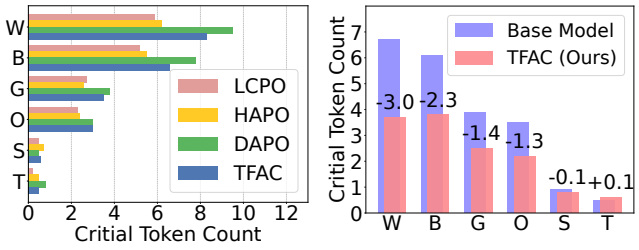
### 5.2 Out-of-Domain Performance

| Methods | Entropy | Avg@32 ↑ | Len@32 ↓ | $A/L$ ↑ |
|---|---|---|---|---|
| *Base Model: Qwen3-8B* | | | | |
| LCPO | 0.47 | 60.31 | 3435 | 17.56 |
| HAPO | 0.56 | 61.35 | 3689 | 16.63 |
| DAPO w. fork | 0.66 | 64.92 | 3828 | 16.96 |
| TFAC (Ours) | 0.60 | **66.65** | 3645 | **18.29** |
| *Base Model: R1-Distill-32B* | | | | |
| LCPO | 0.46 | 58.59 | 4385 | 13.36 |
| HAPO | 0.50 | 59.04 | 4579 | 12.89 |
| DAPO w. fork | 0.64 | 63.52 | 4832 | 13.15 |
| TFAC (Ours) | 0.55 | **65.35** | 4643 | **14.08** |

Table 3: Out-of-domain (OOD) performance on the GPQA dataset. Our method, TFAC, outperforms all baselines by achieving the highest accuracy (Avg@32) and the best accuracy-to-length efficiency (A/L), demonstrating its superior generalization to unseen problems.

We use the GPQA dataset to evaluate the models' OOD generalization capabilities. As illustrated in Table 3, we reveal the trade-off between accuracy and inference cost by calculating the efficiency score (A/L). TFAC achieves the highest efficiency score on both models, indicating that it provides the best balance between performance and computational cost. This efficiency is supported by the entropy metrics, where TFAC maintains a moderate entropy level (0.60 and 0.55). This balanced approach enables effective exploration, avoiding the performance degradation seen in low-entropy, length-constrained models (LCPO, HAPO) and the potential instability of DAPO with only trained on high-entropy tokens.

### 5.3 Critical Token Distribution




(a) Critical Token Distribution (b) Critical Token Compare

Figure 4: Distribution of critical tokens per 1k tokens after training the Qwen3-8B model using different methods, and a comparison of the reduction in token quantity of our method compared to the initial policy model.

Motivated by previous work (Tu et al. 2025), we calculate the frequency of critical tokens associated with reasoning within AIME 2024. This analysis aims to characterize the distinct reasoning patterns that different methodologies exhibit. As Fig.4a illustrates, our method shows a moderate frequency of reflection tokens compared to other approaches, while the frequencies of summarization tokens remain relatively consistent across all methods. This suggests that by maintaining a moderate level of reflection token usage, our approach strikes an optimal balance between insufficient contemplation and excessive deliberation. In addition, Fig.4b reveals that our method significantly reduces the reflection token frequency after training. This observation indicates that appropriately reducing reflection tokens enhances the efficiency of the language model's chain-of-thought reasoning and effectively mitigates the issue of overthinking.

### 5.4 Ablation Study

| Methods | Entropy | Avg@32 ↑ | Len@32 ↓ |
|---|---|---|---|
| *Base Model: Qwen3-8B* | | | |
| – | 0.50 | 73.9 | 15108 |
| w/ dynamic | 0.48 | 74.2 | 15589 |
| w/o time-count | 0.55 | 71.4 | 17241 |
| w/o freq-clip | 0.59 | 70.8 | 19183 |
| *Base Model: R1-Distill-32B* | | | |
| – | 0.41 | 71.2 | 12718 |
| w/ dynamic | 0.42 | 72.9 | 12365 |
| w/o time-count | 0.53 | 68.4 | 17446 |
| w/o freq-clip | 0.59 | 68.9 | 18523 |

Table 4: Ablation study of TFAC's key components. Removing either the historical token count (w/o time-count) or the intra-path frequency clipping (w/o freq-clip) significantly degrades performance.

The ablation study in Table 4 highlights the critical role of TFAC's core components. Removing either the historical critical token count (*w/o time-count*) or in-path advantage clipping (*w/o freq-clip*) significantly degrades accuracy and increases solution length. The performance drop is most severe without frequency clipping, emphasizing its importance in preventing inefficient reasoning. The full TFAC model substantially outperforms the ablated versions, confirming its components work synergistically to balance accuracy and computational cost. Additionally, dynamic sampling (*w/ dynamic*) provides further incremental gains.

## 6 Conclusion

We propose TFAC (Time-Frequency token Advantage Clipping), a novel framework designed to address inefficiency issues in LRMs' long CoT reasoning. TFAC implements a dynamic token-level advantage clipping mechanism which is guided by two dimensions: intra-path critical token frequency and token budget allocation based on historical successful trajectories. It effectively suppresses redundant deliberation while preserving the LRM's exploratory capacity. Experiments conducted across two LRMs demonstrate that TFAC significantly enhances accuracy compared to robust baseline models while drastically reducing inference costs.

## Acknowledgements

The research of Leszek Rutkowski was supported by the program "Excellence initiative - research university" for the AGH University of Krakow, as well as the ARTIQ project UMO-2021/01/2/ST6/00004 and ARTIQ/0004/2021, and by Polish Ministry of Science and Higher Education funds assigned to the AGH University of Krakow. Dr Tao's research is partially supported by NTU RSR and Start Up Grants.

## References

Aggarwal, P.; and Welleck, S. 2025. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. *CoRR*, abs/2503.04697.

An, S.; Wang, R.; Zhou, T.; and Hsieh, C.-J. 2025. Don't Think Longer, Think Wisely: Optimizing Thinking Dynamics for Large Reasoning Models. arXiv:2505.21765.

Bai, Y.; Jones, A.; Ndousse, K.; et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862.

Bao, R.; Yu, D.; Fan, K.; and Liao, M. 2025. Fixing Distribution Shifts of LLM Self-Critique via On-Policy Self-Play Training. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 17680–17700. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Bogdan, P. C.; Macar, U.; Nanda, N.; and Conmy, A. 2025. Thought Anchors: Which LLM Reasoning Steps Matter? *CoRR*, abs/2506.19143.

Bowman, S. R.; Hyun, J.; Perez, E.; et al. 2022. Measuring Progress on Scalable Oversight for Large Language Models. *CoRR*, abs/2211.03540.

Chen, W.; Yuan, J.; Jin, T.; Ding, N.; Chen, H.; Liu, Z.; and Sun, M. 2025. The Overthinker's DIET: Cutting Token Calories with DIfficulty-AwarE Training. arXiv:2505.19217.

Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2024. Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. *CoRR*, abs/2412.21187.

Cui, G.; Yuan, L.; Wang, Z.; Wang, H.; Li, W.; He, B.; Fan, Y.; Yu, T.; Xu, Q.; Chen, W.; Yuan, J.; Chen, H.; Zhang, K.; Lv, X.; Wang, S.; Yao, Y.; Han, X.; Peng, H.; Cheng, Y.; Liu, Z.; Sun, M.; Zhou, B.; and Ding, N. 2025a. Process Reinforcement through Implicit Rewards. *CoRR*, abs/2502.01456.

Cui, G.; Zhang, Y.; Chen, J.; Yuan, L.; Wang, Z.; Zuo, Y.; Li, H.; Fan, Y.; Chen, H.; Chen, W.; Liu, Z.; Peng, H.; Bai, L.; Ouyang, W.; Cheng, Y.; Zhou, B.; and Ding, N. 2025b. The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models. *CoRR*, abs/2505.22617.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948.

DeepSeek-AI; Liu, A.; Feng, B.; Xue, B.; et al. 2024. DeepSeek-V3 Technical Report. *CoRR*, abs/2412.19437.

Fang, G.; Ma, X.; and Wang, X. 2025. Thinkless: LLM Learns When to Think. *CoRR*, abs/2505.13379.

Gandhi, K.; Chakravarthy, A.; Singh, A.; Lile, N.; and Goodman, N. D. 2025. Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs. *CoRR*, abs/2503.01307.

Huang, C.; Zhang, Z.; and Cardie, C. 2025. HAPO: Training Language Models to Reason Concisely via History-Aware Policy Optimization. *CoRR*, abs/2505.11225.

Jaech, A.; Kalai, A.; Lerer, A.; et al. 2024. OpenAI o1 System Card. *CoRR*, abs/2412.16720.

Jain, N.; Han, K.; Gu, A.; Li, W.; Yan, F.; Zhang, T.; Wang, S.; Solar-Lezama, A.; Sen, K.; and Stoica, I. 2025. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Jiang, G.; Quan, G.; Ding, Z.; Luo, Z.; Wang, D.; and Hu, Z. 2025a. FlashThink: An Early Exit Method For Efficient Reasoning. *CoRR*, abs/2505.13949.

Jiang, L.; Wu, X.; Huang, S.; Dong, Q.; Chi, Z.; Dong, L.; Zhang, X.; Lv, T.; Cui, L.; and Wei, F. 2025b. Think Only When You Need with Large Hybrid-Reasoning Models. *CoRR*, abs/2505.14631.

Li, D.; Cao, S.; Griggs, T.; Liu, S.; Mo, X.; Tang, E.; Hegde, S.; Hakhamaneshi, K.; Patil, S. G.; Zaharia, M.; Gonzalez, J. E.; and Stoica, I. 2025. LLMs Can Easily Learn to Reason from Demonstrations Structure, not content, is what matters! *CoRR*, abs/2502.07374.

Liang, G.; Zhong, L.; Yang, Z.; and Quan, X. 2025. ThinkSwitcher: When to Think Hard, When to Think Fast. *CoRR*, abs/2505.14183.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. arXiv:2305.20050.

Lin, Z.; Liang, T.; Xu, J.; Lin, Q.; Wang, X.; Luo, R.; Shi, C.; Li, S.; Yang, Y.; and Tu, Z. 2025. Critical Tokens Matter: Token-Level Contrastive Estimation Enhances LLM's Reasoning Capability. arXiv:2411.19943.

Liu, W.; Zhou, R.; Deng, Y.; Huang, Y.; Liu, J.; Deng, Y.; Zhang, Y.; and He, J. 2025a. Learn to Reason Efficiently with Adaptive Length-based Reward Shaping. *CoRR*, abs/2505.15612.

Liu, Y.; Wu, J.; He, Y.; Gao, H.; Chen, H.; Bi, B.; Zhang, J.; Huang, Z.; and Hooi, B. 2025b. Efficient Inference for Large Reasoning Models: A Survey. *CoRR*, abs/2503.23077.

Luo, H.; Shen, L.; He, H.; Wang, Y.; Liu, S.; Li, W.; Tan, N.; Cao, X.; and Tao, D. 2025. O1-Pruner: Length-Harmonizing Fine-Tuning for O1-Like Reasoning Pruning. *CoRR*, abs/2501.12570.

Ma, X.; Wan, G.; Yu, R.; Fang, G.; and Wang, X. 2025. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, 6025–6035. Association for Computational Linguistics.

McAleese, N.; Pokorny, R. M.; Uribe, J. F. C.; Nitishinskaya, E.; Trebacz, M.; and Leike, J. 2024. LLM Critics Help Catch LLM Bugs. *CoRR*, abs/2407.00215.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Pallagani, V.; Muppasani, B. C.; Roy, K.; Fabiano, F.; Loreggia, A.; Murugesan, K.; Srivastava, B.; Rossi, F.; Horesh, L.; and Sheth, A. P. 2024. On the Prospects of Incorporating Large Language Models (LLMs) in Automated Planning and Scheduling (APS). In Bernardini, S.; and Muise, C., eds., *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2024, Banff, Alberta, Canada, June 1-6, 2024*, 432–444. AAAI Press.

Peng, S.; Hu, X.; Yi, Q.; Zhang, R.; Guo, J.; Huang, D.; Tian, Z.; Chen, R.; Du, Z.; Guo, Q.; Chen, Y.; and Li, L. 2024. Hypothesis, Verification, and Induction: Grounding Large Language Models with Self-Driven Skill Learning. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 14599–14607. AAAI Press.

Qian, C.; Liu, D.; Wen, H.; Bai, Z.; Liu, Y.; and Shao, J. 2025. Demystifying Reasoning Dynamics with Mutual Information: Thinking Tokens are Information Peaks in LLM Reasoning. *CoRR*, abs/2506.02867.

Qu, X.; Li, Y.; Su, Z.; Sun, W.; Yan, J.; Liu, D.; Cui, G.; Liu, D.; Liang, S.; He, J.; Li, P.; Wei, W.; Shao, J.; Lu, C.; Zhang, Y.; Hua, X.; Zhou, B.; and Cheng, Y. 2025. A Survey of Efficient Reasoning for Large Reasoning Models: Language, Multimodality, and Beyond. *CoRR*, abs/2503.21614.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *First Conference on Language Modeling*.

Saunders, W.; Yeh, C.; Wu, J.; Bills, S.; Ouyang, L.; Ward, J.; and Leike, J. 2022. Self-critiquing models for assisting human evaluators. *CoRR*, abs/2206.05802.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *CoRR*, abs/2408.03314.

Sui, Y.; Chuang, Y.; Wang, G.; Zhang, J.; Zhang, T.; Yuan, J.; Liu, H.; Wen, A.; Zhong, S.; Chen, H.; and Hu, X. B. 2025. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. *CoRR*, abs/2503.16419.

Tu, S.; Lin, J.; Zhang, Q.; Tian, X.; Li, L.; Lan, X.; and Zhao, D. 2025. Learning When to Think: Shaping Adaptive Reasoning in R1-Style Models via Multi-Stage RL. arXiv:2505.10832.

Valmeekam, K.; Marquez, M.; Sreedharan, S.; and Kambhampati, S. 2023. On the Planning Abilities of Large Language Models : A Critical Investigation. arXiv:2305.15771.

Vassoyan, J.; Beau, N.; and Plaud, R. 2025. Ignore the KL Penalty! Boosting Exploration on Critical Tokens to Enhance RL Fine-Tuning. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, 6108–6118. Association for Computational Linguistics.

Wang, J.; Li, J.; Wu, L.; and Zhang, M. 2025a. Efficient Reasoning for LLMs through Speculative Chain-of-Thought. *CoRR*, abs/2504.19095.

Wang, S.; Yu, L.; Gao, C.; Zheng, C.; Liu, S.; Lu, R.; Dang, K.; Chen, X.; Yang, J.; Zhang, Z.; Liu, Y.; Yang, A.; Zhao, A.; Yue, Y.; Song, S.; Yu, B.; Huang, G.; and Lin, J. 2025b. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. *CoRR*, abs/2506.01939.

Wang, Y.; Liu, Q.; Xu, J.; Liang, T.; Chen, X.; He, Z.; Song, L.; Yu, D.; Li, J.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2025c. Thoughts Are All Over the Place: On the Underthinking of o1-Like LLMs. *CoRR*, abs/2501.18585.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yang, A.; Li, A.; Yang, B.; et al. 2025. Qwen3 Technical Report. *CoRR*, abs/2505.09388.

Yang, A.; Yang, B.; Zhang, B.; et al. 2024. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; Lin, H.; Lin, Z.; Ma, B.; Sheng, G.; Tong, Y.; Zhang, C.; Zhang, M.; Zhang, W.; Zhu, H.; Zhu, J.; Chen, J.; Chen, J.; Wang, C.; Yu, H.; Dai, W.; Song, Y.; Wei, X.; Zhou, H.; Liu, J.; Ma, W.; Zhang, Y.; Yan, L.; Qiao, M.; Wu, Y.; and Wang, M. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *CoRR*, abs/2503.14476.

Yuan, D.; Xie, T.; Huang, S.; Gong, Z.; Zhang, H.; Luo, C.; Wei, F.; and Zhao, D. 2025. Efficient RL Training for Reasoning Models via Length-Aware Optimization. *CoRR*, abs/2505.12284.

Zhang, A.; Chen, Y.; Pan, J.; Zhao, C.; Panda, A.; Li, J.; and He, H. 2025. Reasoning Models Know When They're Right: Probing Hidden States for Self-Verification. arXiv:2504.05419.