Sprint or Delve: A Distribution-Aware Approach to Efficient Reasoning

Zehui Ling[1,3], **Deshu Chen**[1,2,3], **Hongwei Zhang**[1,2,3], **Yifeng Jiao**[1,3], **Xin Guo**[1,3], **Zenglin Xu**[1,3] and **Yuan Cheng**[1,3]
[1]Artificial Intelligence Innovation and Incubation Institute, Fudan University
[2]School of Data Science, Fudan University
[3]Shanghai Academy of Artificial Intelligence for Science
a51325516@gmail.com, 22110980001@m.fudan.edu.cn, zhanghw.hongwei@gmail.com, johnsonzn@163.com, guoxin@sais.org.cn, zenglin@gmail.com, cheng_yuan@fudan.edu.cn

Abstract

Reasoning chains in Large Language Models (LLMs) often exhibit heavy-tailed length distributions, yet existing efficiency methods rely on suboptimal linear penalties that suppress complex reasoning, limiting both accuracy and generalization. To address this, we first empirically observe that reasoning lengths are well approximated by a log-normal distribution, and provide an intuitive explanation for this phenomenon. Based on this insight, we propose the Powered Length Penalty (PLP), an adaptive regularizer that penalizes redundancy in short sequences while gradually reducing penalties for longer sequences, preserving deep reasoning. Trained solely on the elementary GSM8K dataset, PLP significantly improves reasoning efficiency by reducing inference costs on GSM8K and MATH500, while simultaneously enhancing accuracy on the challenging AIME2024 benchmark. Furthermore, PLP transfers effectively to diverse domains, including MMLU and GPQA. These results suggest that modeling reasoning length distributions and adapting penalties accordingly can mitigate the typical trade-off between efficiency and performance, enabling more reliable and cost-effective reasoning across tasks.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, largely driven by Chain-of-Thought (CoT) prompting mechanisms [Zhou *et al.*, 2023; Chen *et al.*, 2024; Nye *et al.*, 2022; Wei *et al.*, 2022]. However, this performance enhancement often comes with a relatively static computational paradigm: in practice, models tend to allocate similar generation budgets, measured in output token length, regardless of the underlying problem difficulty. For instance, current state-of-the-art reasoning models, such as DeepSeek-R1 [Guo *et al.*, 2025], often generate extensive reasoning chains even for trivial arithmetic instances commonly found in reasoning benchmarks, resulting in significant redundancy and inference latency [Wan *et al.*, 2024; Sprague *et al.*, 2025]. This inefficiency highlights a critical misalignment between the model's computational expenditure and the actual difficulty of the task.

To mitigate this overhead, recent research has explored methods to incentivize concise reasoning [Sui *et al.*, 2025]. While prompt engineering [Han *et al.*, 2024; Ding *et al.*, 2024] and supervised fine-tuning on variable-length data [Xia *et al.*, 2025b; Ye *et al.*, 2025; Ma *et al.*, 2025] offer partial solutions, Reinforcement Learning (RL) with length-based reward shaping has emerged as a promising direction [Arora and Zanette, 2025; Luo *et al.*, 2025; Li *et al.*, 2025a; Xiang *et al.*, 2025; Liu *et al.*, 2025; Zhang *et al.*, 2025; Li *et al.*,].

Despite their success, most existing approaches rely on **linear or normalized length penalties**. We argue that such formulations are fundamentally limited because they ignore the variation in problem difficulty. Blindly compressing reasoning chains may lead to more concise responses for simple problems; however, applying the same degree of compression to complex tasks often degrades the model's reasoning capability and can even **suppress its ability to perform deep, multi-step reasoning**. This leads to a critical dilemma: a penalty strong enough to compress simple tasks will inadvertently suppress the deep reasoning required for hard tasks, whereas a penalty weak enough to preserve accuracy on hard tasks fails to reduce redundancy on easy ones. This distributional mismatch raises an important research question:

Can we design a length penalty mechanism that naturally adapts to the variance inherent in reasoning chains of varying complexities?

Empirically, our analysis of reasoning chain lengths yields two primary observations.

First, we find that reasoning length serves as a quantifiable proxy for cognitive complexity. As illustrated in Figure 1, there is a clear positive correlation between the length of reference solutions and problem difficulty across varying datasets. This aligns with established findings in cognitive science, where complex problems necessitate deeper, more structured, and multi-step reasoning processes [Kahneman, 2011; Perez *et al.*, 2020].

Second, we observe significant variability in the reasoning lengths generated for identical problems. Rather than clustering tightly, the length distribution for a single prompt exhibits a pronounced heavy-tailed behavior. Through statistical fitting, we demonstrate that this variability is well-

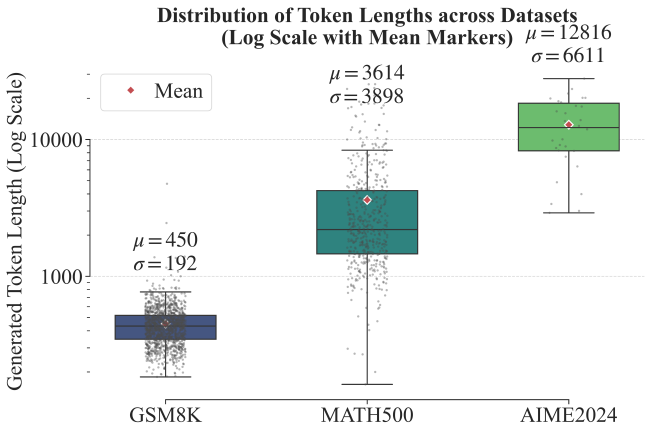


Figure 1: Distribution of reasoning chain lengths from DeepSeek-R1-Distill-Qwen-7B across three benchmarks. The visualizations on a logarithmic scale highlight the multiplicative nature of reasoning depth as task difficulty increases.

approximated by a Log-Normal distribution across diverse benchmarks.

Motivated by above findings, we treat reasoning length as an implicit indicator of task complexity. In contrast to prior approaches, we design a reinforcement learning framework that leverages the variance structure of the Log-Normal distribution to adaptively regulate reasoning length. Rather than imposing a fixed compression strength, our approach estimates task difficulty from the generated response length itself and dynamically adjusts the penalty applied to the model. This design aims to reduce redundant computation on simple problems while preserving, or even enhancing, performance on more complex reasoning tasks.

Our contributions can be summarized as follows:

- We empirically demonstrate that reasoning lengths across diverse models approximate a Log-Normal distribution, and we propose a tree-structured divide-and-conquer framework to explain this phenomenon.
- We propose the **Powered Length Penalty (PLP)**, a reinforcement learning objective **motivated by the variance structure of the Log-Normal distribution**. PLP scales penalty strength inversely with intrinsic task complexity, enabling models to internalize problem difficulty without relying on static heuristics or additional inference-time overhead.
- Extensive experiments against multiple baselines demonstrate that PLP achieves consistent improvements in reasoning efficiency on both mathematical and non-mathematical tasks, while maintaining or even enhancing accuracy.

2 Related Work

Large Language Models for Reasoning. In recent years, large language models have demonstrated remarkable potential in tackling complex reasoning tasks. Pioneering approaches like CoT and Tree-of-Thought [Yao *et al.*, 2023] have enabled these models to perform multi-step logical inference through step-by-step reasoning processes. Meanwhile, models such as ChatGPT [Achiam *et al.*, 2023], DeepSeek-R1 QwQ-preview , and Kimi have further enhanced their reasoning abilities via large-scale reinforcement learning, equipping them with advanced capabilities like branching and validation, which substantially boost their overall reasoning performance.

Efficient reasoning. Although reasoning models and the CoT approach enhance model reasoning capabilities, they also entail increased computational costs. To address this, various methods have been proposed to improve reasoning efficiency. Token-Budget [Han *et al.*, 2024] estimates the minimum number of tokens needed to answer a question and incorporates a prompt to guide the model to keep its reasoning within this token limit, thereby reducing computational overhead. However, this method incurs additional token usage when estimating the problem's difficulty. TokenSkip [Xia *et al.*, 2025b] evaluates the importance of each token in the response, marks them accordingly, and fine-tunes the model on a dataset where a portion of less important tokens are removed. This often leads to answers that omit conjunctions, resulting in less coherent responses. Another approach, training language models for efficient reasoning [Arora and Zanette, 2025], modifies the reward function in RLOO by adding a length penalty based on the number of actions α, followed by reinforcement learning on a mixed dataset. However, it does not differentiate the length penalty between difficult and easy samples. Although the aforementioned methods effectively reduce the number of tokens used by the model, for complex problems the compressed reasoning chains are often insufficient to support correct answers, leading to a decline in accuracy. Motivated by this limitation, we propose the PLP, which aims to reduce the model's output only on simple problems while preserving its capacity for deep reasoning on complex tasks.

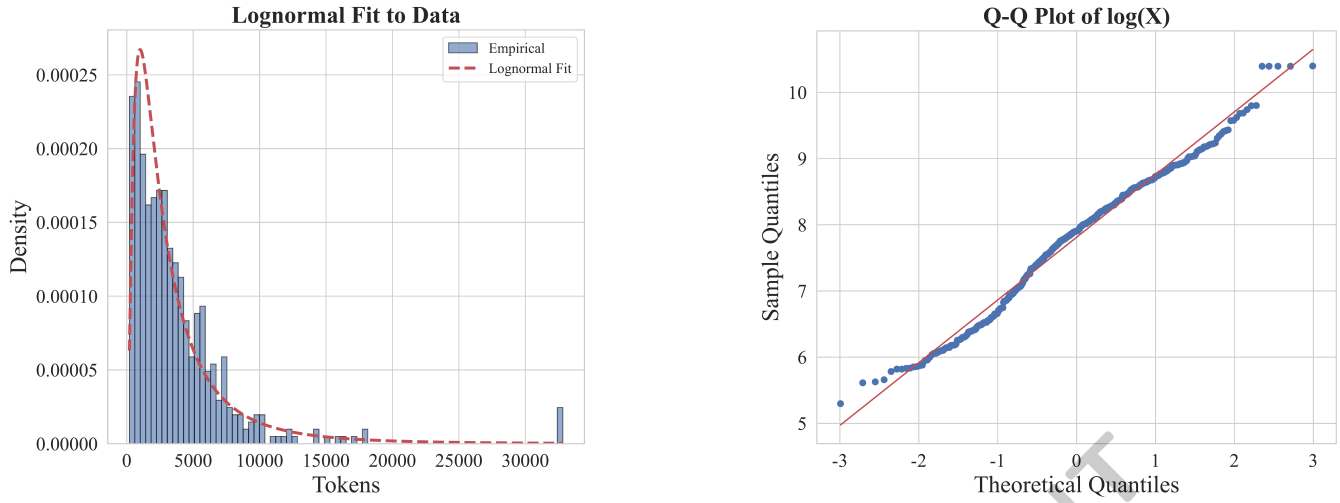
3 Methodology

3.1 Statistical Perspective on Reasoning Length

To motivate the design of our distribution-aware penalty, we first analyze the statistical properties of reasoning chain generation.

Empirical Observation of Length Variability. We first examine the distribution of reasoning lengths generated by LLMs when solving complex problems. By performing Monte Carlo sampling to generate 500 distinct responses for the same prompt using DeepSeek-R1-Distill-Qwen-7B on a single representative problem from the GSM8K dataset, we observe significant variability in the physical length of reasoning chains. Rather than clustering tightly around a mean, the lengths exhibit a pronounced heavy-tailed behavior.

As illustrated in Figure 2, this empirical distribution is robustly approximated by a **Log-Normal distribution**. The histogram (Figure 2a) displays the characteristic positive skew, while the Q-Q plot (Figure 2b) confirms the goodness-of-fit against the theoretical quantiles. The Kolmogorov–Smirnov (K-S) test further supports this fit with a statistic of 0.0597 and a p-value of 0.0544. We note a minor



(a) Histogram of Reasoning Lengths

(b) Q-Q Plot Analysis

Figure 2: Empirical distribution of reasoning token lengths. Results are derived from 500 independent sampling runs using DeepSeek-R1-Distill-Qwen-7B on a single representative problem from the GSM8K dataset. (a) The histogram exhibits a heavy-tailed pattern consistent with Log-Normal behavior. (b) The Q-Q plot confirms the Log-Normal fit. The Kolmogorov–Smirnov (K-S) test further supports this fit with statistic = 0.0597 and p-value = 0.0544.

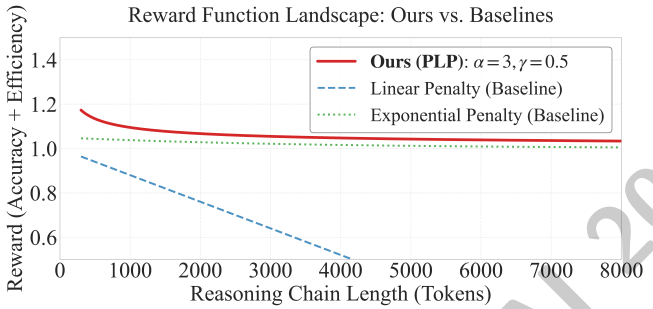


Figure 3: Linear penalty, power-law penalty, and exponential penalty comparison.

deviation in the extreme upper tail (Figure 2b), which is an artifact of the maximum generation length cap (set at 32,694 tokens), causing a truncation effect. However, this does not undermine the goodness-of-fit in the bulk region.

Theoretical Mechanism: Recursive Task Decomposition. To provide a principled explanation for this observed Log-Normal regularity, we model the reasoning process not merely as linear sequence generation, but as the *traversal* of a hierarchical solution tree. We view CoT reasoning as a recursive divide-and-conquer process. Complex problems are decomposed into intermediate steps, which further branch into finer-grained verification or correction substeps. Statistically, this can be modeled as a multiplicative branching process. Let X_k denote the stochastic branching factor (or expansion rate) at depth k. The number of active reasoning nodes at this depth expands as $N_k \approx N_{k-1} \cdot X_k$. Although the total length L is physically the sum of tokens across all depths ($L = \sum N_k$), for any non-trivial branching process, the total magnitude is **dominated by the multiplicative expansion** of the tree's width. Thus, the scale of the reasoning chain can be approximated by the cumulative product of these expansion factors:

$$L \propto \prod_{k=1}^{D} X_k \quad \Rightarrow \quad \ln L \approx \sum_{k=1}^{D} \ln X_k + C. \tag{1}$$

Here, X_k represents an independent random variable capturing the local reasoning complexity at depth k. As shown in [Limpert *et al.*, 2001], the cumulative effect of a multiplicative process converges to a Log-Normal distribution. Consequently, the length L itself follows a Log-Normal distribution:

$$L \sim \text{LogNormal}(\mu, \sigma^2)\,. \tag{2}$$

This theoretical result explains why the multiplicative accumulation of uncertainty in a tree-structured reasoning process naturally gives rise to the heavy-tailed distributions observed in our experiments.

3.2 Difficulty-Aware Length Penalty

Assuming a log-normal distribution for generation lengths, the variance is determined by both the log-mean μ and the log-standard deviation σ:

$$\text{Var}(L) = \left(e^{\sigma^2} - 1\right) e^{2\mu+\sigma^2}. \tag{3}$$

In standard reinforcement learning settings, the sampling temperature is typically held constant to ensure training stability, implying that σ remains effectively invariant. Under this condition, Eq. (3) reveals that the variance scales quadratically with the expectation, i.e., $\text{Var}(L) \propto (\mathbb{E}[L])^2$.

This scaling law reveals a critical insight: uncertainty accumulates polynomially with reasoning depth. Consequently, standard linear penalty terms fail to capture the heavy-tailed

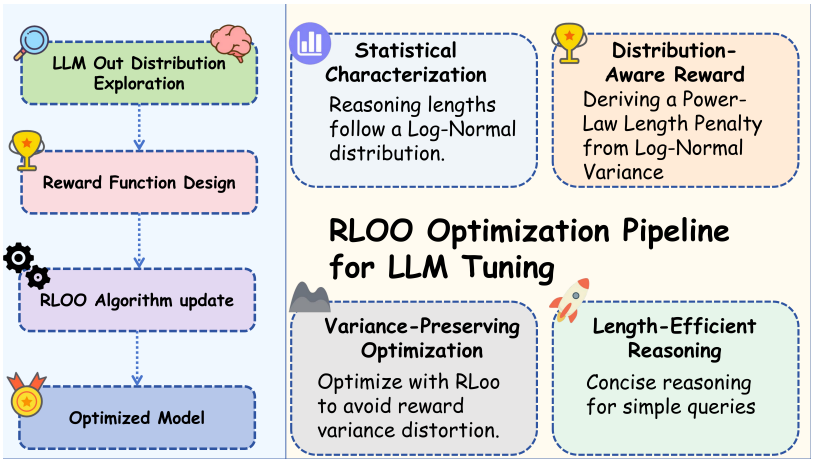


Figure 4: The pipeline of our work. Our pipeline leverages the variance scaling of Log-Normal reasoning distributions to derive a Powered Length Penalty, optimized via RLOO for efficient inference.

nature of the variance distribution. This theoretical property dictates that the penalty term must follow a Power-Law Formulation, $L^{-\gamma}$, to structurally align with the uncertainty growth.

While a strict inverse-variance normalization would theoretically imply fixing $\gamma \approx 2$, directly applying L^{-2} creates a significant optimization bottleneck. In reasoning tasks where L often spans 10^2 to 10^4, a quadratic decay suppresses the penalty magnitude to a negligible range (e.g., 10^{-6}), resulting in vanishing gradients that provide little effective signal.

To resolve the limitation, we propose a Relaxed Power-Law Formulation. We retain the polynomial decay profile mandated by the distribution theory but treat the curvature γ as a tunable hyperparameter. The relaxation offers a dual advantage, as visualized in Figure 3.

- **Structurally (Shape Matching):** Unlike linear penalties which impose uniform marginal costs, or exponential penalties that decay too aggressively (leading to premature saturation), the power-law curve aligns with the heavy-tailed distribution. It provides high resolution for short trajectories while naturally compressing distinctions in the long-length regime.
- **Functionally (Signal-to-Noise Trade-off):** By tuning γ, we can balance the signal magnitude against variance reduction. This mechanism mitigates the high uncertainty inherent in long reasoning chains without excessively diminishing the reward scale, thereby maintaining a robust learning signal during policy optimization.

Accordingly, we define the reward function $r(\cdot)$ as:

$$r\Big(\mathrm{len}\Big(y^{(i)}\Big)\Big) = \left(1 + \frac{\alpha}{\mathrm{len}\big(y^{(i)}\big)^{\gamma}}\right) \cdot \mathbb{I}\Big[\mathrm{acc}^{(i)} = 1\Big] \quad (4)$$

where α modulates the incentive magnitude, and γ calibrates the sensitivity decay. Maximizing this reward effectively imposes a *soft constraint on verbosity*, guiding the model toward the *principle of parsimony* via diminishing marginal returns.

3.3 Choice of RL Algorithm

Selecting an appropriate reinforcement learning algorithm is critical for effectively minimizing the proposed PLP. We analyze three representative approaches widely used in reasoning-oriented fine-tuning: PPO [Schulman *et al.*, 2017], GRPO [Shao *et al.*, 2024], and RLOO [Ahmadian *et al.*, 2024], and justify our choice of RLOO.

PPO relies on a learned critic and incurs substantial computational overhead. In contrast, both GRPO and RLOO are critic-free and employ explicit reward functions, making them more suitable for efficiency-aware optimization.

In GRPO, the advantage is computed via group-wise normalization:

$$A_i = \frac{r_i - \mathrm{mean}\big(\{r_1, \ldots, r_G\}\big)}{\mathrm{std}\big(\{r_1, \ldots, r_G\}\big)}. \quad (5)$$

This normalization rescales rewards according to within-group statistics, which can compress variance on simple problems while amplifying differences on complex ones. When combined with length-sensitive reward shaping, such adaptive rescaling may introduce additional optimization variance.

To mitigate these effects, we adopt the RLOO algorithm:

$$A^{\mathrm{RLOO}}\big(y^{(i)}, x\big) = r\big(y^{(i)}, x\big) - \frac{1}{k-1}\sum_{j \neq i} r\big(y^{(j)}, x\big). \quad (6)$$

Here, A^{RLOO} denotes the leave-one-out advantage. This subtraction-based baseline centers rewards without rescaling their magnitude, thereby preserving the relative variance structure induced by PLP and making RLOO well-suited for length-aware optimization across varying difficulty levels. An overview of our pipeline is presented in Figure 4

3.4 Experimental Setup

In this section, we present experimental results to evaluate the effectiveness of our method.

Models. We conduct experiments using the DeepSeek-R1 framework [Guo *et al.*, 2025], evaluating three models from

In-domain evaluation		GSM8K			MATH500			AIME2024		
Model	Method	Acc↑	Tokens↓	AE↑	Acc	Tokens↓	AE↑	Acc↑	Tokens↓	AE↑
DeepSeek-R1-Distill Qwen-1.5B	Original	76.5%	720	–	82.9%	5446	–	27.7%	15643	–
	PE	75.1%	738	-0.12	83.1%	5054	0.08	33.3%	15892	0.59
	TokenSkip	75.5%	463	0.29	77%	4784	-0.09	23.3%	17925	-0.94
	SFT	57.5%	507	-0.95	63.4%	4024	-0.92	17%	13433	-1.79
	Efficient	85.4%	702	0.37	83.2%	3641	0.34	33.3%	15087	0.64
	Shorterbetter	71.3%	**165**	0.43	81.4%	4271	0.13	31.7%	14837	0.48
	ThinkPrune	77.5%	347	0.56	82.2%	4332	0.16	27.3%	15212	-0.04
	Ours	**86.3%**	411	**0.81**	**85.1%**	**3606**	**0.42**	**33.7%**	**13327**	**0.80**
DeepSeek-R1-Distill Qwen-7B	Original	**92.6%**	1629	–	**92.4%**	4141	–	54.7%	12816	–
	PE	87.7%	619	0.36	91.4%	3614	0.07	49.7%	13783	-0.53
	Tokenskip	84.5%	3703	-1.71	91.6%	4312	-0.08	36.7%	15984	-1.89
	SFT	75.3%	**140**	-0.02	58%	2358	-1.43	40%	11278	-1.22
	Efficient	89.2%	226	0.68	90.7%	2329	0.35	48.7%	9721	-0.31
	Shorterbetter	91.9%	1042	0.32	92.2%	3854	0.06	53%	12864	-0.16
	ThinkPrune	90.8%	986	0.30	93%	3807	0.10	51.1%	12704	-0.32
	Ours	91.4%	363	**0.71**	91.4%	**2035**	**0.45**	**56.7%**	**7857**	**0.50**
DeepSeek-R1-Distill Llama-8B	Original	80.5%	834	–	84.7%	3408	–	48.3%	14151	–
	PE	80.8%	826	0.02	**90.6%**	3718	0.12	43.3%	12631	-0.41
	Tokenskip	77.9%	1573	-0.98	84.8%	4840	-0.42	43.3%	15289	-0.60
	SFT	53.1%	**185**	-0.92	66.6%	**787**	-0.83	43.3%	10310	-0.25
	Efficient	**87.6%**	379	0.81	86%	2670	0.26	36.7%	12903	-1.11
	Shorterbetter	78.2%	263	0.54	84.4%	3140	0.07	50%	12026	0.26
	Thinkprune	84.7%	262	0.84	86.6%	3265	0.11	50%	11461	0.30
	Ours	87.1%	309	**0.88**	87.2%	2418	**0.44**	**53.3%**	**11066**	**0.53**

Table 1: Model performance evaluation on in-domain data. PE denotes Prompt Engineering. The AE scores are computed using the hyperparameters $\phi = 1$, $\eta = 3$, and $\theta = 5$, as specified in the original paper

this family: DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, and DeepSeek-R1-Distill-LLaMA-8B. The first two models are based on the Qwen architecture [Hui *et al.*, 2024], while the third is based on LLaMA[Dubey *et al.*, 2024]. These models vary in both scale and pretraining architecture, enabling us to assess the generality of our method.

Datasets. The primary training dataset is GSM8K [Cobbe *et al.*, 2021]. We randomly select 3,200 examples from the GSM8K training set, ensuring a balanced distribution of difficulty levels. To evaluate generalization and robustness across diverse domains and difficulty levels, we test our models on six datasets: GSM8K, MATH500 [Lightman *et al.*, 2023], AIME2024, GPQA [Rein *et al.*, 2024], MMLU [Hendrycks *et al.*, 2020], and HumanEval [Chen *et al.*, 2021].

Baselines. In our experiments, we evaluate six baseline methods grouped into three categories: prompt-based approaches such as **Prompt Engineering**; supervised fine-tuning techniques including **SFT** and **TokenSkip** [Xia *et al.*, 2025b]; and reinforcement learning-based methods like **Efficient** [Arora and Zanette, 2025], **ShorterBetter** [Yi *et al.*, 2025], and **ThinkPrune** [Hou *et al.*, 2025]. To ensure a fair comparison, all methods are trained on the same dataset of 3,200 GSM8K samples, using identical prompt formats and evaluation metrics. More details about the baseline settings can be found in the Appendix G.

Implementation details. Training is performed using the OpenRLHF framework [Hu *et al.*, 2024], with efficient evaluation carried out via vLLM [Kwon *et al.*, 2023]. We employ a learning rate of 5×10^{-6} for the 1.5B model, and a lower learning rate of 2×10^{-6} for both the 7B and 8B models. In this study, we perform experiments by fixing $\gamma = 0.5$ and adjusting the value of α. The maximum number of tokens is set to 32,768 for all experiments. To account for variability and ensure robust results, we perform three independent training runs for each configuration and report the average performance. More details can be found in the Appendix E.

Evaluation Setup. During the evaluation process, experiments are conducted using vLLM. The temperature is set to 0.6 and the seed is set to 42, while all other parameters followed the default settings of vLLM. The prompts employed for the different datasets are provided in the Appendix D.

Evaluation Metrics. To comprehensively assess model performance, we employ three evaluation metrics: Accuracy, Tokens, and AE (Accuracy-Efficiency). Accuracy measures the proportion of test cases in which the model produces the correct final answer, serving as the primary indicator of reasoning reliability. Tokens refers to the total number of generated tokens per response, including intermediate reasoning steps and the final answer, and is used to evaluate the efficiency of the model's reasoning process. Together, these metrics provide a balanced assessment of both correctness and computational efficiency. The Accuracy-Efficiency metric, directly adopted from the O1-Prune methodology [Luo *et al.*, 2025], quantifies the trade-off between accuracy and conciseness. It offers a unified measure that captures both the effectiveness and brevity of model outputs, making it particularly well suited for evaluating reasoning models under efficiency constraints. The detailed information regarding the calculation formula for AE can be found in the Appendix H.

3.5 Results

As shown in Table 1, our method consistently outperforms baselines on in-domain datasets, achieving the highest AE scores across all cases. These results highlight the effectiveness of our approach in facilitating efficient reasoning. Fur-

Out-of-Domain Evaluation		GPQA			Humaneval			MMLU		
Model	Method	Acc↑	Tokens↓	AE↑	Acc	Tokens↓	AE↑	Acc↑	Tokens↓	AE↑
DeepSeek-R1-Distill Qwen-1.5B	Original	34.3%	8675	–	65.2%	5343	–	48.7%	1739	–
	PE	32.3%	9091	-0.22	61.0%	8178	-0.85	45.6%	1912	-0.67
	TokenSkip	34.8%	10584	-0.18	64%	5087	0.0	47.8%	2354	-0.45
	SFT	27.3%	8031	-0.54	59.8%	**4616**	-0.11	39%	1668	-0.64
	Efficient	32.8%	**7327**	-0.06	70%	5031	0.28	47.8%	1506	0.04
	Shorterbetter	34.8%	7416	0.19	68.9%	4900	0.25	48.2%	1415	0.13
	ThinkPrune	28.3%	7624	-0.75	72.6%	5083	0.39	48.4%	**1247**	0.25
	Ours	**36.4%**	8343	**0.22**	**75%**	4852	**0.54**	**50.1%**	1390	**0.29**
DeepSeek-R1-Distill Qwen-7B	Original	48.5%	8467	–	86.6%	5752	–	63.2%	1509	–
	PE	48%	8028	0.00	77.4%	4993	-0.19	65.1%	1358	0.19
	TokenSkip	44.4%	12706	-0.92	86%	5579	0.01	62.4%	4465	-2.02
	SFT	24.7%	10347	-2.68	45.7%	4832	-2.20	35.9%	5013	-4.4
	Efficient	47.5%	4448	0.37	86.8%	4091	0.30	62.4%	718	0.46
	Shorterbetter	48.5%	7355	0.13	**88.4%**	5050	0.18	64.9%	1232	0.26
	ThinkPrune	52.5%	8042	0.30	87.8%	4965	0.18	**65.4%**	1296	0.25
	Ours	**53%**	**4321**	**0.77**	84.8%	**3418**	**0.30**	64.6%	**694**	**0.61**
DeepSeek-R1-Distill Llama-8B	Original	40.4%	7377	–	84.1%	4683	–	**73.9%**	2049	–
	PE	42.4%	8782	-0.04	79.7%	4114	-0.14	72.4%	1893	-0.03
	TokenSkip	42.4%	12843	-0.59	77%	5402	-0.42	66.8%	2154	-0.53
	SFT	40.9%	**5412**	0.30	79.9%	2747	0.16	62.7%	2140	-1.43
	Efficient	**49%**	7278	0.65	86.6%	4255	0.18	71.6%	1476	0.12
	Shorterbetter	45.5%	8192	0.27	83.5%	4633	-0.02	73.3%	1823	0.07
	ThinkPrune	48%	7485	0.55	84%	4616	0.01	72.4%	1471	0.18
	Ours	47.5%	6288	**0.67**	**88.4%**	**3329**	**0.54**	70.7%	**953**	**0.32**

Table 2: Model performance evaluation on out-of-domain data. PE denotes Prompt Engineering. The AE scores are computed using the hyperparameters $\phi = 1$, $\eta = 3$, and $\theta = 5$, as specified in the original paper.

Model	GSM8K			MATH500			AIME2024			Average	
	Acc↑	Tokens↓	AE↑	Acc↑	Tokens↓	AE↑	Acc↑	Tokens↓	AE↑	Acc↑	AE↑
Qwen3-32B	96.1%	1658	–	95.1%	4565	–	81.7%	12764	–	91.0%	–
WHISPER*	96.2%	**435**	**0.74**	95.3%	**2295**	**0.50**	73.3%	**8700**	-0.2	88.3%	0.35
Ours	**97.0%**	915	0.48	**96.6%**	2768	0.44	**83.3%**	9287	**0.33**	**92.0%**	**0.42**

Table 3: Performance of Qwen3-32B on math datasets of varying difficulty.* denotes results reported in the original paper

thermore, as demonstrated in Table 2, our method generalizes well to out-of-domain datasets, consistently obtaining the highest AE scores across various models and datasets. This underscores its robust generalization capability.

Furthermore, we observe that when the base model exhibits limited capability on an evaluation dataset, reflected by an accuracy below 50%, our method consistently outperforms the base model in accuracy while simultaneously reducing the number of output tokens. Conversely, for datasets where the base model performs strongly, our method prioritizes token length reduction, sometimes at the cost of a slight accuracy decline. Importantly, despite not being explicitly trained on these datasets, our method effectively adheres to the principle of "fast on easy, deep on hard", dynamically adjusting its behavior based on task difficulty to strike a balance between efficiency and accuracy.

3.6 Scalability Analysis

To evaluate the scalability of our approach across both model architecture and model size, we further conduct experiments on the large-scale Qwen3-32B model [Yang *et al.*, 2025]. As shown in Table 3, our method consistently achieves the highest accuracy across all benchmarks while substantially reducing inference token usage compared to the baseline. Notably, although WHISPER [Xia *et al.*, 2025a] aggressively compresses reasoning tokens, it suffers from a significant performance drop on the challenging AIME2024 benchmark. (this result is directly quoted from the WHISPER paper) In contrast, our method preserves deep reasoning on hard problems while effectively eliminating redundancy on simpler ones, resulting in the best overall efficiency–performance trade-off.

3.7 Empirical Analysis

We employed Gemini 2.5 Pro [Comanici *et al.*, 2025] to assign difficulty scores, ranging from 2 to 9, to each problem in three mathematics datasets of varying difficulty levels. Based on these scores, we analyzed the models' performance across different difficulty tiers. As shown in Figure 5, the figure consolidates multiple aspects: Our fine-tuned model demonstrates adaptive efficiency: on simple tasks, it achieves drastic token reduction for rapid inference, with only a minor trade-off in accuracy. Notably, as difficulty increases, the model recovers and even surpasses the baseline on intermediate tasks, while maintaining performance parity on the most challenging problems. This confirms the model's ability to accelerate simple queries without compromising its reasoning capabilities on complex ones. This observation demonstrates that our approach aligns with the *"fast on easy, deep on hard"* principle.

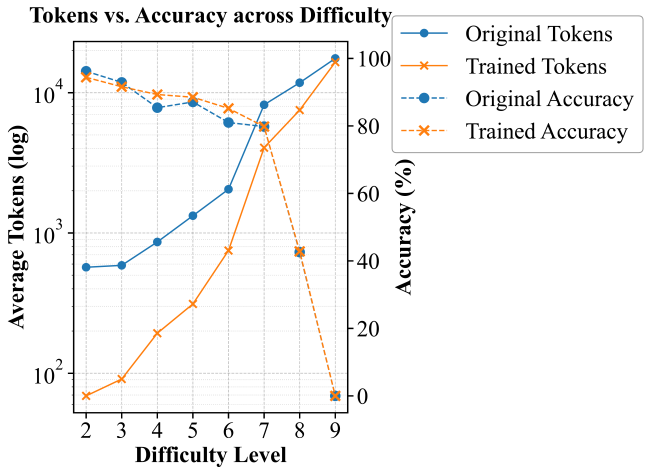


Figure 5: Performance comparison between the original and trained models across difficulty levels.

3.8 Impact on Reasoning Composition

To investigate how the PLP alters the internal structure of reasoning chains, we conducted a fine-grained component analysis comparing model outputs before and after training. We utilized `Gemini-2.5-Pro` as an impartial evaluator to classify reasoning steps into four distinct categories: *Pivotal Reasoning*, *Productive Elaboration*, *Redundant Calculation*, and *Non-Substantive Statement*.

As illustrated in Figure 6, the training process induces a substantial shift in the distribution of reasoning components:

- **Densification of Valid Reasoning:** The proportion of essential reasoning steps—comprising *Pivotal Reasoning* and *Productive Elaboration*—increased significantly from 65.4% to 83.9%. This indicates that the model learns to allocate its generation budget more efficiently towards effective problem-solving states.
- **Suppression of Noise:** Conversely, components associated with inefficiency were markedly suppressed. *Non-Substantive Statements* dropped by nearly half , and *Redundant Calculations* were virtually eliminated

Robustness Verification. To mitigate potential biases inherent to a single evaluator , we replicated this component analysis using other state-of-the-art models. We observed highly consistent distribution shifts across different evaluators, confirming that the densification of reasoning is a robust phenomenon rather than an artifact of a specific judge. Detailed comparative results from these auxiliary experiments are provided in Appendix M. These results suggest that PLP does not merely constrain length; rather, it implicitly optimizes the **information density** of the generated chain. By penalizing variance-induced redundancy while protecting deep thinking, our method encourages the model to prune irrelevant tokens while reinforcing the logical backbone required for complex reasoning.

3.9 Ablation Studies

Effect of the Length Penalty Parameters. We have conducted an ablation study on α and γ. For α, we test values from 1 to 30 and find optimal settings of 5, 3, and 5 for the 1.5B, 7B, and 8B models, respectively. Our results indicate that as α increases, the number of tokens generally decreases. We ablate γ over $\{0.05, 0.1, 0.25, 0.5, 1\}$ and observe that $\gamma = 0.5$ yields the most stable performance across all models. Details can be found in Appendix O .

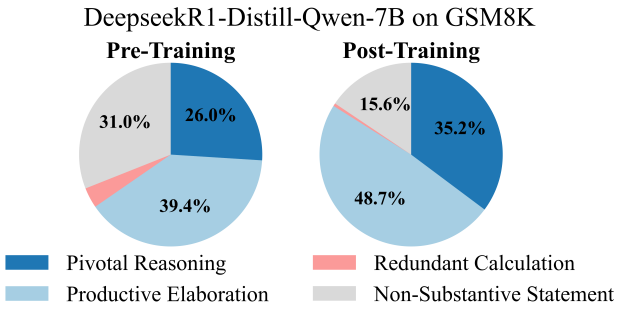


Figure 6: Evolution of Reasoning Components: Pre vs. Post Training

4 Discussion

While PLP outperforms baselines such as TokenSkip and ThinkPrune, it is instructive to examine its theoretical distinctions within adaptive reasoning. Constraint-based methods like LEASH [Li *et al.*, 2025b] enforce predefined length targets, which can conflict with the long-tailed nature of complex reasoning and degrade performance on challenging tasks. Outcome-based methods such as ALP adjust penalties according to empirical success, but risk signal collapse as models saturate in accuracy. In contrast, PLP leverages the intrinsic statistics of the reasoning process, deriving penalties from log-normal length distributions. Its curvature-controlled power-law naturally decays with sequence length, penalizing redundancy on simple tasks while asymptotically preserving flexibility for complex reasoning, ensuring robust adaptation across difficulty levels.

5 Conclusion

In this work, we introduce a novel reinforcement learning framework that leverages a powered length penalty to adaptively control the reasoning depth of large language models based on input difficulty. Our approach promotes concise responses for simple problems while allowing more elaborate reasoning for complex ones, thus adhering to the principle of "fast on the easy, deep on the hard.". Remarkably, despite being trained solely on the simplest reasoning benchmark, GSM8K, our method demonstrates strong generalization to more challenging tasks and even out-of-domain datasets.

Contribution Statement

Zehui Ling, Deshu Chen, and Hongwei Zhang contributed equally to this work. Yuan Cheng is the corresponding author.

References

[Achiam *et al.*, 2023] 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.

[Ahmadian *et al.*, 2024] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

[Arora and Zanette, 2025] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

[Chen *et al.*, 2021] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[Chen *et al.*, 2024] Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky Ho, Phil Mui, Silvio Savarese, Caiming Xiong, et al. Language models are hidden reasoners: Unlocking latent reasoning capabilities via self-rewarding. *arXiv preprint arXiv:2411.04282*, 2024.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[Comanici *et al.*, 2025] 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.

[Ding *et al.*, 2024] Mengru Ding, Hanmeng Liu, Zhizhang Fu, Jian Song, Wenbo Xie, and Yue Zhang. Break the chain: Large language models can be shortcut reasoners. *arXiv preprint arXiv:2406.06580*, 2024.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[Han *et al.*, 2024] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

[Hendrycks *et al.*, 2020] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[Hou *et al.*, 2025] 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.

[Hu *et al.*, 2024] Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

[Hui *et al.*, 2024] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[Kahneman, 2011] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[Kwon *et al.*, 2023] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[Li *et al.*,] Long Li, Jiaran Hao, Jason Klein Liu, Zhijian Zhou, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, et al. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward, 2025. *URL https://arxiv. org/abs/2509.07430*, 1.

[Li *et al.*, 2025a] Ruosen Li, Ziming Luo, Quan Zhang, Ruochen Li, Ben Zhou, Ali Payani, and Xinya Du. Aalc: Large language model efficient reasoning via adaptive accuracy-length control. *arXiv preprint arXiv:2506.20160*, 2025.

[Li *et al.*, 2025b] Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. Leash: Adaptive length penalty and reward shaping for efficient large reasoning model. *arXiv preprint arXiv:2512.21540*, 2025.

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

[Limpert *et al.*, 2001] Eckhard Limpert, Werner A Stahel, and Markus Abbt. Log-normal distributions across the sciences: keys and clues: on the charms of statistics, and how mechanical models resembling gambling machines offer

a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability—normal or log-normal: that is the question. *BioScience*, 51(5):341–352, 2001.

[Liu *et al.*, 2025] 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.

[Luo *et al.*, 2025] 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.

[Ma *et al.*, 2025] 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.

[Nye *et al.*, 2022] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. In *ICLR 2022 Deep Learning for Code Workshop,*, 2022.

[Perez *et al.*, 2020] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880. Association for Computational Linguistics, November 2020.

[Rein *et al.*, 2024] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Shao *et al.*, 2024] 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.

[Sprague *et al.*, 2025] Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Sui *et al.*, 2025] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

[Wan *et al.*, 2024] Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus Mcaleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 49890–49920. PMLR, 21–27 Jul 2024.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[Xia *et al.*, 2025a] Heming Xia, Cunxiao Du, Rui Li, Chak Tou Leong, Yongqi Li, and Wenjie Li. Merlin's whisper: Enabling efficient reasoning in llms via black-box adversarial prompting. *arXiv e-prints*, pages arXiv–2510, 2025.

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

[Xiang *et al.*, 2025] Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

[Yang *et al.*, 2025] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[Yao *et al.*, 2023] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[Ye *et al.*, 2025] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[Yi *et al.*, 2025] Jingyang Yi, Jiazheng Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*, 2025.

[Zhang *et al.*, 2025] Xuan Zhang, Ruixiao Li, Zhijian Zhou, Long Li, Yulei Qin, Ke Li, Xing Sun, Xiaoyu Tan, Chao Qu, and Yuan Qi. Count counts: Motivating exploration in llm reasoning with count-based intrinsic rewards. *arXiv preprint arXiv:2510.16614*, 2025.

[Zhou *et al.*, 2023] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.