Efficient Reasoning for Large Reasoning Language Models via Certainty-Guided Reflection Suppression

Jiameng Huang[1*], Baijiong Lin[2*], Guhao Feng[1], Jierun Chen[3], Di He[1†], Lu Hou[3†]

[1] National Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] Huawei Technologies Co., Ltd.
{huangjm, fenguhao}@stu.pku.edu.cn, {bj.lin.email, jierunchen}@gmail.com, dihe@pku.edu.cn, houlu3@huawei.com

Abstract

Recent Large Reasoning Language Models (LRLMs) employ long chain-of-thought reasoning with complex reflection behaviors, typically signaled by specific trigger words (e.g., “Wait” and “Alternatively”) to enhance performance. However, these reflection behaviors can lead to the overthinking problem where the generation of redundant reasoning steps that unnecessarily increase token usage, raise inference costs, and reduce practical utility. In this paper, we propose **Certainty-Guided Reflection Suppression (CGRS)**, a novel method that mitigates overthinking in LRLMs while maintaining reasoning accuracy. CGRS operates by dynamically suppressing the model’s generation of reflection triggers when it exhibits high confidence in its current response, thereby preventing redundant reflection cycles without compromising output quality. Our approach is model-agnostic, requires no retraining or architectural modifications, and can be integrated seamlessly with existing autoregressive generation pipelines. Extensive experiments across four reasoning benchmarks (i.e., AIME24, AMC23, MATH500, and GPQA-D) demonstrate CGRS’s effectiveness: it reduces token usage by an average of 18.5% to 41.9% while preserving accuracy and also achieves the optimal balance between length reduction and performance compared to state-of-the-art baselines. These results hold consistently across model architectures (e.g., DeepSeek-R1-Distill series, QwQ-32B, and Qwen3 family) and scales (4B to 32B parameters), highlighting CGRS’s practical value for efficient reasoning.

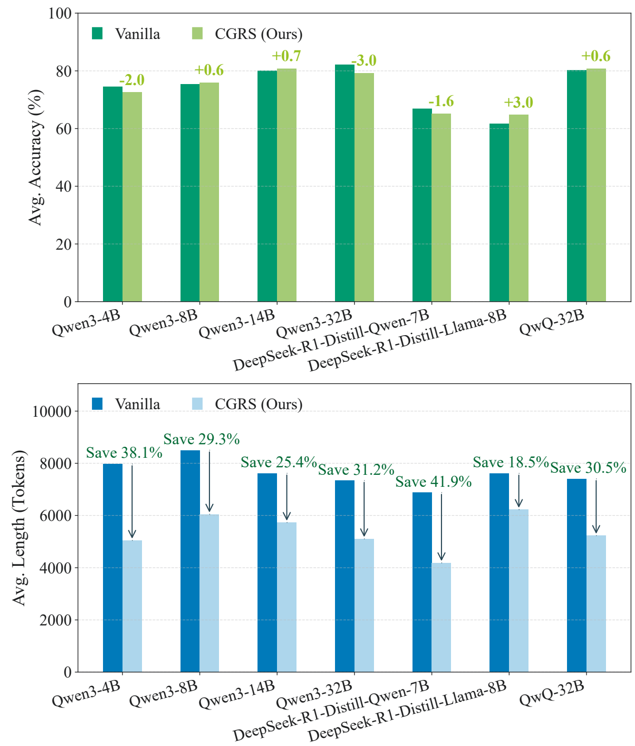

Figure 1: Accuracy and output length of the Vanilla and proposed CGRS methods across different models, averaged over three mathematical reasoning benchmarks (AIME24, AMC23, and MATH500) and one scientific reasoning benchmark (GPQA-D). CGRS achieves an average token reduction of $18.5 - 41.9\%$ while maintaining performance.

1 Introduction

Large Reasoning Language Models (LRLMs), including OpenAI’s o1/o3 (OpenAI 2024, 2025) and DeepSeek-R1 (Guo et al. 2025), have demonstrated remarkable performance on demanding benchmarks, especially in advanced mathematics and program synthesis. A key factor behind their success is their slow-thinking capability, which combines step-by-step deduction with complex reflection behaviors, including backtracking, exploring alternative strategies, and self-verification of results (Zhu and Li 2025).

Although these reflection behaviors enable models to self-correct and obtain accurate answers, they often suffer from a significant **overthinking** problem (Cuadron et al. 2025; Sui et al. 2025; Fan et al. 2025), where LRLMs persistently continue reasoning even after arriving at correct solutions. This

*These authors contributed equally.
†Corresponding to Lu Hou <houlu3@huawei.com> and Di He <dihe@pku.edu.cn>.

phenomenon leads to substantial increases in token consumption, higher inference costs, and degraded user experience due to unnecessary delays (Sun et al. 2025; Wang et al. 2025b). In extreme cases, excessively prolonged responses may exceed context window limits, resulting in truncated critical information and compromised accuracy (Li et al. 2025a). Addressing this overthinking problem and developing efficient reasoning mechanisms, therefore represents a crucial challenge for improving LRLM performance and practicality.

To address this challenge, training-free methods have recently emerged as lightweight solutions that mitigate overthinking without requiring additional training (Zhu and Li 2025). For instance, prompt-guided approaches like TALE (Han et al. 2025) prompts model responses within predefined token budgets through carefully designed instructions. While straightforward, these methods rely heavily on the model's innate instruction-following capabilities. Decoding-manipulation techniques, such as Dynasor (Fu et al. 2024) and DEER (Yang et al. 2025b), dynamically adjust the decoding process to eliminate redundancy by terminating generation when the model reaches sufficient confidence. However, their effectiveness remains highly sensitive to the design of early-exit conditions. Consequently, current methods lack adaptive mechanisms to properly balance reflection suppression with reasoning quality, often either excessively suppressing valid corrections or inadequately curbing redundant reflections.

We propose Certainty-Guided Reflection Suppression (**CGRS**), a novel method that mitigates overthinking in LRLMs while preserving reasoning accuracy. Previous work has demonstrated that reflection behaviors are typically signaled by specific **reflection triggers**, i.e., keywords like "Wait", "Alternatively", "But", and "Hmm" (Muennighoff et al. 2025; Zhang et al. 2025; Ding et al. 2025). Our CGRS approach proactively suppresses the model's generation of these reflection triggers when it exhibits high certainty in its current response, thereby preventing redundant reflection cycles and effectively alleviating overthinking while preserving reasoning quality.

Specifically, CGRS operates through two key components: (i) **Certainty estimation**: We first identify logical breakpoints in the reasoning process using structural delimiters (e.g., `\n\n`), then probe tentative final answers by injecting the prompt `**Final Answer: \boxed`. The model's certainty in its current response is quantified as the average token entropy over these tentative answers. Note that a lower entropy indicates higher confidence, suggesting a diminished need for further reflection. (ii) **Suppression of dynamic reflection triggers**: Based on the certainty score, we probabilistically suppress reflection triggers (e.g. "Wait", "But") by setting their logits to large negative values during sampling, effectively preventing unnecessary reflection cycles while preserving valid corrective behaviors.

We evaluate CGRS across diverse open-source LRLMs spanning multiple architectures and scales: the *DeepSeek-R1-Distill* series (Qwen-7B and Llama-8B) (Guo et al. 2025), *QwQ-32B* (Team 2025), and the *Qwen3* family (4B, 8B, 14B, and 32B) (Yang et al. 2025a). Our evaluation covers four reasoning benchmarks, including mathematical tasks (AIME24 (MAA Committees 2025), AMC23 (AI-MO 2024), and MATH500 (Lightman et al. 2023)) and scientific reasoning (GPQA Diamond (Rein et al. 2024)). As shown in Fig. 1, CGRS achieves significant token reduction ranging from 18.5% to 41.9%, while maintaining comparable accuracy. Furthermore, CGRS outperforms existing efficient reasoning approaches, including both prompt-guided and decoding-manipulation methods, by delivering greater token efficiency with minimal accuracy impact. These results solidify CGRS as a superior solution for enabling efficient reasoning in LRLMs.

Our contributions are summarized as follows:

- We propose CGRS, a training-free efficient reasoning method that alleviates the overthinking problem in LRLMs by dynamically suppressing reflection behaviors based on the model's internal certainty signals;
- We introduce two key components: (i) a certainty estimation mechanism that quantifies model confidence through entropy analysis of tentative answers, and (ii) a dynamic reflection trigger suppression technique that selectively disables unproductive reflection cycles;
- We validate our method against state-of-the-art efficient reasoning methods spanning multiple model scales (4B to 32B parameters) on four reasoning benchmarks. Across all evaluations, CGRS demonstrates superior token efficiency with minimal impact on accuracy, positioning it as a superior solution for efficient reasoning in LRLMs.

2 Related Work

LRLMs have significantly advanced long Chain-of-Thought (CoT) reasoning through supervised fine-tuning and reinforcement learning (Pan et al. 2025). However, prolonged CoT often introduces redundant steps that increase computational costs, a phenomenon known as **overthinking** (Cuadron et al. 2025; Sui et al. 2025; Fan et al. 2025). To address this issue, many works focus on compressing verbose CoTs into more concise reasoning traces (Sui et al. 2025; Qu et al. 2025; Feng et al. 2025; Wang et al. 2025b; Zhu and Li 2025), which can be broadly categorized into two approaches: training-based methods (Arora and Zanette 2025; Aggarwal and Welleck 2025; Wang et al. 2025a; Ma et al. 2025b; Munkhbat et al. 2025; Chen et al. 2025b; Zhu et al. 2025) and training-free methods (Ma et al. 2025a; Yu, Yu, and Wang 2025; Han et al. 2025; Yang et al. 2025c; Team et al. 2025; Wu et al. 2025; Fu et al. 2024; Jiang et al. 2025; Yang et al. 2025b; Li et al. 2025b; Aytes, Baek, and Hwang 2025; Yang et al. 2025c). While effective, training-based methods often require retraining, careful data curation, and may compromise the general capabilities of the pretrained model. In this work, we focus on training-free methods, which eliminate the need for additional training and can be further divided into four categories: prompt-guided, pipeline-based, model-merging, and decoding-manipulation methods.

Prompt-guided Methods. This type of method (Ma et al. 2025a; Yu, Yu, and Wang 2025; Han et al. 2025) compress lengthy CoT reasoning through prompt engineering. For instance, TALE (Han et al. 2025) simply prompts the model to solve the problem within token budgets. While simple, these methods heavily depend on the model's inherent instruction-following capability.

Pipeline-based Methods. This category of methods (Ong et al. 2025; Aytes, Baek, and Hwang 2025; Yang et al. 2025c) distribute queries or reasoning stages across multiple LRLMs, leveraging the efficiency of small models and the advanced reasoning of larger ones. However, these approaches introduce deployment complexity and additional inference overhead from auxiliary modules.

Model Merging. Such methods (Team et al. 2025; Wu et al. 2025) combine slow- and fast-thinking models via weight averaging to enable long-to-short reasoning. While effective for medium-sized models, this approach underperforms at extreme scales (very large or small models) and lacks precise control over reasoning depth (Zhu and Li 2025).

Decoding-manipulation Methods. This type of method (Fu et al. 2024; Jiang et al. 2025; Yang et al. 2025b; Li et al. 2025b) dynamically adjusts the decoding process to eliminate redundancy. For example, Dynasor (Fu et al. 2024) and DEER (Yang et al. 2025b) terminate decoding early when the model reaches sufficient confidence, though their performance depends critically on the exit condition design.

In this work, we focus on decoding manipulation for efficient reasoning. Our proposed CGRS method dynamically intervenes in token logits to suppress reflective behaviors based on the model's internal certainty signals.

3 Method

In this section, we present our Certainty-Guided Reflection Suppression (**CGRS**) method for efficient reasoning in LRLMs.

3.1 Preliminaries

Large reasoning language models demonstrate unique generation patterns during inference. Given a problem $\mathbf{q}$ (with prompt included), the output $\mathbf{o}$ is generated token by token as $\mathbf{o}_t \sim \boldsymbol{\pi}(\cdot|\mathbf{q}, \mathbf{o}_{<t})$, where $\boldsymbol{\pi}$ is the model. LRLMs organize their output using thinking delimiters (i.e., <think> and </think>), splitting the response into two key parts: slow thinking and conclusion. During slow thinking, LRLMs perform detailed, step-by-step reasoning before summarizing their thought process and delivering the final answer in the conclusion (Zhang et al. 2025).

Within slow thinking, models engage in complex reflection behaviors, including backtracking, exploring alternative approaches, and verifying results. These behaviors are often signaled by specific reflection triggers, i.e., keywords like "Wait", "Alternatively", "But", and "Hmm" (Muennighoff et al. 2025; Zhang et al. 2025; Ding et al. 2025).

Notably, these reflection behaviors cause models to continue reasoning even after reaching an initial answer. They may perform extra validation steps or switch reasoning strategies in later stages, either to double-check previous results or explore different paths. However, this can sometimes lead to unproductive overthinking, where the model enters unnecessary reasoning loops, repeatedly validating or re-evaluating without meaningful progress (Chen et al. 2025a).

To address this, our CGRS method proactively suppresses the model's tendency to generate reflection triggers when it exhibits high certainty in its current response. This prevents redundant reflection behavior and avoids overthinking. CGRS operates in two phases: (i) certainty estimation through checkpoint probing, and (ii) dynamic trigger suppression. We detail these components as follows.

3.2 Certainty Estimation

To estimate the necessity of reflective behaviors during inference, we introduce the **certainty score** that quantifies the LLM's confidence in its currently generated response. Specifically, we establish **checkpoints** along the LLM's reasoning path, identified by specific tokens. `\n\n` is used as the checkpoint marker, as this structural delimiter naturally indicates a thinking breakpoint (Yang et al. 2025c). At each checkpoint, we probe tentative final answers by appending a prompt $\mathbf{c}$ = `**Final Answer: \boxed` to the current generation $\mathbf{o}_{<t}$, then concurrently generating a response $\mathbf{a}$. This probing runs independently of the main decoding process, preserving the integrity of the primary reasoning trajectory. Then we calculate the certainty score C on the probed result $\mathbf{a}$ as follows:

$$C = 1 - \left(\frac{\frac{1}{n}\sum_{i=1}^{n} \mathcal{H}(\mathbf{p}_{\mathbf{a}_i})}{\log(|\mathbf{V}|)} \right), \tag{1}$$

where $n = |\mathbf{a}|$ is the token length of the probed results, $\mathbf{p}_{\mathbf{a}_i} = \boldsymbol{\pi}(\cdot|\mathbf{q}, \mathbf{o}_{<t}, \mathbf{c}, \mathbf{a}_{<i})$ represents the probability distribution of $\mathbf{a}_i$, $\mathcal{H}(\mathbf{p}) = -\mathbf{p}^\top \log(\mathbf{p})$ denotes the token entropy, and $|\mathbf{V}|$ is the total number of tokens in the LLM's vocabulary. The term $\log(|\mathbf{V}|)$ represents the maximum possible entropy (i.e., when the probability distribution is uniform across all tokens), serving as a normalization factor. This normalization bounds the certainty score C to $[0, 1]$, where a higher C indicates higher LLM confidence (i.e., lower entropy in $\mathbf{a}$), suggesting a reduced need for subsequent reflection behaviors.

3.3 Reflection Suppression

After estimating the LLM's certainty regarding its current generation, the certainty score C guides the suppression of subsequent reflection behaviors.

Reflection Triggers. Following prior work (Yang et al. 2025c; Fu et al. 2024; Zhang et al. 2025), we consider three trigger categories: (i) core hesitation and transition words (e.g., "But" and "Wait"), (ii) alternative proposal markers (e.g., "Alternatively"), and (iii) colloquial contemplation cues (e.g., "Hmm"), all of which signal potential shifts in the LLM's reasoning mode. Since a single trigger word has multiple natural language variants (e.g., "wait" as a lowercase variant of "Wait") that map to different tokens in the vocabulary, we construct a comprehensive trigger token set S_{trigger} through frequency analysis. Specifically, we identify

Algorithm 1: The token prediction process in CGRS.

Require: LRLM $\boldsymbol{\pi}$, input prompt $\mathbf{q}$, generated tokens $\mathbf{o}_{<t}$, checkpoint prompt $\mathbf{c}$, checkpoint marker `\n\n`, set of reflection triggers tokens S_{trigger}, previous suppression probability p (set to 0 if not exist), threshold δ;

compute probability distribution $\mathbf{p}_t = \boldsymbol{\pi}(\cdot|\mathbf{q}, \mathbf{o}_{<t})$;
sample $r \sim \mathrm{Bernoulli}(p)$;
if $r = 1$ **then**
  modify $\mathbf{p}_t$: set logits of tokens in S_{trigger} to a large negative value and re-normalize;
end if
sample the next token $\mathbf{o}_t$ from $\mathbf{p}_t$;
if checkpoint marker appears **then**
  obtain tentative answers $\mathbf{a}$ given the input $(\mathbf{q}, \mathbf{o}_{<t}, \mathbf{c})$;
  compute certainty score C using $\mathbf{a}$ via Eq. (1);
  compute suppression probability p using C and δ via Eq. (2);
end if
return: the next token $\mathbf{o}_t$, the suppression probability p.

all possible variants of each reflection trigger in the tokenizer's vocabulary and then analyze their generation frequencies in reasoning traces. This analysis was performed using reasoning traces from the *DeepSeek-R1-Distill-Qwen-7B* model (Guo et al. 2025) on both the AIME24 (MAA Committees 2025) and AMC23 (AI-MO 2024) benchmarks. To ensure robustness, we conducted four independent inference runs per dataset, aggregating statistics across all executions. Additional details and the complete list of trigger words and their variants are available in Appendix (Huang et al. 2025).

Suppressing Triggers Generation via Certainty. We reduce the reflection behaviors by suppressing the generation of triggers according to the certainty score. Specifically, we first calculate the suppression probability p as follows,

$$p = \max\left(0, \frac{C-\delta}{1-\delta}\right), \tag{2}$$

where $\delta \in [0, 1]$ is the confidence threshold. Then, with probability p, we set the logits of trigger tokens in S_{trigger} to a large negative value, thereby effectively excluding them from sampling. Eq. (2) indicates that trigger suppression occurs only when $C > \delta$. Besides, higher C values yield more frequent trigger suppression, as the LLM has exhibited high confidence in current responses, thereby requiring fewer subsequent reflection behaviors.

In summary, during inference, CGRS inserts checkpoints to probe intermediate answers for certainty calculation, then suppresses reflection tokens accordingly. This dynamic control mechanism reduces redundant reflection behaviors while preserving necessary corrections, enabling efficient reasoning. Moreover, CGRS is model-agnostic, requires no retraining or architectural modifications, and can be seamlessly integrated into existing autoregressive generation pipelines. The token prediction process of CGRS is shown in Algorithm 1.

4 Experiments

In this section, we evaluate the proposed CGRS method on multiple benchmark datasets and models to demonstrate its effectiveness in enabling efficient reasoning.

4.1 Experimental Setup

Benchmark Datasets. We evaluate on three mathematical reasoning benchmarks: AIME24 (MAA Committees 2025) that contains 30 high-difficulty problems from the 2024 American Invitational Mathematics Examination, AMC23 (AI-MO 2024) that includes 40 problems from the 2023 American Mathematics Competitions, and MATH500 (Lightman et al. 2023) that has 500 multi-step problems covering algebra, geometry, and probability, curated by OpenAI, and one scientific reasoning benchmark: GPQA Diamond (abbreviated as GPQA-D) (Rein et al. 2024), a dataset of 198 multiple-choice questions on biology, chemistry, and physics, at the post-graduate level.

Models. We conduct experiments on the a series of open-source models spanning different architectures and scales, including *DeepSeek-R1-Distill* series of models (Qwen-7B and Llama-8B) (Guo et al. 2025), *QwQ-32B* (Team 2025), and the *Qwen3* family (4B, 8B, 14B, and 32B) (Yang et al. 2025a).

Implementation Details. All experiments are conducted using the open-source `vLLM` framework (Kwon et al. 2023) to ensure high-throughput and memory-efficient inference. All decoding use temperature 0.6 and top-p 0.95. Each experiment is repeated three times and the average results are reported. The reflection suppression threshold δ in our CGRS method is set to 0.9.

Baselines. The proposed CGRS method is compared with three types of baselines: (i) **Vanilla** that performs standard decoding without any intervention, (ii) prompt-guided methods, including **NoThinking** (Ma et al. 2025a) that prompts the model to bypass intermediate reasoning entirely and generate the final answer directly, and **TALE** (Han et al. 2025) that prompts the model to solve the problem within token budgets (in our experiments, we set this budget based on the actual token length generated by CGRS), and (iii) decoding-manipulation methods, including **Dynasor** (Fu et al. 2024) that periodically requests intermediate answers at fixed token intervals and exits early if multiple consecutive answers match and **DEER** (Yang et al. 2025b) dynamically truncates chain-of-thought generation by detecting high-confidence intermediate answers at transition cues such as "Wait".

Evaluation Metrics. We evaluate performance using three metrics: (i) pass@1 accuracy (**Acc**) that is the proportion of problems correctly solved on the first attempt, (ii) average output token length (**Len**) that serves as a proxy for reasoning cost during inference, and (iii) the average length reduction ratio (**LR**) that measures the percentage decrease in output token length compared to the Vanilla method, with higher values indicating higher compression.

Method	AIME24			AMC23			MATH500			GPQA-D			AVG	
	Acc↑	Len↓	LR ↑	Acc↑	Len↓	LR ↑	Acc↑	Len↓	LR ↑	Acc↑	Len↓	LR ↑	Acc↑	LR ↑
Qwen3-4B														
Vanilla	60.0	11449	-	91.7	7449	-	92.7	4796	-	53.5	8195	-	74.5	-
TALE	48.9	9727	15.0%	86.7	5107	31.4%	89.1	2657	44.6%	36.2	4938	39.7%	65.2	32.7%
NoThinking	24.4	4504	60.7%	70.0	1710	77.0%	84.9	988	79.4%	47.5	1471	82.1%	56.7	**74.8%**
Dynasor	54.3	9912	13.4%	86.7	6233	16.3%	90.1	3877	19.2%	50.5	4398	46.3%	70.4	23.8%
DEER	50.0	6873	40.0%	80.0	3231	56.6%	87.0	1854	61.3%	54.9	7033	14.2%	68.0	43.0%
CGRS (**ours**)	56.7	7893	31.1%	86.7	4351	41.6%	91.3	2704	43.6%	55.2	5229	36.2%	**72.5**	38.1%
Qwen3-8B														
Vanilla	61.1	11924	-	89.4	7876	-	92.9	5104	-	57.7	9104	-	75.3	-
TALE	68.9	10942	8.2%	88.3	6872	12.7%	92.3	3885	23.9%	59.1	7113	21.9%	**77.2**	16.7%
NoThinking	30.0	5967	50.0%	72.5	2426	69.2%	87.1	1239	75.7%	54.2	1546	83.0%	61.0	**69.5%**
Dynasor	62.2	10174	14.7%	89.2	6457	18.0%	91.7	3841	24.7%	57.7	5965	34.5%	75.2	23.0%
DEER	45.6	7443	37.6%	79.2	3715	52.8%	88.7	1935	62.1%	59.3	7837	13.9%	68.2	41.6%
CGRS (**ours**)	61.1	8792	26.3%	89.2	5595	29.0%	93.3	3507	31.3%	59.8	6302	30.8%	75.9	29.3%
Qwen3-14B														
Vanilla	68.9	11316	-	93.3	7190	-	94.1	4551	-	64.0	7411	-	80.1	-
TALE	71.1	10860	6.4%	92.5	5951	17.2%	93.7	3389	25.5%	63.8	6091	17.8%	80.3	16.2%
NoThinking	27.8	3689	67.4%	77.5	1616	77.5%	87.0	853	81.3%	56.9	1268	82.9%	62.3	**77.3%**
Dynasor	65.6	9775	13.6%	90.0	6030	16.1%	84.4	3667	19.4%	64.3	5775	22.1%	76.1	17.8%
DEER	56.7	6755	40.3%	90.8	4079	43.3%	91.7	1956	57.0%	58.2	6338	14.5%	74.4	38.8%
CGRS (**ours**)	70.0	8662	23.5%	93.3	5076	29.4%	94.5	3235	28.9%	65.2	5953	19.7%	**80.8**	25.4%
Qwen3-32B														
Vanilla	67.8	11022	-	95.8	6794	-	93.9	4473	-	71.4	7062	-	82.2	-
TALE	67.8	10688	3.0%	93.3	6533	3.8%	93.6	3857	13.8%	67.0	5916	16.2%	**80.4**	9.2%
NoThinking	41.1	5635	48.9%	75.0	2221	67.3%	87.0	1054	76.4%	56.6	917	87.0%	64.9	**69.9%**
Dynasor	64.4	9518	13.6%	92.5	5521	18.7%	85.2	3486	22.1%	58.1	3161	55.2%	75.0	27.4%
DEER	68.9	8697	21.1%	88.2	4878	28.2%	93.2	2533	43.4%	69.6	6310	10.6%	80.0	25.8%
CGRS (**ours**)	65.6	8128	26.3%	94.2	4766	29.8%	93.1	2993	33.1%	64.0	4535	35.8%	79.2	31.2%
DeepSeek-R1-Distill-Qwen-7B														
Vanilla	52.2	10662	-	87.5	5861	-	91.3	3787	-	36.2	7191	-	66.8	-
TALE	48.9	9727	8.8%	86.7	5107	19.9%	89.1	2657	29.8%	36.2	4938	31.3%	**65.2**	20.7%
NoThinking	32.2	6680	37.3%	75.8	2499	57.4%	80.9	1173	65.4%	37.9	1312	81.8%	56.7	**61.4%**
Dynasor	47.8	8334	21.8%	84.2	5201	11.3%	81.8	2070	45.3%	22.2	561	92.2%	59.0	42.7%
DEER	47.8	9288	12.9%	88.3	4670	20.3%	89.6	2272	40.0%	33.1	6457	10.2%	64.7	20.9%
CGRS (**ours**)	52.2	7597	28.7%	88.3	3406	41.9%	87.6	1867	50.7%	32.8	3876	46.1%	**65.2**	41.9%
DeepSeek-R1-Distill-Llama-8B														
Vanilla	37.7	11898	-	84.2	6374	-	85.7	4087	-	39.2	8096	-	61.7	-
TALE	40.0	11141	6.4%	85.0	5915	7.2%	84.0	3541	13.4%	46.1	5573	31.2%	63.8	14.5%
NoThinking	40.0	11242	5.5%	82.5	5796	9.1%	83.3	2405	41.2%	36.2	6638	18.0%	60.5	18.4%
Dynasor	37.7	10368	12.9%	84.2	5681	10.9%	85.0	3585	12.3%	31.8	3095	61.8%	59.7	24.4%
DEER	42.2	9778	17.8%	80.8	5480	14.0%	82.3	2722	33.4%	41.8	7434	8.2%	56.4	**35.7%**
CGRS (**ours**)	47.8	9536	19.9%	86.7	4899	23.1%	84.7	3254	20.4%	39.6	7221	10.8%	**64.7**	18.5%
QwQ-32B														
Vanilla	71.1	11026	-	89.1	7210	-	94.2	4216	-	66.4	7269	-	80.2	-
TALE	61.1	10888	1.3%	90.8	6522	9.5%	94.0	3533	16.2%	65.5	6482	10.8%	77.8	9.5%
NoThinking	62.2	11688	-6.0%	88.3	7493	-3.9%	94.2	4276	-1.4%	65.2	7604	-4.6%	77.5	-4.0%
Dynasor	64.4	9733	11.7%	90.0	7185	0.3%	94.0	4156	1.4%	41.9	2478	65.9%	72.6	19.9%
DEER	65.6	10015	9.2%	92.5	6324	12.3%	94.1	3359	20.3%	66.5	6453	11.2%	79.7	13.3%
CGRS (**ours**)	68.9	8202	25.6%	93.3	4771	33.8%	94.2	2810	33.3%	67.0	5141	29.3%	**80.8**	**30.5%**

Table 1: Comparison across models of different scales and multiple methods. Each experiment is repeated three times and the average results are reported. "Acc" (%) and "Len" (in tokens) denote the accuracy and response length, respectively. "LR" is the average length reduction ratio relative to Vanilla. ↑ (↓) indicates that the higher (lower) the result, the better the performance. The best results (except for Vanilla) are highlighted in **bold**.

4.2 Comparison with State-of-the-art Methods

In Table 1, we compare the proposed CGRS method with previous efficient reasoning methods in terms of accuracy and output length. The evaluation covers diverse model scales and architectures, including *DeepSeek-R1-Distill-Qwen-7B*, *DeepSeek-R1-Distill-Llama-8B*, *QwQ-32B*, and *Qwen3-4/8/14/32B* across four reasoning benchmark datasets (i.e., AIME24, AMC23, MATH500, and GPQA-D).

As can be seen, CGRS consistently achieves the optimal balance between average length reduction ratio and average accuracy preservation across diverse models, demonstrating its effectiveness for efficient reasoning. For example, on the *Qwen3-14B* model, CGRS reduces output length by 25.4% while maintaining accuracy. In contrast, while baseline methods achieve greater length reduction (77.3% of NoThinking and 38.8% of DEER), they suffer significant accuracy degradation (17.8% and 5.7% drops respectively). TALE maintains comparable accuracy but achieves less length reduction than CGRS (16.2% vs. 25.4%).

Moreover, the proposed CGRS method demonstrates reliable effectiveness in every test case, consistently achieving 18.5% to 41.9% length reduction with negligible accuracy drop (up to 3%), while the baseline methods often perform unsatisfactorily. For example, NoThinking prompts models to bypass slow thinking processes, which always yields high compression rates but severely degrades accuracy (e.g., 65.4% length reduction and 9.4% accuracy drop on MATH500 with *DeepSeek-R1-Distill-Qwen-7B*). Early-exit approaches like Dynasor and DEER also sometimes underperform, e.g., on AMC23 with *DeepSeek-R1-Distill-Llama-8B*, they only achieve 10.9% and 14.0% compression, respectively, with DEER additionally reducing accuracy. Remarkably, in the same test case, CGRS attains 23.1% length reduction while slightly improving accuracy.

Notably, on the *QwQ-32B* model, the baseline methods (TALE, NoThinking, Dynasor, and DEER) achieve only minor reduction ratios across all four benchmark datasets. This limitation occurs because these methods rely on hard-coded reflection boundaries that assume reasoning terminates at the </think> token, while *QwQ-32B* often continues reflection behaviors beyond this token (Yang et al. 2025b). In contrast, the proposed CGRS method dynamically regulates reflection triggers based on confidence scores, which is independent on the end-of-think token, enabling CGRS to maintain strong compression rates (30.5%) while preserving accuracy in this case.

In summary, these results demonstrate the superior performance of the proposed CGRS method over all baseline approaches across various models and datasets, indicating its effectiveness for efficient reasoning in LRLMs.

4.3 Effectiveness of Reflection Suppression

As introduced in Section 3, CGRS proactively suppresses redundant reflection behaviors to alleviate the overthinking problem. To evaluate the effectiveness of this mechanism, in this section, we analyze two key metrics: (i) the frequency of reflection triggers (specifically the words "Wait", "But", "Alternatively", and "Hmm"), and (ii) the answer length distribution, comparing CGRS against the Vanilla baseline. The experiments employ the *DeepSeek-R1-Distill-Qwen-7B* model on the AIME24 benchmark, following the same experimental setup described in Section 4.1.

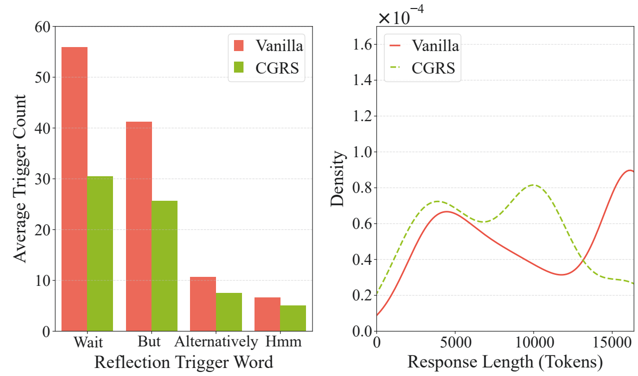

Figure 2: Comparison of Vanilla and CGRS methods in terms of the frequency of reflection triggers (**left**) and the answer length distribution (**right**).

	Acc↑	Len↓
$p = 0$ (the Vanilla baseline)	87.5	5861
$p = 0.25$	81.7	3729
$p = 0.5$	80.0	3266
$p = 1.0$	76.7	2373
Certainty-Guided (Eq. (2))	88.3	3406

Table 2: Comparison of certainty-guided suppression probability (Eq. (2)) with fixed probabilities.

The results are presented in Figure 2. Compared to the Vanilla baseline, CGRS demonstrates: (i) a significant reduction in reflection trigger words (particularly "Wait" and "But"), and (ii) a more concentrated and significantly shorter length distribution. These findings clearly indicate effective suppression of reflection behaviors. When combined with the findings in Table 1 showing that CGRS achieves 28.7% length reduction without accuracy decrease on AIME24 using the *DeepSeek-R1-Distill-Qwen-7B* model, these results demonstrate that CGRS effectively alleviate the overthinking problem and enable efficient reasoning.

4.4 Effectiveness of Certainty-Guided Probability

As described in Section 3, CGRS dynamically suppresses the LRLM's tendency to generate reflection triggers based on the certainty of its current response. In this section, we evaluate the effectiveness of this certainty-guided suppression probability (Eq. (2)) by comparing it against fixed suppression probabilities: $p = 0$ (the Vanilla baseline described in Section 4.1), 0.25, 0.5, and 1 (where no reflection triggers are sampled during inference). We conduct experiments on the AMC23 dataset using the *DeepSeek-R1-Distill-Qwen-7B* model, following the same setup as in Section 4.1.

The results are shown in Table 2. As can be seen, as p increases, the response length decreases significantly while the accuracy also drops considerably. This indicates that important reflection behaviors are being suppressed in the slow thinking process. In contrast, our certainty-guided method achieves a 41.9% reduction in length without compromising

An Example from MATH500 Dataset

Problem:
Consider the geometric sequence $\frac{125}{9}, \frac{25}{3}, 5, 3, \ldots$. What is the eighth term of the sequence? Express your answer as a common fraction.
Ground Truth: $\frac{243}{625}$

Response:
I'll write down my thought process as I go through the problem.
First, I need to understand the problem. I have a geometric sequence given as 125/9, 25/3, 5, 3, ...
... [Omitted]
Subtract 78,125 once: 781,250 - 78,125 = 703,125
So, denominator is 703,125 [**High Certainty Score:** 0.926]

Continue (Vanilla):

Therefore, $a_8 = \frac{273{,}375}{703{,}125}$
Now, we can simplify this fraction.
... [Omitted]
Therefore, the eighth term $a_8 = \frac{243}{625}$ **[Correct Anwser]**
Wait a second [Reflection Triggered], let me verify the calculations again to make sure I didn't make a mistake.
Starting from ... **[Overthinking]**
Therefore, the eighth term is 243/625.

Final Answer The eighth term of the sequence is $\boxed{\frac{243}{625}}$.

`</think>`
... [Omitted]

Continue (CGRS):

Therefore, $a_8 = \frac{273{,}375}{703{,}125}$
Now, simplify this fraction.
... [Omitted]
Therefore, the simplified fraction is $\frac{243}{625}$ **[Correct Anwser]**
So, the eighth term of the sequence is $\frac{243}{625}$
`</think>`
... [Omitted]

Figure 3: Comparison of the responses from Vanilla and the proposed CGRS (with $\delta = 0.9$) methods on an example from the MATH500 dataset using the *DeepSeek-R1-Distill-Qwen-7B* model.

accuracy, demonstrating that this certainty-based control effectively suppresses unproductive reflections while preserving necessary ones.

4.5 Case Study

In Figure 3, we present a representative case from the MATH 500 dataset, solved by the *DeepSeek-R1-Distill-7B* model, to demonstrate CGRS's mechanism and advantages.

During the initial reasoning phase, the model executes routine computations. Since the certainty scores for these intermediate steps remain below a predefined threshold, CGRS remains inactive, permitting the model to freely explore and revise its reasoning path—similar to standard autoregressive inference.

The turning point occurs when a high certainty score of 0.926 is detected, triggering the CGRS mechanism. In the Vanilla baseline, the model performs redundant reflection even after reaching a correct intermediate answer—reexamining previous steps (e.g., common ratio, exponentiation, and multiplication in this case). This unnecessary verification increases computational overhead without enhancing accuracy. In contrast, CGRS promotes efficient and goal-directed behavior. It suppresses reflection-inducing tokens, preventing re-evaluation of outputs it considers reliable. Instead, the model proceeds directly to the next logical operation (e.g., fraction simplification). This targeted intervention ensures high-confidence reasoning proceeds without unnecessary detours, significantly reducing both token count and inference time.

This case study offers qualitative evidence that CGRS effectively alleviate overthinking by utilizing internal certainty estimates to suppress redundant reasoning steps.

5 Conslusion

In this paper, we propose Certainty-Guided Reflection Suppression (CGRS), a novel approach to address the overthinking problem in Large Reasoning Language Models (LRLMs) and enable efficient reasoning. CGRS is a lightweight, certainty-guided decoding strategy that dynamically suppresses reflection triggers when the model exhibits high confidence in its current reasoning trajectory. Our method is model-agnostic, requires no retraining or architectural modifications, and can be seamlessly integrated into existing autoregressive generation pipelines. Extensive experiments on open-source LRLMs across four reasoning benchmarks demonstrate that CGRS achieves token usage reductions of up to 41.9% while preserving answer accuracy. These results establish that reflective behaviors in LRLMs can be effectively modulated during inference through certainty-aware decoding. Our work advances the development of efficient reasoning systems by demonstrating how model behavior can be aligned with internal certainty signals, providing a practical pathway toward more scalable LLM deployments.

Acknowledgments

This work is supported in part by the State Key Laboratory of General Artificial Intelligence. DH is supported by National Science Foundation of China (NSFC62376007) and Beijing Natural Science Foundation (Z250001).

References

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

AI-MO. 2024. AMC 2023. https://huggingface.co/datasets/AI-MO/aimo-validation-amc. Accessed: 2025-07-26.

Arora, D.; and Zanette, A. 2025. Training Language Models to Reason Efficiently. arXiv:2502.04463.

Aytes, S. A.; Baek, J.; and Hwang, S. J. 2025. Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching. arXiv:2503.05179.

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. 2025a. Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. arXiv:2412.21187.

Chen, X.; Zhao, A.; Xia, H.; Lu, X.; Wang, H.; Chen, Y.; Zhang, W.; Wang, J.; Li, W.; and Shen, X. 2025b. Reasoning Beyond Language: A Comprehensive Survey on Latent Chain-of-Thought Reasoning. arXiv:2505.16782.

Cuadron, A.; Li, D.; Ma, W.; Wang, X.; Wang, Y.; Zhuang, S.; Liu, S.; Schroeder, L. G.; Xia, T.; Mao, H.; Thumiger, N.; Desai, A.; Stoica, I.; Klimovic, A.; Neubig, G.; and Gonzalez, J. E. 2025. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. arXiv:2502.08235.

Ding, B.; Chen, Y.; Wang, F.; Ming, L.; and Lin, T. 2025. Do Thinking Tokens Help or Trap? Towards More Efficient Large Reasoning Model. arXiv:2506.23840.

Fan, C.; Li, M.; Sun, L.; and Zhou, T. 2025. Missing Premise exacerbates Overthinking: Are Reasoning Models losing Critical Thinking Skill? arXiv:2504.06514.

Feng, S.; Fang, G.; Ma, X.; and Wang, X. 2025. Efficient Reasoning Models: A Survey. arXiv:2504.10903.

Fu, Y.; Chen, J.; Zhu, S.; Fu, Z.; Dai, Z.; Zhuang, Y.; Ma, Y.; Qiao, A.; Rosing, T.; Stoica, I.; and Zhang, H. 2024. Efficiently Scaling LLM Reasoning with Certaindex. arXiv:2412.20993.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv:2501.12948.

Han, T.; Wang, Z.; Fang, C.; Zhao, S.; Ma, S.; and Chen, Z. 2025. Token-Budget-Aware LLM Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 24842–24855. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Huang, J.; Lin, B.; Feng, G.; Chen, J.; He, D.; and Hou, L. 2025. Efficient Reasoning for Large Reasoning Language Models via Certainty-Guided Reflection Suppression. arXiv:2508.05337.

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

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Li, G.; Gao, Y.; Li, Y.; and Wu, Y. 2025a. ThinkLess: A Training-Free Inference-Efficient Method for Reducing Reasoning Redundancy. arXiv:2505.15684.

Li, J.; Zhao, W.; Zhang, Y.; and Gan, C. 2025b. Steering LLM Thinking with Budget Guidance. arXiv:2506.13752.

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. In *International Conference on Learning Representations*.

Ma, W.; He, J.; Snell, C.; Griggs, T.; Min, S.; and Zaharia, M. 2025a. Reasoning Models Can Be Effective Without Thinking. arXiv:2504.09858.

Ma, X.; Wan, G.; Yu, R.; Fang, G.; and Wang, X. 2025b. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. arXiv:2502.09601.

MAA Committees. 2025. AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. Accessed: 2025-07-26.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. arXiv:2501.19393.

Munkhbat, T.; Ho, N.; Kim, S. H.; Yang, Y.; Kim, Y.; and Yun, S.-Y. 2025. Self-Training Elicits Concise Reasoning in Large Language Models. arXiv:2502.20122.

Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.-L.; Wu, T.; Gonzalez, J. E.; Kadous, M. W.; and Stoica, I. 2025. RouteLLM: Learning to Route LLMs with Preference Data. arXiv:2406.18665.

OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2025-07-26.

OpenAI. 2025. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-08-01.

Pan, Q.; Ji, W.; Ding, Y.; Li, J.; Chen, S.; Wang, J.; Zhou, J.; Chen, Q.; Zhang, M.; Wu, Y.; and He, L. 2025. A Survey of Slow Thinking-based Reasoning LLMs using Reinforced Learning and Inference-time Scaling Law. *ArXiv*, abs/2505.02665.

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.-S.; Zhou, B.; and Cheng, Y. 2025. A Survey of Efficient Reasoning for Large Reasoning Models: Language, Multimodality, and Beyond. arXiv: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 (COLM)*.

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

Sun, G.; Wang, Z.; Zhao, X.; Tian, B.; Shen, Z.; He, Y.; Xing, J.; and Li, A. 2025. Invisible Tokens, Visible Bills: The Urgent Need to Audit Hidden Operations in Opaque LLM Services. arXiv:2505.18471.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; Tang, C.; Wang, C.; Zhang, D.; Yuan, E.; Lu, E.; Tang, F.; Sung, F.; Wei, G.; Lai, G.; Guo, H.; Zhu, H.; Ding, H.; Hu, H.; Yang, H.; Zhang, H.; Yao, H.; Zhao, H.; Lu, H.; Li, H.; Yu, H.; Gao, H.; Zheng, H.; Yuan, H.; Chen, J.; Guo, J.; Su, J.; Wang, J.; Zhao, J.; Zhang, J.; Liu, J.; Yan, J.; Wu, J.; Shi, L.; Ye, L.; Yu, L.; Dong, M.; Zhang, N.; Ma, N.; Pan, Q.; Gong, Q.; Liu, S.; Ma, S.; Wei, S.; Cao, S.; Huang, S.; Jiang, T.; Gao, W.; Xiong, W.; He, W.; Huang, W.; Xu, W.; Wu, W.; He, W.; Wei, X.; Jia, X.; Wu, X.; Xu, X.; Zu, X.; Zhou, X.; Pan, X.; Charles, Y.; Li, Y.; Hu, Y.; Liu, Y.; Chen, Y.; Wang, Y.; Liu, Y.; Qin, Y.; Liu, Y.; Yang, Y.; Bao, Y.; Du, Y.; Wu, Y.; Wang, Y.; Zhou, Z.; Wang, Z.; Li, Z.; Zhu, Z.; Zhang, Z.; Wang, Z.; Yang, Z.; Huang, Z.; Huang, Z.; Xu, Z.; Yang, Z.; and Lin, Z. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. arXiv:2501.12599.

Team, Q. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning. https://qwenlm.github.io/blog/qwq-32b/. Accessed: 2025-07-26.

Wang, J.; Qiang, W.; Song, Z.; Zheng, C.; and Xiong, H. 2025a. Learning to Think: Information-Theoretic Reinforcement Fine-Tuning for LLMs. arXiv:2505.10425.

Wang, R.; Wang, H.; Xue, B.; Pang, J.; Liu, S.; Chen, Y.; Qiu, J.; Wong, D. F.; Ji, H.; and Wong, K.-F. 2025b. Harnessing the Reasoning Economy: A Survey of Efficient Reasoning for Large Language Models. arXiv:2503.24377.

Wu, H.; Yao, Y.; Liu, S.; Liu, Z.; Fu, X.; Han, X.; Li, X.; Zhen, H.-L.; Zhong, T.; and Yuan, M. 2025. Unlocking Efficient Long-to-Short LLM Reasoning with Model Merging. arXiv:2503.20641.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025a. Qwen3 Technical Report. arXiv:2505.09388.

Yang, C.; Si, Q.; Duan, Y.; Zhu, Z.; Zhu, C.; Li, Q.; Lin, Z.; Cao, L.; and Wang, W. 2025b. Dynamic Early Exit in Reasoning Models. arXiv:2504.15895.

Yang, W.; Yue, X.; Chaudhary, V.; and Han, X. 2025c. Speculative Thinking: Enhancing Small-Model Reasoning with Large Model Guidance at Inference Time. arXiv:2504.12329.

Yu, Y.; Yu, Y.; and Wang, H. 2025. PREMISE: Scalable and Strategic Prompt Optimization for Efficient Mathematical Reasoning in Large Models. arXiv:2506.10716.

Zhang, J.; Dong, R.; Wang, H.; Ning, X.; Geng, H.; Li, P.; He, X.; Bai, Y.; Malik, J.; Gupta, S.; and Zhang, H. 2025. AlphaOne: Reasoning Models Thinking Slow and Fast at Test Time. arXiv:2505.24863.

Zhu, J.; and Li, H. 2025. Towards Concise and Adaptive Thinking in Large Reasoning Models: A Survey. arXiv:2507.09662.

Zhu, R.-J.; Peng, T.; Cheng, T.; Qu, X.; Huang, J.; Zhu, D.; Wang, H.; Xue, K.; Zhang, X.; Shan, Y.; Cai, T.; Kergan, T.; Kembay, A.; Smith, A.; Lin, C.; Nguyen, B.; Pan, Y.; Chou, Y.; Cai, Z.; Wu, Z.; Zhao, Y.; Liu, T.; Yang, J.; Zhou, W.; Zheng, C.; Li, C.; Zhou, Y.; Li, Z.; Zhang, Z.; Liu, J.; Zhang, G.; Huang, W.; and Eshraghian, J. 2025. A Survey on Latent Reasoning. arXiv:2507.06203.