Let LLMs Break Free from Overthinking via Self-Braking Tuning

Haoran Zhao[1,2,*] **Yuchen Yan**[1,*] **Yongliang Shen**[1,†] **Haolei Xu**[1] **Wenqi Zhang**[1]

Kaitao Song[3] **Jian Shao**[1] **Weiming Lu**[1] **Jun Xiao**[1] **Yueting Zhuang**[1]

[1] Zhejiang University, [2] Tianjin University, [3] Microsoft Research Asia
ran159753@tju.edu.cn, {yanyuchen, syl}@zju.edu.cn

GitHub: https://github.com/CCAI-Lab/Self-Braking-Tuning
Project: https://CCAI-Lab.github.io/SBT

Abstract

Large reasoning models (LRMs), such as OpenAI o1 and DeepSeek-R1, have significantly enhanced their reasoning capabilities by generating longer chains of thought, demonstrating outstanding performance across a variety of tasks. However, this performance gain comes at the cost of a substantial increase in redundant reasoning during the generation process, leading to high computational overhead and exacerbating the issue of overthinking. Although numerous existing approaches aim to address the problem of overthinking, they often rely on external interventions. In this paper, we propose a novel framework, **Self-Braking Tuning**(SBT), which tackles overthinking from the perspective of allowing the model to regulate its own reasoning process, thus eliminating the reliance on external control mechanisms. We construct a set of overthinking identification metrics based on standard answers and design a systematic method to detect redundant reasoning. This method accurately identifies unnecessary steps within the reasoning trajectory and generates training signals for learning self-regulation behaviors. Building on this foundation, we develop a complete strategy for constructing data with adaptive reasoning lengths and introduce an innovative braking prompt mechanism that enables the model to naturally learn when to terminate reasoning at an appropriate point. Experiments across mathematical benchmarks (AIME, AMC, MATH500, GSM8K) demonstrate that our method reduces token consumption by up to 60% while maintaining comparable accuracy to unconstrained models.

1 Introduction

Large reasoning models (LRMs) such as OpenAI's o1 [3], Deepseek-R1 [4], QwQ [5], Gemini 2.0 Flash Thinking [6] and Kimi-1.5 [7], excel at mathematical and logical tasks by generating detailed multi-step reasoning, boosting accuracy on complex benchmarks [8]. However, this often results in excessively long inference trajectories, frequently consuming thousands of tokens per problem [9, 10],

* The first two authors have equal contributions. This work was done when the first author was an intern at Zhejiang University.

† Corresponding author.

Preprint. Under review.

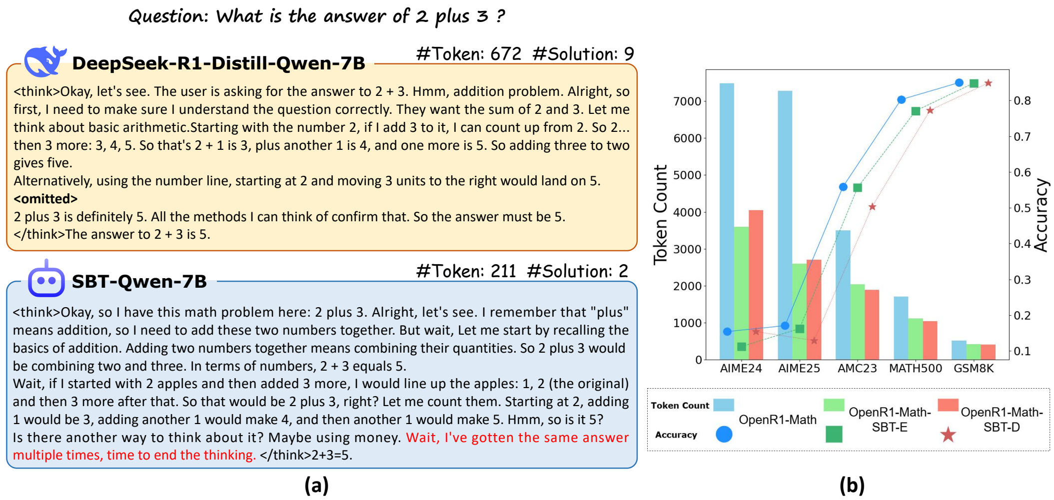


Figure 1: Demonstration of Self-Braking Tuning Effectiveness. In the single-example case (a), the self-braking tuned model exhibits spontaneous termination of overthinking and significantly reduces token usage. On major mathematical benchmarks (b), compared to using OpenR1-Math [1] as the SFT dataset, the self-braking tuned Qwen2.5-Math-1.5B-Instruct [2] achieves a substantial reduction in tokens consumed during inference while maintaining comparable accuracy.

leading to increased computational cost, latency, and redundant reasoning that can obscure core solutions [11]. This "overthinking" [12] poses a significant challenge for practical deployment.

Many recent studies have focused on addressing the problem of overthinking [13, 14], which can be broadly categorized into three approaches: (1) *Model optimization:* Apply reinforcement learning (RL) or supervised fine-tuning (SFT) to equip models with the ability to control reasoning length [15–17]; (2) *Reasoning output optimization*: Dynamically reducing the number of reasoning steps and output length during inference [18–20]; (3) *Adding external restrictions*: Imposing external constraints, such as token budgets, to reduce overthinking [21, 22]. Most existing methods follow the paradigm of external intervention, relying on complex optimization strategies or introducing additional constraint mechanisms, and have yet to fully explore the intrinsic ability of the model to mitigate overthinking on its own.

This reliance on external control prompts a fundamental question: ***Can we enable large reasoning models to autonomously recognize excessive reasoning and terminate their thinking process appropriately?*** Ideally, a model should intrinsically understand when additional reasoning becomes redundant and halt its thought process without external triggers, similar to how humans naturally conclude their reasoning when reaching sufficient certainty.

To address this challenge, we propose **Self-Braking Tuning(SBT)**, a novel framework that teaches LRMs to autonomously identify and terminate redundant reasoning. Unlike previous approaches that impose external constraints, SBT fundamentally reshapes how models perceive and regulate their own reasoning processes. Our key insight is that LRMs can be trained to develop an internal braking mechanism that recognizes when further reasoning becomes unproductive, enabling them to naturally conclude the thought process and transition to formulating the final solution.

Our approach begins with a systematic methodology for identifying overthinking patterns in reasoning trajectories. By combining metrics such as reasoning efficiency ratio and overthinking label ratio, we precisely pinpoint the transition point at which the model shifts from effective reasoning to redundant computation. Based on this analysis, we develop two complementary data construction strategies: (1) Self-Braking Tuning Exact (SBT-E), which strictly removes redundant reasoning segments based on predefined braking points, allowing the model to enter the conclusion phase earlier; (2) Self-Braking Tuning Dynamic (SBT-D), which implements step-level monitoring and dynamically halts reasoning when overthinking patterns emerge.

Building upon the high-quality OpenR1-Math [1] dataset of reasoning trajectories, we constructed two specialized training datasets using these strategies: OpenR1-Math-SBT-E and OpenR1-Math-SBT-D. To further enhance the model's self-awareness of its reasoning state, we introduce braking

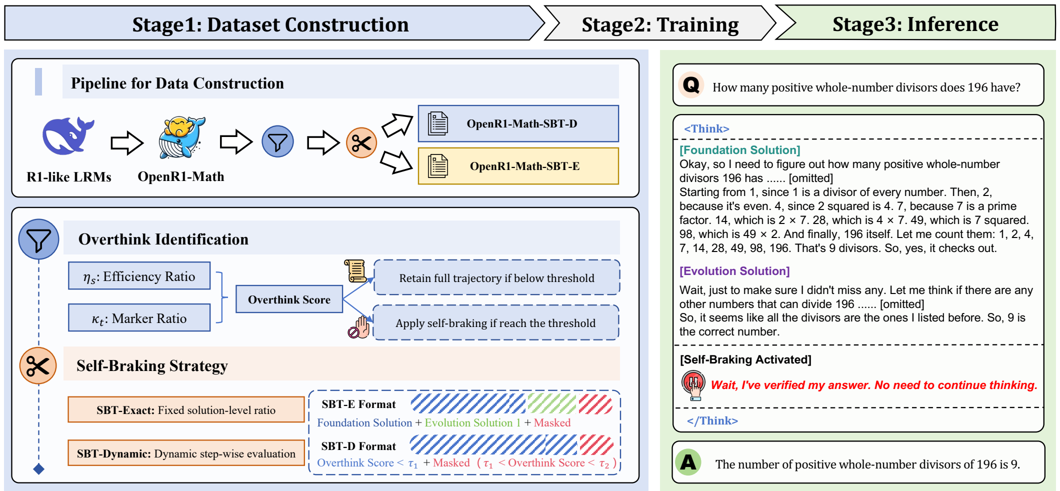


Figure 2: Overview of Self-Braking Tuning. (a) Data construction process with overthinking identification and self braking truncation strategies. (b) An example of automatic reasoning termination in a trained Self-Braking LLM.

prompts at the identified braking points, explicitly simulating the recognition of having sufficiently completed the reasoning process. These prompts enable the model to naturally express an awareness of having reached adequate reasoning depth, thus promoting autonomous termination without the need for external signals. Our experimental results demonstrate consistent improvements in reasoning efficiency across mathematical benchmarks of varying difficulty. While maintaining high accuracy, token consumption was reduced by 30% to 60%. Our contributions can be summarized as follows:

- We introduce a novel tuning framework that enables LRMs to self-regulate reasoning length without external constraints. Self-Braking Tuning cultivates models' intrinsic ability to recognize and inhibit excessive reasoning, fundamentally improving inference efficiency and response quality.
- We propose a systematic methodology for identifying overthinking patterns and develop two complementary data construction strategies: SBT-E and SBT-D, resulting in specialized training datasets for addressing the overthinking problem. These datasets systematically prune redundant reasoning while preserving essential thinking steps.
- We demonstrate that models trained with our SBT framework maintain original accuracy levels while reducing token consumption by up to 60% across multiple benchmarks, confirming the effectiveness and generalizability of our approach in enhancing reasoning efficiency.

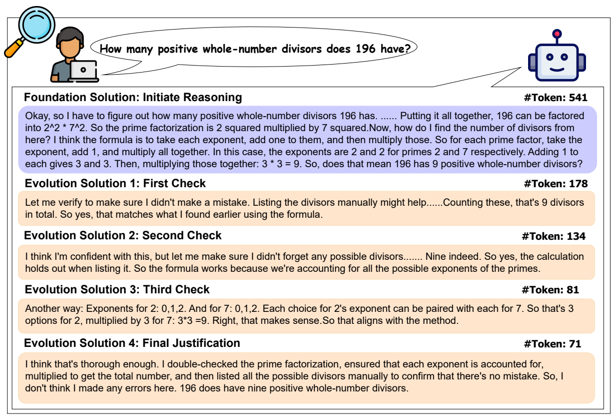
2 Methods

In this section, we begin by analyzing the reasoning trajectories of LRMs to understand the patterns of overthinking (Section 2.1). Based on this analysis, we propose metrics to quantify overthinking (Section 2.2). We then introduce our Self-Braking Tuning framework, which includes two data construction strategies (Section 2.3) and a braking prompt mechanism (Section 2.4).

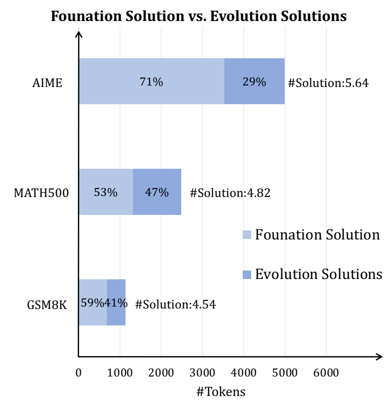
2.1 Reasoning Trajectory Analysis in R1-like Models

Understanding the reasoning structure of LRMs is key to addressing overthinking. Analysis of trajectories from models like DeepSeek-R1 reveals a common pattern: multiple distinct solution attempts are often generated for a single problem. Based on their role and position, these solution segments can be grouped into two main types:

1. **Foundation Solution:** This is the first solution at the beginning of its reasoning process. After comprehending the problem, it proceeds with a step-by-step solution. This forms the foundation of the reasoning process and guides the development of subsequent Evolution Solutions.



(a)



(b)

Figure 3: In panel (a), we present a representative example from DeepSeek-R1-Distill-Qwen-7B. In panel (b), we analyze the model's performance across GSM8K, MATH500, and AIME benchmarks, showing that the Foundation Solution plays a critical role across tasks of varying difficulty.

2. **Evolution Solution:** These solutions appear in the later stages of the model's reasoning process and are often introduced with cues such as "Wait," "Alternatively," or "However." Evolution solutions primarily reflect, refine, supplement, or summarize the foundational solution, and may propose new approaches. While this part of the reasoning grants the model self-correction and improvement capabilities, it is also where overthinking most frequently occurs.

To illustrate how LRMs behave across varying difficulty levels, we constructed a set of math evaluation benchmarks with graduated difficulty and conducted experiments on DeepSeek-Distill-Qwen-7B. The distribution of correct reasoning trajectories is reported in Figure 3. Additionally, a representative example from the MATH500 [23, 24] task is presented to demonstrate the concrete forms of Foundation Solution and Evolution Solution.

2.2 Overthinking Identification

Identifying overthinking in reasoning trajectories is crucial for designing effective self-braking mechanisms. Based on our structural analysis of reasoning, we propose a quantitative framework for detecting and measuring redundancy in the OpenR1-Math dataset, which spans a range of mathematical problems and difficulty levels. To distinguish essential reasoning from unnecessary computation, we introduce two complementary metrics capturing different forms of redundancy, integrated through a composite scoring mechanism.

Reasoning Efficiency Ratio Our first metric addresses a fundamental observation in Section 2.1: LRMs frequently derive correct answers relatively early in their reasoning process but continue generating additional solution attempts. To quantify this inefficiency, we introduce the Reasoning Efficiency Ratio, denoted as

$$\eta_s = \frac{FS}{TS} \tag{1}$$

where FS (First Correct Steps) represents the number of reasoning steps required to reach the first correct answer within the thinking segment (enclosed by <think> and </think> tags) and TS (Total Thinking Steps) represents the total number of steps in the entire thinking segment.

This ratio provides a direct measure of reasoning efficiency: values closer to 1 indicate that the model spent most of its reasoning steps before arriving at the correct answer, reflecting a focused and efficient reasoning process. In contrast, values closer to 0 suggest that the model continued reasoning extensively after reaching the correct answer, which may indicate overthinking or redundant computation. The step-based calculation enables us to assess the structural efficiency of the reasoning process independently of implementation-specific token counts.

Overthinking Marker Ratio While η_s captures structural inefficiency, it does not account for linguistic patterns characteristic of overthinking. Through analysis of high-quality DeepSeek-R1 reasoning trajectories, we identified a set of linguistic markers strongly associated with overthinking behaviors—terms that signal reconsideration, verification, or alternative approach exploration. We formalize this observation through the Overthinking Marker Ratio, denoted as κ_t:

$$\kappa_t = \frac{1}{TT} \sum_{i=1}^{TT} \mathbb{I}[w_i \in \mathcal{M}], \quad \mathbb{I}[w_i \in \mathcal{M}] = \begin{cases} 1, & \text{if } w_i \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{M}$ represents our curated lexicon of overthinking marker terms (complete list in Appendix B), TT (Total Tokens) represents the total number of tokens in the thinking segment, $\mathbb{I}[\cdot]$ is the indicator function that equals 1 when w_i belongs to $\mathcal{M}$ and 0 otherwise.

This ratio quantifies the linguistic footprint of overthinking within a reasoning trajectory. Higher values of κ_t indicate a greater presence of reconsideration and verification language, which typically correlates with redundant reasoning patterns.

Overthink Score To develop a comprehensive assessment of overthinking that leverages both structural and linguistic indicators, we introduce the Overthink Score, a weighted combination of our two metrics:

$$\text{Overthink Score} = \beta \times \kappa_t + (1 - \beta) \times (1 - \eta_s) \tag{3}$$

Note that we transform η_s to $(1 - \eta_s)$ to ensure directional consistency, as higher values of η_s indicate greater efficiency (less overthinking), while higher values of κ_t suggest stronger overthinking patterns. The weighting parameter $\beta \in [0, 1]$ balances the contribution of each component to the final score.

In our implementation, we set $\beta = 0.1$ based on both theoretical considerations and empirical validation. This parameter choice reflects two key insights:

- **Reasoning efficiency dominance**: Early arrival at correct answers significantly reduces computational resources and latency, making $(1 - \eta_s)$ the primary component (weighted at 90%). This prioritization aligns with our objective of minimizing unnecessary computation while preserving reasoning quality.
- **Linguistic indicator robustness**: While κ_t provides valuable signals about overthinking, it exhibits greater sensitivity to variations in prompt formulation, corpus characteristics, and model-specific output patterns. Assigning a lower weight (10%) to κ_t mitigates potential noise amplification from these stylistic fluctuations.

2.3 Adaptive Inference Data Construction

Building on our quantitative framework for overthinking identification, we introduce two complementary strategies for constructing adaptive inference length datasets: Self-Braking Tuning Exact (SBT-E) and Self-Braking Tuning Dynamic (SBT-D). Both approaches aim to preserve reasoning depth while cultivating the model's ability to terminate excessive thinking.

Self-Braking Tuning Exact SBT-E implements a solution-level truncation strategy with consistent reasoning structure across all training examples. For each trajectory exhibiting overthinking, we preserve the Foundation Solution plus one Evolution Solution, followed by a small masked segment from subsequent reasoning. This structured approach ensures the model learns clear boundaries between necessary reasoning and excessive computation.

The Foundation Solution captures the initial structured approach to the problem, while the additional Evolution Solution preserves self-correction capabilities. The masked segment, comprising the beginning of the next Evolution Solution, serves as a braking indicator that signals where further reasoning becomes redundant. During training, this masked content does not contribute to the loss function, thereby avoiding reinforcement of overthinking patterns. Algorithm 1 in Appendix presents the formal procedure for SBT-E construction.

Self-Braking Tuning Dynamic While SBT-E uses uniform truncation, SBT-D adopts a step-wise adaptive strategy that tailors reasoning length to each problem. It incrementally analyzes each reasoning step to determine individualized termination points.

The process begins with the full preservation of the Foundation Solution. Subsequent steps are added one by one, with the Overthink Score recalculated after each. Reasoning continues until the score surpasses a primary threshold τ_1 (set to 0.2), allowing complex problems to retain more steps and simpler ones to terminate earlier.

A masking segment is then defined from steps with Overthink Scores between τ_1 and a secondary threshold τ_2 (set to $\tau_1 + 5\%$). This segment is excluded from loss computation but retained to expose the model to overthinking patterns without reinforcing them. The full procedure is outlined in Algorithm 2 in Appendix.

Together, SBT-E and SBT-D yield the OpenR1-Math-SBT-E and OpenR1-Math-SBT-D datasets, each with 92,064 examples, designed to train models to autonomously terminate redundant reasoning.

2.4 Self-Regulating Braking Strategy

Beyond generating adaptive-length reasoning, we introduce complementary training mechanisms to enhance the model's ability to stop reasoning autonomously. Our Self-Regulating Braking Strategy includes two components: masked redundant thinking and natural language braking signals—both aimed at fostering self-awareness of reasoning efficiency.

Masked Redundant Thinking While both SBT-E and SBT-D identify optimal truncation points, simply cutting off reasoning there doesn't help models learn to detect overthinking. Instead, we retain a small portion of redundant reasoning and apply loss masking to prevent it from affecting training. For each SBT-E or SBT-D sample, we append this masked segment right after the preserved valid reasoning: in SBT-E, it's the start of the second Evolution Solution; in SBT-D, it includes steps with Overthink Scores between thresholds τ_1 and τ_2. This consistent strategy across both methods ensures balanced exposure.

By presenting overthinking patterns without calculating their loss, the model learns to distinguish between productive reasoning and redundancy. The masked segment serves as a soft boundary cue, encouraging the model to stop reasoning autonomously during inference.

Natural Language Guidance We further enhance self-regulation by adding clear natural language cues at reasoning stop points. These braking signals are self-reflective statements that show awareness of finishing reasoning. For example, *"Wait, I've gotten the same answer multiple times, time to end the thinking."*

Placed at the boundary between preserved and masked content, these cues act as linguistic anchors for stopping decisions. Unlike special tokens or external rules, natural language signals fit naturally with the model's abilities, provide explicit metacognitive hints, and keep the reasoning fluent while clearly indicating when to stop.

3 Experiments

We conduct extensive experiments to evaluate the effectiveness of Self-Braking Tuning across various model architectures and mathematical reasoning tasks. Our evaluation aims to answer three key questions: (1) How effectively does SBT reduce token consumption while preserving accuracy? (2) How does performance vary across different model sizes and architectures? (3) How do the two SBT variants (SBT-E and SBT-D) compare in practice?

3.1 Experimental Setup

Datasets and Training. We curate a dataset of 92K high-quality instances from OpenR1-Math [1][1] by applying a 16K context limit and filtering out problematic samples (e.g., those with multiple

[1]The OpenR1-Math dataset is licensed under the Apache 2.0 License. We adhere to its terms of use and do not redistribute the dataset but build upon it for experimental purposes.

Base Model	Method	GSM8K		MATH500		AIME		AMC23		AVERAGE	
		Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok
Qwen2.5-Math-1.5B-Instruct	Baseline	85.00	514	80.25	1712	16.25	7381	55.94	3503	**59.36**	3277
	SBT-E	84.85	426	77.10	1121	13.75	3101	55.63	2044	57.83	**1673**
	SBT-D	84.87	414	77.30	1046	14.17	3381	50.31	1888	56.66	1682
Qwen2.5-Math-7B-Instruct	Baseline	96.11	1460	92.67	3816	40.83	11904	83.13	6937	**78.19**	6029
	SBT-E	95.45	997	90.77	2501	38.75	8772	77.19	4443	75.54	**4178**
	SBT-D	95.37	956	91.15	2629	38.38	9778	80.06	5208	76.24	4643
Llama-3.2-1B-Instruct	Baseline	41.85	1639	25.22	6624	1.25	13150	9.38	10210	19.43	7906
	SBT-E	39.96	1056	24.35	3180	0.42	6615	9.06	4708	18.45	3890
	SBT-D	41.21	698	25.07	2591	1.04	6821	13.13	4388	**20.11**	**3624**
Llama-3.1-8B-Instruct	Baseline	88.03	1593	59.98	9304	9.58	13663	36.75	9742	48.59	8576
	SBT-E	85.03	777	57.60	2292	6.84	5658	33.44	4045	45.73	3193
	SBT-D	88.27	997	62.60	3847	7.70	5845	38.12	6476	**49.17**	**4291**

Table 1: Performance of different models with Self-Braking Tuning applied, evaluated across GSM8K, MATH500, AMC23, and AIME (including AIME24 and AIME25) benchmarks.

</think> tags). This filtered dataset serves as our baseline. We then construct our SBT-E and SBT-D variants following the methodology detailed in Section 2.3.

We perform supervised fine-tuning on both mathematical specialists (Qwen2.5-Math-1.5B/7B-Instruct [2]) and general-purpose models (Llama-3.2-1B and Llama-3.1-8B-Instruct [25]). All models are trained using Megatron-LM for 3 epochs with a 1e-5 initial learning rate, cosine decay schedule, 0.03 warm-up ratio, and 16,384-token maximum sequence length. Training is conducted on 64 Ascend H910B-64G hardware.

Evaluation Benchmarks. We evaluate performance across four mathematical reasoning benchmarks of varying difficulty: AIME (24&25, competition-level algebraic problems), AMC23 (pre-collegiate mathematics), MATH500 [23, 24] (diverse mathematical problems), and GSM8K [26] (grade school math word problems). For inference, we use vLLM [27] with temperature 0.7, generating 8 samples per question and reporting average accuracy. All inference are performed on NVIDIA A100 GPUs.

3.2 Main Results

Table 1 presents the performance of models trained with our Self-Braking Tuning approaches compared to baselines trained on the unmodified dataset. We observe several significant trends:

Substantial Token Reduction with Preserved Acc. Both SBT variants achieve remarkable reductions in token consumption while maintaining comparable accuracy to baseline models. For the Qwen2.5-Math-7B-Instruct model, SBT-E and SBT-D reduce token usage by 30.7% and 23.0% respectively, with accuracy drops of only 2.65% and 1.95%. Even more impressively, when applied to the Llama-3.1-8B-Instruct model, SBT-E reduces token consumption by 62.8% while preserving 94.1% of the baseline accuracy.

Scaling Dynamics Across Model Types Efficiency gains from Self-Braking Tuning (SBT) vary by model type. For general-purpose models like Llama, larger models benefit more—token reductions improve from 54.2% (1B) to 62.8% (8B). In math-specialized models, however, larger models see smaller gains (30.7% for 7B vs. 48.9% for 1.5B), suggesting that specialized models already have more focused and efficient reasoning, leaving less room for further compression. These findings indicate that the self-regulation benefits from SBT depend not only on model scale but also on whether the model is trained for general-purpose or domain-specific tasks.

SBT-E vs. SBT-D Performance. The two proposed variants show distinct performance characteristics. SBT-E generally achieves greater token reductions (averaging 48.3% across all models compared to 43.9% for SBT-D) but with slightly larger accuracy drops. SBT-D demonstrates more balanced performance, particularly on the most challenging AIME and MATH500 benchmarks. Notably, for

Method	Threshold	Acc	#Tok
Baseline	–	59.36	3278
SBT-Exact	0.2	**57.83**	**1673**
	0.3	56.70	1755
	0.4	57.38	1834
SBT-Dynamic	0.2	56.66	**1682**
	0.3	**57.47**	1917
	0.4	57.36	1902

Table 2: Performance across overthink score thresholds. Detailed results shown in Appendix Table 6.

Reservations & Masked Content	Acc	#Tok
Baseline	59.36	3277
1 solution & A few sentences	56.95	1700
1 solution & 1 solution	57.69	1697
2 solutions & A few sentences	**57.83**	**1673**
2 solutions & 1 solution	57.45	1684

Table 3: Performance with different configurations of preserved reasoning and masked redundant content. Detailed results shown in Appendix Table 7.

the Llama-3.1-8B model, SBT-D actually improves accuracy on MATH500 by 2.62 % while reducing tokens by 58.7%, suggesting that dynamic truncation may help eliminate not just redundant reasoning but potentially harmful overthinking in some cases.

4 Analysis

4.1 Impact of Overthinking Thresholds

The threshold for classifying overthinking instances significantly impacts both dataset composition and model performance. We experimented with thresholds of 0.2, 0.3, and 0.4, which classified approximately 60%, 50%, and 40% of samples as overthinking cases, respectively. As shown in Table 2, a threshold of 0.2 yields the best performance for SBT-E, achieving an optimal balance between token reduction (49% fewer tokens than baseline) and accuracy preservation (97.4% of baseline accuracy).

This finding reveals a crucial insight: aggressive overthinking identification (lower thresholds) leads to more substantial efficiency gains without proportional accuracy losses. This suggests that a significant portion of reasoning in LRMs truly is redundant and can be eliminated without compromising problem-solving capabilities. The consistent pattern across both SBT variants indicates that identifying and addressing overthinking in approximately 60% of reasoning instances represents an optimal operating point for Self-Braking Tuning.

4.2 Preserved Reasoning and Redundancy Masking Trade-off

To develop effective self-braking behavior, models learn both when to continue reasoning and when to stop. We investigated different configurations of preserved (unmasked) and masked content to understand this balance. Shown in Table 3, preserving two complete solutions while masking only a few additional sentences yields optimal performance, reducing tokens by 49% while maintaining 97.4% of baseline accuracy.

This finding provides two key insights. First, solution repetition serves as a natural termination signal: when a model derives the same answer twice, it learns this is a strong indication to conclude reasoning. Second, we observe an inverse relationship between preserved and masked content: with more preserved reasoning (two solutions), less masked content is optimal; with less preserved reasoning (one solution), more masked content performs better.

This relationship suggests that models require a certain "reasoning quota" to develop robust problem-solving capabilities, which can be satisfied either through more preserved reasoning or more exposure to masked reasoning patterns. However, the superior performance of the "two solutions with minimal masked content" configuration indicates that clearly delineated, complete reasoning paths provide stronger learning signals than exposure to additional masked content.

4.3 Step-Level vs. Token-Level Overthinking Detection

The granularity of overthinking detection, whether operating at the reasoning step level or token level, impacts both the coherence of preserved reasoning and overall model performance. We compared our step-level approach with a token-level alternative using a token efficiency ratio defined as $\eta_t = \frac{FT}{TT}$, where FT represents tokens until first correct answer and TT represents total tokens.

Level	Acc	#Tok
Baseline	59.36	3277
Step-Level	**56.66**	**1682**
Token-Level	56.24	1753

Table 4: Overthinking detection granularity comparison. Detailed results in Appendix Table 8.

Guiding Mode	Acc	#Tok
Baseline	59.36	3277
Natural Language	**56.66**	**1682**
Special Token	56.61	1797

Table 5: Guiding mode comparison. Detailed results shown in Appendix Table 9.

Results in Table 4 demonstrate that step-level detection outperforms token-level approaches, achieving both higher accuracy and lower token usage. This confirms our hypothesis that reasoning coherence is better preserved when entire logical steps are maintained intact. Token-level truncation, while more granular, risks breaking logical units of reasoning, potentially creating disjointed or incomplete thinking patterns that are harder for models to learn from or reproduce effectively.

This finding highlights the importance of respecting the inherent structure of reasoning when developing overthinking mitigation strategies: models benefit from complete logical units rather than more aggressive but potentially incoherent truncation approaches.

4.4 Natural Language Guidance vs. Special Token Guidance

A fundamental aspect of Self-Braking Tuning is the mechanism used to signal reasoning termination. We compared our natural language guidance approach (epiphany sentences like "I've verified my answer, no need to continue...") with a special token approach using <stop_overthinking> as an explicit control signal. As shown in Table 5, natural language guidance demonstrates superior performance, achieving equivalent accuracy with significantly fewer tokens (1682 vs. 1797). This suggests that metacognitive self-reflection embedded in natural language provides more effective learning signals than explicit control tokens.

This insight aligns with the inherent nature of LRMs, which are fundamentally language models trained to understand semantic and contextual relationships. Natural language guidance leverages the model's existing capabilities to recognize logical transitions and inference completion, rather than introducing artificial control mechanisms that require learning new conventions. This finding suggests that future approaches to controlling LRM behavior may benefit from working with, rather than around, their foundational linguistic capabilities.

5 Related Works

Large Reasoning Models Large reasoning models (LRMs) extend traditional language models with advanced reasoning capabilities, often enabled by reinforcement learning. OpenAI's o1 series [3] marked a key milestone, followed by DeepSeek-R1 [4], which matched o1's performance through a cost-efficient combination of supervised fine-tuning and RL. Subsequent models like Kimi-k1.5 [7] and QwQ-32B [5] further solidified the LRM era. Concurrently, alternative approaches have emerged to reduce reliance on RL, leveraging supervised fine-tuning and data distillation. Models such as DeepSeek Distill, OpenR1-Math-7B [28], Sky-T1 [29], and LIMO [30] have shown that strong reasoning performance can also be achieved without RL, broadening the design space for LRMs.

Efficient Reasoning Overthinking is a common behavior of large reasoning models, where models generate unnecessarily long answers instead of stopping at the right time. Existing studies addressing overthinking [13] can be grouped into three categories: (1) *Model Optimization:* This line improves model behavior via post-training techniques, primarily reinforcement learning (RL) and supervised fine-tuning (SFT). RL-based methods design length-sensitive rewards to limit output [16, 17, 15, 31]. SFT-based methods use variable-length chain-of-thought (CoT) data and auxiliary constraints to shorten reasoning paths [32–36]. (2) *Reasoning Output Optimization:* This direction reduces reasoning length at inference time by altering generation strategies. *lightthinker* [20] compresses intermediate steps; *DEER* [19] halts once high confidence is reached; *NoThinking* [18] skips reasoning entirely via prompting. These efforts are representative, with many other studies exploring similar directions [37–40]. (3) These methods impose constraints like token budgets or prompt controls to regulate reasoning behavior. In addition to representative works such as Token-Budget [21] and CoD [22], numerous other studies have explored similar constraint-based strategies [41–43].

6 Conclusion

In this paper, we propose a novel endogenous approach, Self-Braking Tuning (SBT), to mitigating overthinking in large language models. SBT aims to stimulate the model's ability to autonomously identify and stop redundant reasoning. We construct a data framework with adaptive reasoning lengths, where overthinking characteristics are extracted through step-level analysis and keyword annotation. Based on this, we design two data generation strategies, SBT-E and SBT-D, to help the model learn when to stop and how to simplify its reasoning process. During supervised fine-tuning, we introduce redundancy masking and epiphany sentences to preserve the reasoning paradigm while enhancing the model's sensitivity to redundant thought patterns. Experimental results show that SBT models significantly reduce token consumption by 30%–60% on multiple mathematical benchmarks, with minimal impact on accuracy. Our work demonstrates the potential for large reasoning models to self-regulate their reasoning and offers a promising direction for more efficient long-chain reasoning.

References

[1] OpenR1 Team. OpenR1-Math-220k Dataset. `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`, 2025.

[2] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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

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

[5] Team Qwen. Qwq: Reflect deeply on the boundaries of the unknown. `https://qwenlm.github.io/blog/qwq-32b-preview/`, 2024.

[6] Google DeepMind. Gemini flash thinking. `https://deepmind.google/technologies/gemini/flash-thinking/`, 2025.

[7] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[8] Ehsan Latif, Yifan Zhou, Shuchen Guo, Lehong Shi, Yizhu Gao, Matthew Nyaaba, Arne Bewerdorff, Xiantong Yang, and Xiaoming Zhai. Can openai o1 outperform humans in higher-order cognitive thinking? *arXiv preprint arXiv:2412.05753*, 2024.

[9] OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`, 2024.

[10] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

[11] Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

[12] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

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

[14] Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. *arXiv preprint arXiv:2503.24377*, 2025.

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

[16] 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.

[17] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

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

[19] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025.

[20] Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*, 2025.

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

[22] Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

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

[24] 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.

[25] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[26] 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.

[27] 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.

[28] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025.

[29] NovaSky Team. Sky-t1: Fully open-source reasoning model with o1-preview performance in $450 budget, 2025.

[30] 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.

[31] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*, 2025.

[32] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[33] Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*, 2025.

[34] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320, 2025.

[35] 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.

[36] 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.

[37] Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*, 2025.

[38] Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*, 2024.

[39] Rohin Manvi, Anikait Singh, and Stefano Ermon. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*, 2024.

[40] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.

[41] Ayeong Lee, Ethan Che, and Tianyi Peng. How well do llms compress their own chain-of-thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*, 2025.

[42] Matthew Renze and Erhan Guven. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 476–483. IEEE, 2024.

[43] Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*, 2025.

[44] Ximing Lu, Seungju Han, David Acuna, Hyunwoo Kim, Jaehun Jung, Shrimai Prabhumoye, Niklas Muennighoff, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, et al. Retro-search: Exploring untaken paths for deeper and efficient reasoning. *arXiv preprint arXiv:2504.04383*, 2025.

[45] Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *arXiv preprint arXiv:2504.06514*, 2025.

[46] Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G Patil, Matei Zaharia, et al. Llms can easily learn to reason from demonstrations structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*, 2025.

A Limitations

While *Self-Braking Tuning* (SBT) has demonstrated its ability to reduce overthinking by enabling models to autonomously regulate reasoning length, several limitations remain:

- **Domain Generalization.** Our study focuses primarily on math reasoning tasks (e.g., GSM8K, MATH), which, while structurally rich and sensitive to overthinking, do not cover the full diversity of reasoning challenges. The applicability of SBT to open-ended, commonsense, logical, or multimodal reasoning remains unverified. These domains may exhibit distinct redundancy patterns that require specialized adaptation strategies.
- **Data Scalability and Adaptivity.** Due to computational constraints, our experiments are conducted on a dataset of approximately 92K examples. The impact of scaling to larger datasets (e.g., millions of samples) remains unexplored. Moreover, current data construction strategies (SBT-E/D) rely on fixed threshold parameters for overthinking detection, which may require manual tuning across different tasks and hinder dynamic adaptation.
- **Overthinking Signal Definition.** We use reasoning efficiency (η_s) and overthink marker ratio (κ_t) to estimate redundancy, but these metrics may not capture all subtle or latent forms of reasoning utility. Some steps that appear redundant may actually serve a hidden purpose, such as helping the model organize its reasoning or implicitly represent abstract patterns.
- **Interpretability and Controllability.** While SBT introduces a soft constraint via loss masking, the internal process by which a model decides to terminate reasoning remains opaque. There is no explicit mechanism to trace or control this decision, limiting transparency and reliability—especially in applications requiring high interpretability, such as education, healthcare, or scientific domains.
- **Potential Trade-offs in Complex Tasks.** For tasks requiring deep, multi-step reasoning (e.g., theorem proving), prematurely terminating reasoning may risk omitting critical steps. Although SBT preserves accuracy in most cases, it may underperform in scenarios where full-chain reasoning is essential. Adaptive or progressive braking strategies may be needed to better balance efficiency and completeness.

Future work could address these limitations by (i) expanding SBT to open-ended and multimodal tasks, (ii) scaling to larger and more diverse datasets, (iii) developing adaptive thresholding and automatic data construction pipelines, (iv) improving multilingual and domain-robust prompt designs, and (v) integrating interpretable or feedback-driven stopping mechanisms that better align with complex reasoning dynamics.

B Overthink Markers

In the study of overthinking behaviors, several prior works have highlighted that certain words associated with reflection, hesitation, or backtracking play a critical role in identifying and guiding redundant reasoning processes [17, 44–46]. Building on these insights, we compile a set of common Overthink Markers, including:

Another	Backtrack	But
Check	Going back	Hmm
Hmmm	However	Hold on
Instead of	Just to be thorough	Just to make sure
Let me check	Let me just double-check	Let me try another
Let me verify	Maybe	Maybe I can consider
Maybe I should consider	Might	Not sure
Perhaps	Recheck	Retry
Trace back	Wait	

These terms frequently co-occur with behaviors such as repeated verification, alternative hypothesis formulation, or reasoning path retracing, and are therefore treated as linguistic indicators of redundant cognitive load during reasoning. Based on this, we construct the set $\mathcal{M}$ as part of our overthinking detection metric, used to compute redundancy density at the linguistic level and help identify potential efficiency bottlenecks in deep reasoning.

C Algorithms of Self-Braking Tuning

To facilitate reproducibility and offer a transparent view into our data construction pipeline, we provide the formal procedures for the Self-Braking Tuning Exact (SBT-E) and Self-Braking Tuning Dynamic (SBT-D) methods in this appendix. These two strategies represent complementary approaches for curating training data that teach models to regulate their own reasoning length and avoid excessive computation.

SBT-E adopts a uniform truncation scheme, where each reasoning trajectory is truncated at a consistent structural boundary: the Foundation Solution and the first Evolution Solution are preserved, followed by a small masked portion of subsequent reasoning. This approach provides a clean and interpretable signal for identifying the onset of overthinking.

In contrast, SBT-D offers a fine-grained, adaptive strategy that dynamically determines the optimal stopping point for each problem based on the model's own overthinking scores. It incrementally evaluates reasoning steps, retaining those below a primary overthinking threshold, and masks additional steps that exceed this threshold but remain below a secondary cutoff—effectively preserving problem-specific nuances in reasoning depth.

Crucially, both methods employ loss masking on the redundant segments: the model sees these overthinking patterns during training but receives no gradient updates from them. This enables the model to implicitly recognize and learn to avoid overthinking, without being rewarded for producing verbose or unnecessary steps.

The full algorithmic details are outlined in Algorithms 1 and 2, which serve as a blueprint for implementing the Self-Braking Tuning framework.

Algorithm 1 Self-Braking Tuning Exact (SBT-E)

Require: Reasoning trajectory T with Foundation Solution FS and Evolution Solutions $ES = [ES_1, ES_2, ...]$
Ensure: Modified trajectory T' with preserved and masked segments
PreservedSegment $\leftarrow FS + ES_1$
if $|ES| > 1$ **then**
    MaskedSegment $\leftarrow$ First 10-20% of ES_2
else
    MaskedSegment $\leftarrow \emptyset$
end if
$T' \leftarrow$ PreservedSegment + MaskedSegment {With loss masking on MaskedSegment}
return T'

Algorithm 2 Self-Braking Tuning Dynamic (SBT-D)

Require: Reasoning trajectory T with steps $[S_1, S_2, ..., S_n]$, thresholds τ_1 and τ_2
Ensure: Modified trajectory T' with preserved and masked segments
PreservedThinking $\leftarrow$ Foundation Solution from T
$i \leftarrow$ index of first step after Foundation Solution
while $i \leq n$ **and** CalculateOverthinkScore(PreservedThinking $+ S_i) < \tau_1$ **do**
    PreservedThinking $\leftarrow$ PreservedThinking $+ S_i$
    $i \leftarrow i + 1$
end while
MaskedThinking $\leftarrow \emptyset$
while $i \leq n$ **and** CalculateOverthinkScore(PreservedThinking + MaskedThinking $+ S_i) < \tau_2$ **do**
    MaskedThinking $\leftarrow$ MaskedThinking $+ S_i$
    $i \leftarrow i + 1$
end while
$T' \leftarrow$ PreservedThinking + MaskedThinking {With loss masking on MaskedThinking}
return T'

D Detailed Experimental Results

In Section 4, to improve the readability of the main text, we only present the average results across the datasets. Here, we provide the specific data and evaluation results.

Method	Threshold	GSM8K		MATH500		AIME		AMC23		AVERAGE	
		Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok
Baseline	-	85.00	514	80.25	1712	16.25	7381	55.94	3503	59.36	3277
SBT-Exact	0.2	84.85	426	77.10	1121	13.75	3101	55.63	2044	**57.83**	**1673**
	0.3	85.16	424	77.25	1113	15.63	3353	48.75	2132	56.70	1755
	0.4	84.73	421	77.40	1130	12.71	3795	54.69	1988	57.38	1834
SBT-Dynamic	0.2	84.87	414	77.30	1046	14.17	3381	50.31	1888	56.66	**1682**
	0.3	84.58	410	78.00	1125	12.92	3710	54.37	2422	**57.47**	1917
	0.4	85.07	407	78.73	1187	14.38	3593	51.25	2421	57.36	1902

Table 6: Performance with varying overthink score thresholds. A lower threshold (e.g., 0.2), which identifies more reasoning instances as overthinking, yields the best overall performance—achieving up to 49% token reduction while preserving 97.4% of baseline accuracy, especially for SBT-Exact. This supports the effectiveness of aggressive pruning in eliminating redundant reasoning across tasks. On easier datasets like GSM8K, accuracy remains stable across all thresholds. For more challenging datasets such as AIME, although 0.2 still performs best overall, slightly higher thresholds (0.3–0.4) show marginal improvements in accuracy at the cost of increased token usage, suggesting that task complexity can influence the sensitivity to pruning aggressiveness.

Reservations & Masked Content	GSM8K		MATH500		AIME		AMC23		AVERAGE	
	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok
Baseline	85.00	514	80.25	1712	16.25	7381	55.94	3503	59.36	3277
1 solution & A few sentences	85.23	416	78.00	1103	12.71	3132	51.88	2148	56.95	1700
1 solution & 1 solution	85.06	432	78.60	1101	13.96	3178	53.12	2076	57.69	1697
2 solutions & A few sentences	84.85	426	77.10	1121	13.75	3101	55.63	2044	**57.83**	**1673**
2 solutions & 1 solution	84.77	411	77.82	1034	12.50	3092	54.69	2197	57.45	1684

Table 7: Performance corresponding to different combinations of preserved and masked reasoning. The best configuration—preserving two complete solutions while masking only a few redundant sentences—achieves the highest average accuracy (57.83%) with 49% fewer tokens than baseline. On simpler datasets like GSM8K, even minimal preservation suffices for learning effective termination, while harder tasks such as AIME benefit more from exposing multiple complete solutions, highlighting that task complexity influences the optimal balance between reasoning exposure and truncation cues.

Level	GSM8K		MATH500		AIME		AMC23		AVERAGE	
	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok
Baseline	85.00	514	80.25	1712	16.25	7381	55.94	3503	59.36	3277
Step-Level	84.87	414	77.30	1046	14.17	3381	50.31	1888	**56.66**	**1682**
Token-Level	85.09	431	78.23	1088	11.34	3399	50.31	2091	56.24	1753

Table 8: Performance comparison between step-level and token-level overthinking detection. Step-level supervision consistently achieves lower token usage across all datasets (e.g., 414 vs. 431 on GSM8K, 1888 vs. 2091 on AMC23), indicating more efficient reasoning truncation. Accuracy-wise, the two methods are comparable on GSM8K and MATH500, but step-level clearly outperforms token-level on more challenging tasks such as AIME (14.17% vs. 11.34%). These results highlight that preserving complete reasoning steps enables better efficiency–accuracy trade-offs, especially for complex problem-solving.

Guiding Mode	GSM8K		MATH500		AIME		AMC		AVERAGE	
	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok	Acc	#Tok
Baseline	85.00	514	80.25	1712	16.25	7381	55.94	3503	59.36	3277
Natural Language	84.87	414	77.30	1046	14.17	3381	50.31	1888	**56.66**	**1682**
Special Token	84.94	413	77.92	1120	12.34	3647	51.25	2007	56.61	1797

Table 9: Performance comparison between natural language and special token guiding modes for reasoning termination. Natural language guidance achieves comparable or better accuracy with consistently lower token usage across datasets—e.g., 3381 vs. 3647 tokens on AIME, and 1888 vs. 2007 on AMC—demonstrating more efficient reasoning control. Accuracy differences are minor on GSM8K and MATH500, but natural language guidance significantly outperforms special tokens on AIME (14.17% vs. 12.34%). These results suggest that semantically aligned, self-reflective cues are more effective than explicit control tokens, especially in complex reasoning scenarios.