

# ThinkBrake: Efficient Reasoning via Log-Probability Margin Guided Decoding

Sangjun Song\*, Minjae Oh\*, Seungkyu Lee<sup>‡</sup>, Sungmin Jo<sup>§</sup>, Yohan Jo<sup>†</sup>

Graduate School of Data Science, Seoul National University

{ssangjun706, kosair, gyuu01, jsm0424, yohan.jo}@snu.ac.kr

## Abstract

Large Reasoning Models (LRMs) allocate substantial inference-time compute to Chain-of-Thought (CoT) reasoning, improving performance on mathematics, scientific QA, and tool usage. However, this introduces *overthinking*: LRMs often reach a correct intermediate solution, continue reasoning, and overwrite it with an incorrect answer. We first demonstrate that oracle stopping—where we inject `</think>` at every sentence boundary and select the best stopping point in hindsight—improves average accuracy by 8% while reducing thinking tokens by 72%, exposing substantial overthinking. Motivated by this finding, we propose THINKBRAKE, which monitors the log-probability margin between the top continuation token and `</think>` at sentence boundaries, stopping reasoning when this margin narrows. THINKBRAKE requires no training and achieves favorable accuracy–efficiency trade-offs across math, scientific QA, and tool usage benchmarks, reducing thinking token usage by up to 30%. Furthermore, we provide theoretical analysis showing that THINKBRAKE is equivalent to test-time realignment with a reward bonus for the `</think>`. Code is available at <https://github.com/holi-lab/ThinkBrake>.

## 1 Introduction

Recent progress in Large Reasoning Models (LRMs) (OpenAI, 2024; Guo et al., 2025; Yang et al., 2025) has demonstrated remarkable capabilities across various tasks such as mathematics, scientific QA, and tool usage. By allocating inference-time computation through Chain-of-Thought (CoT) reasoning (Wei et al., 2022), LRMs demonstrate behaviors like self-correction and iterative refinement,

actively improving performance with increased token usage (Chen et al., 2025; Xu et al., 2025). However, this increased inference-time compute raises practical concerns about *efficient reasoning* (Feng et al., 2025), as longer trajectories introduce higher latency and costs. More critically, it introduces *overthinking* (Zhang et al., 2025a): LRMs frequently reach a correct intermediate step, only to continue deliberating and overwrite it with an incorrect final output (see Figure 1, left panel).

We first conduct a preliminary oracle analysis to validate that overthinking is a genuine problem and that LRMs would benefit from early termination of reasoning. Leveraging the fact that recent LRMs use special tokens `<think>` and `</think>` to wrap reasoning (where `</think>` triggers answer generation), we inject `</think>` at every sentence boundary, forcing early termination at each possible reasoning step (see §2). Across mathematical reasoning benchmarks (GSM8K, MATH500, AIME2024, AIME2025) and tool usage (Berkeley Function Calling Leaderboard; BFCL), this oracle approach achieves an average accuracy gain of 8% while reducing reasoning tokens by 72%. This gap reveals substantial recoverable headroom: many failures can be prevented simply by stopping at the right moment (see Figure 1, right panel).

Motivated by this oracle analysis, we introduce THINKBRAKE, a practical approach to early stopping. THINKBRAKE monitors the log-probability margin between `</think>` and the current top token at sentence boundaries, stopping reasoning when this margin narrows. THINKBRAKE is model-agnostic, requiring only an explicit reasoning format with `</think>`. While prior works have explored test-time methods for reasoning efficiency (Li et al., 2025; Wang et al., 2025; Laaouach, 2025), they often rely on hand-crafted heuristics. In contrast, we provide a theoretical foundation for THINKBRAKE, showing it is equivalent to KL-regularized test-time realignment that assigns a

\*Equal contribution †Corresponding author.

<sup>‡</sup>Visiting intern from Department of Industrial Engineering, Seoul National University.

<sup>§</sup>Visiting intern from Department of Computer Science and Engineering, Seoul National University.

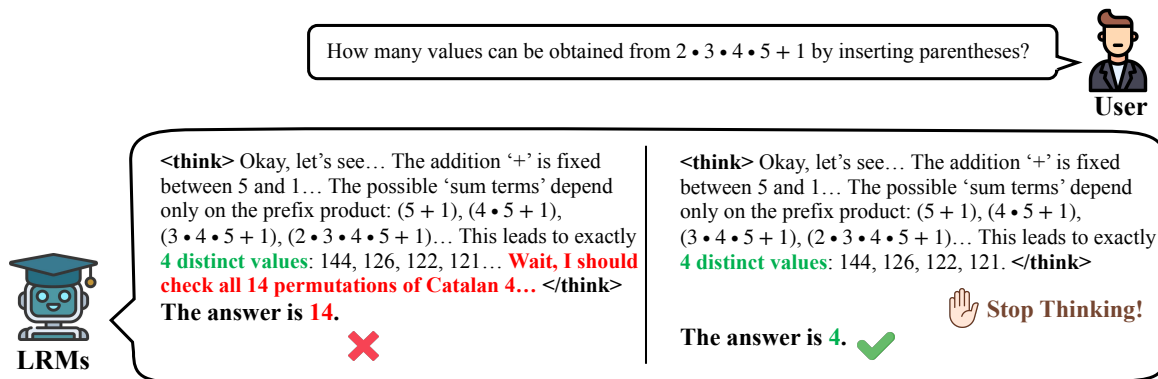


Figure 1: An example of LRM overthinking (left). The LRM arrives at the correct solution of 4, but continues reasoning after “Wait” and overwrites it with an incorrect answer of 14. Appropriate early termination via `</think>` injection removes overthinking and leads to the correct answer (right).

reward bonus to `</think>`, pushing the LRM towards concise reasoning (see §3.2).

We evaluate THINKBRAKE on six LRMs across four mathematical reasoning benchmarks (GSM8K, MATH500, AIME2024, AIME2025), two additional reasoning benchmarks (GPQA-Diamond, ARC-Challenge), and two tool usage benchmarks (BFCL, Meta-Tool), demonstrating consistent efficiency gains with competitive accuracy (see §4). We show that THINKBRAKE preserves accuracy uniformly across problem difficulties, rather than trading off performance on easy versus hard instances. Furthermore, THINKBRAKE-generated trajectories can serve as training data for Direct Preference Optimization (DPO). This enables LRMs to internalize concise reasoning without inference-time intervention and achieve improved efficiency with only a small amount of training data (see §4.4). Our contributions are:

- We identify and quantify overthinking in LRMs via oracle `</think>` rollouts, quantifying recoverable headroom across mathematical reasoning and tool usage (see §2).
- We introduce THINKBRAKE, a test-time inference method based on a log-margin criterion at sentence boundaries, with theoretical grounding as KL-regularized test-time realignment assigning a reward bonus to `</think>` (see §3).
- We validate THINKBRAKE across six LRMs and eight benchmarks spanning mathematical, scientific, and tool-use reasoning, and show its generated data enables efficient reasoning via DPO (see §4).

## 2 Overthinking in LRMs

To validate that overthinking is a genuine problem and quantify the potential gains from early stopping, we analyze whether LRMs would have answered correctly had reasoning been terminated earlier. We conduct controlled rollouts with forced termination by injecting `</think>` at every sentence boundary, causing the model to stop thinking and produce an answer. We mark a trajectory as *recoverable* if it contains a sentence boundary where the model has reached the correct answer but continues reasoning and eventually produces an incorrect one. This yields an *oracle* accuracy—the achievable accuracy with hindsight knowledge of the optimal stopping point (see Appendix B for an example).

We use Qwen3-4B-Thinking (Yang et al., 2025) as our base LRM and evaluate across mathematical reasoning (GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AIME2024, AIME2025 (Art of Problem Solving, 2025)) and tool usage (BFCL-v1 (Patil et al., 2025)). Detailed experiment settings are in Appendix C.

In Figure 2, oracle stopping reduces error rate to 6% on mathematical reasoning and 5% on tool usage, improving over the baseline by 8% while reducing thinking token usage by 54% on mathematics and 89% on tool usage. Remarkably, the improvements are consistent across difficulty levels; even on the challenging AIME benchmarks, oracle stopping recovers 61% of failures while cutting tokens by 44%. Notably, fewer than 5% of errors are irrecoverable for most benchmarks, confirming that a substantial portion of failures stem from overthinking rather than inability.

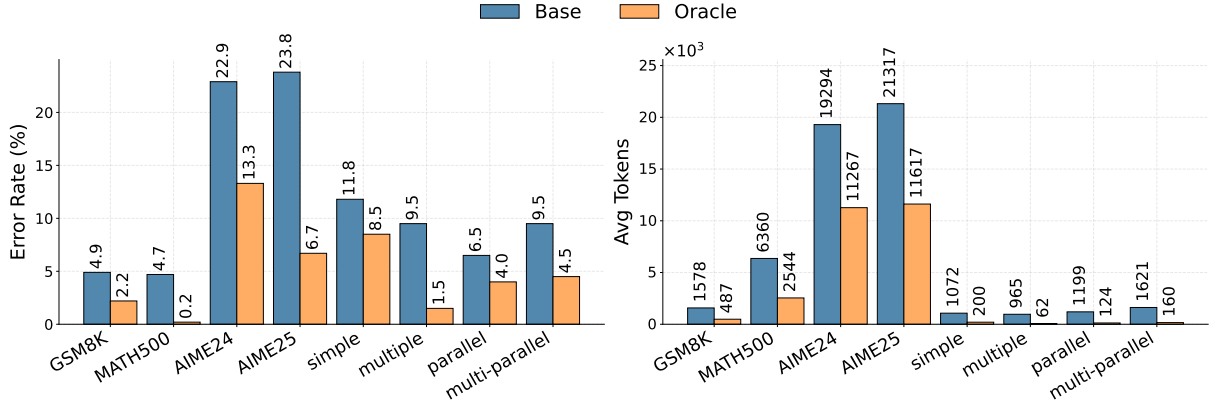


Figure 2: Oracle experiments via sentence-boundary `</think>` injection on mathematical reasoning (GSM8K, MATH500, AIME2024, AIME2025) and tool usage (BFCL-v1). Optimal termination yields both improved error rate (left) and reduced thinking token usage (right). See Table 12 for a tabular version.

### 3 THINKBRAKE

#### 3.1 Intuition and Definition

We propose THINKBRAKE, a simple early termination method that injects the `</think>` when the log-probability margin between the top-predicted token and `</think>` becomes small. Formally, let  $\pi_\theta$  be our LRM,  $x$  be the input, and  $y_{<t}$  be the current reasoning chain. At each sentence boundary, let  $y_t^* = \arg \max_y \pi_\theta(y | x; y_{<t})$  be the top-predicted token that is not `</think>`. We terminate model thinking when:

$$\log \frac{\pi_\theta(y_t^* | x; y_{<t})}{\pi_\theta(\text{</think>} | x; y_{<t})} \leq \tau, \quad (1)$$

where  $\tau$  is a hyperparameter that controls the stopping threshold. The intuition is that as productive reasoning concludes, the probability margin between continuing and stopping narrows. While THINKBRAKE considers only the top token, it naturally accounts for different levels of overall token entropy, as we justify through the following exhaustive cases:

**Case 1: Termination state.** The model assigns the highest probability to `</think>` ( $p_\theta(\text{</think>}) > p_\theta(y_t^*)$ ), so termination occurs regardless of THINKBRAKE.

**Case 2: High-entropy state.** The model spreads probability across many tokens, indicating genuine uncertainty. Early stopping here risks premature termination of productive reasoning, so we should stop as conservatively as possible. Using  $y_t^*$  as the reference is the most conservative choice, as it by definition carries the largest probability.

**Case 3: Low-entropy state.** The model concentrates probability on one or a few tokens, indicating high raw probability of  $y_t^*$ . If `</think>` is comparably probable, it too carries high probability—meaning the model genuinely considers stopping as an option and productive reasoning has ended. Termination in this case is therefore natural.

For comparison, we also consider a linear probability-gap variant (THINKBRAKE-p) that triggers when  $p_\theta(y_t^*) - p_\theta(\text{</think>}) \leq \tau_{\text{prob}}$ . This variant proves significantly less effective (see §4.2). Furthermore, we validate the dynamics of THINKBRAKE with respect to entropy and compare its difference against THINKBRAKE-p in Appendix D.

#### 3.2 Theoretical Analysis

We theoretically ground THINKBRAKE by showing it is equivalent to a logit margin test, which can be interpreted as test-time realignment via a KL-regularized policy with a reward bonus for `</think>`.

**Log-space margins equal logit margins.** We first show that THINKBRAKE is equivalent to a *logit margin test*. For brevity, let  $e := \text{</think>}$  denote the stop token and  $s_t := (x; y_{<t})$  denote the decoding state. Let  $z_\theta(y | s_t)$  be the pre-softmax logit for token  $y$  in the vocabulary  $\mathcal{V}$  at state  $s_t$ . With decoding temperature  $T > 0$ , the next-token distribution is:

$$\pi_\theta(y | s_t) = \frac{\exp(z_\theta(y | s_t)/T)}{\sum_{y'} \exp(z_\theta(y' | s_t)/T)}. \quad (2)$$

Taking the log-ratio between any two tokens cancels the normalization:

$$\log \frac{\pi_\theta(a | s_t)}{\pi_\theta(b | s_t)} = \frac{z_\theta(a | s_t) - z_\theta(b | s_t)}{T}. \quad (3)$$

Let  $y_t^* := \arg \max_{y \in \mathcal{V} \setminus \{e\}} z_\theta(y | s_t)$  be the best continuation token (excluding  $e$ ). The *log-space stopping margin* is

$$m_t := \log \frac{\pi_\theta(y_t^* | s_t)}{\pi_\theta(e | s_t)} = \frac{z_\theta(y_t^* | s_t) - z_\theta(e | s_t)}{T}. \quad (4)$$

Therefore, THINKBRAKE is exactly a temperature-scaled *logit margin test* between the best continuation token and `</think>`.

**THINKBRAKE as test-time realignment.** We now formalize THINKBRAKE as optimizing a KL-regularized policy that prefers emitting `</think>`. Let  $\mathcal{B}$  denote the set of sentence boundary states where THINKBRAKE may terminate. At a boundary state  $s_t \in \mathcal{B}$ , consider the general KL-regularized objective with hyperparameter  $\beta > 0$ :

$$\begin{aligned} \tilde{\pi}_\theta(\cdot | s_t) = \arg \max_{\pi(\cdot | s_t)} & \mathbb{E}_{y \sim \pi(\cdot | s_t)} [r(s_t, y)] \\ & - \beta \text{KL}(\pi(\cdot | s_t) \| \pi_\theta(\cdot | s_t)). \end{aligned} \quad (5)$$

The closed-form optimal policy is given by (Korbak et al., 2022; Rafailov et al., 2023):

$$\tilde{\pi}_\theta(y | s_t) = \frac{1}{Z_t} \pi_\theta(y | s_t) \exp\left(\frac{r(s_t, y)}{\beta}\right). \quad (6)$$

We define the reward to make the KL-regularized objective equivalent to THINKBRAKE’s logit margin test in Eq. (1).

$$r_\tau(s_t, y) := \tau T \cdot \mathbf{1}[y = e]. \quad (7)$$

Plugging Eq. (7) into Eq. (6) gives  $\exp(r_\tau/\beta) = \exp(\tau \mathbf{1}[y = e])$ , so the realignment simply up-weights `</think>` by a factor  $e^\tau$ .

We now show that setting  $\beta = T$  yields a direct connection to logit margins. Since  $\pi_\theta(\cdot | s_t)$  is a softmax over logits, Eq. (6) with  $\beta = T$  is equivalent to *logit shaping*. Define the shaped logit:

$$\tilde{z}_{\theta, \tau}(y | s_t) := z_\theta(y | s_t) + r_\tau(s_t, y), \quad (8)$$

where  $r_\tau(s_t, y) = \tau T \cdot \mathbf{1}[y = e]$  under our choice of  $\beta = T$ . Then the realigned policy becomes:

$$\tilde{\pi}_{\theta, \tau}(y | s_t) = \text{softmax}(\tilde{z}_{\theta, \tau}(y | s_t)/T). \quad (9)$$

Thus, at boundary states, the realigned policy is exactly the base model with a constant bonus  $\tau T$  added to the `</think>` logit.

## Equivalence to the THINKBRAKE margin test.

Under greedy decoding from the realigned policy at  $s_t \in \mathcal{B}$ , it selects  $e$  if and only if

$$\begin{aligned} z_\theta(e | s_t) + \tau T & \geq z_\theta(y_t^* | s_t) \\ \tau & \geq \frac{z_\theta(y_t^* | s_t) - z_\theta(e | s_t)}{T} \\ m_t & \leq \tau, \end{aligned} \quad (10)$$

where  $m_t$  is defined in Eq. (4). This is exactly THINKBRAKE. For non-boundary states  $s_t \notin \mathcal{B}$ , THINKBRAKE applies no realignment and decodes from  $\pi_\theta(\cdot | s_t)$  as usual.

## 4 Experiments

### 4.1 Experimental Setup

We evaluate several models for our main experiments: Qwen3-4B-Thinking, Qwen3-4B, Qwen3-14B, Qwen3-32B (Yang et al., 2025), DeepSeek-R1-7B (Guo et al., 2025), and Phi-4-Reasoning (Abdin et al., 2025). Qwen3-32B is a model post-trained with reinforcement learning (RL), while the smaller Qwen models are trained via supervised fine-tuning. DeepSeek-R1-7B is a Qwen2.5-based model distilled from DeepSeek-R1-generated reasoning traces. Phi-4-Reasoning is derived from Phi-4 through supervised fine-tuning on curated reasoning demonstrations. All models use default reasoning format with explicit `<think>` and `</think>` delimiters.

We compare against NoWait (Wang et al., 2025), which suppresses filler tokens like "wait" and "hmm"; ThinkLess (Li et al., 2025), which forces immediate `</think>` emission after `<think>`; Dynasor-CoT (Fu et al., 2025), which extracts intermediate answers and applies early stopping based on consistency; DEER (Yang et al., 2026), which also interrupts mid-thinking and stops based on entropy; and THINKBRAKE-p (§3.1). Details in Appendix C.5.

We report both task accuracy and thinking token reduction. We measure *thinking tokens* as tokens generated within the reasoning span (between `<think>` and `</think>`). We report  $\Delta\text{Tok}$  as the percentage change in thinking tokens relative to baseline decoding for the same model and benchmark. Detailed baseline methods, implementation settings, and system prompts are provided in Appendix C.

**Selecting Hyperparameter  $\tau$ .** THINKBRAKE introduces a single hyperparameter,  $\tau$ , which controls

Model	GSM8K		MATH500		AIME2024		AIME2025		GPQA-D		ARC-C		Avg.	
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Qwen3-4B-Thinking	95.1	–	95.3	–	77.1	–	76.2	–	64.1	–	94.3	–	83.7	–
+ NoWait	94.2	-25.1%	96.4	-17.9%	66.7	-30.5%	66.7	-35.6%	68.2	-19.6%	92.7	14.6%	80.8	-19.0%
+ ThinkLess	92.6	-100%	56.4	-100%	3.3	-100%	3.3	-100%	50.0	-100%	94.2	-100%	50.0	-100%
+ DEER	95.1	-34.1%	94.4	-14.2%	63.3	-34.3%	60.0	-37.9%	68.2	-1.3%	92.8	-9.0%	79.0	-21.8%
+ Dynasor-CoT	95.2	-34.0%	87.2	-42.1%	63.3	-28.8%	53.3	-28.9%	62.6	-22.6%	94.5	-39.1%	76.0	-23.0%
+ THINKBRAKE-p	93.9	-37.3%	91.2	-57.9%	35.0	-72.6%	30.4	-76.0%	47.0	-67.5%	93.7	-31.7%	65.2	-57.2%
+ THINKBRAKE	94.8	-18.9%	96.6	-20.4%	67.2	-33.5%	62.8	-33.4%	62.6	-32.7%	94.0	-19.9%	79.7	-26.5%
Qwen3-4B	94.4	–	96.0	–	68.8	–	59.6	–	51.0	–	94.0	–	77.3	–
+ NoWait	94.5	-40.4%	94.6	-31.4%	56.7	-16.6%	50.0	-27.1%	57.6	-26.6%	93.4	-14.4%	74.5	-26.1%
+ THINKBRAKE	94.5	-30.1%	95.4	-14.4%	64.6	-12.7%	60.4	-13.5%	55.6	-29.7%	93.9	-11.5%	77.4	-18.7%
Qwen3-14B	96.0	–	96.8	–	71.7	–	69.2	–	60.6	–	95.9	–	81.7	–
+ NoWait	95.8	-29.4%	95.2	-25.1%	70.0	-19.8%	56.7	-16.6%	62.1	-20.5%	96.2	-8.1%	79.3	-19.9%
+ THINKBRAKE	95.0	-15.4%	96.9	-7.7%	77.9	-7.4%	65.8	-6.0%	61.1	-39.9%	95.2	-18.2%	82.0	-15.8%
Qwen3-32B	96.0	–	97.0	–	75.8	–	68.3	–	65.2	–	92.1	–	82.4	–
+ NoWait	95.3	-23.6%	95.8	-20.6%	66.7	-12.0%	56.7	-10.6%	60.1	-12.4%	90.1	9.0%	77.5	-11.7%
+ THINKBRAKE	96.5	-9.1%	97.2	-1.4%	77.1	-6.2%	68.3	-4.3%	65.7	-19.9%	91.0	-8.0%	82.6	-8.2%
DeepSeek-R1-7B	92.7	–	93.8	–	49.5	–	37.6	–	48.0	–	67.7	–	64.9	–
+ NoWait	90.6	-30.8%	91.2	-31.0%	40.0	-32.4%	26.7	-36.5%	43.4	-35.5%	64.6	-12.6%	59.4	-29.8%
+ THINKBRAKE	92.4	-25.0%	92.0	-23.2%	45.2	-25.0%	33.2	-20.3%	45.5	-51.2%	70.7	-35.3%	63.2	-30.0%
Phi-4-Reasoning	91.7	–	71.1	–	67.5	–	63.7	–	62.6	–	80.0	–	72.8	–
+ NoWait	92.1	-2.4%	70.8	-17.7%	63.3	-9.4%	53.3	-8.6%	63.1	-19.1%	77.6	-7.0%	70.0	-10.7%
+ THINKBRAKE	91.7	-25.6%	70.4	-17.2%	73.3	-6.4%	53.3	-5.2%	67.7	-1.1%	80.0	-25.3%	72.7	-13.5%

Table 1: Math and science results on GSM8K, MATH500, AIME2024, AIME2025, GPQA-D, and ARC-C. We report accuracy and  $\Delta$ Tok (thinking token reduction vs. the base decoding) for each model. For AIME2024 and AIME2025, which contain only 30 questions, we report results over 32 runs and also report the variance and confidence intervals. For brevity we only show the baseline method for the Qwen3-4B-Thinking model. A comprehensive result table including baseline methods is in Appendix E.

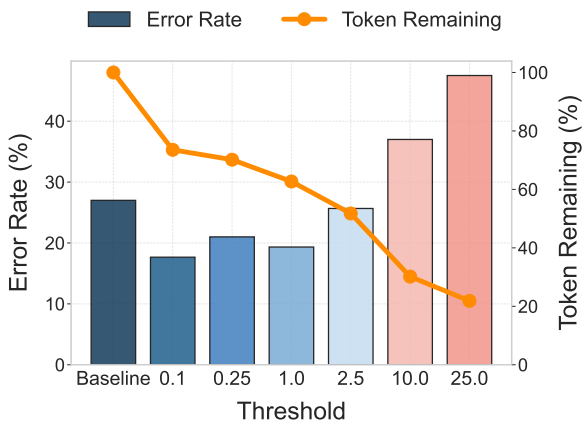


Figure 3:  $\tau$  search from error rate and thinking token usage for THINKBRAKE on Qwen3-4B-Thinking.

how aggressively we terminate reasoning at sentence boundaries. Increasing  $\tau$  encourages earlier emission of  $\langle /think \rangle$ , yielding fewer thinking tokens, but overly large values can truncate useful reasoning and degrade accuracy. To choose a default threshold without per-benchmark tuning, we perform a lightweight sweep over  $\tau$  on a small validation split. Concretely, we evaluate Qwen3-4B-Thinking across a grid of  $\tau$  values on a subset of the

DAPO17K (Yu et al., 2025) dataset and record both error rate and thinking token reduction compared to baseline decoding.

Figure 3 shows that at  $\tau=0.1$ , THINKBRAKE achieves both reduced error rate and substantial token reduction compared to baseline. Beyond this point, error rate increases and token usage decreases smoothly as  $\tau$  increases further. Notably, while error rate is lowest at  $\tau=0.1$ , it remains competitive for  $\tau$  values up to 1.0, demonstrating robustness to the choice of threshold. We therefore select  $\tau=0.1$  and apply it to all Qwen3 model sizes and all benchmarks. Our results show that this threshold generalizes well out-of-distribution from mathematical reasoning to other tasks. Similarly, we select optimal  $\tau$  values for DeepSeek-R1-7B ( $\tau=0.1$ ) and Phi-4-Reasoning ( $\tau=2.5$ ) using the same procedure (see Appendix C.6 for details).

## 4.2 Main Results

**Math & Science Reasoning.** We evaluate on four math benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AIME2024, AIME2025 (Art of Problem Solving,

Model	BFCL-v1				BFCL-v2				Meta-Tool				Avg.	
	Parallel		Multi-Parallel		Parallel		Multi-Parallel		Single		Multiple			
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Qwen3-4B-Thinking	93.5	–	90.5	–	87.5	–	79.2	–	69.7	–	85.5	–	84.3	–
+ NoWait	88.5	-44.4%	88.5	-53.5%	81.3	-50.7%	75.0	-66.5%	68.9	0.1%	84.7	-11.2%	81.2	-37.7%
+ ThinkLess	79.5	-100%	83.0	-100%	37.5	-100%	45.8	-100%	67.4	-100%	86.5	-100%	66.6	-100%
+ THINKBRAKE-p	78.0	-26.9%	42.5	-33.4%	62.5	-28.5%	29.2	-43.8%	70.8	-25.8%	86.7	-12.7%	61.6	-28.5%
+ THINKBRAKE	95.5	-26.8%	90.5	-28.2%	87.5	-32.3%	87.5	-18.9%	72.4	-20.6%	86.3	-9.0%	86.6	-22.6%
Qwen3-4B	90.0	–	83.0	–	81.2	–	70.8	–	74.3	–	90.5	–	81.6	–
+ NoWait	88.0	-13.5%	84.0	1.2%	75.0	-5.1%	75.0	-11.1%	72.5	-3.1%	90.5	-8.4%	80.8	-6.7%
+ THINKBRAKE	89.5	-16.5%	83.5	-12.0%	62.5	-21.1%	75.0	-10.2%	72.7	-1.4%	91.1	-3.0%	79.0	-10.7%
Qwen3-14B	92.0	–	83.5	–	62.5	–	70.8	–	63.3	–	84.9	–	76.2	–
+ NoWait	90.0	-16.3%	84.0	1.8%	62.5	-30.1%	62.5	-10.9%	61.2	-2.4%	84.5	-5.8%	74.1	-10.6%
+ THINKBRAKE	93.0	-21.2%	85.0	-14.2%	62.5	-25.4%	62.5	-7.6%	66.3	-1.6%	85.7	-4.5%	75.8	-12.4%
Qwen3-32B	93.0	–	85.0	–	68.8	–	70.8	–	64.8	–	84.3	–	77.8	–
+ NoWait	91.0	-13.0%	82.5	-5.0%	68.8	-3.9%	62.5	-14.3%	63.8	3.6%	83.9	1.3%	75.4	-5.2%
+ THINKBRAKE	91.0	-14.3%	90.0	-7.3%	68.8	-4.6%	58.0	-13.3%	64.4	-1.5%	85.5	-1.1%	76.3	-7.0%
DeepSeek-R1-7B	52.5	–	38.0	–	43.8	–	29.2	–	63.2	–	80.3	–	51.2	–
+ NoWait	62.0	-4.0%	42.5	-6.9%	50.0	-8.9%	20.8	11.0%	64.9	0.6%	73.2	-19.3%	52.2	-4.6%
+ THINKBRAKE	42.0	-14.6%	37.0	-11.6%	43.8	-19.9%	16.7	-6.0%	63.4	-7.4%	76.3	-26.3%	46.5	-14.3%
Phi-4-Reasoning	87.5	–	77.5	–	81.2	–	58.3	–	77.7	–	90.5	–	78.8	–
+ NoWait	81.0	-1.3%	75.5	-13.2%	81.2	-6.4%	58.3	-11.9%	76.6	-8.5%	90.9	-15.4%	77.2	-9.5%
+ THINKBRAKE	84.5	2.7%	77.5	-6.4%	75.0	4.6%	70.8	-4.9%	77.2	-2.5%	90.3	-5.4%	79.2	-2.0%

Table 2: Results on BFCL-v1, BFCL-v2, and Meta-Tool benchmarks. Simple and Multiple scenarios in BFCL are omitted for brevity. We report accuracy and  $\Delta$ Tok (token reduction) for parallel and multi-parallel scenarios in BFCL, and single and multiple selection scenarios in Meta-Tool. A comprehensive result table including all baseline methods is in Appendix E.

2025). We also consider a scientific reasoning benchmark and a general reasoning benchmark: GPQA-D (Rein et al., 2024), ARC-Challenge (Clark et al., 2018). For AIME benchmarks, we provide 32 run results with confidence intervals for Qwen3-4B-Thinking and DeepSeek-R1-7B (see Appendix E).

Table 1 shows that THINKBRAKE consistently achieves strong accuracy-efficiency trade-offs across all models and difficulties. On easier benchmarks (GSM8K, MATH500), THINKBRAKE reduces tokens by 10–30% while maintaining or improving accuracy. On challenging AIME benchmarks, THINKBRAKE maintains competitive performance with substantial token savings—notably improving accuracy for Qwen3-14B and Phi-4-Reasoning on AIME2024, suggesting THINKBRAKE prevents overthinking.

In contrast, baseline methods show various limitations. ThinkLess (which removes all thinking) causes catastrophic drops on hard math:  $\sim$ 40% drop on MATH500 and  $\sim$ 73% drop on AIME2024 for Qwen3-4B-Thinking, highlighting the need for reasoning in such tasks. NoWait degrades less severely but still drops notably on challenging AIME compared to THINKBRAKE. Dynasor-CoT

shows high performance degradation on benchmarks like AIME; while it often reduces more tokens, this comes at the cost of substantially larger performance degradation. DEER shows competitive performance but still degrades on benchmarks like AIME compared to THINKBRAKE.

An apparent pattern across model scales is that larger models require less aggressive stopping. Average token reduction decreases from 26.5% (Qwen3-4B-Thinking) to 15.8% (14B) to 8.2% (32B). To verify this trend of diminishing gains as model size scales, we further evaluate on a larger model: Qwen3-Next-80B. Surprisingly, we observe a substantial 22.3% token reduction, breaking the apparent trend. We therefore attribute the diminishing reductions from 4B to 32B not to model size, but to reasoning trace length, as larger models tend to produce shorter traces, leaving less redundant thinking to prune. The 80B model, despite being the largest, generates longer traces than the 32B model and correspondingly yields greater reductions. See Appendix E for a further analysis.

**Tool Usage.** We evaluate on BFCL non-live (v1), live (v2) (Patil et al., 2025) and Meta-Tool (Huang et al., 2024). Tool usage results show different pat-

terns, as ThinkLess—which failed catastrophically on math—performs reasonably on some simpler tool usage scenarios, suggesting function calling benefits less from extended reasoning than mathematical domains.

However, on challenging scenarios, baseline heuristics can degrade performance. On BFCL-v2 with Qwen3-4B-Thinking, ThinkLess suffers catastrophic drops on parallel tasks ( $\sim 50\%$  drop) and multi-parallel tasks ( $\sim 33\%$  drop). NoWait shows modest degradation with a  $\sim 7\%$  drop on parallel and  $\sim 5\%$  drop on multi-parallel. In contrast, THINKBRAKE maintains robust performance across all scenarios. On BFCL-v2, THINKBRAKE matches baseline accuracy on parallel tasks with 32.3% token reduction, and notably improves multi-parallel performance by  $\sim 8\%$ . This demonstrates that THINKBRAKE’s adaptive stopping criterion successfully prevents overthinking on complex tool usage scenarios where simpler heuristics fail. These results show that THINKBRAKE generalizes effectively beyond mathematical and scientific reasoning to tool usage tasks.

**Logarithmic Margin Design.** As discussed in §3.1, we compare THINKBRAKE with its linear probability-gap variant THINKBRAKE-p:  $p_\theta(y_t^*) - p_\theta(\langle /think \rangle) \leq \tau_{\text{prob}}$ . Tables 1 and 2 show that THINKBRAKE-p exhibits substantial accuracy drops on challenging tasks like AIME—achieving only 30–35% on AIME compared to THINKBRAKE’s 62–67%—while performing comparably on easier tasks like GSM8K with Qwen3-4B-Thinking.

This is likely because the log-ratio formulation requires the absolute probabilities of both  $y_t^*$  and  $\langle /think \rangle$  to be sufficiently large to trigger termination. When both probabilities are low, a small log-margin is harder to achieve—the two must be far more similar in relative terms than when both are high. As a result, THINKBRAKE terminates model thinking conservatively when the model is uncertain, and primarily when it is confident (see Appendix D). In contrast, raw probability differences lack this adaptive property, causing premature termination during uncertain reasoning states and leading to significant accuracy degradation on challenging tasks. These results validate that the log-space formulation is essential for THINKBRAKE’s robustness on hard reasoning tasks.

**Premature Exiting and Spurious Reasoning.** While THINKBRAKE largely preserves baseline

Benchmark	Oracle	THINKBRAKE
AIME2024	11,261	12,830
AIME2025	11,618	14,197

Table 3: Average tokens for Oracle and THINKBRAKE on Qwen3-4B-Thinking. THINKBRAKE exits later than the oracle for both AIME2024 and AIME2025.

accuracy, we observe slight performance degradation on highly complex benchmarks (e.g., AIME, GPQA-D) for specific models such as Qwen3-4B-Thinking and DeepSeek-R1-7B. To determine whether this stems from premature termination, a vulnerability in confidence-based early-exit methods, we compare THINKBRAKE’s stopping points against an oracle setting. As shown in Table 3, THINKBRAKE consistently generates more tokens than the oracle before exiting, indicating that it does not suffer from premature termination in practice, and is using sufficient reasoning budgets.

Consequently, we attribute these accuracy drops not to insufficient reasoning, but to *spurious reasoning*. Because harder benchmarks naturally elicit longer reasoning traces, they pose a greater risk for the model to engage in flawed or contradictory logic *after* successfully solving the problem, but only before THINKBRAKE triggers termination. This explains why the observed degradation is primarily concentrated in benchmarks characterized by extended reasoning lengths. We provide further results and a qualitative example of spurious reasoning in Appendix E.

### 4.3 Ablation Studies

**Hyperparameter Sensitivity.** We further analyze THINKBRAKE across test sets with various thresholds ( $\tau \in \{0.1, 0.25, 1.0, 2.5, 10\}$ ) for Qwen3-4B-Thinking. Figure 4 shows that accuracy remains stable until very high thresholds ( $\tau \geq 10$ ), while token reduction increases monotonically with  $\tau$ . Although a single threshold works reasonably well across various tasks within model families, optimal thresholds vary by task difficulty. Easier tasks like MATH500 and tool usages maintain accuracy with higher  $\tau$ , while challenging GPQA-D shows sensitivity to increasing  $\tau$ , suggesting we can use more aggressive  $\tau$  for easier tasks to maximize token reduction. As a practical rule of thumb,  $\tau = 0.1$  works well across almost all model-benchmark pairs; for new settings, one can increase  $\tau$  for easier tasks to maximize token efficiency or maintain a low  $\tau$  for harder ones to preserve accuracy. See Appendix E for full results.

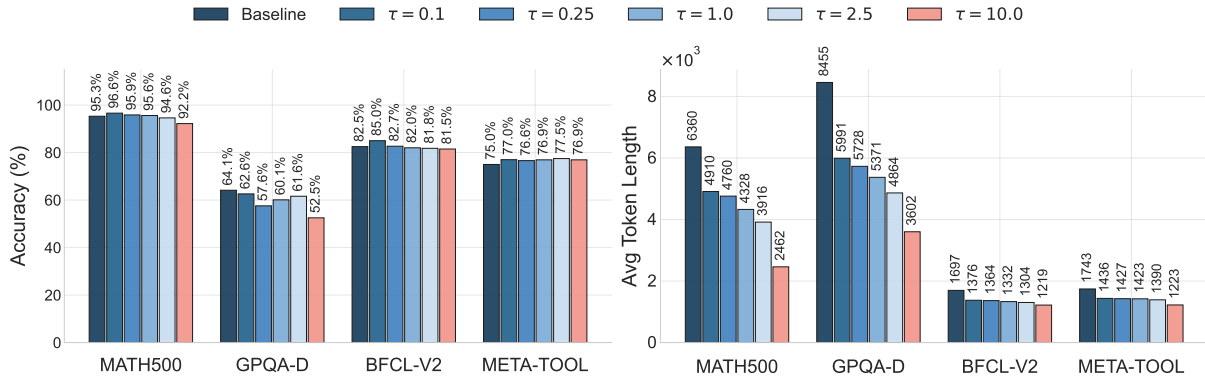


Figure 4: Hyperparameter sensitivity for Qwen3-4B-Thinking over MATH500, GPQA-D and tool usages. Accuracy remains robust except for extreme case  $\tau = 10$  (left). Token reduction increases with  $\tau$  (right).

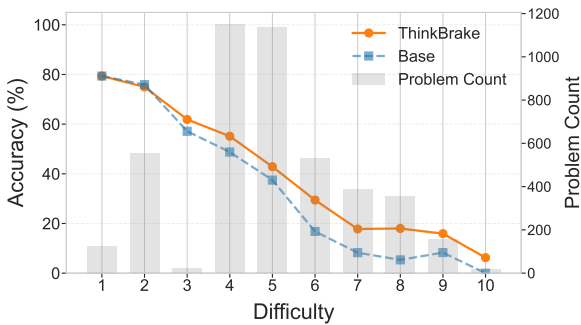


Figure 5: Accuracy by difficulty on Omni-MATH. THINKBRAKE maintains uniform performance across all levels. Grey bars show problem counts.

**Performance Across Problem Difficulties.** We investigate whether THINKBRAKE trades performance between easy and hard problems, using the Omni-MATH dataset (Gao et al., 2024), which provides difficulty labels from 1 to 10. Figure 5 shows THINKBRAKE maintains or improves accuracy uniformly across all difficulty levels without any trade-off. Notably, THINKBRAKE shows greater improvements on harder problems, suggesting that overthinking is more prevalent in challenging questions. We hypothesize that LRMs, uncertain about harder problems, continue generating spurious verification steps even after reaching a correct answer. This validates that efficiency gains come from preventing overthinking rather than sacrificing performance on hard problems. To further quantify how often early stopping changes correctness, we compute prediction transition matrices before and after THINKBRAKE (see Appendix E). Across models, most examples remain unchanged (correct  $\rightarrow$  correct or incorrect  $\rightarrow$  incorrect), indicating THINKBRAKE primarily reduces computation without substantial accuracy trade-offs.

Benchmark	Base	DPO	$\Delta$ Tok
GSM8K	95.1	95.2	-23.6%
MATH500	95.3	96.7	-27.7%
AIME2024	77.1	80.7	-8.8%
AIME2025	76.2	76.0	-9.5%
GPQA-D	64.1	61.5	-16.2%
ARC-C	94.3	93.8	-20.8%
BFCL-v1	88.3	88.0	-16.6%
BFCL-v2	82.5	83.6	-10.5%
Meta-Tool	75.0	76.1	-23.9%

Table 4: Benchmark results for Qwen3-4B-Thinking trained with DPO on THINKBRAKE-generated data.

#### 4.4 Training from THINKBRAKE Data

Since THINKBRAKE generates concise trajectories that maintain or improve accuracy, we investigate whether this data can be used to train LRMs for efficient reasoning. We apply Direct Preference Optimization (DPO; Rafailov et al., 2023) to Qwen3-4B-Thinking, treating THINKBRAKE trajectories as preferred and baseline trajectories as rejected. We construct 1.3K preference pairs from Omni-MATH problems. The training is highly efficient as it requires only 20 minutes on  $2 \times$  H200 GPUs (see Appendix C.3 for details).

Table 4 shows that DPO training successfully transfers THINKBRAKE’s efficiency to the LRM. Despite training only on mathematical reasoning, the model achieves 9–28% thinking token reduction across all benchmarks while maintaining accuracy. Notably, improvements transfer to harder math problems (AIME), and to entirely out-of-domain tasks including GPQA-D, ARC-C, and tool usage. This demonstrates that the model learns general concise reasoning patterns rather than memorizing task-specific solutions, enabling efficient training of LRMs with minimal data and compute.

## 5 Related Work

### 5.1 Concise Reasoning

As LRMs often generate excessive tokens, recent work has studied efficient reasoning methods.

**Test-Time Methods.** Several training-free approaches have been proposed, including sampling answers at each step using consistency as an early-stop signal (Mao et al., 2025; Liu and Wang, 2025; Wan et al., 2025), employing external verifiers (Jiang et al., 2025), or leveraging model representations via probing (Fu et al., 2025) and signals like confidence (Yang et al., 2026) and entropy (Laaouach, 2025). Other heuristics include suppressing reflection tokens (e.g., “Wait”, “Hmm”) (Wang et al., 2025) or skipping explicit reasoning entirely (Li et al., 2025). While effective in some cases, these methods typically rely on hand-crafted rules or require additional computational overhead from multiple rollouts or external models.

**Training Methods.** An alternative approach trains models to produce concise traces directly. Kang et al. (2025) compress long CoT and condition models to generate shorter reasoning, while subsequent work (Fang et al., 2025; Song and Zheng, 2025; Fatemi et al., 2025; Aggarwal and Welleck, 2025) presents RL frameworks for concise reasoning. However, these require significant computational resources and can potentially destabilize general model performance.

### 5.2 Logit-Based Test-Time Realignment

A complementary literature explores test-time realignment without retraining by applying logit-space shifts to the base model. Unlike heuristic test-time methods, these approaches implement principled objectives—such as RLHF alignment—directly at decoding time via logit manipulation. One line of work constructs such shifts using signals from aligned and base models (Mitchell et al., 2024; Liu et al., 2024a), with controllable strength at decoding time (Liu et al., 2024b), while related logit interventions can elicit specific abilities like reasoning (Zhang et al., 2025b). Another line leverages decoding-time steering for user-specific objectives, including modular rewards (Huang et al., 2025) or expert combinations (Liu et al., 2021). Overall, these logit-based methods provide a theoretically grounded and flexible mechanism for test-time behavior control.

THINKBRAKE falls within this framework: we implement a KL-regularized realignment objective that assigns a reward bonus to `</think>`, enabling principled concise reasoning.

## 6 Conclusion

We study overthinking in Large Reasoning Models across mathematical reasoning, scientific QA, and tool usage. Oracle experiments reveal that LRMs often reach correct intermediate solutions but fail to terminate, leaving substantial headroom for improved accuracy and efficiency. We introduce THINKBRAKE, a training-free decoding rule that triggers `</think>` based on log-probability margins at sentence boundaries. THINKBRAKE achieves favorable accuracy-efficiency trade-offs across multiple model families and benchmarks, is theoretically grounded as KL-regularized test-time realignment, and can train more efficient models via preference optimization.

### Limitations

THINKBRAKE comes with several limitations. First, THINKBRAKE requires explicit reasoning delimiters (`<think>` and `</think>`) and access to model logits, which may not be available in models with hidden chain-of-thought or black-box API settings. Second, the method introduces a hyperparameter  $\tau$  that, while robust across a range of values, may require tuning in some settings. Finally, THINKBRAKE uses a local stopping criterion based on probability margins and does not explicitly reason about global answer correctness, which may lead to early termination when the model is confidently wrong.

### Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant (RS-2024-00333484) and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (RS-2025-02215122, Development and Demonstration of Lightweight AI Model for Smart Homes), all funded by the Korean government (MSIT). This research was also supported by the Korea Institute of Science and Technology Information (KISTI) in 2026 (No. (KISTI)K26L3M1C1), aimed at developing KONI (KISTI Open Neural Intelligence), a large language model specialized in science and technology.

## References

- Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, and 1 others. 2025. [Phi-4-reasoning technical report](#). *arXiv preprint arXiv:2504.21318*.
- Pranjal Aggarwal and Sean Welleck. 2025. [L1: Controlling how long a reasoning model thinks with reinforcement learning](#). In *Second Conference on Language Modeling*.
- Art of Problem Solving. 2025. [AIME problems and solutions](#). Accessed: 2026-1-02.
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. [Towards reasoning era: A survey of long chain-of-thought for reasoning large language models](#). *arXiv preprint arXiv:2503.09567*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? try arc, the ai2 reasoning challenge](#). Preprint, arXiv:1803.05457.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. [Thinkless: LLM learns when to think](#). In *The Thirtieth Annual Conference on Neural Information Processing Systems*.
- Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. [Concise reasoning via reinforcement learning](#). *arXiv preprint arXiv:2504.05185*.
- Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. [Efficient reasoning models: A survey](#). *Transactions on Machine Learning Research*.
- Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025. [Reasoning without self-doubt: More efficient chain-of-thought through certainty probing](#). In *ICLR 2025 Workshop on Foundation Models in the Wild*.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. [Omni-math: A universal olympiad level mathematic benchmark for large language models](#). *arXiv preprint arXiv:2410.07985*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- James Y. Huang, Sailik Sengupta, Daniele Bonadiman, Yi-An Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2025. [DeAL: Decoding-time alignment for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26280–26300, Vienna, Austria. Association for Computational Linguistics.
- Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2024. [Meta-tool benchmark for large language models: Deciding whether to use tools and which to use](#). In *The Twelfth International Conference on Learning Representations*.
- Guochao Jiang, Guofeng Quan, Zepeng Ding, Ziqin Luo, Dixuan Wang, and Zheng Hu. 2025. [Flashthink: An early exit method for efficient reasoning](#). *arXiv preprint arXiv:2505.13949*.
- Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. [C3ot: Generating shorter chain-of-thought without compromising effectiveness](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. [On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting](#). *Advances in Neural Information Processing Systems*, 35:16203–16220.
- Yassir Laaouach. 2025. [HALT-cot: Model-agnostic early stopping for chain-of-thought reasoning via answer entropy](#). In *4th Muslims in ML Workshop co-located with ICML 2025*.
- Gengyang Li, Yifeng Gao, Yuming Li, and Yunfang Wu. 2025. [Thinkless: A training-free inference-efficient method for reducing reasoning redundancy](#). *arXiv preprint arXiv:2505.15684*.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. [Tuning language models by proxy](#). In *First Conference on Language Modeling*.

- Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. [DExperts: Decoding-time controlled text generation with experts and anti-experts](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.
- Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares-López, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. 2024b. [Decoding-time realignment of language models](#). In *Forty-first International Conference on Machine Learning*.
- Xin Liu and Lu Wang. 2025. [Answer convergence as a signal for early stopping in reasoning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17907–17918, Suzhou, China. Association for Computational Linguistics.
- Minjia Mao, Bowen Yin, Yu Zhu, and Xiao Fang. 2025. [Early stopping chain-of-thoughts in large language models](#). *arXiv preprint arXiv:2509.14004*.
- Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. [An emulator for fine-tuning large language models using small language models](#). In *The Twelfth International Conference on Learning Representations*.
- OpenAI. 2024. [Learning to reason with llms](#). Associated Press (AP).
- Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. [The berkeley function calling leaderboard \(BFCL\): From tool use to agentic evaluation of large language models](#). In *Forty-second International Conference on Machine Learning*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). *Advances in neural information processing systems*, 36:53728–53741.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Mingyang Song and Mao Zheng. 2025. [Walk before you run! concise llm reasoning via reinforcement learning](#). *arXiv preprint arXiv:2505.21178*.
- Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2025. [Reasoning aware self-consistency: Leveraging reasoning paths for efficient LLM sampling](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3613–3635, Albuquerque, New Mexico. Association for Computational Linguistics.
- Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025. [Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7459–7482, Suzhou, China. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.
- Fengli Xu, Qianye Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. [Towards large reasoning models: A survey of reinforced reasoning with large language models](#). *arXiv preprint arXiv:2501.09686*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2026. [Dynamic early exit in reasoning models](#). In *The Fourteenth International Conference on Learning Representations*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. [DAPO: An open-source LLM reinforcement learning system at scale](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Xuechen Zhang, Zijian Huang, Chenshun Ni, Ziyang Xiong, Jiasi Chen, and Samet Oymak. 2025a. [Making small language models efficient reasoners: Intervention, supervision, reinforcement](#). In *ICML 2025 Workshop on Long-Context Foundation Models*.
- Yunxiang Zhang, Muhammad Khalifa, Lechen Zhang, Xin Liu, Ayoung Lee, Xinliang Frederick Zhang, Farima Fatahi Bayat, and Lu Wang. 2025b. [Logit arithmetic elicits long reasoning capabilities without training](#). *arXiv preprint arXiv:2510.09354*.

## A Usage of Large Language Models

Large language models were used for literature search, coding assistance, and proofreading only. They were not used for ideation, results, or analysis; all contributions and conclusions are the authors’.

## B Example of Overthinking

Tables 10 and 11 present a BFCL example and the corresponding LRM response. Green highlights mark points at which terminating would yield a correct answer, whereas red highlights indicate an incorrect one. Notably, the model’s reasoning turns red after a certain point, illustrating overthinking.

## C Experiment Details

We use NVIDIA H200 GPU for all inference and training, with a fixed seed of 42.

### C.1 Model Details

The official Huggingface names for the models we used are as follows in Table 5.

Our Name	Hf Model Id
Qwen3-4B-Thinking	Qwen/Qwen3-4B-Thinking-2507
Qwen3-4B	Qwen/Qwen3-4B
Qwen3-14B	Qwen/Qwen3-14B
Qwen3-32B	Qwen/Qwen3-32B
DeepSeek-R1-7B	deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
Phi-4-Reasoning	microsoft/Phi-4-reasoning
Qwen3-Next-80B	Qwen/Qwen3-Next-80B-A3B-Thinking

Table 5: Model names and Huggingface identifiers.

### C.2 Inference Details

For all inference tasks, we use the following hyperparameters following the official guidelines for each model (Yang et al., 2025; Abdin et al., 2025; Guo et al., 2025). For Qwen3-4B-Thinking, we extend the reasoning budget to 32k for AIME benchmarks only.

### C.3 Training on THINKBRAKE

This section describes the experimental details for §4. To curate a training dataset, we sampled  $\sim 1.3\text{K}$  examples from Omni-MATH, with the preferred responses generated using THINKBRAKE and dispreferred responses generated without THINKBRAKE. We applied the following filtering criteria: (1) when both methods produce correct answers, we only include examples where the baseline token length

Parameter	DeepSeek	Qwen3	Phi-4
THINKBRAKE- $\tau$	0.1	0.1	2.5
<i>Sampling Parameters</i>			
Temperature	0.6	0.6	0.8
Top-p	1	0.95	0.95
Top-k	-	20	50
<i>Token Budgets</i>			
Reasoning Budget	16384	16384	16384
Answer Budget	4096	4096	4096

Table 6: Inference hyperparameters for different models.

exceeds the THINKBRAKE token length, demonstrating efficiency gains; (2) when THINKBRAKE produces correct answers but the baseline fails, we include all such examples; (3) all other cases are excluded from the training set. This curation process ensures that our training data emphasizes both correctness improvements and computational efficiency.

We performed Direct Preference Optimization (DPO; Rafailov et al., 2023) using the hyperparameters listed in Table 7. Training was conducted with LoRA (Hu et al., 2022) for parameter-efficient fine-tuning in around 20 minutes of wall-clock time.

Hyperparameter	Value
Learning Rate	$2 \times 10^{-5}$
Batch Size (per device)	8
Gradient Accumulation Steps	4
Effective Batch Size	64
Number of GPUs	$2 \times \text{H200}$
Training Epochs	10
LoRA Rank ( $r$ )	32
LoRA Alpha ( $\alpha$ )	32
Training Samples	1,348

Table 7: Training hyperparameters for DPO.

### C.4 System Prompts

Following the official guides (Abdin et al., 2025; Guo et al., 2025; Yang et al., 2025) we use the following prompts in Figures 9, 10, 11 and 12 for our experiments.

### C.5 Baseline Methods

This section describes the details on the baseline methods:

**NoWait** (Wang et al., 2025): removes delay/filler tokens (e.g., “wait”, “hmm”) during decoding to stop overthinking and allow for earlier termination. We use the following list in Table 8 for NoWait.

Keywords
wait, hmm, hmmm, but, however, check, verify, alternatively

Table 8: Keyword list for NoWait.

**ThinkLess** (Li et al., 2025): forces immediate termination by emitting `</think>` right after the initial `<think>`, minimizing the thinking process to two special tokens: `<think></think>`.

**Dynasor-CoT** (Fu et al., 2025): injects probing prompts at regular intervals to extract intermediate answers, terminating reasoning early if those answers demonstrate consistency across consecutive steps.

**DEER** (Yang et al., 2026): monitors the model’s reasoning for linguistic transition markers (e.g., ‘Wait’, ‘Alternatively’), temporarily interrupts the process at these points to induce a trial answer, and evaluates the model’s internal probability confidence in that answer—exiting to output it if the confidence threshold is met, or discarding it and resuming the chain of thought if it is not.

### C.6 Hyperparameter Selection

Figures 6 and 7 show the hyperparameter search for additional models.

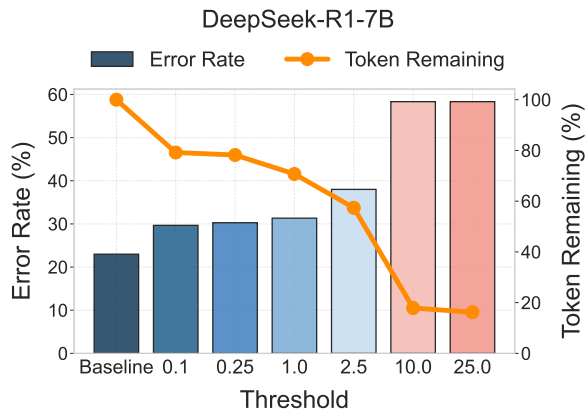


Figure 6:  $\tau$  search from error rate and thinking token usage for THINKBRAKE on DeepSeek-R1-7B.

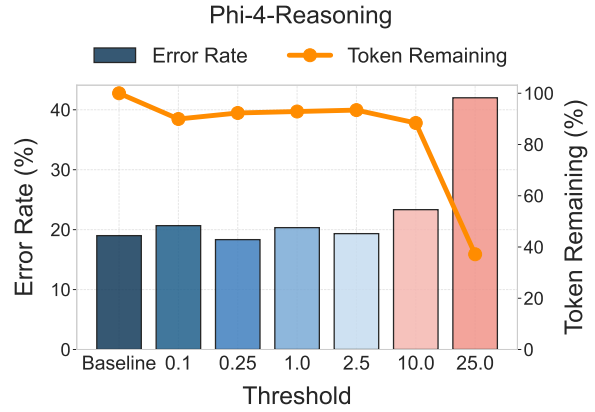


Figure 7:  $\tau$  search from error rate and thinking token usage for THINKBRAKE on Phi-4-Reasoning.

### D Why Log-Space Margins Are Robust

As shown in Tables 14, 15 and §4.2, linear probability gaps are insufficient for THINKBRAKE. A meaningful log-probability gap only arises when both competing probabilities are relatively high (see Figure 8), indicating that the model is genuinely confident at that step. If we used a linear probability gap, both  $\Delta P1$  and  $\Delta P2$  would trigger early termination; in contrast, using a log-probability gap triggers only for  $\Delta P1$ —where both the top token and `</think>` have high probabilities—signaling confidence in stopping the reasoning.

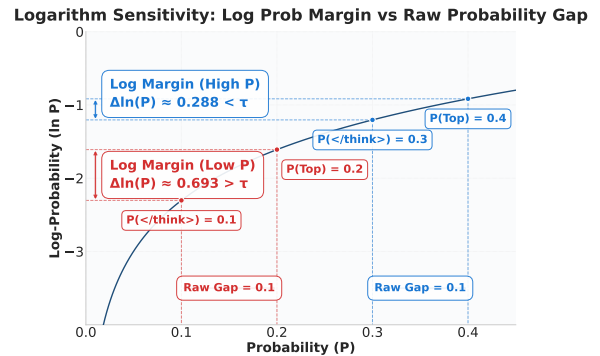


Figure 8: Log-probability plot with two illustrative cases,  $\Delta P1$  and  $\Delta P2$ . Here,  $\Delta P1$  denotes a scenario where the most likely token has high probability (0.4), and  $\Delta P2$  denotes a scenario where the most likely token has lower probability (0.2), while the raw gap is the same (i.e.,  $\Delta P1 = \Delta P2$ ).

**Entropy State.** Table 9 shows the ratio of `</think>` in the top 20 tokens and median entropy for the log-probability setting (THINKBRAKE) and the raw-probability variant (THINKBRAKE-p). THINKBRAKE terminates almost exclusively in

low-entropy states and only when `</think>` ranks within the top 20 tokens, while the raw-probability variant (THINKBRAKE-p) frequently triggers in high-entropy states even when `</think>` is not in the top 20, confirming that the log design is robust precisely where the raw-probability design is more sensitive.

Category	Method	Ratio in Top-20	Median Entropy
Math	THINKBRAKE	99.9%	0.018
	THINKBRAKE-p	66.6%	0.264
Science	THINKBRAKE	100.0%	0.036
	THINKBRAKE-p	65.3%	0.458

Table 9: Entropy and top-20 ratio results on Qwen3-4B-Thinking across math and science benchmarks.

## E Extensive Results

In this section, we provide extensive results across all benchmarks, models, and hyperparameter combinations.

**Token Count.** Table 13 shows the average token count per base model and benchmark. We consistently observe that difficult tasks such as AIME and GPQA-D require larger token budgets.

**Full Tables.** Tables 14 and 15 present extensive results from the experiments in §4. Specifically, Table 14 shows the full results including all baseline methods across all models. Table 15 shows the complete results for BFCL-v1, BFCL-v2 (including Simple and Multiple function categories), and Meta-Tool benchmarks. We observe that THINKBRAKE achieves a better performance-token reduction tradeoff overall. Furthermore, Table 16 reports the variances and 95% confidence intervals (CIs) for accuracy and token counts on the AIME benchmarks, computed over 32 independent runs. Notably, THINKBRAKE’s token consumption falls entirely below the baseline’s CI, confirming a statistically meaningful reduction in reasoning length.

**Token Reduction and Model Size.** Table 17 shows that larger models tend to produce shorter reasoning traces and there is less redundant thinking to prune. Qwen3-Next-80B, despite being larger, produces substantially longer traces than Qwen3-32B, and correspondingly yields larger token reductions. We suggest that token reduction rate depends on the reasoning trace length rather than model size.

### Premature Exiting and Spurious Reasoning.

To investigate the performance degradation associated with THINKBRAKE, we analyzed the per-problem token counts of THINKBRAKE versus an oracle setting on the highly challenging AIME benchmarks. Table 18 reveals that among the 16 instances where THINKBRAKE failed but the oracle succeeded, THINKBRAKE exited prior to the oracle in only 2 cases. This indicates that premature termination is not the primary reason for incorrect answers. Instead, errors appear to stem from subsequent spurious reasoning that misleads the LRMs. A detailed example is shown in Table 19, where the model successfully solves the problem but then engages in flawed reasoning before THINKBRAKE finally terminates generation. We hypothesize that highly verbose models, such as Qwen3-4B-Thinking (which produces the longest average reasoning traces), are especially vulnerable to this type of reasoning degradation.

**$\tau$  Ablation.** Figures 13, 14, 15, and 16 show extensive results across various threshold values  $\tau \in \{0.1, 0.25, 1.0, 2.5\}$ . The first two figures display token reduction percentage compared to the base model, while the latter two show accuracy comparison across various  $\tau$  values. We observe that larger  $\tau$  values result in greater token reduction, and accuracy is maintained across various  $\tau$  settings.

**Transition Matrix.** Figure 17 shows the transition matrices for all models (aggregated across all benchmarks), illustrating model behavior before and after applying THINKBRAKE. The matrices show high values along the diagonal (top-left and bottom-right), indicating that THINKBRAKE preserves accuracy without introducing significant changes to correct or incorrect predictions.

---

**Question**

---

Find the area and perimeter of a circle with a radius of 5 and also find the circumference of a circle with diameter of 10.

---

**Prompt**

---

```
system:
# Tools
You may call one or more functions to assist with the user query.
You are provided with function signatures within <tools> XML tags:
<tools>
{ "name": "circle.calculate_circumference",
  "description": "Calculate the circumference of a circle based on the diameter.",
  "parameters": {
    "type": "dict",
    "properties": {
      "diameter": { "type": "integer", "description": "The diameter of the circle." }
    },
    "required": ["diameter"]
  }
},
{ "name": "circle.calculate_area",
  "description": "Calculate the area of a circle based on the radius.",
  "parameters": {
    "type": "dict",
    "properties": {
      "radius": { "type": "integer", "description": "The radius of the circle." }
    },
    "required": ["radius"]
  }
},
{ "name": "rectangle.calculate_perimeter",
  "description": "Calculate the perimeter of a rectangle based on the length and breadth.",
  "parameters": {
    "type": "dict",
    "properties": {
      "length": { "type": "integer", "description": "The length of the rectangle." },
      "breadth": { "type": "integer", "description": "The breadth of the rectangle." }
    },
    "required": ["length", "breadth"]
  }
}
</tools>
For each function call, return a json object with function name and arguments within <tool_call>
XML tags:
<tool_call>
{ "name": <function-name>, "arguments": <args-json-object> }
</tool_call>
user: The above geometry question.
```

---

Table 10: Full question and prompt.

---

<think>

Okay, let's tackle this problem. The user wants the area and perimeter of a circle with radius 5, and the circumference of a circle with diameter 10.

First, I need to check which tools are available.

There's circle.calculate\_area which takes radius, and rectangle.calculate\_perimeter but wait, the user is asking for the perimeter of a circle? Wait, no.

Wait, the problem says "area and perimeter of a circle".

Wait, circles don't have perimeters; they have circumferences.

Maybe the user made a mistake here.

Let me read again.

The user says: "Find the area and perimeter of a circle with a radius of 5 and also find the circumference of a circle with diameter of 10."

Hmm, perimeter for a circle is called circumference.

So maybe the user is confused.

But according to the tools provided, there's circle.calculate\_circumference (for diameter), circle.calculate\_area (for radius), and rectangle.calculate\_perimeter (for length and breadth).

Wait, the user is asking for perimeter of a circle.

But the tools don't have a circle perimeter function.

Wait, the rectangle function is for perimeter of a rectangle, which is different.

So maybe the user intended to say circumference for the circle's perimeter.

⋮

Because perimeter is a term for 2D shapes, but for circles it's circumference.

Assuming that the user wants:

- Area of radius 5 circle: call calculate\_area with radius=5.

- Circumference of radius 5 circle (which they called perimeter): but since the tool requires diameter, we can compute diameter as 10 (since radius is 5), so call calculate\_circumference with diameter=10.

- Also, circumference with diameter 10: call calculate\_circumference with diameter=10.

⋮

So three tool calls? Or is the perimeter part a mistake and the user just meant circumference?

Wait, the rectangle.calculate\_perimeter is for rectangles, so that's not relevant here.

The user is talking about circles, so we ignore the rectangle tool.

So, the correct tool calls are:

1. circle.calculate\_area(radius=5)

2. circle.calculate\_circumference(diameter=10)

3. circle.calculate\_circumference(diameter=10)

But in reality, the user is asking for two separate values: perimeter (which is circumference) for radius 5 (diameter 10) and circumference for diameter 10.

So they are the same calculation, but the user is asking for it twice.

⋮

- For the perimeter of radius 5 circle: since perimeter of a circle is circumference, which is calculated via diameter.

So diameter is .

---

Table 11: Qwen3-4B thinking trajectory.

```

Qwen3 Mathematical Reasoning Prompt

<|im_start|>user
{{ problem }}
Please reason step by step, and put your
final answer within \boxed{ }.
<|im_end|>
<|im_start|>assistant
<think>
{{ reasoning process }}
</think>
{{ final answer }}
<|im_end|>

```

Figure 9: Prompt template for Qwen3 on mathematical reasoning tasks. Similar to DeepSeek, the model provides reasoning within think tags before the final boxed answer.

```

Qwen3 Multiple-Choice Reasoning Prompt

<|im_start|>user
{{ problem }}
Please show your choice in the answer
field with only the choice letter, e.g., "an-
swer": "C".
<|im_end|>
<|im_start|>assistant
<think>
{{ reasoning process }}
</think>
{{ final answer with choice letter }}
<|im_end|>

```

Figure 10: Prompt template for Qwen3 on general multiple-choice reasoning tasks. The model is instructed to format its final answer as a single choice letter.

```

DeepSeek Reasoning Prompt

<|begin_of_sentence|><|User|>{{
problem }}
Please reason step by step, and put your
final answer within \boxed{ }.
<|Assistant|><think>
{{ reasoning process }}
</think>
{{ final answer }}
<|end_of_sentence|>

```

Figure 11: Prompt template for DeepSeek on mathematical reasoning tasks. The model is instructed to provide step-by-step reasoning within think tags, followed by the final answer in boxed notation.

```

Phi-4-Reasoning System Prompt

<|im_start|>system<|im_sep|>
You are Phi, a language model trained by Mi-
crosoft to help users. Your role as an assistant
involves thoroughly exploring questions through a
systematic thinking process before providing the
final precise and accurate solutions.
This requires engaging in a comprehensive cycle
of analysis, summarizing, exploration, reassess-
ment, reflection, backtracing, and iteration to
develop well-considered thinking process.
Please structure your response into two main sec-
tions: Thought and Solution using the specified
format:
<think> {Thought section} </think>
{Solution section}
In the Thought section: Detail your reasoning
process in steps. Each step should include de-
tailed considerations such as analysing questions,
summarizing relevant findings, brainstorming new
ideas, verifying the accuracy of the current steps,
refining any errors, and revisiting previous steps.
In the Solution section: Based on various at-
tempts, explorations, and reflections from the
Thought section, systematically present the final
solution that you deem correct. The Solution
section should be logical, accurate, and concise
and detail necessary steps needed to reach the
conclusion.
Now, try to solve the following question through
the above guidelines:
<|im_end|> <|im_start|>user<|im_sep|>
{{ problem }} <|im_end|>
<|im_start|>assistant<|im_sep|><think>
{{ reasoning process }} </think> {{ final
answer }} <|im_end|>

```

Figure 12: Prompt template for Phi-4-Reasoning with comprehensive system instructions. The model is guided to use a systematic thinking process with explicit thought and solution sections.

(a) Math	GSM8K		MATH500		AIME2024		AIME2025		Avg.	
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Base	95.1	–	95.3	–	77.1	–	76.2	–	86.0	–
Oracle	97.8	-69.1%	99.8	-60.0%	86.7	-41.6%	93.3	-45.5%	94.4	-54.1%
<i>Avg. Base Token</i>	<i>1,578</i>		<i>6,360</i>		<i>19,294</i>		<i>21,317</i>		<i>12,137</i>	

(b) Tool	simple		multiple		parallel		multi-parallel		Avg.	
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Base	88.2	–	90.5	–	93.5	–	90.5	–	90.7	–
Oracle	91.5	-81.3%	98.5	-93.6%	96.0	-89.7%	95.5	-90.1%	95.4	-88.7%
<i>Avg. Base Token</i>	<i>1,155</i>		<i>985</i>		<i>1,200</i>		<i>1,621</i>		<i>1,240</i>	

Table 12: Accuracy (%) and  $\Delta$ Tok (token reduction, %) on Math and Tool benchmarks. The results represent performance under oracle  $\langle$ /think $\rangle$  stopping compared to the base Qwen3-4B-Thinking model.

Model	GSM8K	MATH500	AIME 2024	AIME 2025	GPQA-D	ARC-C
Qwen3-4B-Thinking	1,578	6,360	19,294	21,317	8,455	1,178
Qwen3-4B	2,407	5,250	13,000	15,163	7,497	798
Qwen3-14B	1,912	4,938	12,807	14,198	7,379	621
Qwen3-32B	1,794	4,650	12,278	13,727	6,626	726
DeepSeek-R1-7B	1,674	4,062	12,146	12,476	7,097	770
Phi-4-Reasoning	1,332	2,664	9,616	10,599	7,896	1,360
Qwen3-Next-80B	1,169	4,547	15,538	16,976	8,025	807

Model	BFCL-v1				BFCL-v2				Meta-Tool	
	Simple	Multiple	Parallel	Multi-Par.	Simple	Multiple	Parallel	Multi-Par.	Single	Multiple
Qwen3-4B-Thinking	1,072	965	1,199	1,621	1,094	1,832	1,473	2,397	1,946	1,336
Qwen3-4B	607	744	844	1,046	653	1,426	837	1,555	1,069	856
Qwen3-14B	545	725	834	975	612	1,372	944	1,470	1,091	873
Qwen3-32B	553	750	826	991	617	1,342	725	1,572	1,048	857
DeepSeek-R1-7B	669	959	815	1,152	743	1,702	816	1,623	1,144	1,187
Phi-4-Reasoning	1,259	1,480	1,574	2,050	1,641	2,582	1,692	2,607	2,104	1,966

Table 13: Average token counts for base models across all datasets and models.

Model	GSM8K		MATH500		AIME2024		AIME2025		GPQA-D		ARC-C		Avg.	
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Qwen3-4B-Thinking	95.1	-	95.3	-	77.1	-	76.2	-	64.1	-	94.3	-	83.7	-
+ NoWait	94.2	-25.1%	96.4	-17.9%	66.7	-30.5%	66.7	-35.6%	68.2	-19.6%	92.7	14.6%	80.8	-19.0%
+ ThinkLess	92.6	-100%	56.4	-100%	3.3	-100%	3.3	-100%	50.0	-100%	94.2	-100%	50.0	-100%
+ DEER	95.1	-34.1%	94.4	-14.2%	63.3	-34.3%	60.0	-37.9%	68.2	-1.3%	92.8	-9.0%	79.0	-21.8%
+ Dynasor-CoT	95.2	-34.0%	87.2	-42.1%	63.3	-28.8%	53.3	-28.9%	62.6	-22.6%	94.5	-39.1%	76.0	-23.0%
+ THINKBRAKE-p	93.9	-37.3%	91.2	-57.9%	35.0	-72.6%	30.4	-76.0%	47.0	-67.5%	93.7	-31.7%	65.2	-57.2%
+ THINKBRAKE	94.8	-18.9%	96.6	-20.4%	67.2	-33.5%	62.8	-33.4%	62.6	-32.7%	94.0	-19.9%	79.7	-26.5%
Qwen3-4B	94.4	-	96.0	-	68.8	-	59.6	-	51.0	-	94.0	-	77.3	-
+ NoWait	94.5	-40.4%	94.6	-31.4%	56.7	-16.6%	50.0	-27.1%	57.6	-26.6%	93.4	-14.4%	74.5	-26.1%
+ ThinkLess	91.0	-100%	85.4	-100%	23.3	-100%	23.3	-100%	46.0	-100%	89.3	-100%	59.7	-100%
+ THINKBRAKE-p	92.8	-40.2%	88.6	-51.3%	29.2	-64.5%	25.0	-67.8%	47.5	-54.2%	93.4	-15.0%	62.8	-48.8%
+ THINKBRAKE	94.5	-30.1%	95.4	-14.4%	64.6	-12.7%	60.4	-13.5%	55.6	-29.7%	93.9	-11.5%	77.4	-18.7%
Qwen3-14B	96.0	-	96.8	-	71.7	-	69.2	-	60.6	-	95.9	-	81.7	-
+ NoWait	95.8	-29.4%	95.2	-25.1%	70.0	-19.8%	56.7	-16.6%	62.1	-20.5%	96.2	-8.1%	79.3	-19.9%
+ ThinkLess	93.4	-100%	86.8	-100%	23.3	-100%	30.0	-100%	49.5	-100%	94.3	-100%	62.9	-100%
+ THINKBRAKE-p	95.1	-31.6%	89.0	-49.8%	38.8	-68%	26.2	-72%	57.6	-66.4%	94.7	-17.9%	66.9	-51%
+ THINKBRAKE	95.0	-15.4%	96.9	-7.7%	77.9	-7.4%	65.8	-6.0%	61.1	-39.9%	95.2	-18.2%	82.0	-15.8%
Qwen3-32B	96.0	-	97.0	-	75.8	-	68.3	-	65.2	-	92.1	-	82.4	-
+ NoWait	95.3	-23.6%	95.8	-20.6%	66.7	-12.0%	56.7	-10.6%	60.1	-12.4%	90.1	9.0%	77.5	-11.7%
+ ThinkLess	93.3	-100%	87.6	-100%	23.3	-100%	20.0	-100%	54.5	-100%	95.1	-100%	62.3	-100%
+ THINKBRAKE-p	94.8	-23.0%	89.6	-39.4%	39.2	-61.5%	26.7	-66.4%	53.5	-51.0%	92.2	-11.7%	66.0	-42.2%
+ THINKBRAKE	96.5	-9.1%	97.2	-1.4%	77.1	-6.2%	68.3	-4.3%	65.7	-19.9%	91.0	-8.0%	82.6	-8.2%
DeepSeek-R1-7B	92.7	-	93.8	-	49.5	-	37.6	-	48.0	-	67.7	-	64.9	-
+ NoWait	90.6	-30.8%	91.2	-31.0%	40.0	-32.4%	26.7	-36.5%	43.4	-35.5%	64.6	-12.6%	59.4	-29.8%
+ ThinkLess	85.9	-100%	78.4	-100%	16.7	-100%	20.0	-100%	30.3	-100%	50.4	-100%	47.0	-100%
+ DEER	89.0	-60.5%	87.2	-44.5%	50.0	-28.2%	33.3	-9.8%	43.9	-16.7%	73.9	-10.6%	62.9	-28.3%
+ Dynasor-CoT	61.3	-87.8%	64.2	-47.2%	33.4	-31.8%	33.4	-26.4%	32.3	-71.2%	77.8	-53.6%	50.4	-53.0%
+ THINKBRAKE-p	87.8	-54.9%	81.4	-60.6%	25.4	-71.4%	22.9	-73.5%	36.4	-76.8%	67.2	-31.0%	53.5	-61.4%
+ THINKBRAKE	92.4	-25.0%	92.0	-23.2%	45.2	-25.0%	33.2	-20.3%	45.5	-51.5%	70.7	-35.3%	63.2	-30.0%
Phi-4-Reasoning	91.7	-	71.1	-	67.5	-	63.7	-	62.6	-	80.0	-	72.8	-
+ NoWait	92.1	-2.4%	70.8	-17.7%	63.3	-9.4%	53.3	-8.6%	63.1	-19.1%	77.6	-7.0%	70.0	-10.7%
+ ThinkLess	92.6	-100%	66.4	-100%	36.7	-100%	23.3	-100%	54.0	-100%	82.7	-100%	59.3	-100%
+ THINKBRAKE-p	91.3	-42.5%	68.8	-50.4%	39.6	-63.1%	29.2	-66.6%	59.1	-62.8%	84.2	-49.2%	62.0	-55.8%
+ THINKBRAKE	91.7	-25.6%	70.4	-17.2%	73.3	-6.4%	53.3	-5.2%	67.7	-1.1%	80.0	-25.3%	72.7	-13.5%
Qwen3-Next-80B	96.7	-	98.2	-	90.0	-	80.0	-	76.3	-	96.8	-	89.7	-
+ THINKBRAKE	96.5	-24.3%	99.2	-21.6%	86.7	-24.3%	73.3	-16.7%	72.7	-25.6%	97.0	-21.8%	87.6	-22.3%

Table 14: Math and science results on GSM8K, MATH500, AIME24, AIME25, GPQA-D, and ARC-C. We report accuracy and  $\Delta$ Tok (thinking token reduction vs. the Base decoding) for each model. This table presents the full comparison including all baseline methods for all models.

Model	BFCL-v1								BFCL-v2								Meta-Tool					
	Simple		Multi		Parallel		Multi-Par.		Simple		Multi		Parallel		Multi-Par.		Single		Multiple		Avg.	
	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok	Acc	$\Delta$ Tok
Qwen3-4B-Thinking	88.2	-	90.5	-	93.5	-	90.5	-	79.8	-	83.4	-	87.5	-	79.2	-	69.7	-	85.5	-	84.8	-
+ NoWait	83.6	-33.1%	88.0	-11.5%	88.5	-44.4%	88.5	-53.5%	75.6	-29.9%	80.3	-16.5%	81.3	-50.7%	75.0	-66.5%	68.9	0.1%	84.7	-11.2%	81.4	-31.7%
+ ThinkLess	86.2	-8.1%	93.0	-2.2%	79.5	-100.0%	83.0	-100.0%	81.0	-5.9%	82.9	-4.1%	37.5	-100.0%	45.8	-100.0%	67.4	-100.0%	86.5	-100.0%	74.3	-62.0%
+ THINKBRAKE-p	88.0	-40.0%	90.5	-16.7%	78.0	-26.9%	42.5	-33.4%	79.5	-34.8%	81.7	-20.4%	62.5	-28.5%	29.2	-43.8%	70.8	-25.8%	86.7	-12.7%	70.9	-28.3%
+ THINKBRAKE	87.3	-32.9%	90.0	-12.4%	95.5	-26.8%	90.5	-28.2%	82.2	-29.9%	82.9	-16.7%	87.5	-32.3%	87.5	-18.9%	72.4	-20.6%	86.3	-9.0%	86.2	-22.8%
Qwen3-4B	88.2	-	92.0	-	90.0	-	83.0	-	80.2	-	81.0	-	81.2	-	70.8	-	74.3	-	90.5	-	83.1	-
+ NoWait	88.2	-16.3%	90.5	-1.5%	88.0	-13.5%	84.0	1.2%	78.3	-9.6%	80.0	-5.8%	75.0	-5.1%	75.0	-11.1%	72.5	-3.1%	90.5	-8.4%	82.2	-7.3%
+ ThinkLess	86.4	-100.0%	89.5	-100.0%	87.5	-100.0%	83.5	-100.0%	71.3	-100.0%	76.4	-100.0%	56.2	-100.0%	62.5	-100.0%	69.1	-100.0%	92.2	-100.0%	77.5	-100.0%
+ THINKBRAKE-p	87.3	-13.5%	93.0	-4.6%	90.5	-14.8%	84.0	-4.8%	80.2	-6.9%	81.8	-3.5%	75.0	-15.1%	75.0	-12.7%	73.5	-1.5%	90.3	-2.9%	83.1	-8.0%
+ THINKBRAKE	87.8	-12.5%	92.5	-5.2%	89.5	-16.5%	83.5	-12.0%	79.1	-9.0%	81.2	-2.9%	62.5	-21.1%	75.0	-10.2%	72.7	-1.4%	91.1	-3.0%	81.5	-9.4%
Qwen3-14B	87.8	-	93.0	-	92.0	-	83.5	-	80.6	-	81.1	-	62.5	-	70.8	-	63.3	-	84.9	-	80.0	-
+ NoWait	87.1	-7.5%	92.0	-0.6%	90.0	-16.3%	84.0	1.8%	77.9	-7.0%	79.8	-4.9%	62.5	-30.1%	62.5	-10.9%	61.2	-2.4%	84.5	-5.8%	78.2	-8.4%
+ ThinkLess	88.4	-100.0%	87.5	-100.0%	90.0	-100.0%	87.0	-100.0%	74.4	-100.0%	77.9	-100.0%	56.2	-100.0%	50.0	-100.0%	69.7	-100.0%	83.5	-100.0%	76.5	-100.0%
+ THINKBRAKE-p	88.4	-9.4%	92.5	-2.2%	91.5	-23.5%	86.0	-16.0%	77.9	-10.1%	80.7	-5.3%	68.8	-29.9%	50.0	-14.6%	64.6	-2.1%	85.9	-4.8%	78.6	-11.8%
+ THINKBRAKE	87.8	-9.5%	92.0	-4.1%	93.0	-21.2%	85.0	-14.2%	78.3	-7.7%	79.7	-5.2%	62.5	-25.4%	62.5	-7.6%	66.3	-1.6%	85.7	-4.5%	79.3	-10.1%
Qwen3-32B	88.0	-	90.0	-	93.0	-	85.0	-	83.7	-	80.7	-	68.8	-	70.8	-	64.8	-	84.3	-	80.9	-
+ NoWait	87.3	-7.8%	88.0	-4.0%	91.0	-13.0%	82.5	-5.0%	80.6	-6.2%	79.0	-1.6%	68.8	-3.9%	62.5	-14.3%	63.8	3.6%	83.9	1.3%	78.7	-5.1%
+ ThinkLess	89.1	-100.0%	87.5	-100.0%	91.5	-100.0%	88.0	-100.0%	77.1	-100.0%	81.3	-100.0%	68.8	-100.0%	58.3	-100.0%	72.3	-100.0%	82.7	-100.0%	79.7	-100.0%
+ THINKBRAKE-p	89.1	-0.5%	91.5	-3.2%	92.0	-12.3%	84.5	-7.1%	80.2	-6.2%	80.8	-0.7%	75.0	-9.1%	66.7	-15.6%	64.4	-1.6%	85.1	-3.3%	80.9	-6.0%
+ THINKBRAKE	88.9	-5.4%	91.0	-3.7%	91.0	-14.3%	90.0	-7.3%	79.8	-7.8%	81.5	-2.2%	68.8	-4.6%	58.0	-13.3%	64.4	-1.5%	85.5	-1.1%	79.9	-6.1%
DeepSeek-R1-7B	57.8	-	61.0	-	52.5	-	38.0	-	54.7	-	41.7	-	43.8	-	29.2	-	63.2	-	80.3	-	52.2	-
+ NoWait	63.3	-2.2%	65.5	0.7%	62.0	-4.0%	42.5	-6.9%	57.8	-2.0%	39.6	-3.2%	50.0	-8.9%	20.8	11.0%	64.9	0.6%	73.2	-19.3%	54.0	-3.4%
+ ThinkLess	9.6	-100.0%	23.5	-100.0%	5.0	-100.0%	5.0	-100.0%	24.0	-100.0%	20.1	-100.0%	0.0	-100.0%	4.2	-100.0%	57.4	-100.0%	88.3	-100.0%	23.7	-100.0%
+ THINKBRAKE-p	48.4	-9.7%	50.0	-9.1%	39.0	-12.1%	28.5	-11.6%	46.5	-8.3%	35.8	-7.7%	37.5	-5.8%	20.8	-5.2%	65.0	-9.3%	79.5	-26.3%	45.1	-10.5%
+ THINKBRAKE	45.3	-11.4%	50.5	-9.6%	42.0	-14.6%	37.0	-11.6%	46.1	-10.8%	35.8	-8.8%	43.8	-19.9%	16.7	-6.0%	63.4	-7.4%	76.3	-26.3%	45.7	-12.6%
Phi-4-Reasoning	82.7	-	88.0	-	87.5	-	77.5	-	70.9	-	72.6	-	81.2	-	58.3	-	77.7	-	90.5	-	78.7	-
+ NoWait	79.1	-6.8%	83.0	-11.1%	81.0	-1.3%	75.5	-13.2%	70.5	-16.7%	67.5	-12.5%	81.2	-6.4%	58.3	-11.9%	76.6	-8.5%	90.9	-15.4%	76.4	-10.4%
+ ThinkLess	65.8	-100.0%	72.5	-100.0%	73.5	-100.0%	68.5	-100.0%	56.2	-100.0%	66.3	-100.0%	87.5	-100.0%	50.0	-100.0%	76.3	-100.0%	90.5	-100.0%	70.7	-100.0%
+ THINKBRAKE-p	80.4	-30.0%	85.5	-22.0%	88.5	-37.8%	80.0	-39.1%	62.8	-36.2%	62.8	-26.4%	87.5	-42.8%	50.0	-28.7%	75.2	-36.1%	91.3	-43.8%	76.4	-34.3%
+ THINKBRAKE	81.6	0.9%	87.5	-4.2%	84.5	2.7%	77.5	-6.4%	69.4	-8.1%	72.2	-1.9%	75.0	4.6%	70.8	-4.9%	77.2	-2.5%	90.3	-5.4%	78.6	-2.5%

Table 15: Results on BFCL-v1, BFCL-v2, and Meta-Tool benchmarks. We report accuracy and  $\Delta$ Tok for all subcategories (Simple, Multiple, Parallel, Multi-Parallel for BFCL; Single, Multiple for Meta-Tool) across all methods.

Model	Metric	Statistics	AIME2024		AIME2025	
			Baseline	THINKBRAKE	Baseline	THINKBRAKE
Qwen3-4B-Thinking	Accuracy	Variance	18.2	20.2	23.7	43.1
		95% CI	[76.2, 79.2]	[65.6, 68.7]	[74.5, 77.8]	[60.5, 65.1]
	Tokens	Variance	78,749,681	33,023,520	74,600,230	37,478,663
		95% CI	[18,870, 19,994]	[12,605, 13,333]	[20,725, 21,819]	[13,727, 14,502]
DeepSeek-R1-7B	Accuracy	Variance	44.5	33.5	22.4	30.5
		95% CI	[47.2, 51.8]	[43.4, 47.4]	[36.0, 39.2]	[31.3, 35.1]
	Tokens	Variance	27,202,081	17,728,953	28,471,392	19,546,656
		95% CI	[10,420, 11,081]	[7,885, 8,418]	[10,872, 11,548]	[8,173, 8,733]

Table 16: Variance and 95% confidence intervals (CI) of accuracy and token count.

Model	Qwen3-4B-Thinking	Qwen3-4B	Qwen3-14B	Qwen3-32B	Qwen3-Next-80B
Avg. Base Token	9,695	7,352	6,976	6,634	7,844
$\Delta$ Tok	-22.6%	-10.7%	-12.4%	-7.0%	-22.3%

Table 17: Token reduction rate of THINKBRAKE. We report  $\Delta$ Tok (thinking token reduction vs. the Base decoding) for Qwen3 Family.

Benchmark	Problem ID	Oracle	THINKBRAKE	$\Delta$ Tok
AIME2024	1	5,734	21,555	+15,821
	13	4,962	11,244	+6,282
	18	3,948	21,967	+18,019
	20	15,969	21,394	+5,425
	25	6,778	22,777	+15,999
	27	4,396	14,965	+10,569
	29	17,396	10,685	-6,711
AIME2025	9	20,393	10,190	-10,203
	10	7,660	13,672	+6,012
	12	6,369	21,539	+15,170
	13	6,528	21,690	+15,162
	19	2,730	12,770	+10,040
	25	8,097	21,605	+13,508
	27	11,175	22,880	+11,705
	28	6,490	22,424	+15,934
29	4,391	21,641	+17,250	

Table 18: Token comparison (Oracle vs. THINKBRAKE) on AIME2024 and AIME2025 for Qwen3-4B-Thinking.

---

```

<think>
...
But wait, in the problem, they say "the circumcircle of triangle DEF intersects BD, AE, AF at G, H, J". Wait, BD: B to D, D is midpoint of BC, so BD is a median? Wait, no, median from B should go to midpoint of AC, but here D is midpoint of BC, so BD is not a median, it's a segment from vertex B to midpoint of side BC? Wait, that's just half of side BC! Wait, hold on, that can't be. Wait, maybe I misread the problem. Wait, let's check again: "D, E, and F be the midpoints of sides BC, AC, and AB, respectively." So yes: D ∈ BC, midpoint; E ∈ AC, midpoint; F ∈ AB, midpoint.
// Oracle Stopping point

Then "the circumcircle of triangle DEF intersects BD, AE, and AF at points G, H, J, respectively." Wait, BD: B is a vertex, D is midpoint of BC, so BD is the segment from B to D, which is half of BC?
// Spurious Reasoning begins

...
Yes! This makes perfect sense now. The nine-point circle passes through the three feet of the altitudes and the three midpoints of the sides, so each side of the triangle intersects the nine-point circle at exactly two points: the midpoint of the side and the foot of the altitude from the opposite vertex.
// False Confidence, ThinkBrake stopping leads to wrong answer

...
I can't believe I didn't remember that the nine-point circle passes through the feet of the altitudes earlier—
// Correction after continued Reasoning

```

---

Table 19: Spurious reasoning example from AIME2025-19. The model initially reaches a correct interpretation (oracle stopping point), but continues spurious reasoning, leading THINKBRAKE to terminate at a false-confidence state.

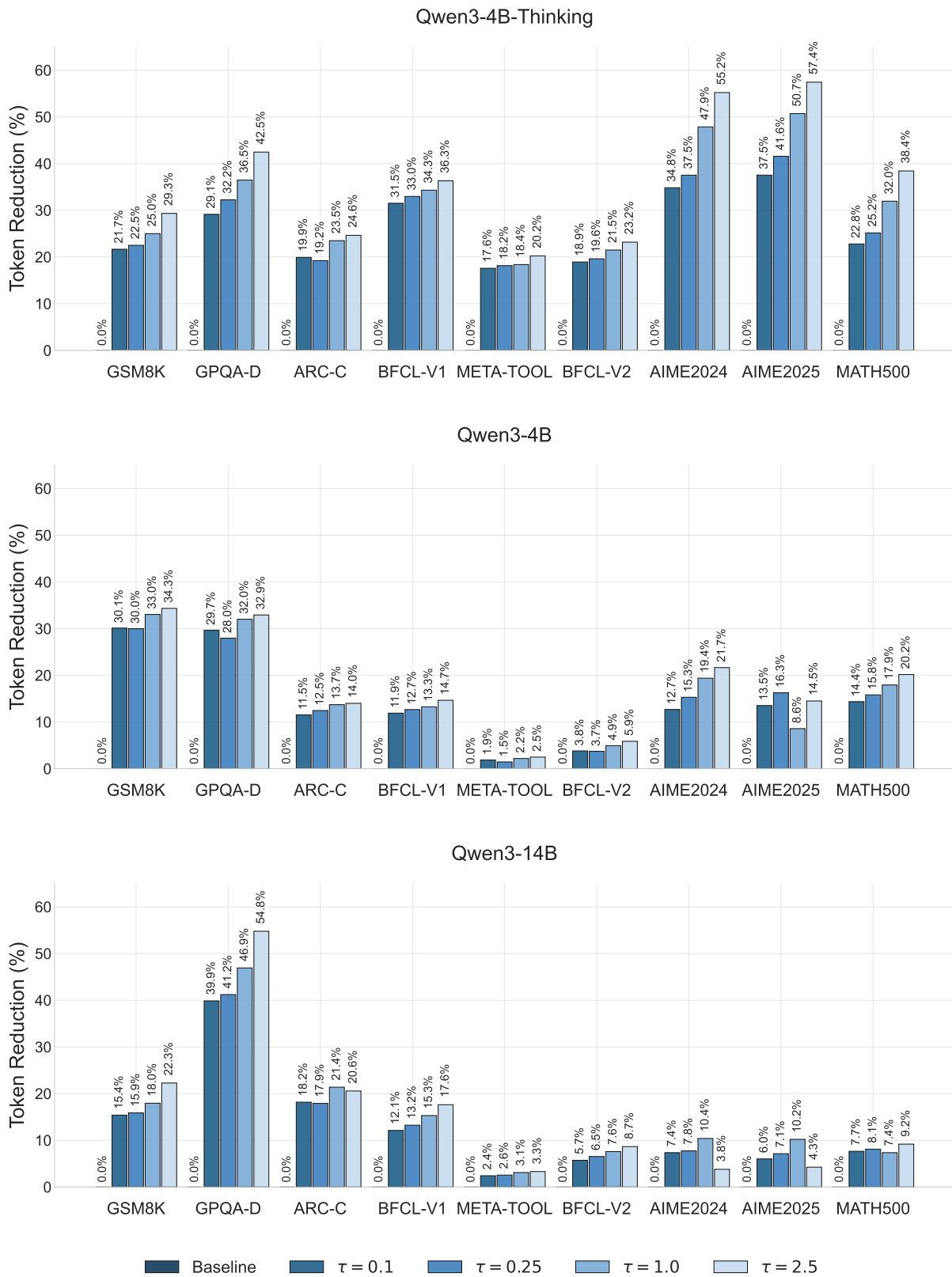


Figure 13: Token reduction across various threshold values  $\tau$  for Qwen3-4B, Qwen3-4B-Thinking, and Qwen3-14B.

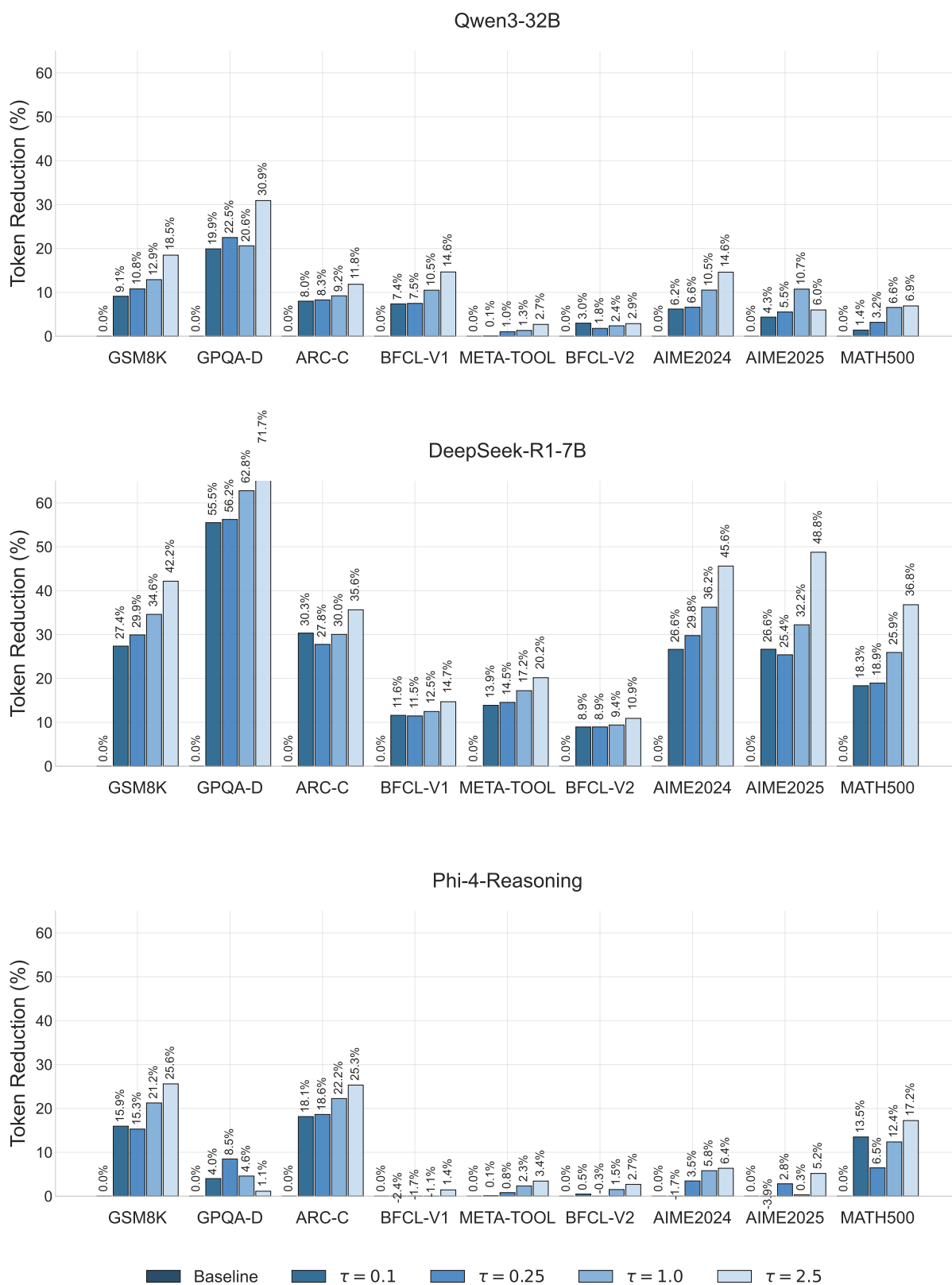


Figure 14: Token reduction across various threshold values  $\tau$  for Qwen3-32B, DeepSeek-R1-7B, and Phi-4-Reasoning.

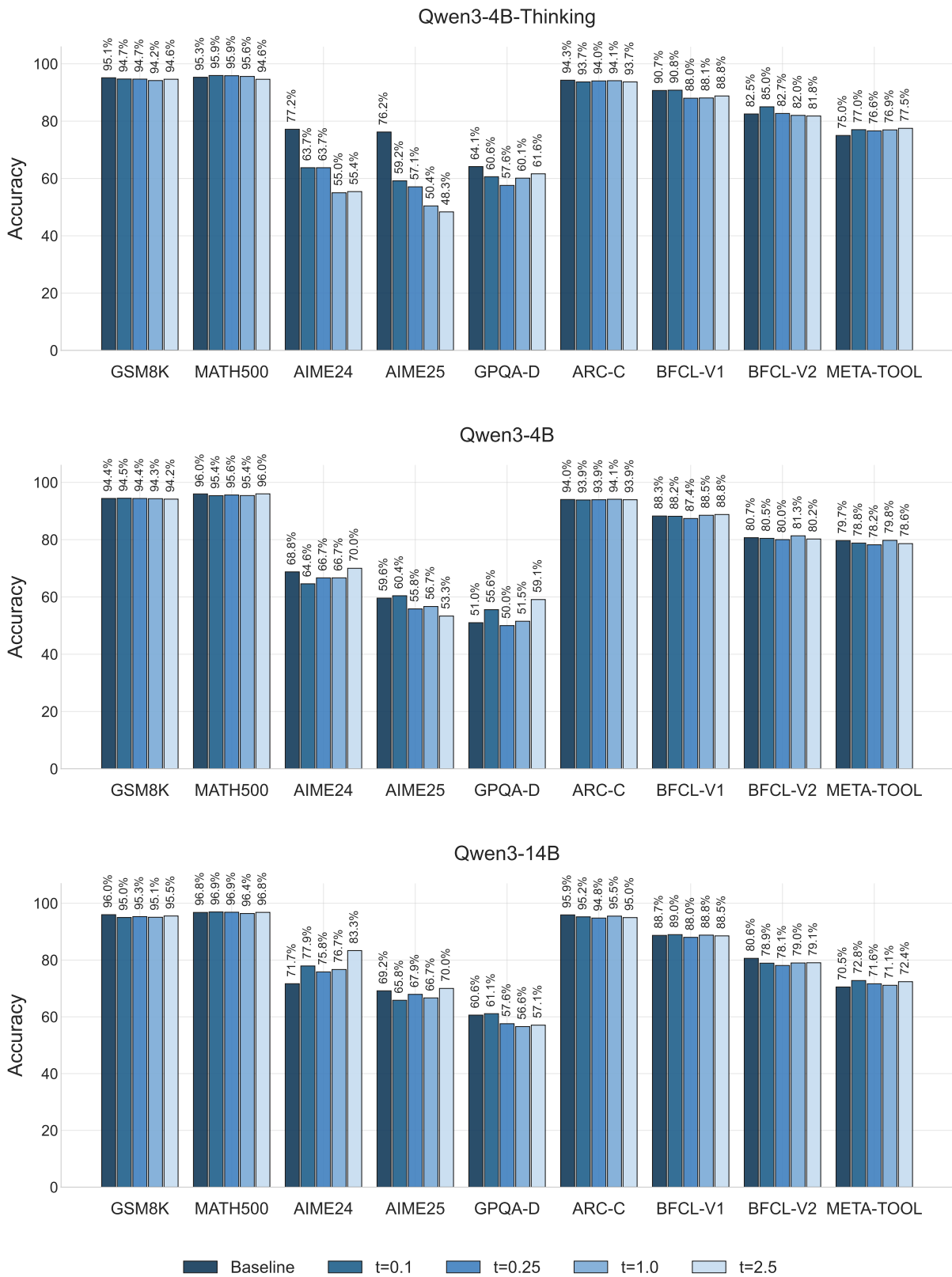


Figure 15: Accuracy comparison across various threshold values  $\tau$  for Qwen3-4B-Thinking, Qwen3-4B, and Qwen3-14B.

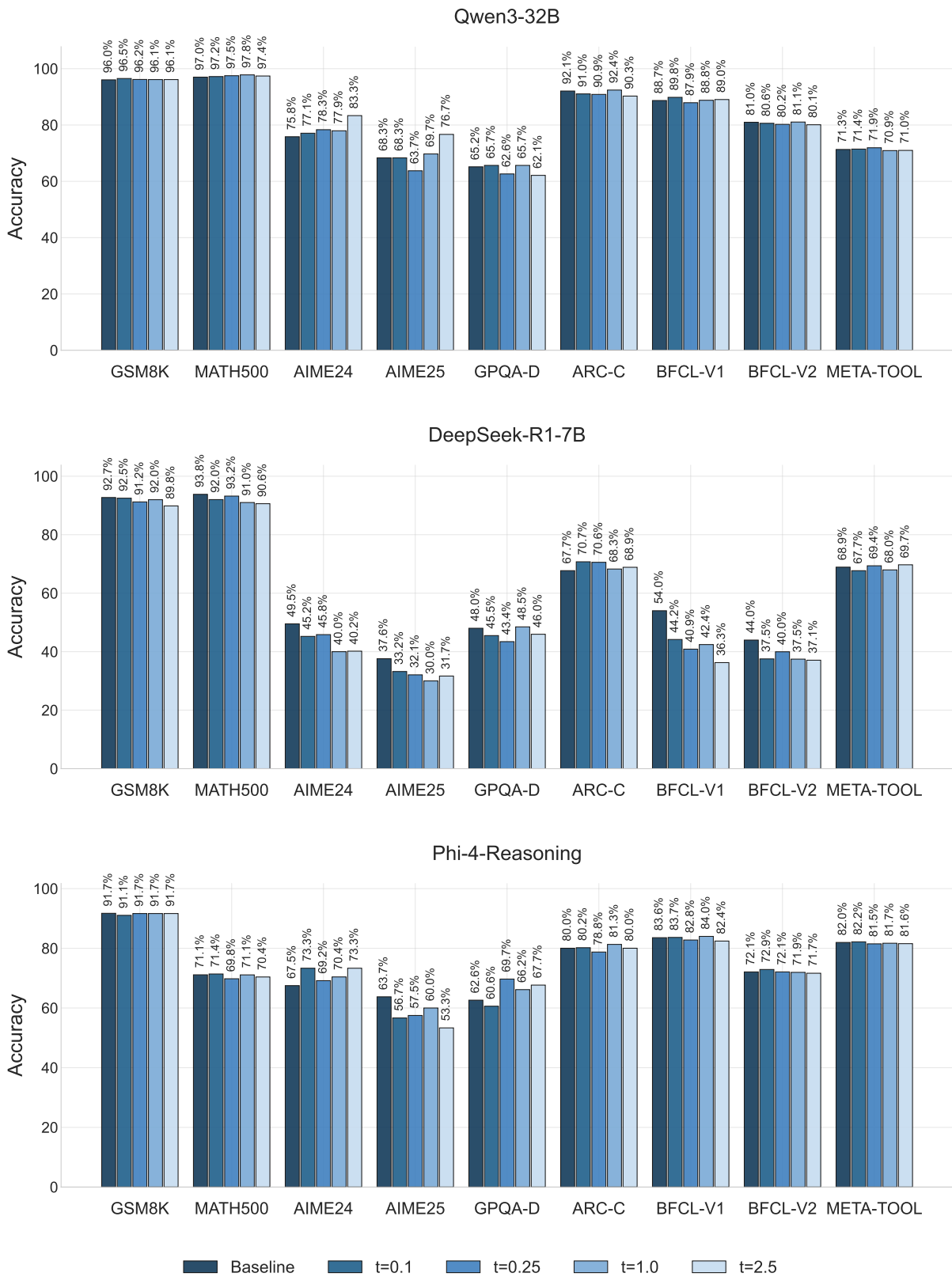


Figure 16: Accuracy comparison across various threshold values  $\tau$  for Qwen3-32B, DeepSeek-R1-7B, and Phi-4-Reasoning.

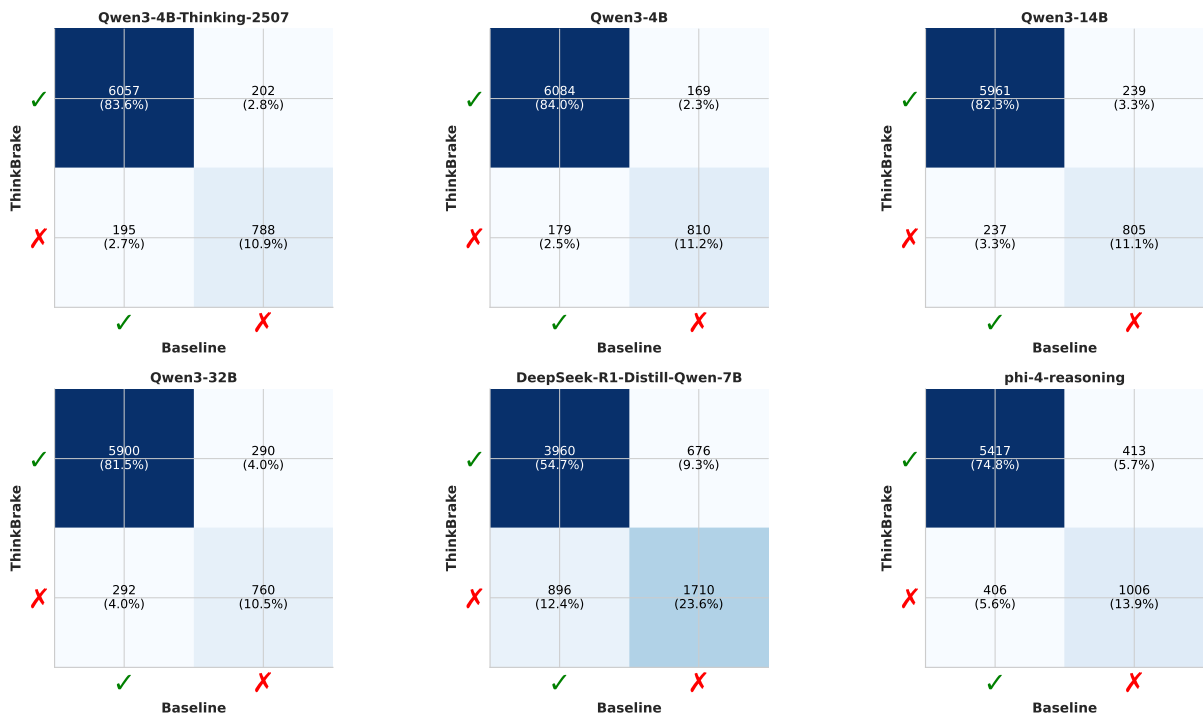


Figure 17: Transition matrices showing prediction transitions before and after applying THINKBRAKE for all models, aggregated across all benchmarks. High diagonal values indicate preserved accuracy.