SABER: Stability-Aware Early Exit for LLM Reasoning via Adversarial Branch Probing

Wanli Cheng*, **Haiya Xiang***, **Juntao Li**†, **Hongling Wang, Wenliang Chen**
Soochow University, China
{wlchengsmk666,hyxiang}@stu.suda.edu.cn, ljt@suda.edu.cn

Abstract

Large Reasoning Models (LRMs) achieve strong reasoning capabilities, yet long-chain reasoning becomes inefficient once the intermediate answer stabilizes across reasoning steps: additional reasoning yields little marginal benefit while incurring substantial inference cost. Existing early-exit methods based on confidence or entropy poorly capture reasoning stability, while consistency-based approaches rely on multi-step trajectory agreement, requiring sequential evaluations that delay exit. To better balance efficiency and reliability, we propose SABER, a training-free framework for stability-aware early exit via adversarial branch probing. SABER constructs simple yet effective semantic perturbations around intermediate reasoning states to form adversarial branches, and applies lightweight probing to estimate their likely final outcomes without full trajectory rollouts. When the probed outcomes remain consistent across branches, SABER exits early; otherwise, it continues reasoning. Experiments across multiple reasoning benchmarks and model architectures show that SABER reduces reasoning token consumption by 30.2%–39.8% on average while maintaining competitive accuracy with full-length reasoning.[1]

1 Introduction

In recent years, Large Reasoning Models (LRMs), such as OpenAI o1 (Jaech et al., 2024), DeepSeek R1 (Guo et al., 2025), and Qwen3 (Yang et al., 2025a), have made remarkable progress on complex tasks, including mathematical reasoning and code generation, by leveraging test-time scaling and long Chain-of-Thought (CoT) reasoning (Snell et al., 2024). However, prior work shows that LRMs often exhibit an overthinking phenomenon, generating redundant verification steps or exploring unnecessary reasoning paths even after sufficient information has been obtained (Chen et al., 2025b; Sui et al., 2025). Such behavior reduces inference efficiency and may introduce superfluous reasoning steps or error accumulation during decoding. These challenges motivate early exit as a promising approach to efficient LRM inference.

To mitigate inference inefficiency caused by long reasoning trajectories, recent work has explored various early-exit strategies for adaptive reasoning. Existing methods such as DEER (Yang et al., 2025b) and EAT (Wang et al., 2026) determine stopping points using confidence or entropy signals from intermediate reasoning states. However, LRMs often suffer from severe confidence miscalibration, remaining highly confident even when their reasoning is incorrect (Mei et al., 2026; Slobodkin et al., 2023; Simhi et al., 2025; Kadavath et al., 2022). Beyond confidence-based approaches, another line of work exploits consistency as a stopping signal. Representative methods include Dynasor (Fu et al., 2025), which leverages agreement among intermediate answers for adaptive stopping. While more reliable, such methods typically require consistency checks across multiple subsequent reasoning steps, delaying early exit and introducing extra inference overhead. Therefore, constructing a stopping signal that can reliably reflect convergence without requiring expensive long-horizon consistency accumulation remains an open challenge.

Answer convergence (Liu and Wang, 2025) provides an effective signal for early exit in long-chain reasoning. Building on this insight, we hypothesize that reasoning trajectories approaching convergence exhibit stability under local semantic perturbations, causing perturbed reasoning branches to yield consistent predictions. We therefore propose **Stability-Aware Early Exit for LLM Reasoning via Adversarial Branch Probing (SABER)**, a

*Equal contribution.
†Corresponding author.
[1]Code is available at https://github.com/Bl1nding/SABER.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 13294–13313
October 24-29, 2026 ©2026 Association for Computational Linguistics

lightweight and training-free framework that enables adaptive early exit by estimating reasoning stability across perturbed branches.

Inspired by self-consistency (Wang et al., 2022), SABER constructs parallel neutral and adversarially perturbed branches from the same intermediate reasoning state, and performs lightweight probing using short answer-focused continuations rather than full trajectory rollouts. Instead of relying solely on single-pass confidence signals or long-horizon consistency accumulation, SABER directly probes local reasoning stability under perturbation. Specifically, SABER estimates reasoning stability from two complementary signals: **Semantic Consistency (SC)**, which quantifies prediction consistency between neutral and adversarial branches, and **Confidence Stability (CS)**, which captures the stability of model confidence across branches. These signals are combined into a unified **Reasoning Stability Score (RSS)** for adaptive early-exit decisions. By jointly modeling semantic and confidence stability, SABER achieves a better trade-off between inference efficiency and reasoning quality.

Our contributions are summarized as follows:

- We introduce a stability-aware perspective for adaptive early exit, and empirically show that reasoning trajectories leading to correct solutions exhibit substantially stronger stability under semantic perturbations.
- We propose SABER, a lightweight and training-free early-exit framework that probes reasoning stability through adversarial branching and jointly models semantic consistency and confidence stability for adaptive stopping.
- Extensive experiments across diverse reasoning benchmarks and model architectures demonstrate that SABER consistently reduces reasoning token consumption by **30.2%–39.8%** while maintaining competitive overall accuracy.

2 Reasoning Stability under Adversarial Perturbations

Recent studies suggest that behavioral stability under perturbations provides informative signals for model reliability and uncertainty estimation (Khanmohammadi et al., 2025; Saadat and Nemzer, 2026). Building on these findings, we analyze how intermediate reasoning states evolve under adversarial perturbations throughout multi-step reasoning.

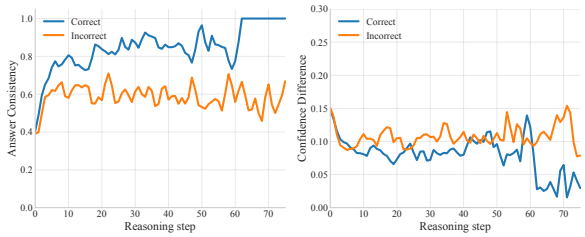

Figure 1: Dynamics of reasoning behavior under adversarial perturbations across reasoning steps. **Left:** Answer consistency. Trajectories leading to correct solutions progressively become more stable and consistent under perturbation, whereas trajectories leading to incorrect solutions remain unstable. **Right:** Confidence variation. Trajectories leading to correct solutions exhibit progressively reduced confidence variation under perturbation, while incorrect trajectories remain highly volatile.

We formalize a reasoning trajectory generated by an LLM as a sequence of tokens $t_1, t_2, \ldots, t_n$. The trajectory is segmented into intermediate reasoning steps $S_1, S_2, \ldots, S_K$ using transition trigger phrases such as "Wait", "Alternatively", and "\n\n". Additional trigger phrases are provided in Appendix B.3.

At each reasoning step S_i, we construct two parallel reasoning branches by appending the following probing prompts to the current reasoning prefix:

- **Neutral Probing Prompt:** *"Wait, let me summarize. The answer is* `\boxed{`*"*
- **Adversarial Probing Prompt:** *"Wait, I think my previous reasoning was incorrect. After correcting it, the answer is* `\boxed{`*"*

The adversarial probe injects misleading corrective signals into the reasoning process, allowing us to examine how intermediate reasoning states respond to perturbations.

For each reasoning step S_i, we perform one stochastic sample under both the neutral and adversarial branches to obtain the corresponding predicted answer and confidence score. We conduct adversarial probing throughout reasoning trajectories sampled from OlympiadBench (He et al., 2024) to analyze the evolution of reasoning behaviors under perturbations.

As shown in Figure 1, correct and incorrect reasoning trajectories exhibit markedly different dynamics throughout reasoning. As reasoning progresses, correct trajectories exhibit progressively increasing stability: predictions generated under the two probes increasingly converge, while confi-

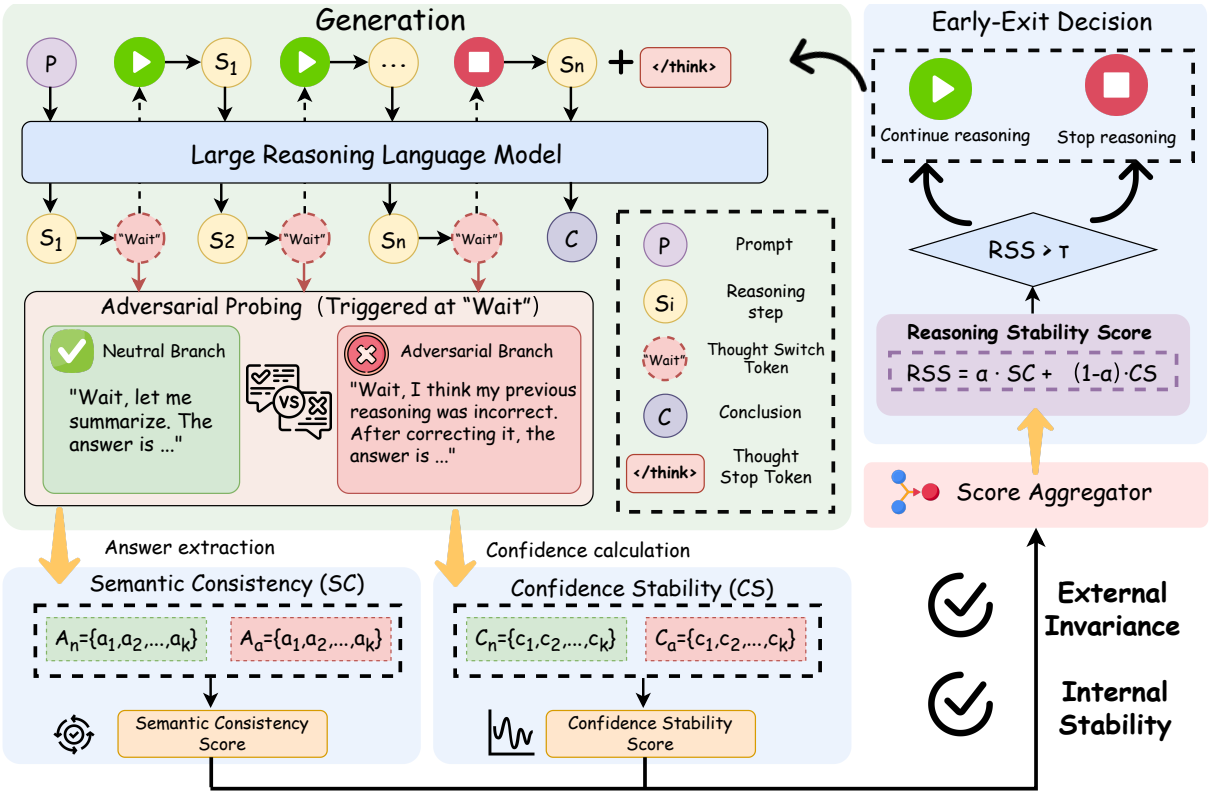

Figure 2: Overview of SABER. The framework performs adversarial probing by constructing neutral and adversarial branches at intermediate reasoning steps to estimate reasoning stability under perturbations. Semantic consistency and confidence stability are jointly integrated into the RSS, which dynamically determines whether reasoning should continue or terminate early.

dence variations steadily decrease. In contrast, incorrect trajectories remain unstable throughout the reasoning process, exhibiting persistent answer inconsistency and large confidence fluctuations even at later reasoning stages.

These observations indicate that successful reasoning trajectories progressively converge toward stable intermediate reasoning states. This stability consistently manifests in both generated answers and confidence dynamics, providing a potential criterion for characterizing reasoning convergence.

3 Method

Motivated by the observation that reasoning trajectories gradually become more stable as the reasoning process approaches a correct solution, we propose SABER, a training-free framework for adaptive early exit based on reasoning stability. As illustrated in Figure 2, SABER performs adversarial branch probing at intermediate reasoning steps by constructing neutral and adversarial branches from the same reasoning prefix. Based on the responses from these branches, SABER estimates reasoning stability and aggregates multiple signals into a unified Reasoning Stability Score (RSS), which is used to dynamically determine whether reasoning should continue or terminate early. The detailed formulations of each stability component are provided in the following subsections.

3.1 Adversarial Branch Probing

Following the probing setup introduced in Section 2, SABER performs adversarial probing during reasoning. Specifically, when the token "Wait" appears in the reasoning trajectory, the current reasoning prefix is branched into neutral and adversarial continuations.

For each branch, we perform k stochastic decoding samples to obtain answer multisets:

$$\mathcal{A}_n = \{a_n^{(j)}\}_{j=1}^{k}, \quad \mathcal{A}_a = \{a_a^{(j)}\}_{j=1}^{k}, \qquad (1)$$

where $\mathcal{A}_n$ and $\mathcal{A}_a$ denote the sampled answers under the neutral and adversarial branches, respectively.

For each sampled continuation, we additionally estimate its generation confidence based on token-level predictive probabilities, yielding two corre-

sponding confidence sets:

$$\mathcal{C}_n = \{c_n^{(j)}\}_{j=1}^k, \quad \mathcal{C}_a = \{c_a^{(j)}\}_{j=1}^k. \tag{2}$$

This design enables controlled semantic perturbations of intermediate reasoning states for stability estimation.

3.2 Reasoning Stability Estimation

We characterize reasoning trajectory stability from two complementary perspectives: semantic consistency (SC) and confidence stability (CS). Semantic consistency measures the consistency of generated predictions under perturbations, while confidence stability captures the variation of model confidence across perturbed branches. Together, these two signals provide a more reliable estimate of whether the current reasoning state has converged toward a stable solution.

Semantic Consistency. We quantify semantic consistency using the multiset Jaccard similarity between predictions from the neutral and adversarial branches:

$$SC = \frac{\sum_{a \in \mathcal{A}_n \cup \mathcal{A}_a} \min(f_n(a), f_a(a))}{\sum_{a \in \mathcal{A}_n \cup \mathcal{A}_a} \max(f_n(a), f_a(a))}, \tag{3}$$

where $f_n(a)$ and $f_a(a)$ denote the occurrence frequencies of answer a in $\mathcal{A}_n$ and $\mathcal{A}_a$, respectively.

Confidence Stability. For each sampled continuation, we compute a confidence score as the length-normalized geometric mean of token-level maximum predictive probabilities:

$$c_j = \left(\prod_{t=1}^{T} \max_{v \in \mathcal{V}} P(v \mid w_{<t}) \right)^{1/T}, \tag{4}$$

where $\mathcal{V}$ denotes the vocabulary, $w_{<t}$ represents the preceding token sequence, and T is the length of the generated continuation.

The average confidence under the neutral and adversarial branches is then computed as:

$$\bar{P}_n = \frac{1}{k} \sum_{j=1}^{k} c_n^{(j)}, \quad \bar{P}_a = \frac{1}{k} \sum_{j=1}^{k} c_a^{(j)}. \tag{5}$$

The confidence stability score is defined as:

$$CS = e^{-\gamma \cdot |\bar{P}_n - \bar{P}_a|}, \tag{6}$$

where γ controls sensitivity to confidence fluctuations. The exponential form amplifies instability under large deviations.

Higher SC and CS jointly indicate greater stability of the reasoning trajectory under adversarial perturbations, suggesting that further reasoning is unlikely to substantially change the final prediction.

3.3 Early-Exit Decision

We combine SC and CS into a unified score that reflects reasoning stability:

$$RSS = \alpha \cdot SC + (1 - \alpha) \cdot CS, \tag{7}$$

where $\alpha \in [0, 1]$ controls the relative contribution of the two signals. The resulting RSS serves as a unified indicator of reasoning stability and convergence under adversarial perturbations.

During reasoning, SABER continuously monitors the RSS at intermediate reasoning steps. Once the score exceeds a predefined threshold τ, the framework terminates the reasoning process and directly triggers final answer generation. For reasoning models with explicit reasoning delimiters, reasoning is terminated by injecting the closure token `</think>`.

The underlying intuition is that reasoning trajectories approaching correct solutions tend to become increasingly stable under perturbation. Once the reasoning state reaches a sufficiently stable regime, additional reasoning steps are unlikely to substantially change the final prediction. SABER therefore enables adaptive early exit while largely preserving reasoning quality.

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics. We evaluate our method on six widely used mathematical reasoning benchmarks—GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), AMC23 (AIMO Project and Project Numina, 2025), OlympiadBench (He et al., 2024), AIME 2024, and AIME 2025 (MAA Committees)—as well as the scientific reasoning benchmark GPQA Diamond (Rein et al., 2023). We report three primary evaluation metrics: *Accuracy* (**Acc**), which measures final answer correctness; *Token Count* (**Tok**), representing the average number of generated tokens per sample; and *Compression Ratio* (**CR**), which measures token consumption relative to vanilla reasoning, indicates inference efficiency, where lower values correspond to better performance.

Method	GSM8K		MATH-500		MATH AMC23		Olympiad		AIME24,25		SCIENCE GPQA-D		Overall	
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	CR↓
DeepSeek-R1-Distill-Qwen-7B														
Vanilla	**91.2**	1,164	90.4	3,797	**90.6**	6,032	**56.3**	7,461	46.3	10,683	31.8	7,437	67.8	100%
NoThinking	84.6	231	79.4	970	73.1	1,807	41.0	1,998	22.5	5,551	28.8	638	54.9	**30.6%**
Dynasor	89.8	773	89.4	2,987	86.7	4,689	54.8	6,477	42.5	9,726	31.3	1,471	65.8	71.4%
DEER	89.2	686	90.4	2,157	86.9	4,220	54.4	5,614	45.0	9,374	34.3	6,586	66.7	78.3%
SABER *(Ours)*	91.0	662	**91.2**	2,401	90.0	4,256	56.1	5,560	**50.9**	9,333	**34.8**	3,326	**69.0**	69.8%
Qwen3-4B														
Vanilla	94.7	2,113	92.4	4,796	**92.5**	7,396	59.7	9,066	**56.3**	12,399	53.5	8,139	74.8	100%
NoThinking	92.1	369	80.0	2,079	85.0	4,033	52.2	4,472	43.8	7,982	38.9	565	65.3	**44.4%**
Dynasor	94.2	1,998	89.0	4,189	83.1	6,457	58.5	7,658	52.1	10,648	54.6	6,687	71.9	85.7%
DEER	94.3	1,204	91.4	3,420	89.4	5,714	60.1	6,680	52.1	9,706	53.5	7,012	73.5	76.8%
SABER *(Ours)*	**94.8**	1,389	**92.8**	2,743	90.6	4,824	**61.9**	5,270	54.2	9,327	**56.1**	6,894	**75.1**	69.3%
Qwen3-8B														
Vanilla	**96.4**	2,192	**92.8**	5,135	**90.6**	7,871	59.0	9,447	**60.5**	12,501	55.1	9,047	75.7	100%
NoThinking	91.3	383	85.8	2,372	86.3	4,692	51.9	4,939	43.8	7,907	50.5	3,423	68.3	**51.3%**
Dynasor	95.1	1,740	89.8	4,060	86.9	6,225	57.9	7,415	52.5	10,475	52.0	4,420	72.4	74.3%
DEER	95.3	1,069	91.6	2,958	89.4	5,630	59.6	6,674	54.9	9,547	57.6	8,002	74.7	73.3%
SABER *(Ours)*	96.0	1,258	92.4	2,303	90.0	4,173	**61.5**	4,892	58.8	10,119	**57.6**	5,056	**76.0**	60.2%

Table 1: Experimental results across multiple reasoning benchmarks and model scales. "Acc" denotes accuracy, "Tok" denotes reasoning token count, and "CR" denotes compression ratio.↑(↓) indicates higher (lower) values are better.The best results are highlighted in **bold**.

Models. We conduct experiments on a diverse set of reasoning LLMs spanning multiple scales and architectures, including **Qwen3-4B**, **Qwen3-8B** (Yang et al., 2025a), and **DeepSeek-R1-Distilled-Qwen-7B** (Guo et al., 2025). For each model, we adopt the recommended reasoning prompts and decoding configurations.

Baselines. We compare **SABER** against vanilla reasoning and several representative training-free efficient reasoning methods, including NoThinking (Ma et al., 2025), DEER (Yang et al., 2025b), and Dynasor (Fu et al., 2025). **Vanilla reasoning** generates complete reasoning trajectories without early termination. **NoThinking** directly produces final answers, bypassing explicit reasoning. **DEER** performs adaptive early exit using confidence-based stopping signals, while **Dynasor** performs adaptive early exit based on intermediate answer consistency estimated through probing.

Implementation Details. All experiments are conducted using the vLLM framework on a server equipped with 8 NVIDIA RTX 3090 GPUs. Due to the relatively small sizes of AMC 2023, AIME 2024, and AIME 2025, each instance is evaluated four times and results are averaged across runs. The sensitivity coefficient γ in the confidence stability score is set to 3 across all experiments. The early-exit threshold τ is set to 0.9 for DeepSeek-R1-Distilled-Qwen-7B and 0.95 for the Qwen3-series models. Additional implementation details and hyperparameter settings are provided in Appendix A.1.

4.2 Main Results

Overall Performance. Table 1 shows that SABER consistently achieves strong efficiency–accuracy trade-offs across different model scales and reasoning benchmarks. Compared with vanilla reasoning, SABER reduces reasoning tokens by 30.2%–39.8% while maintaining comparable or improved accuracy. In particular, SABER improves the overall accuracy of DeepSeek-R1-Distilled-Qwen-7B from 67.8 to 69.0 while using only 69.8% of the original reasoning tokens, and also achieves the best overall accuracy on both Qwen3-4B and Qwen3-8B. These results suggest that our stability-aware early exit approach can effectively reduce redundant reasoning while preserving quality.

Comparison with Baselines. Compared with representative training-free efficient reasoning baselines, SABER consistently achieves stronger performance across diverse benchmarks and model scales. NoThinking attains high compression ratios

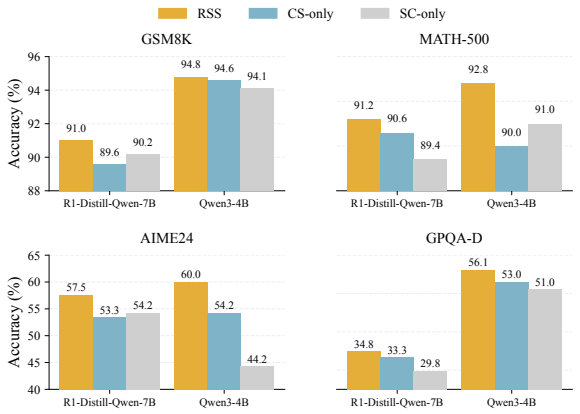

Figure 3: Accuracy of RSS early-exit on mathematical reasoning benchmarks: component ablation of semantic consistency (SC) and confidence stability (CS). Results are shown for RSS (our full method) compared with SC-only and CS-only variants on two representative models.

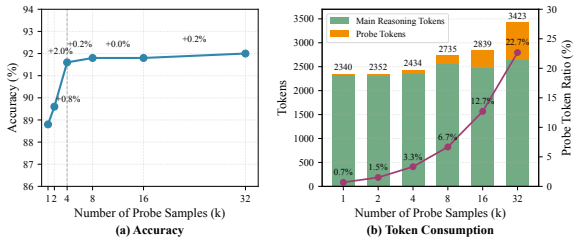

Figure 4: Sampling number ablation on MATH-500 using Qwen3-8B. Left: Accuracy improvement with increasing k. Right: Probe overhead growth and its proportion of total generated tokens as a function of k.

by bypassing explicit reasoning, but suffers substantial accuracy degradation. Dynasor and DEER rely on prediction consistency and confidence-based early exiting, respectively, yet both often produce unreliable exit decisions when intermediate reasoning states remain uncertain. In contrast, SABER jointly integrates semantic consistency and confidence stability under adversarial perturbations, enabling more reliable reasoning convergence estimation. The gains are particularly evident on challenging benchmarks such as OlympiadBench and GPQA-D.

5 Analysis

5.1 Ablation Study

Component Ablation. To examine the contributions of the two stability signals in SABER, we compare three configurations: (1) **SC-only**, using only semantic consistency as the early-exit signal, (2) **CS-only**, using only confidence stability as the early-exit signal, and (3) **RSS**, our full method. Figure 3 shows the accuracy of two reasoning models across four benchmarks. Overall, RSS outperforms both single-signal variants across all tasks, highlighting the importance of jointly modeling semantic consistency and confidence stability. On relatively simple tasks such as GSM8K and MATH-500, all methods achieve comparable performance. However, on more challenging reasoning tasks such as AIME24 and GPQA-D, both SC-only and CS-only degrade noticeably. This reflects the complementary nature of the two signals: SC-only may trigger premature exits on semantically consistent yet incorrect answers, whereas CS-only is vulnerable to stable but incorrect reasoning trajectories. By jointly considering semantic consistency and confidence stability, RSS more reliably identifies genuinely converged reasoning states.

Model	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
GSM8K			
R1-Distill-Qwen-7B	90.4/739	90.1/652	**91.0**/662
Qwen3-4B	94.3/1,635	94.5/1,503	**94.8**/1,389
Qwen3-8B	95.3/1,604	95.1/1,394	**96.0**/1,258
OlympiadBench			
R1-Distill-Qwen-7B	**56.1**/5,560	54.4/5,462	53.8/5,122
Qwen3-4B	**61.9**/5,270	59.6/5,121	59.6/4,889
Qwen3-8B	**61.5**/4,892	58.8/4,330	58.4/4,316

Table 2: Sensitivity analysis of weight α on GSM8K and OlympiadBench. Each cell shows accuracy (%) and average token count.

Scoring Function Ablation. To disentangle the contribution of the two-branch probing framework from that of the specific scoring function, we keep the neutral/adversarial two-branch probing framework unchanged and replace only the stopping score with two alternatives: (1) **SC · CS**, which multiplicatively combines semantic consistency and confidence stability, and (2) **Branch-UQ Diff**, which estimates the uncertainty of each branch using answer-set entropy and perplexity, and uses their difference as the stopping signal. Detailed formulations and results are provided in Appendix C.2. Both alternatives remain competitive under the same probing framework, indicating that SABER's gains are not tied to the specific formulation of RSS and suggesting that the two-branch probing framework is the primary source of the gains. Nevertheless, RSS achieves better performance, demonstrating its effectiveness as a stopping criterion within this framework.

5.2 Impact of Sampling Number

The sampling number k plays a critical role in balancing accuracy and computational overhead in SABER. We evaluate the effect of different sampling numbers on Qwen3-8B using six configurations with $k \in \{1, 2, 4, 8, 16, 32\}$. Figure 4 presents the trends of accuracy, main reasoning tokens, and probe overhead on MATH-500. As k increases, model accuracy generally improves, indicating that larger sample sizes yield more stable convergence estimation. Raising k from 1 to 4 increases accuracy from 88.8% to 91.6%, showing that a small number of additional samples significantly enhances the reliability of early-exit decisions. However, further increasing k leads to rapidly diminishing returns. Beyond $k = 4$, accuracy improves by only 0.4% even when the sampling number is raised to 32. Meanwhile, probe overhead grows approximately linearly, with its proportion of total tokens increasing from 3.3% to 22.7%, resulting in higher computational cost. These results suggest diminishing returns from increasing sampling budget, with most benefits achieved at small k values.

5.3 Effect of Weight α

To investigate the impact of the balancing weight α, we conduct a sensitivity analysis across different models and datasets, with results shown in Table 2. On GSM8K, $\alpha = 0.7$ consistently achieves the highest accuracy with fewer tokens across all three models, indicating that semantic consistency plays a dominant role in supporting reliable early exit on relatively simple reasoning tasks. In contrast, on OlympiadBench, smaller α values consistently yield higher accuracy, suggesting that difficult reasoning tasks rely more heavily on confidence stability, since semantic consistency alone may produce stable yet incorrect reasoning trajectories. Overall, simple tasks favor larger α values, while more challenging tasks benefit from smaller α values. These observations further demonstrate the complementarity of semantic consistency and confidence stability, and highlight a clear task-dependent preference in balancing the two signals across different levels of reasoning difficulty.

5.4 Threshold Sensitivity τ

We conduct threshold sensitivity analysis on the challenging OlympiadBench benchmark, since difficult reasoning tasks are generally more sensitive to early-exit decisions. Figure 5 shows the accuracy and average token count of three models under different stopping thresholds with $\alpha = 0.3$. As τ increases, the model becomes more selective, postponing early exit until stronger stability is achieved, thereby reducing premature exits on ambiguous reasoning steps. Consequently, model accuracy improves, accompanied by higher token consumption. Within $\tau \in [0.8, 0.95]$, performance remains relatively stable, indicating that SABER is robust to threshold selection within practical ranges and does not require delicate tuning.

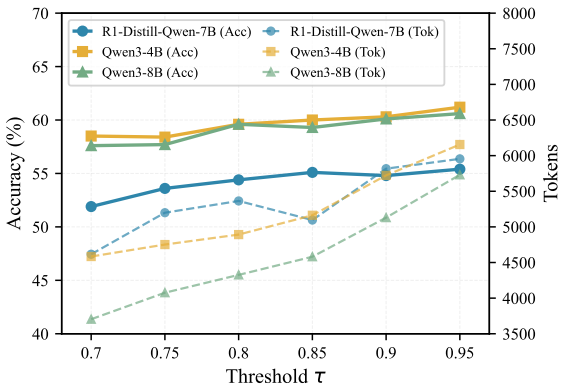

Figure 5: Accuracy (left y-axis) and average token count (right y-axis) on OlympiadBench under different stopping thresholds τ with $\alpha = 0.3$. Results are shown for three reasoning models.

Model	Probe	Total	Ratio (%)
R1-Distill-Qwen-7B	221	4,477	4.9%
Qwen3-4B	147	5,221	2.8%
Qwen3-8B	181	4,815	3.8%
Average	183	4,838	3.8%

Table 3: Probe overhead under the default configuration. Probe tokens account for only a small fraction of total generated tokens.

5.5 Probe Overhead Analysis

We analyze the additional computational overhead introduced by the probing mechanism in SABER. As shown in Table 3, which reports the average probing ratios across all test datasets, the ratios for the three models are 4.9%, 2.8%, and 3.8%, respectively, with an average of only 3.8%. This indicates that adversarial probing constitutes a relatively small proportion of the overall reasoning process, and the associated overhead remains manageable. More importantly, the token savings achieved through early exit substantially outweigh the ad-

Model	Vanilla	SABER	Reduction
R1-Distill-Qwen-7B	149.4	92.9	37.8%
Qwen3-4B	190.1	107.0	43.7%
Qwen3-8B	243.0	98.5	59.5%
Average	194.2	99.5	48.8%

Table 4: Average inference latency (in seconds) of SABER across GSM8K, AIME24, and GPQA-D.

ditional cost introduced by adversarial probing, demonstrating a favorable cost-benefit trade-off.

5.6 Inference Latency

We evaluate the end-to-end inference latency of SABER on GSM8K, AIME24, and GPQA-D, covering reasoning tasks of varying difficulty levels. Table 4 reports the average inference latency across three models. The results show that SABER significantly reduces inference latency (48.8% on average). The reduction is particularly pronounced on more challenging tasks with longer reasoning trajectories, where late-stage computation dominates runtime. These latency gains are consistent with our token reduction analysis, suggesting that fewer generated tokens contribute to lower wall-clock time. In summary, SABER achieves substantial wall-clock speedups, highlighting its practical efficiency in real-world inference scenarios.

6 Related Work

Efficient Reasoning and Early Exit. To mitigate the overthinking problem, researchers have proposed various efficient reasoning strategies, which can be broadly categorized into three types. **Post-training methods** enable models to adaptively generate reasoning chains of varying lengths through supervised fine-tuning on variable-length CoT data (Zhao et al., 2025; Yu et al., 2024; Kang et al., 2025) or length rewards in reinforcement learning (Lou et al., 2025; Aggarwal and Welleck, 2025). **Prompt-guided methods** rely on carefully designed prompts, such as "Be concise" (Renze and Guven, 2024; Xu et al., 2025) or token budget constraints (Han et al., 2025), to encourage shorter reasoning. **Inference-time methods** dynamically control reasoning behavior during generation to reduce redundant computation without modifying model parameters, offering greater flexibility. We focus on inference-time methods and study how to determine when reasoning has been sufficient for early termination.

Inference-time Early Exit Signals. The core of inference-time early exit methods lies in constructing a reliable **stopping signal**. One representative line of work is based on **uncertainty signals**, such as confidence or entropy. EAT (Wang et al., 2026) leverages token-level entropy, HALT-CoT (Laaouach, 2025) computes the entropy of answer distributions, while DEER (Yang et al., 2025b) and CGRS (Huang et al., 2026) decide whether to exit at specific checkpoints based on confidence. Another line of work relies on **consistency signals**. LRMs often converge to stable final answers at early stages, which can be directly used as a stopping signal (Liu and Wang, 2025). Dynasor (Fu et al., 2025) triggers early exit by detecting consistency in intermediate answers, and TRACE (Li et al., 2026) further combines consistency and confidence within a sliding window, at the cost of multi-step aggregation and additional latency. In addition, **hidden-state methods** attempt to estimate reasoning status directly from internal representations. TPV (Eisenstadt et al., 2025) encodes reasoning progress using hidden states, SEAL (Chen et al., 2025a) steers reasoning via activation vectors, and probe-based methods (Zhang et al., 2025) train probes on hidden states to predict answer correctness. However, these methods typically require additional training and model-specific adaptation. Overall, existing methods primarily rely on output-level or internal signals for early exit, while leaving the stability of intermediate reasoning trajectories underexplored as a factor for decision making.

Multi-path Reasoning and Perturbation. To obtain more reliable predictions than a single reasoning chain, one line of research uses multi-path reasoning. Self-Consistency (Wang et al., 2022) generates multiple reasoning paths via repeated sampling and aggregates final predictions through voting, based on the insight that semantic agreement across paths reflects answer reliability. However, this approach requires generating full-length paths in parallel and is not directly applicable to early exit. A related line of research studies reasoning path perturbation, which injects neutral or adversarial perturbations into intermediate CoT steps (von Recum et al., 2026; Aravindan and Kejriwal, 2026), revealing the fragility and limited recovery ability of reasoning models under perturbations. Meanwhile, prior studies have shown that behavioral stability under perturbations can

provide useful signals for model reliability and uncertainty estimation (Khanmohammadi et al., 2025; Saadat and Nemzer, 2026). However, existing perturbation-based studies primarily use perturbations for robustness evaluation or behavioral analysis, rather than online reasoning control during generation. Building on these insights, we view perturbation-induced multi-branch reasoning as a promising source of stability signals for early exit.

7 Conclusion

We propose SABER, a training-free framework for stability-aware early exit via adversarial branch probing. Rather than relying solely on confidence or entropy, SABER estimates reasoning convergence by measuring the stability of intermediate reasoning states under lightweight semantic perturbations. This approach efficiently evaluates the consistency of potential outcomes without requiring additional full-generation rollouts. Across diverse reasoning benchmarks and model architectures, SABER substantially reduces reasoning token consumption while maintaining or even improving reasoning accuracy. Overall, our results suggest that reasoning stability provides an effective signal for adaptive inference in large reasoning models and offers a promising direction for efficient inference-time scaling.

Limitations

Despite the promising results of SABER in improving reasoning efficiency, our evaluation is subject to several limitations. Due to computational constraints, we conduct experiments on text-based reasoning benchmarks using models ranging from 4B to 8B parameters, and do not investigate larger-scale models. In addition, we focus exclusively on text-based settings and do not evaluate SABER in multimodal or agent-based scenarios, where intermediate representations and confidence dynamics may differ significantly. Exploring the applicability of SABER in these broader settings remains an important direction for future work.

Acknowledgments

We would like to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (NSFC No. 62576232) and the Key Laboratory of Data Intelligence and Advanced Computing, Soochow University.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*.

AI-MO Project and Project Numina. 2025. Aimo validation amc: Amc math validation set. https://huggingface.co/datasets/AI-MO/aimo-validation-amc. Extracted from AMC12 2022 and AMC12 2023 problems from the Art of Problem Solving Wiki and adapted to integer outputs.

Ashwath Vaithinathan Aravindan and Mayank Kejriwal. 2026. Fragile thoughts: How large language models handle chain-of-thought perturbations. *Preprint*, arXiv:2603.03332.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. 2025a. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Roy Eisenstadt, Itamar Zimerman, and Lior Wolf. 2025. Overclocking llm reasoning: Monitoring and controlling thinking path lengths in llms. *arXiv preprint arXiv:2506.07240*.

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*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan

Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *Preprint*, arXiv:2402.14008.

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. *Preprint*, arXiv:2103.03874.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2026. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31176–31184.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

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*, volume 39, pages 24312–24320.

Reza Khanmohammadi, Erfan Miahi, Mehrsa Mardikoraem, Simerjot Kaur, Ivan Brugere, Charese H. Smiley, Kundan Thind, and Mohammad M. Ghassemi. 2025. Calibrating llm confidence by probing perturbed representation stability. *Preprint*, arXiv:2505.21772.

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*.

Jiakun Li, Xingwei He, Kefan Li, Hongzheng Chai, Hongyue Yu, and Yuan Yuan. 2026. Efficient test-time scaling via temporal reasoning aggregation. *arXiv preprint arXiv:2604.17304*.

Xin Liu and Lu Wang. 2025. Answer convergence as a signal for early stopping in reasoning. *Preprint*, arXiv:2506.02536.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.

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

MAA Committees. Aime problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`. Accessed: 2026-05-07.

Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Sho, and Anirudha Majumdar. 2026. Reasoning about uncertainty: Do reasoning models know when they don't know? In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3408–3458.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. *arXiv preprint arXiv:2401.05618*.

Mohammadreza Saadat and Steve Nemzer. 2026. Certainty robustness: Evaluating llm stability under self-challenging prompts. *Preprint*, arXiv:2603.03330.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, i'm wrong: Llms hallucinate with certainty despite knowing the answer. *arXiv preprint arXiv:2502.12964*.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un) answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. 2025. Uncertainty quantification for llms through minimum bayes risk: Bridging confidence and consistency. *Preprint*, arXiv:2502.04964.

Alexander von Recum, Leander Girrbach, and Zeynep Akata. 2026. Are reasoning llms robust to interventions on their chain-of-thought? *Preprint*, arXiv:2602.07470.

Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. 2026. Entropy after </think> for reasoning model early exiting. *Preprint*, arXiv:2509.26522.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025b. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

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

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

Haoran Zhao, Yuchen Yan, Yongliang Shen, Haolei Xu, Wenqi Zhang, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. Let lrms break free from overthinking via self-braking tuning. *arXiv preprint arXiv:2505.14604*.

A Experimental Details

A.1 Hyperparameter Settings

We provide detailed hyperparameter settings of SABER, including configurations for adversarial probing, stability estimation, and early-exit decisions. During adversarial probing, we sample short continuations to estimate reasoning trajectory stability under perturbations. To reduce the additional computational overhead introduced by probing, each probing branch is restricted to generate at most 10 tokens. At each intermediate reasoning step, we perform stochastic decoding with sampling number $k = 4$ to obtain sampled answer sets and corresponding confidence sets, as defined in Eq. 1 and Eq. 2. Unless otherwise specified, we use a temperature of 0.6 and a top-p of 0.95.

For all experiments, the maximum context length and maximum generation length are set to 32k and 16k tokens, respectively. For the confidence stability score in Eq. 6, we set the sensitivity coefficient $\gamma = 3$ across all experiments to better capture confidence fluctuations under adversarial perturbations. For the RSS in Eq. 7, we perform hyperparameter search over $\alpha \in \{0.3, 0.5, 0.7\}$, with 0.3 or 0.7 selected depending on the model and dataset. Finally, we perform hyperparameter search over $\tau \in \{0.85, 0.9, 0.93, 0.95, 0.98\}$, with the selected threshold set to 0.9 for DeepSeek-R1-Distilled-Qwen-7B and 0.95 for the Qwen3-series models.

A.2 Dataset Details

The mathematical and scientific reasoning datasets used in our evaluation are described below. We strictly follow the licenses and evaluation protocols specified in the original papers.

GSM8K (Cobbe et al., 2021) is a widely used benchmark for grade-school mathematical reasoning. It consists of diverse multi-step arithmetic word problems that require sequential reasoning and intermediate numerical calculations, making it a standard benchmark for evaluating chain-of-thought reasoning capability.

MATH-500 (Hendrycks et al., 2021) is a curated subset of the MATH benchmark containing 500 high-difficulty mathematics problems. The dataset spans multiple domains, including algebra, calculus, geometry, and probability, and is widely used for evaluating mathematical reasoning and test-time scaling behavior.

AMC23 (AI-MO Project and Project Numina, 2025) is a validation benchmark constructed from AMC 12 problems released in 2023. The dataset focuses on high-school-level mathematics and includes problems involving algebraic reasoning, geometry, and combinatorial counting. Following prior work, all answers are normalized to integer outputs for consistent evaluation.

OlympiadBench (He et al., 2024) is a challenging benchmark designed to evaluate advanced mathematical reasoning at the Olympiad level. It consists of competition-style problems covering algebra, geometry, number theory, and combinatorics. The benchmark typically requires long-form multi-step reasoning and symbolic manipulation, making

it particularly challenging for large reasoning models.

AIME 2024 and **AIME 2025** (MAA Committees) are constructed from the American Invitational Mathematics Examination problems from 2024 and 2025, respectively. These datasets consist of short-answer competition math problems that require deep multi-step reasoning and nontrivial mathematical insights.

GPQA Diamond (Rein et al., 2023) is the difficult split of the Graduate-Level Google-Proof QA benchmark. It contains expert-written multiple-choice questions spanning physics, chemistry, and biology, where solving the problems requires domain knowledge and multi-step scientific reasoning rather than memorization.

All datasets are evaluated under the zero-shot inference setting. Their requirement for long reasoning chains and complex intermediate computations makes them well-suited for studying early-exit strategies and reasoning efficiency.

A.3 Prompt Templates

We use a unified prompting format across all benchmarks to ensure fair comparison between different early-exit methods. For all datasets, the model is instructed to explicitly generate intermediate reasoning steps before producing the final answer.

Base Inference Prompt. The following prompt is used for standard reasoning generation during inference:

```
Please reason step by step, and put
your final answer within \boxed{}.
```

Neutral Probing Prompt. The neutral probing prompt encourages the model to continue its current reasoning trajectory and summarize the inferred answer:

```
Wait, let me summarize. The answer is
\boxed{
```

Adversarial Probing Prompt. The adversarial probing prompt encourages the model to suspect that its previous reasoning trajectory may be incorrect, and then guides the model to regenerate and summarize the final answer under the perturbed reasoning context:

```
Wait, I think my previous reasoning
was incorrect. After correcting it,
the answer is \boxed{
```

B Implementation Details

B.1 Overview

SABER is implemented as a training-free decoding-time intervention that operates seamlessly on top of the standard auto-regressive generation process. During inference, the framework continuously monitors the reasoning trajectory for predefined transition points. Once triggered, the current reasoning state is branched into lightweight neutral and adversarial probing continuations, where stochastic multi-sampling is performed in parallel. Based on the resulting Reasoning Stability Score (RSS), SABER dynamically determines whether reasoning has sufficiently converged: if the RSS exceeds a predefined threshold τ, the framework injects the closure token `</think>` to terminate reasoning early and directly generate the final answer; otherwise, the probing branches are discarded and standard step-by-step reasoning resumes. To ensure practical efficiency, all probing branches are strictly length-constrained and parallelized within the inference engine.

B.2 Pseudo-code

Algorithm 1 summarizes the SABER procedure.

B.3 Reasoning Transition Points

As discussed in Section 2, we segment reasoning trajectories into intermediate steps using several transition trigger phrases, including "Wait", "Alternatively", "But", "So", and "Let me double-check". These phrases commonly indicate self-correction, verification, or subgoal transitions, and empirically provide effective decomposition of long reasoning trajectories into coherent reasoning steps.

In subsequent evaluations, to reduce probing overhead, we only perform adversarial perturbation probing when the token "Wait" appears, as it is the most representative indicator of reasoning revision and uncertainty reflection.

C Additional Experimental Results

C.1 Comparison with Confidence-Only Early-Exit

To isolate the contribution of adversarial branch probing, we compare SABER with a confidence-

Algorithm 1 SABER: Stability-Aware Early Exit for LLM Reasoning via Adversarial Branch Probing

Require: Query q; model $\mathcal{M}$; transition tokens $\mathcal{T}$ (e.g., "Wait", "Alternatively"); sample size k; sensitivity γ; weight α; threshold τ; max length $L_{\max}$

Ensure: Final generated answer y

$\mathcal{P}_{\text{probe_n}} \leftarrow$ "Wait, let me summarize. The answer is \boxed{" ▷ Define neutral probe template
$\mathcal{P}_{\text{probe_a}} \leftarrow$ "Wait, I think my previous reasoning was incorrect..." ▷ Define adversarial probe template
$\mathcal{P} \leftarrow q$ ▷ Initialize reasoning prefix state $\mathcal{P}$
while $|\mathcal{P}| < L_{\max}$ **do**
  Generate next token $t \sim \mathcal{M}(\mathcal{P})$ ▷ Auto-regressive generation
  $\mathcal{P} \leftarrow \mathcal{P} \oplus t$
  if $t \in \mathcal{T}$ **then** ▷ Intercept at transition points
    $\mathcal{P}_n \leftarrow \mathcal{P} \oplus \mathcal{P}_{\text{probe_n}}$ ▷ Neutral probe prefix
    $\mathcal{P}_a \leftarrow \mathcal{P} \oplus \mathcal{P}_{\text{probe_a}}$ ▷ Adversarial probe prefix
    $\mathcal{A}_n, \mathcal{A}_a \leftarrow \emptyset$ ▷ Initialize answer multisets, Eq. (1)
    $\mathcal{C}_n, \mathcal{C}_a \leftarrow \emptyset$ ▷ Initialize confidence sets, Eq. (2)
    for $j = 1$ to k **do**
      Sample $a_n^{(j)} \sim \mathcal{M}(\mathcal{P}_n)$; compute confidence $c_n^{(j)}$
      Sample $a_a^{(j)} \sim \mathcal{M}(\mathcal{P}_a)$; compute confidence $c_a^{(j)}$
      $\mathcal{A}_n \leftarrow \mathcal{A}_n \cup \{a_n^{(j)}\},\ \mathcal{C}_n \leftarrow \mathcal{C}_n \cup \{c_n^{(j)}\}$
      $\mathcal{A}_a \leftarrow \mathcal{A}_a \cup \{a_a^{(j)}\},\ \mathcal{C}_a \leftarrow \mathcal{C}_a \cup \{c_a^{(j)}\}$
    end for
    $SC \leftarrow \textsc{MultisetJaccard}(\mathcal{A}_n, \mathcal{A}_a)$ ▷ Semantic Consistency, Eq. (3)
    $\bar{P}_n \leftarrow \frac{1}{k}\sum_{j=1}^{k} c_n^{(j)}, \quad \bar{P}_a \leftarrow \frac{1}{k}\sum_{j=1}^{k} c_a^{(j)}$ ▷ Average confidences
    $CS \leftarrow e^{-\gamma \cdot |\bar{P}_n - \bar{P}_a|}$ ▷ Confidence Stability, Eq. (6)
    $RSS \leftarrow \alpha \cdot SC + (1-\alpha) \cdot CS$ ▷ Reasoning Stability Score, Eq. (7)
    if $RSS > \tau$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \oplus$ `</think>` ▷ Inject closure token
      break ▷ Early exit: reasoning state has converged
    end if
  end if
end while
$y \sim \mathcal{M}(\mathcal{P})$ ▷ Trigger final answer generation
return y

only early-exit variant that removes the adversarial branch and performs stopping decisions solely based on confidence estimated from the neutral branch.

Figure 6 shows the efficiency–accuracy trade-off on MATH-500 under different stopping thresholds. SABER consistently achieves a better Pareto frontier across nearly all operating regions and remains substantially closer to the oracle upper bound.

These results suggest that confidence alone is insufficient for reliable early exit, since reasoning models may remain highly confident even when intermediate reasoning states have not fully converged. By evaluating reasoning stability under adversarial perturbations through semantic consistency and confidence stability, SABER provides more reliable reasoning stability estimation. Notably, SABER surpasses Vanilla reasoning under several thresholds while using fewer reasoning tokens.

C.2 Scoring Function Ablation

We provide detailed formulations and results for the alternative scoring functions evaluated under the same neutral/adversarial two-branch probing framework.

For **SC** · **CS**, we replace the weighted-additive formulation of RSS with a multiplicative combina-

Scoring Function	GSM8K		MATH-500		Olympiad		AIME24,25		GPQA-D		Overall	
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	CR↓
Vanilla	**91.2**	1,164	90.4	3,797	**56.3**	7,461	46.3	10,683	31.8	7,437	63.2	100%
RSS (Ours)	91.0	662	91.2	**2,401**	56.1	5,560	**50.9**	9,333	**34.8**	3,326	**64.8**	69.7%
SC · CS	91.1	**659**	**91.4**	2,509	56.1	**5,505**	48.4	9,697	31.8	4,576	63.8	75.1%
Branch-UQ Diff	90.1	864	90.6	2,826	55.0	5,639	47.1	**8,788**	31.3	**2,998**	62.8	**69.1%**

Table 5: Ablation of different scoring functions under the same neutral/adversarial probing framework on DeepSeek-R1-Distill-Qwen-7B.

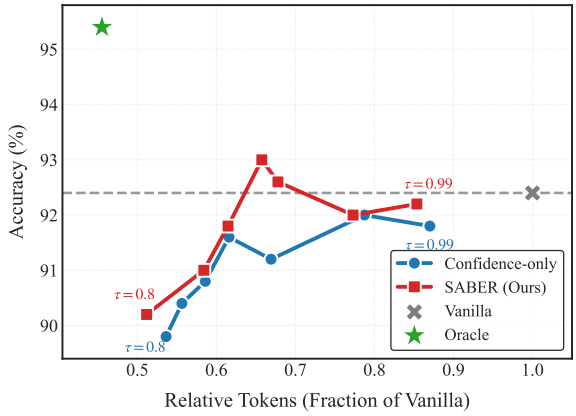

Figure 6: Comparison between SABER and a confidence-only early-exit strategy on MATH-500. SABER consistently achieves a better efficiency–accuracy trade-off and remains closer to the oracle upper bound across different stopping thresholds.

tion of the same two stability signals:

$$S_{mult} = SC \cdot CS. \tag{8}$$

For **Branch-UQ Diff**, we instead adopt a branch-level uncertainty-based score. Inspired by prior uncertainty quantification methods (Vashurin et al., 2025), we estimate the uncertainty of each branch as

$$UQ_b = H_b + \log PPL_b, \tag{9}$$

where H_b denotes the answer-set entropy and PPL_b denotes the perplexity of branch b. The stopping score is then defined as the absolute uncertainty difference between the two branches:

$$S_{UQ} = |UQ_n - UQ_a|. \tag{10}$$

For a fair comparison, we independently tune the stopping threshold for each alternative: $\tau \in \{0.75, 0.80, 0.85, 0.90, 0.95\}$ for SC · CS and $\tau \in \{0.05, 0.10, 0.20, 0.30\}$ for Branch-UQ Diff.

Table 5 reports the results. Both SC · CS and Branch-UQ Diff remain effective under the same two-branch probing framework, despite using different scoring formulations. RSS achieves better performance because it additionally models semantic consistency between the two branches, rather than relying only on their uncertainty gap.

C.3 Adversarial Probing Prompt Style Analysis

To systematically study the effect of perturbation behaviors, we design a perturbation prompt family consisting of four intervention styles: *Self-Correction* (explicitly acknowledging and correcting a potential mistake), *Reflection* (reconsidering the current reasoning state), *Alternative Path* (encouraging an alternative reasoning trajectory), and *Verification* (checking the existing reasoning process), as shown in Table 6. SABER uses a *Self-Correction* style probe as the default configuration, whereas the *Self-Correction* variant adopts an alternative prompt from the same perturbation family.

We then evaluate how different perturbation styles affect the behavior of SABER. Table 7 compares our default SABER configuration against four perturbation variants using Qwen3-4B. Among the four variants, *Self-Correction* achieves the highest average accuracy with moderate token consumption, while *Verification* is the most token-efficient with competitive accuracy. Notably, our default SABER configuration consistently outperforms all variants, achieving the best average accuracy with reasonable token efficiency.These results suggest that different perturbation behaviors lead to different efficiency–accuracy trade-offs.

Overall, all perturbation styles achieve stable and competitive results, indicating that SABER does not rely on a specific manually designed perturbation prompt. Instead, its effectiveness primarily depends on the perturbation behavior induced by the prompt rather than its exact wording.

Category	Perturbation Prompt
Self-Correction	"Wait, I might have made a mistake. Let me reconsider, the answer is \boxed{"
Reflection	"Wait, let me reconsider the reasoning, the answer is \boxed{"
Alternative Path	"Wait, let me try a different approach, the answer is \boxed{"
Verification	"Wait, let me verify the reasoning once more, the answer is \boxed{"

Table 6: Example perturbation probes. Each category represents a family of possible prompts; only one example is shown per category. SABER defaults to the Self-Correction style (not separately marked).

Perturbation Strategy	GSM8K		MATH-500		Olympiad		GPQA-D		Overall	
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓
SABER (Ours)	**94.4**	1,349	**92.0**	2,744	60.0	5,918	**57.6**	6,502	**76.0**	4,128
Self-Correction	94.1	1,098	91.6	2,591	59.4	5,837	57.0	5,497	75.5	3,756
Reflection	94.2	1,085	90.8	2,647	58.5	5,600	57.6	**5,279**	75.3	3,653
Alternative Path	94.1	1,173	90.2	2,765	**61.0**	5,875	52.5	6,029	74.5	3,961
Verification	94.2	**1,023**	91.6	**2,538**	58.7	**5,198**	52.0	5,315	74.1	**3,519**

Table 7: Comparison of different perturbation styles on four benchmarks using Qwen3-4B.

C.4 Trigger Sensitivity Analysis

To assess the sensitivity of SABER to the specific lexical trigger "Wait", we replace it with a more general and semantically neutral double newline (\n\n), while keeping all other settings unchanged. As shown in Table 8, the newline-based trigger achieves nearly identical overall accuracy (70.7% vs. 70.8%) and comparable token compression (71.9% vs. 66.4%). This suggests that SABER is not strongly tied to the specific lexical cue "Wait". The trigger primarily provides a practical way to locate potential reasoning transition points for probe generation.

D Case Study

We present several representative cases generated by Qwen3-8B under the default SABER setting ($\alpha = 0.7, \tau = 0.95$). These cases illustrate how SABER adaptively identifies stable reasoning states through adversarial branch probing, thereby reducing unnecessary reasoning and mitigating overthinking.

Case 1: Detecting Unstable Intermediate Reasoning States. Figure 7 illustrates how SABER dynamically evaluates reasoning stability during multi-step mathematical reasoning on GSM8K. In the early reasoning stages, although the neutral branch already produces highly consistent answers, the adversarial branch remains unstable under perturbation, indicating that the intermediate reasoning state has not yet stabilized. Consequently, SABER continues the reasoning process instead of exiting early. As reasoning progresses, both branches gradually stabilize and converge to the same correct answer, causing the RSS to exceed the predefined threshold and trigger early termination. This example shows that local answer agreement or high confidence alone may provide misleading stopping signals, whereas SABER evaluates whether reasoning remains stable under adversarial perturbations to better identify stable reasoning states.

Case 2: Avoiding Redundant Overthinking. Figure 8 compares SABER with vanilla reasoning. Although both methods ultimately arrive at the correct answer, Vanilla continues to perform repeated verification and reflection after reaching the correct solution, generating substantial redundant reasoning. SABER terminates reasoning once stable convergence is detected, substantially reducing reasoning token consumption.

Case 3: Preventing Endless Reflection and Reasoning Truncation. Figure 9 shows a case where vanilla reasoning exceeds the context limit due to

Trigger	GSM8K		MATH-500		AIME24		AIME25		Olympiad		GPQA-D		Overall	
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	CR↓
Vanilla	**96.4**	2,192	**92.8**	5,135	**66.7**	12,003	54.2	12,999	59.0	9,447	55.1	9,047	70.7	100%
"Wait"	96.0	**1,258**	92.4	**2,303**	65.0	9,645	52.5	**10,592**	**61.5**	**4,892**	**57.6**	**5,056**	**70.8**	**66.4%**
Newline (\n\n)	95.7	1,312	**92.8**	3,387	65.0	**9,115**	**55.0**	10,797	59.7	6,190	56.1	5,726	70.7	71.9%

Table 8: Trigger sensitivity analysis on Qwen3-8B. Replacing the lexical trigger "Wait" with a semantically neutral double newline (\n\n) yields comparable performance, suggesting limited sensitivity to the choice of trigger.

persistent reflection. Although the model actually arrives at the correct answer early, it repeatedly verifies and reflects, ultimately failing to output the final result. SABER instead stops reasoning once the reasoning state becomes sufficiently stable, thereby avoiding truncation caused by excessive reflection.

Case 4: Overthinking Can Hurt Reasoning Accuracy. In the case shown in Figure 10, vanilla reasoning arrives at the correct answer early, but during subsequent reflection, it modifies its original conclusion and ultimately produces an incorrect answer. In contrast, SABER avoids unnecessary post-hoc reflection and successfully preserves the correct answer. This case further suggests that excessive reflection may degrade reasoning performance.

Question:

A basic manicure starts at $35 while a pedicure starts at $40. A certain salon offers 20% off if you get both. Suppose Karen chooses to do both and would also like to add nail art on each of her fingers, which costs $3 per nail. How much would she pay in all?

Response:

<think>\nOkay, let me try to figure out how much Karen will pay for her manicure, pedicure, and nail art. Hmm, let's start by breaking down the problem step by step. \n\nFirst, the basic manicure starts at $35, and the pedicure starts at $40.*<omitted>*\nTotal before discount: 35 + 40 = $75**\n\nWait,**

Reasoning Stability Estimation

Semantic Consistency(SC): 0.6 An=[75, 75, 75, 75], Am=[85, 75, 75, 75]	*Confidence Stability(CS): 0.71* Avg_Pn=0.78, Avg_Pm=0.67	RSS : 0.63 < 0.95

Continue reasoning

the salon offers a 20% discount if you get both a manicure and a pedicure. So, I need to apply that 20% off to the combined total of $75. Let me calculate 20% of $75 first.*<omitted>*Therefore, nail art on each finger would be 10 nails. Each costing $3. So, 10 * $3 = $30**\n\nWait,**

Reasoning Stability Estimation

Semantic Consistency(SC): 1.0 An=[90, 90, 90, 90], Am=[90, 90, 90, 90]	*Confidence Stability(CS): 0.35* Avg_Pn=0.62, Avg_Pm=0.96	RSS : 0.81 < 0.95

Continue reasoning

but let me make sure I'm not missing something here. The problem says \"nail art on each of her fingers,\"*<omitted>*So, the total cost would be the discounted manicure and pedicure ($60) plus the nail art ($30). \n\nSo total cost: 60 + 30 = $90.**\n\nWait,**

Reasoning Stability Estimation

Semantic Consistency(SC): 1.0 An=[90, 90, 90, 90], Am=[90, 90, 90, 90]	*Confidence Stability(CS): 0.97* Avg_Pn=0.98, Avg_Pm=0.97	RSS : 0.99 > 0.95

Terminate reasoning

\n\n**</think>**\n\nTo determine how much Karen will pay in total, we need to break down the costs step by step:\n\n---\n\n### **1. Base Costs of Services**\n\n- **Manicure:** $35 \n- **Pedicure:** $40*<omitted>***Final Answer:**\n$$\n\\boxed{90}\n$$

Figure 7: A case on GSM8K where SABER identifies unstable reasoning states.

Question: Define\n\\[p = \\sum_{k = 1}^\\infty \\frac{1}{k^2} \\quad \\text{and} \\quad q = \\sum_{k = 1}^\\infty \\frac{1}{k^3}.\\]Find a way to write\n\\[\\sum_{j = 1}^\\infty \\sum_{k = 1}^\\infty \\frac{1}{(j + k)^3}\\]in terms of $$$$ and $$q$$

Answer:

SABER

<think>\nOkay, so I need to find a way to express the double sum $\\sum_{j = 1}^\\infty \\sum_{k = 1}^\\infty \\frac{1}{(j + k)^3}$ in terms of p and q, where $p = \\sum_{k = 1}^\\infty \\frac{1}{k^2}$ and $q = \\sum_{k = 1}^\\infty \\frac{1}{k^3}$. Let me start by understanding what this double sum represents and how I can relate it to p and q.\n\nFirst, let me recall that p is the famous Basel problem, which converges to $\\frac{\\pi^2}{6}$, and q is the Riemann zeta function at 3,
<omitted>
So:\n\n$$\n\\sum_{j=1}^\\infty \\sum_{k=1}^\\infty \\frac{1}{(j + k)^3} = \\sum_{n=2}^\\infty \\frac{n - 1}{n^3}\n$$**\n\nWait**

SC: 1.0 CS: 0.83 RSS : 0.88

, let me check that. If I fix $n = j + k$, then for each n, there are $n - 1$ pairs (j, k), so the double sum becomes:\n\n$$\n\\sum_{n=2}^\\infty \\left(\\sum_{\\substack{j=1 \\\\ k=1 \\\\ j + k = n}}^\\infty 1 \\right) \\cdot \\frac{1}{n^3} = \\sum_{n=2}^\\infty (n - 1) \\cdot \\frac{1}{n^3}\n$$\n\nYes, that seems right.
<omitted>
Similarly, $q = \\sum_{n=1}^\\infty \\frac{1}{n^3}$, so $\\sum_{n=2}^\\infty \\frac{1}{n^3} = q - \\frac{1}{1^3} = q - 1$. \n\nTherefore, substituting back:\n\n$$\n(p - 1) - (q - 1) = p - 1 - q + 1 = p - q\n$$**\n\nWait**

SC: 1.0 CS: 0.99 RSS : 0.99>0.95

</think>

\n\nTo evaluate the double sum\n\n$$\n\\sum_{j = 1}^\\infty \\sum_{k = 1}^\\infty \\frac{1}{(j + k)^3},\
<omitted>
Substituting these into the expression:\n\n$$\n\\sum_{n = 2}^\\infty \\left(\\frac{1}{n^2} - \\frac{1}{n^3} \\right) = (p - 1) - (q - 1) = p - q.\n$$\n\n---\n\n### Final Answer:\n\n$$\n\\boxed{p - q}\n$$

Token number: 1626

Vanilla

<think>\nOkay, so I need to find a way to express the double sum $\\sum_{j = 1}^\\infty \\sum_{k = 1}^\\infty \\frac{1}{(j + k)^3}$ in terms of p and q, where $p = \\sum_{k = 1}^\\infty \\frac{1}{k^2}$ and $q = \\sum_{k = 1}^\\infty \\frac{1}{k^3}$. Hmm, let me think about how to approach this.\n\nFirst, I remember that double sums can sometimes be simplified by changing variables or by recognizing patterns.
<omitted>
Therefore, the sums from $n=2$ to infinity are just $p - 1$ and $q - 1$, respectively. So substituting:\n\n$$\n(p - 1) - (q - 1) = p - 1 - q + 1 = p - q\n$$**\n\nWait**, so does that mean the original double sum is equal to $p - q$? Let me verify this because that seems straightforward, but I want to make sure I didn't make a mistake in the substitution.**\n\nLet me check** with small values. Let's compute partial sums numerically. Let's take the original double sum up to j=1, k=1, then j=1, k=2, etc., and see if it aligns with p - q.\n\nBut maybe that's too tedious. **Alternatively, let me check the steps again.**\n\nOriginal double sum: sum_{j=1}^\\infty sum_{k=1}^\\infty 1/(j + k)^3.\n\nChanging variables to n = j + k. For each n ≥ 2, the number of pairs (j, k) with j + k = n is n - 1. Therefore, sum_{n=2}^\\infty (n - 1)/n^3. Then splitting into 1/n² - 1/n³. Then sum from n=2 to ∞ of 1/n² is p - 1, same with sum 1/n³ is q - 1. Therefore, total is (p - 1) - (q - 1) = p - q.
<omitted>
Yes, that's straightforward. Therefore, this seems to confirm again that the answer is p - q. \n\nTherefore, I think my initial conclusion is correct. Therefore, the answer is $\\boxed{p - q}$.\n\n**Final Answer**\n\\boxed{p - q}\n

</think>

\n\nTo evaluate the double sum\n\n$$\n\\sum_{j = 1}^\\infty \\sum_{k = 1}^\\infty \\frac{1}{(j + k)^3},
<omitted>
Substituting back, we have:\n\n$$\n\\sum_{j=1}^\\infty \\sum_{k=1}^\\infty \\frac{1}{(j + k)^3} = (p - 1) - (q - 1) = p - q.\n$$\n\n---\n\n### Final Answer\n\n$$\n\\boxed{p - q}\n$$

Token number: 5840

Figure 8: Comparison of vanilla reasoning and SABER on a MATH-500 problem. SABER reduces token consumption from 5840 to 1626 while maintaining correctness.

Question: In how many ways can 7 people sit around a round table if no two of the 3 people Pierre, Rosa, and Thomas can sit next to each other? (Seating arrangements which are rotations of each other are treated as the same.)

Answer:

SABER

<think>\nOkay, so I need to figure out how many ways 7 people can sit around a round table, but with the condition that Pierre, Rosa, and Thomas (let's call them P, R, T for short) can't sit next to each other.
<omitted>
So the number of ways would be C(4,3) * 3! **?\n\nWait**

SC: 0.33 CS: 0.91 RSS : 0.73

, but hold on. Let me think again. If we fix the positions of the other 4 people, then there are 4 gaps. To place P, R, T into different gaps, we need to choose 3 gaps out of 4, and then arrange P, R, T in those gaps.
<omitted>
but since we fixed one person's position to account for circular permutations?**\n\nWait**

SC: 0.60 CS: 0.95 RSS : 0.84

, maybe I need to adjust for circular permutations. Let me clarify.\n\nFirst, when arranging people around a circular table, fixing one person's position is a common technique to convert it into a linear permutation problem. So if I fix one person, say, Pierre, then arrange the rest.
<omitted>
So total number of ways would be 6 * 4 * 6 = 144. But wait, is this correct?**\n\nWait**

SC: 1.0 CS: 0.99 RSS : 0.99>0.95

</think>

\n\nTo solve the problem of seating 7 people around a round table such that **Pierre (P), Rosa (R), and Thomas (T)** are **not adjacent to each other**, we proceed step by step using the **gap method** in circular permutations.
<omitted>
\\times \\binom{4}{3} \\times 3! = 6 \\times 4 \\times 6 = 144\n$$\n\n---\n\n### ✅ Final Answer\n\n$$\n\\boxed{144}\n$$

Token number: 1664

Vanilla

<think>\nOkay, so I need to figure out how many ways 7 people can sit around a round table, but with the condition that Pierre, Rosa, and Thomas (let me call them P, R, T for short) can't sit next to each other. And rotations are considered the same, so it's a circular permutation problem with restrictions. \n\nFirst, I remember that for circular permutations, the number of ways to arrange n people is (n-1)! because rotations are considered the same. So normally, without any restrictions, 7 people would have 6! = 720 ways. But here, we have restrictions on P, R, and T. \n\nThe problem says no two of them can sit next to each other. So, I need to subtract the arrangements where at least two of them are sitting together. **But wait,** inclusion-exclusion might come into play here. Let me think.\n\nAlternatively, maybe it's easier to first arrange the other people and then place P, R, T in the gaps?
<omitted>
So the number of ways would be C(4,3) * 3! = 4 * 6 = 24. \n\nTherefore, total number of arrangements would be 3! * 4 * 3! = 6 * 24 = 144.
\n\nWait, but hold on. Let me check again. \n\nFirst, arranging the 4 people around the table: (4-1)! = 6. Then, placing the 3 people into the gaps: choosing 3 gaps out of 4, which is C(4,3) = 4, and then arranging P, R, T in those gaps, which is 3! = 6. So total is 6 * 4 * 6 = 144.
\n\nBut wait, is that correct? Let me think again.
<omitted>
So the number of entities is 5: the trio and the other 4 people. So circular arrangements: (5-1)! = 4! = 24. Then, the trio can be arranged in 3! ways. So |A ∩ B| = 3! * 4! = 6 * 24 = 144. **\n\nWait, but hold on. Wait,** if we have three people together, how do we count? Let me think.
<omitted>
\n\nAs calculated, this is 2 * 5! = 240. \n\nBut if we fix P's position, how many arrangements have P and R together? \n\nWith P fixed, R can be in seat2 or seat7 (adjacent to P). Then, the rest can be arranged in 5! ways? No, with P fixed, the remaining 6 seats are arranged, but if R is fixed next to P, then the remaining 5 people can be arranged in 5! ways. **\n\nWait**, but with

Endless exhaustive search

Token number: 16384

Figure 9: A case on MATH-500 where vanilla reasoning exceeds the context limit due to endless reflection, while SABER exits early and produces the correct answer.

Question: James decided to walk to the store. When he got halfway there he realized he forgot something at home and had to walk back. If his home is 4 miles from the store and he walks 4 miles per hour how long did it take him to reach the store?

Answer:

SABER

<think>\nOkay, let's see. James is walking to the store, which is 4 miles away from home. But when he gets halfway, he realizes he forgot something and has to walk back. Then he probably goes back to home and then goes back to the store again?
<omitted>
\n\nFirst, let me break down the problem. The total distance from home to store is 4 miles. So halfway there would be 2 miles. So James walks 2 miles to the halfway point. Then he realizes he forgot something, so he has to walk back home. That would be another 2 miles. So from the halfway point back home is 2 miles. Then, after getting back home, he needs to walk to the store again, which is 4 miles. So total distance he walked is 2 miles (halfway) + 2 miles (back home) + 4 miles (from home to store) = 8 miles total?**\n\nWait**

SC: 1.0 CS: 0.91 RSS : 0.97>0.95

</think>

\n\nTo determine how long it took James to reach the store, we need to calculate the **total distance** he walked and then use his **walking speed** to find the **total time**.\n\n---\n\n### Step 1: Understand the distances\n\n- The total distance from **home to store** is **4 miles**.\n- James walks **halfway** to the store: \n $$\n \\frac{4}{2} = 2 \\text{ miles}\n $$\n- He then **walks back home** from the halfway point: \n $$\n 2 \\text{ miles}\n $$\n- After returning home, he **walks the full 4 miles** to the store again.\n\n---\n\n### Step 2: Calculate total distance walked\n\n$$\n\\text{Total distance} = 2 \\text{ (to halfway)} + 2 \\text{ (back home)} + 4 \\text{ (to store)} = 8 \\text{ miles}\n$$\n\n---\n\n### Step 3: Use walking speed to find time\n\nJames walks at a speed of **4 miles per hour**.\n\n$$\n\\text{Time} = \\frac{\\text{Distance}}{\\text{Speed}} = \\frac{8 \\text{ miles}}{4 \\text{ miles per hour}} = 2 \\text{ hours}\n$$\n\n---\n\n### Final Answer:\n\n$$\n\\boxed{2} \\text{ hours}\n$$

Token number: 532

Vanilla

<think>\nOkay, let's see. James is walking to the store, right? His home is 4 miles from the store. So the total distance from home to store is 4 miles. But he got halfway there and then realized he forgot something, so he had to walk back.
<omitted>
\n\nWait, maybe the problem is that he walked to the store, but when he got halfway there, he realized he forgot something, so he walked back home, and then walked to the store again. So total distance is 2 miles to halfway, 2 miles back, and then 4 miles to the store. Total distance is 8 miles. Time is 8 / 4 = 2 hours.

Correct Answer

<omitted>
Wait, but he started at home. So halfway is 2 miles. So he walked 2 miles, then turned back. So total time is 1 hour.
<omitted>
Alternatively, maybe the question is asking for the total time he spent walking, which would be 1 hour.
<omitted>
Or maybe the problem is asking for the total time he spent walking, which would be the time to walk to the halfway point and back. Which is 2 miles / 4 mph = 0.5 hours, then 2 miles back / 4 mph = 0.5 hours, total 1 hour.
<omitted>
Or maybe the problem is intended to have him go back to the store, in which case it's 1 hour. Given that, I think the answer is 1 hour.\n\n**Final Answer**\nIt took James \\boxed{1} hour to reach the store.\n

Wrong Answer

</think>

\n\nTo determine how long it took James to reach the store, we need to carefully analyze the scenario and the distances involved.
<omitted>
Final Calculation\n\n- Total distance walked = 4 miles \n- Walking speed = 4 miles/hour \n- Total time = 4 miles ÷ 4 mph = **1 hour**\n\n---\n\n### **Final Answer**\n\n$$\n\\boxed{1}\n$$

Token number: 4163

Figure 10: A GSM8K case where vanilla reasoning degrades the correct answer through over-reflection, while SABER preserves it by timely early exit.