

STARS: State-Aware Reasoning Stopping via Hidden-State Spectral Analysis

Aaron T. Cheng¹, Junda Wu², Sheldon Yu³, Rui Wang², Ryan A. Rossi²,
Sizhe Zhou⁴, Jiawei Han⁴, Julian McAuley³, Lina Yao¹, Tong Yu^{2*}

¹University of New South Wales ²Adobe Research

³University of California San Diego ⁴University of Illinois Urbana-Champaign
{aaron.cheng, lina.yao}@unsw.edu.au {jundaw, ruiwan, ryrossi, tyu}@adobe.com
{ziy040, jmcauley}@ucsd.edu {sizhez, hanj}@illinois.edu

Abstract

Chain-of-thought (CoT) reasoning in large language models often produces redundant steps that increase generation cost and obscure when reasoning has effectively completed. Existing stopping methods rely on output-level or token-level signals, overlooking internal representation dynamics that reflect reasoning progress. We propose STARS (STate-Aware Reasoning Stopping), a framework that leverages spectral analysis of hidden-state representations to detect transitions in reasoning phases. Our key insight is that structural changes in the representation space capture shifts in reasoning states, enabling early and reliable stopping decisions. To address representation instability, where semantically similar steps can induce different hidden states, we focus on spectral structure rather than raw activations, improving robustness to context-dependent drift. Experiments on four mathematical and logical reasoning benchmarks show that STARS reduces generation cost by over 45% under fixed token budgets while improving accuracy over standard CoT baselines.

1 Introduction

We study the problem of identifying and eliminating redundant steps in chain-of-thought (CoT) reasoning produced by large language models. Redundant reasoning increases inference cost, reduces interpretability, and can degrade performance through error accumulation, where later steps deviate from the correct reasoning trajectory. In practice, such redundancy often appears as repeated subgoals or unnecessary recalculations that extend reasoning without improving the final answer. Prior work shows that these effects become more pronounced as reasoning chains grow longer (Wei et al. 2022; Wang et al. 2023).

Existing approaches typically rely on output-based heuristics such as answer convergence (Liu and Wang 2025), confidence (Yang et al. 2026), or entropy signals (Wang et al. 2026) to determine when to terminate reasoning. However, these methods operate at the trajectory level and do not explicitly capture internal representation dynamics, limiting their ability to detect semantically redundant steps.

To address this limitation, we propose an early exit approach based on hidden-state evolution, tracking reasoning progress through changes in the eigenvalue spectrum of hidden-state Gram matrices (Yu et al. 2025). This enables a state-aware stopping mechanism that detects structural transitions in reasoning trajectories. Specifically, we derive stop triggers from spectral transition dynamics and identify redundancy when consecutive steps stabilize in spectral space. We combine three complementary signals, confidence based, temporal persistence-based, and cluster position based criteria to robustly detect redundant reasoning patterns.

The key intuition is that reasoning progresses as a structured trajectory in representation space, where meaningful transitions correspond to global spectral shifts rather than token-level variation. However, hidden states are context-dependent, and even semantically similar steps may induce different representations due to accumulated context and positional effects (van Aken et al., 2019; Sinha et al., 2022), making naive similarity measures unreliable.

To address these challenges, our spectral approach identifies redundancy at the level of hidden-state dynamics. We summarize our contributions as follows:

- We show that CoT reasoning often contains redundant steps with limited internal progress despite textual coherence, motivating representation-level stopping signals.

*Corresponding author.

- We propose STARS, a state-aware early-stopping framework that detects latent state transitions and integrates confidence-based, persistence-based, and location-based signals to terminate redundant reasoning.
- Extensive experiments show that our method reduces generation cost by 45% or more across benchmarks while maintaining or improving accuracy over standard CoT.

2 Related Work

2.1 Early Stopping for Reasoning

To reduce the inefficiency of long chain-of-thought reasoning, recent work has explored adaptive inference strategies that dynamically control reasoning length during generation. Budget-based reasoning methods constrain the amount of reasoning computation by estimating or allocating token budgets based on problem complexity (Han et al. 2025; Li et al. 2025; Wen et al. 2026), enabling models to compress reasoning traces while maintaining accuracy. Confidence-based early-exit approaches terminate reasoning once the model exhibits high confidence in a candidate answer, often using entropy, probability calibration, or uncertainty signals (Valade 2026; Yoo et al. 2026; Yang et al. 2026). Trajectory-based stopping methods detect convergence in reasoning trajectories, for example by monitoring reasoning transitions or answer convergence during generation (Liu and Wang 2025). While these approaches demonstrate that adaptive stopping can significantly reduce token usage and latency, most rely on heuristics rather than exploiting structural signals in the reasoning representation itself. In contrast, we base stopping decisions largely on structural transitions in hidden-state dynamics.

2.2 Monitoring Reasoning Trajectories and State Dynamics

While chain-of-thought (CoT) prompting substantially improves reasoning performance in large language models (LLMs) (Wei et al. 2022; Kojima et al. 2022), recent studies have observed that CoT reasoning often contains redundant or unnecessarily long reasoning traces, increasing inference cost and sometimes degrading accuracy through error accumulation (Guan et al. 2026; Hong et al. 2026; Srivastava et al. 2026). Consequently, several works have analyzed intermediate reasoning trajectories to diagnose issues such as hallucinated

steps, logical inconsistencies, and unstable reasoning processes (Xue et al. 2023; Lightman et al. 2024; Uesato et al. 2023; Cobbe et al. 2021).

Beyond token-level analysis, a growing line of research investigates reasoning through hidden-state dynamics. Wu et al. (2026) show that latent trajectories in representation space exhibit structured transitions corresponding to different reasoning stages, while Yu et al. (2025) demonstrate that hidden-state clusters and state shifts align with key decision points during CoT reasoning. Related studies further reveal that internal representation trajectories correlate with task progress, answer convergence, and reasoning failures (Williams et al. 2026). Together, these findings suggest that reasoning progression can be characterized through latent state evolution rather than surface tokens alone.

Unlike prior work, which primarily uses trajectory analysis to interpret or diagnose reasoning behavior, we leverage hidden-state dynamics to address early stopping, developing state-aware stopping signals that identify when reasoning has sufficiently converged.

3 Preliminaries

3.1 Gram-Spectral Representation of Reasoning

Consider a Chain-of-Thought (CoT) trajectory $s_{1:T}$, where each reasoning step s_t consists of m_t tokens. Let $H_t \in \mathbb{R}^{m_t \times d}$ denote the matrix of last-layer hidden states for all m_t tokens in step t , where d is the hidden dimension. We define the step-wise Gram matrix:

$$\tilde{G}_t = \frac{1}{m_t} H_t^\top H_t \in \mathbb{R}^{d \times d}$$

and accumulate it over time:

$$G_t = G_{t-1} + \tilde{G}_t, \quad G_1 = \tilde{G}_1$$

where $G_t \in \mathbb{R}^{d \times d}$ encodes the aggregated semantic structure across all steps 1 through t .

Let $\lambda_{t,1} \geq \dots \geq \lambda_{t,r}$ be the top- r eigenvalues of G_t . The spectral embedding $\mathbf{e}_t = (\lambda_{t,1}, \dots, \lambda_{t,r})$ provides a low-dimensional representation of the accumulated reasoning geometry up to step t . Unlike first-order statistics such as mean pooling, which aggregate token representations directly, the Gram spectrum captures second-order geometry across all token representations within a step and

is invariant to orthogonal transformations of the hidden space, capturing subspace correlations that first-order statistics do not retain. We compare against a mean-pooled hidden-state variant empirically in Appendix G.

3.2 State Transition Modeling

We cluster spectral embeddings using k-means into K clusters, yielding discrete states $c_t \in \{1, \dots, K\}$. A trajectory is represented as $c = (c_1, \dots, c_T)$.

Following prior work (Yu et al. 2025), we model transitions between clusters as a Markov chain and estimate a transition matrix $P \in \mathbb{R}^{K \times K}$, where

$$P_{i,j} = P(c_{t+1} = j \mid c_t = i) = \frac{C_{i,j}}{\sum_{j'} C_{i,j'}}$$

and $C_{i,j}$ denotes the number of observed transitions from state i to state j . This abstraction captures the sequential structure of reasoning trajectories through transitions between discrete reasoning states and forms the basis for the stopping signals introduced in the following sections.

4 Problem Formulation

We consider the problem of early termination in multi-step LLM reasoning. Given an input query q , a large language model generates a sequence of reasoning steps $s_1, s_2, \dots, s_t$, where each step represents an intermediate reasoning state in the solution process.

Let $H_t \in \mathbb{R}^{m_t \times d}$ denote the hidden state at reasoning step t , obtained from the model’s last-layer latent states. The spectral embeddings of these latent states are discretized into clusters $k \in \{1, \dots, K\}$, forming a trajectory of reasoning stages.

The objective is to design a stopping policy π that determines at each step whether to continue reasoning or terminate and output the final answer. Formally, the policy outputs a binary decision $\pi_t \in \{\text{continue}, \text{stop}\}$.

Let τ^* denote the earliest step at which the reasoning process has effectively reached completion. The goal is to detect a stopping time τ that satisfies two objectives:

1. **Correctness:** The final answer produced at step τ matches the correct solution.

2. **Efficiency:** The stopping time τ occurs as early as possible, minimizing unnecessary reasoning steps.

A key challenge is that the utility of additional reasoning is not uniform across reasoning stages. Our empirical analysis in Section 7.1 shows that early reasoning stages contribute substantial accuracy improvements, while later stages exhibit diminishing returns. Therefore, stopping decisions should depend not only on the current answer confidence, but also on the estimated remaining reasoning utility associated with the current latent state.

5 Methodology

Based on the above insight, our method explicitly models both answer confidence and stage-dependent reasoning utility as complementary signals for determining the stopping decision.

Preprocessing phase. We collect a reference corpus $\mathcal{D} = \{(\tau^{(i)}, a^{*(i)})\}_{i=1}^N$ of complete reasoning trajectories $\tau^{(i)}$ with known answers $a^{*(i)}$, using 100 training examples from GSM8K. Each step is embedded using the Gram-spectral representation (Section 3.1), producing $e_t^{(i)} \in \mathbb{R}^d$. We then apply the k-means clustering to the set of all step embeddings across all trajectories in $\mathcal{D}$, obtaining K cluster centroids $\{\mu_k\}_{k=1}^K$ and a cluster assignment function $\phi : \mathbb{R}^d \rightarrow [K]$. The clusters are then ordered according to the average step index within each cluster (details in Algorithm 3).

Inference phase. At inference time, after generating step s_t , we compute its embedding e_t and assign it to cluster $c_t = \phi(e_t)$. We view early stopping as a sequential decision process over reasoning trajectories. Our method decomposes stopping signals along three complementary axes: (1) confidence in the current state (Wald and Wolfowitz, 1948), (2) temporal persistence within a region (Larralde et al., 2006), and (3) spatial location in the learned state space (Norris, 1998). Accordingly, our method decomposes stopping signals along three complementary axes in Section 5.1, 5.2 and 5.3 respectively. Together, these criteria form a coordinated taxonomy of stopping behaviors that captures epistemic, temporal, and spatial signals of reasoning convergence. Figure 1 illustrates the inference workflow, and Algorithm 1 details the generation procedure.

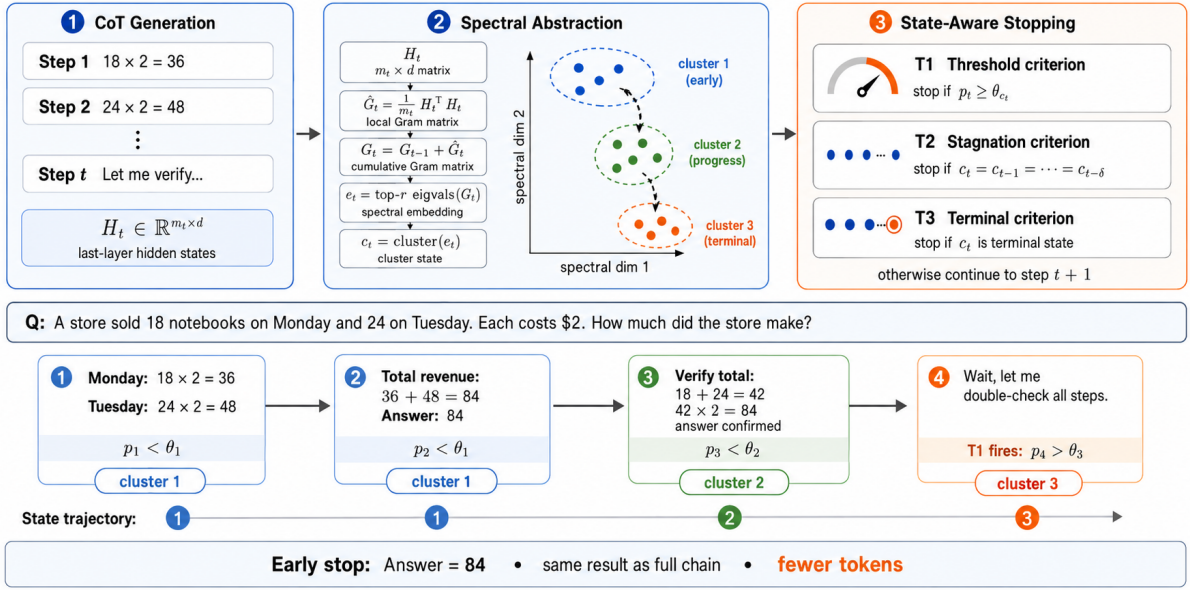


Figure 1: The workflow illustrates how the model tracks latent reasoning states during a long chain-of-thought, and applies a state-aware trigger to stop early while producing the correct answer with fewer tokens.

5.1 Threshold Criterion

Our truncation analysis in Figure 3 shows that early-stage reasoning is significantly more sensitive to premature stopping than later-stage reasoning. This indicates that the marginal utility of additional reasoning decreases as the model progresses through intermediate reasoning states.

To capture this effect, we define a cluster-conditioned stopping policy, where each reasoning state is associated with a cluster-specific threshold that governs termination based on an answer confidence score.

Let $p_t \in [0, 1]$ denote the answer confidence score produced by the LLM judge at step t , representing the estimated probability that the current partial reasoning is sufficient to produce the correct answer. The model stops when

$$p_t \geq \theta_{c_t}$$

where θ_{c_t} is a cluster-specific stopping threshold.

This formulation follows a standard risk-based decision framework, as detailed in Appendix A. It motivates state-dependent thresholds, which we estimate empirically from held-out calibration data in Appendix B.2.

The resulting optimal threshold takes the form:

$$\theta_k^* = \frac{L_I(c_t)}{L_I(c_t) + L_{II}(c_t)}. \quad (1)$$

where $L_I(c_t)$ denotes the premature stopping loss at cluster c_t , and $L_{II}(c_t)$ denotes the continuation loss at cluster c_t . This formulation implies that optimal stopping thresholds are higher in early reasoning clusters, where the expected loss from premature stopping is large, and decrease as reasoning progresses and additional computation yields diminishing returns.

Therefore, early reasoning states adopt more conservative stopping criteria, while later states permit lower confidence requirements. STARS operationalizes this principle by estimating cluster-specific thresholds from data rather than explicitly modeling the underlying loss terms.

5.2 Stagnation Criterion

In addition to detecting termination at cluster boundaries, we introduce a complementary signal that identifies stagnation, i.e., when the reasoning trajectory remains within the same cluster for an extended period. Let c_t denote the cluster assignment at step t . A stagnation event is defined by a consecutive dwell of length δ :

$$c_t = c_{t-1} = \dots = c_{t-\delta}.$$

We interpret cluster transitions as a first-order Markov process over discrete states (Yu et al., 2025). Under this assumption, the probability of remaining in the same cluster is $P(c_{t+1} = c_t)$, and the dwell time T_c follows a geometric distribution

Algorithm 1 State-Aware Reasoning Stopping

Require: Question q , reference cluster centroids $\{\mu_k\}$, thresholds $\{\theta_k^*\}$, judge J , token budget B

Ensure: Final reasoning trace a

```
1: Initialize  $\tau_0 \leftarrow \emptyset, t \leftarrow 1$ 
2: while  $\text{Len}(\tau_{t-1}) < B$  do
3:   Generate step  $s_t \sim \text{LLM}(\cdot \mid q, \tau_{t-1})$ 
4:    $\tau_t \leftarrow (\tau_{t-1}, s_t)$ 
5:   Compute embedding  $e_t \leftarrow \text{Encoder}(s_t)$ 
6:   Assign cluster  $c_t \leftarrow \arg \min_k \|e_t - \mu_k\|_2$ 
7:   Compute judge confidence  $p_t \leftarrow J(\tau_t)$ 
8:   if  $p_t \geq \theta_{c_t}$  then
9:     return  $a(\tau_t)$   $\triangleright$  refer to 5.1
10:  else if  $c_t = c_{t-1} = \dots = c_{t-\delta}$  then
11:    return  $a(\tau_t)$   $\triangleright$  refer to 5.2
12:  else if  $c_t = K$  then
13:    return  $a(\tau_t)$   $\triangleright$  refer to 5.3
14:  end if
15:   $t \leftarrow t + 1$ 
16: end while
17: return  $a(\tau_T)$   $\triangleright$  Max-length reasoning trace
```

with expected value

$$\mathbb{E}[T_c] = \frac{1}{1 - P(c_{t+1} = c_t)}.$$

Long dwell times indicate repeated self-transitions within a cluster and suggest that the trajectory is no longer making meaningful progress across reasoning states. We therefore terminate generation when the dwell time exceeds this threshold, using it as a safeguard against redundant reasoning loops and latent stagnation in the reasoning process.

5.3 Terminal Criterion

We introduce a third stopping signal based on terminal clusters in the abstract state space. Empirically, certain clusters correspond to the final stage of reasoning where the model produces answer-related statements. From the perspective of the Markov chain formulation, such clusters behave similarly to absorbing states, satisfying

$$P(c_{t+1} = K \mid c_t = K) = 1$$

meaning that once the trajectory enters this state K it tends to remain there. Motivated by this observation, we terminate reasoning immediately when a step is assigned to a terminal cluster.

Terminal clusters are identified by computing the average step index of all reasoning steps assigned to each cluster (details in Algorithm 3). The cluster with the largest average step index is designated as the terminal cluster. Upon entry to this cluster, we apply immediate stopping, which serves as a natural heuristic for completion. While this choice is not formally guaranteed, as downstream errors may still occur after entry, it provides a simple and effective rule consistent with the intuition that the terminal cluster represents the final stage of reasoning.

6 Experiments

6.1 Implementation

We evaluate our method on four reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), AQuA (Ling et al., 2017), and LogiQA (Liu et al., 2020), covering both open-ended mathematical reasoning and multiple-choice logical reasoning tasks.

All methods are evaluated under a unified token budget of 4096 tokens to ensure a fair comparison of accuracy–efficiency trade-offs under consistent inference constraints. Note that prior methods were originally designed for different regimes, which may lead to performance deviations from their reported results.

We compare against three representative early-stopping baselines: DEER (Yang et al. 2026) (confidence-based), EAT (Wang et al. 2026) (uncertainty/entropy-based), and TALE (Han et al. 2025) (heuristic budget-based), covering the major paradigms of existing CoT termination strategies.

The preprocessing phase, including spectral embedding construction and k-means clustering, completes in under 3 minutes and is performed once offline prior to inference.

6.2 Metrics

To evaluate the effectiveness of our method, we choose metrics that measure both answer quality and inference cost. **Accuracy (ACC)** quantifies how often the model’s output matches the correct solution, using a standard indicator function to compare predicted vs. true answers. **Generation Length (TOK)** reflects the average number of tokens produced per input, serving as a proxy for inference cost. Both evaluations follow the zero-shot

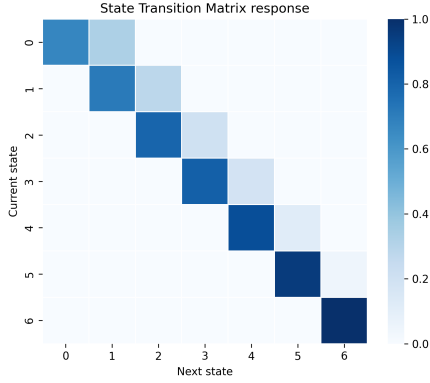


Figure 2: Transition heatmaps for Qwen3-14B. Color intensity represents transition probability. Later-stage clusters exhibit stronger self-loop behavior.

chain-of-thought setup, and averages are computed over the full test sets of the respective benchmarks.

6.3 Model Backbones

We follow DEER (Yang et al. 2026) in selecting the Qwen series as backbone models, using Qwen3-8B, Qwen3-14B, and QwQ-32B. This choice enables direct comparison with prior early-stopping methods while covering a range of model scales and reasoning capabilities. Besides, we add Phi-4-mini-reasoning in Appendix C to evaluate the robustness of our approach across different model families.

7 Results

In this section, we present results to answer three research questions below.

- **RQ1:** How do latent state transition dynamics evolve throughout Chain-of-Thought reasoning, and do later reasoning steps exhibit diminishing state progression that motivates state-aware early stopping?
- **RQ2:** Can the proposed method detect reliable internal signals that indicate when further Chain-of-Thought reasoning becomes unnecessary, enabling effective early stopping without degrading performance?
- **RQ3:** Why can state-aware reasoning stopping improve accuracy under finite generation budgets?

7.1 RQ 1 Results

We estimate a cluster transition matrix from observed reasoning trajectories, where each entry represents the probability of transitioning between latent reasoning states.

Across 100 GSM8K samples and all backbone models, the resulting transition patterns are highly consistent (Figure 2, Figure 6). We observe a structured progression in latent reasoning dynamics: early clusters more frequently transition to different subsequent states, while later clusters increasingly exhibit dominant self-transitions. This indicates a shift from exploratory reasoning dynamics to stable latent regimes, where internal state progression slows despite continued token generation. These findings suggest that later-stage reasoning may contribute limited additional state evolution, motivating state-aware early stopping based on representation dynamics rather than token length alone.

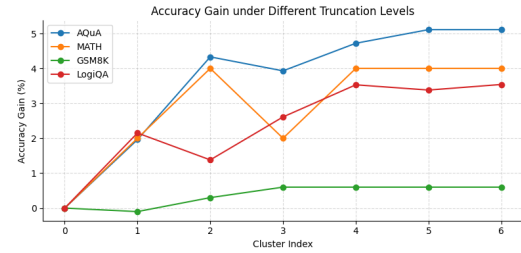


Figure 3: Accuracy gain across four datasets under different truncation clusters using Qwen3-14B. Performance improves rapidly in early clusters and saturates in later stages.

To assess the functional role of later reasoning steps, we perform a truncation experiment in which generation is stopped at a specified latent cluster and the model is prompted to produce the final answer. As shown in Figure 3, accuracy increases sharply in early clusters but quickly saturates. For instance, on AQUA, moving from cluster 0 to 2 yields a notable gain ($\approx 4\%$), whereas extending reasoning to clusters 4–6 produces marginal improvements ($< 1\%$). Beyond this point, additional reasoning steps increase token cost without meaningful accuracy gains.

Together with the observed transition dynamics, these results indicate that later reasoning stages are often characterized by latent stagnation: the model continues generating tokens while internal state evolution has largely stabilized. This mismatch between textual continuation and representation-level progress highlights the limitations of token-based stopping criteria and supports early stopping mechanisms grounded in latent dynamics.

Method	MATH						LOGIC		Overall	
	GSM8K		MATH-500		AQuA		LogiQA			
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓
<i>Qwen3-8B</i>										
Standard CoT	93.7	1294	70.0	2775	78.7	2283	58.0	2555	79.4	1959
EAT	94.3	902	82.4	1604	83.9	1350	66.4	1799	84.5	1287
TALE	95.7	627	82.4	1817	85.0	1380	65.4	1574	85.0	1142
DEER	94.5	760	85.0	1539	89.0	1146	66.4	1466	85.5	1108
STARS	94.8	516	84.2	1065	84.3	946	68.2	1314	85.5	848
<i>Qwen3-14B</i>										
Standard CoT	93.3	1553	67.6	3043	81.9	2268	63.1	2494	80.3	2118
EAT	93.5	1279	73.8	2154	83.7	1599	66.4	1887	82.5	1615
TALE	95.4	466	83.0	1769	85.8	1007	71.1	1274	86.4	948
DEER	95.2	660	87.2	1609	85.0	1066	69.8	1446	86.7	1060
STARS	95.9	579	88.0	1093	86.6	1056	71.9	1464	87.8	929
<i>QwQ-32B</i>										
Standard CoT	94.8	1355	72.2	2788	80.7	2127	64.2	2063	82.0	1859
EAT	94.8	1005	72.8	2566	81.5	1666	65.9	1419	82.6	1452
TALE	95.6	633	79.0	1865	85.4	1408	68.8	1449	85.2	1126
DEER	96.3	873	78.2	2241	85.4	1447	74.2	1565	86.7	1343
STARS	95.5	616	82.8	1423	90.2	1331	74.2	1352	87.6	1007

Table 1: Experimental results across three reasoning models over four datasets. “Acc” denotes accuracy and “Tok” denotes token count. ↑ indicates that higher values are better, while ↓ indicates that lower values are better. Best results are in bold.

7.2 RQ 2 Results

Table 1 reports the performance of different early stopping methods across three reasoning models and benchmarks. We evaluate both reasoning quality and efficiency using accuracy (Acc) and generated token count (Tok), and compute overall metrics as a sample-size-weighted average across datasets:

$$\text{Overall}(m) = \frac{\sum_{d=1}^D N_d m_d}{\sum_{d=1}^D N_d},$$

where m_d is the metric on dataset d and N_d is the number of evaluation instances.

Overall, STARS consistently improves reasoning accuracy while substantially reducing generation cost. Across all models, it achieves the best or competitive overall accuracy with the lowest token usage, indicating that the proposed stopping criteria effectively remove redundant reasoning without harming performance.

In terms of efficiency, STARS reduces token consumption by 45% or more across all three backbones compared to standard CoT, while maintaining or improving accuracy. For example, on QwQ-32B, token usage drops from 1859 to 1007 (45.8%), while accuracy improves from 82.0% to 87.6%. Similar trends are observed across Qwen3-8B and Qwen3-14B, where STARS consistently achieves strong accuracy–efficiency trade-offs.

Compared with existing early stopping methods, STARS achieves a better accuracy–efficiency trade-off than EAT, TALE, and DEER, consistently reducing token usage while maintaining or improving accuracy across benchmarks. Gains are particularly pronounced on logical reasoning tasks such as LogiQA, suggesting that mitigating late-stage reasoning drift is especially beneficial for maintaining logical consistency in long reasoning chains.

Like many adaptive reasoning approaches, STARS introduces an auxiliary decision process for deter-

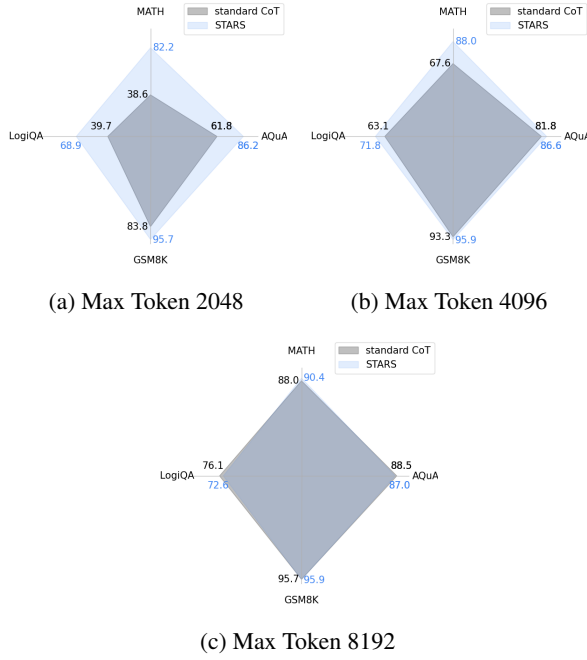


Figure 4: Accuracy comparison between standard CoT and STARS under different token budgets on Qwen3-14B over four datasets.

mining when reasoning should terminate. Such overhead is common in prior work, including methods that require additional model-generated estimates or judgments before producing a final answer (Han et al., 2025). To assess the practical impact, we measured wall-clock runtime on AQuA with Qwen3-14B without vLLM acceleration. STARS completed in 4.23 hours versus 6.67 hours for standard CoT, suggesting that the reduction in reasoning generation can compensate for the cost of the stopping mechanism. The precise runtime trade-off may depend on the task and deployment infrastructure, but our results indicate that the overhead is modest in practice.

7.3 RQ 3 Results

Figure 4 shows the accuracy gap between standard CoT and STARS diminishes as the token budget increases. When the token budget is limited, for example, 2048 in Figure 4 (a) or 4096 in Figure 4 (b), STARS improves accuracy by terminating generation once sufficient reasoning evidence has been accumulated. This reduces the dependence on long reasoning trajectories and allows the model to reach near-optimal performance within a substantially smaller token budget. In contrast, standard CoT often requires much larger budgets to complete its reasoning process, making it more susceptible to

Max Tokens	GSM8K	MATH500	AQuA	LogiQA
2048	21.38	73.00	47.24	54.38
4096	5.99	39.40	18.90	23.96
8192	0.83	11.60	6.69	5.22

Table 2: Percentage of responses that hit the maximum token limit on Qwen3-14B.

truncation when the available budget is limited.

Truncation under limited token budgets can affect the comparison between STARS and standard CoT. To quantify this effect, we measure the fraction of standard-CoT trajectories that reach the maximum generation limit before producing a final answer. For Qwen3-14B, the truncation rates are shown in Table 2.

These results show that standard CoT is substantially affected by the token budget, particularly on more complex reasoning benchmarks. For example, under a 2048-token budget, 73.0% of MATH500 and 54.4% of LogiQA trajectories are truncated before completion. Increasing the budget to 8192 tokens reduces these rates to 11.6% and 5.2%, respectively.

This observation is consistent with our accuracy-budget analysis in Figure 4. As the token budget increases, the accuracy gap between standard CoT and STARS decreases. STARS can terminate generation once sufficient reasoning evidence has been accumulated, whereas standard CoT can recover more performance when given substantially larger generation budgets. Thus, the advantage of STARS is most pronounced under constrained token budgets, where standard CoT is more frequently truncated before reaching a final answer.

8 Ablation

We perform an ablation study by removing each trigger individually while keeping all other components unchanged. Figure 5 shows that the triggers play complementary roles in balancing accuracy and reasoning length. Removing the stagnation trigger leads to longer reasoning and slightly lower accuracy, while removing the confidence threshold results in substantially more token usage. Removing the terminal trigger has only a marginal effect, with nearly unchanged accuracy and reasoning length. This suggests that the other triggers can still terminate most reasoning processes effectively, while the terminal trigger provides an additional

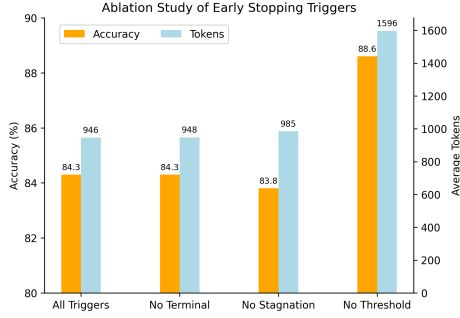


Figure 5: Ablation study on AQuA with Qwen3-8B, showing how different early stopping triggers affect accuracy and token usage.

stopping mechanism for a small subset of examples. Overall, the results demonstrate that the triggers serve complementary roles, with the confidence threshold trigger having the largest impact on reasoning efficiency.

9 Conclusion

We presented STARS, a state-aware early stopping framework that leverages latent-state transition dynamics to identify reasoning completion in chain-of-thought reasoning. By modeling reasoning trajectories as transitions between latent reasoning states and detecting both completion and stagnation patterns, STARS reduces reasoning length while maintaining strong performance across multiple benchmarks and model scales. Our results highlight the value of representation-level signals for understanding and controlling reasoning processes in large language models.

10 Limitations

Our framework focuses on reducing unnecessary reasoning once a valid solution trajectory has emerged. It does not resolve failure modes in which the model fundamentally cannot produce the correct answer for a task, nor cases where an incorrect reasoning trajectory converges to a stable latent state. More broadly, this reflects a limitation of current early-stopping benchmarks (Yang et al., 2026; Han et al., 2025; Wang et al., 2026), which typically assume the existence of a solvable reasoning trajectory and evaluate termination efficiency rather than reasoning capability itself.

11 Acknowledgments

This work is partially supported by NSF IIS-2432486. Jiawei Han’s research was supported

in part by the U.S. National Science Foundation under Awards 2505932, 2537827, and 2118329, and an Adobe gift fund.

During this work, we used AI assistants (including ChatGPT and Claude) for language editing, proofreading, and limited coding assistance. AI-generated suggestions were reviewed and verified by the authors before incorporation. The authors were responsible for all research decisions, experimental design, implementation, analysis, and conclusions reported in this paper.

References

- 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.
- Weixin Guan, Liang Li, Jiapeng Liu, Bing Li, Peng Fu, Chengyang Fang, Xiaoshuai Hao, Can Ma, and Weiping Wang. 2026. [Mitigating overthinking in large reasoning language models via reasoning path deviation monitoring](#). *Preprint*, arXiv:2603.14251.
- 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, Vienna, Austria. Association for Computational Linguistics.
- 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*.
- Anh Khoa Ngo Ho, Martin Chauvin, Simon Gosset, Philippe Cordier, and Boris Gamazaychikov. 2026. [Balancing sustainability and performance: The role of small-scale llms in agentic artificial intelligence systems](#). *Preprint*, arXiv:2601.19311.
- Jialiang Hong, Taihang Zhen, Kai Chen, Jiaheng Liu, Junlan Feng, Wenpeng Zhu, Jing Huo, Yang Gao, Depeng Wang, Haitao Wan, Xi Yang, Boyan Wang, Fanyu Meng, and Yuyao Zhang. 2026. [Reconsidering overthinking: Penalizing internal and external redundancy in cot reasoning](#). *Preprint*, arXiv:2508.02178.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). *Advances in Neural Information Processing Systems*.
- H Larralde, F Leyvraz, and D P Sanders. 2006. [Metastability in markov processes](#). *Journal of Statistical Mechanics: Theory and Experiment*, 2006(08):P08013–P08013.
- Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. 2025. [Selfbudgeter: Adaptive token allocation for efficient LLM reasoning](#).
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program induction by rationale generation: Learning to solve and explain algebraic word problems](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. [Logiqa: A challenge dataset for machine reading comprehension with logical reasoning](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.
- 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 17896–17907, Suzhou, China. Association for Computational Linguistics.
- J. R. Norris. 1998. *Markov Chains*. Cambridge University Press.
- Koustuv Sinha, Amirhossein Kazemnejad, Siva Reddy, Joelle Pineau, Dieuwke Hupkes, and Adina Williams. 2022. [The curious case of absolute position embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4449–4472, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Saurabh Srivastava, Janit Bidhan, Hao Yan, Abhishek Dey, Tanu Kansal, Paras Kath, Sina Mansouri, Mohit Marvania, Vamsi Shankar Simhadri, and Gaurav Singh. 2026. [Reasoning under constraint: How batch prompting suppresses overthinking in reasoning models](#). In *ICLR 2026 Workshop on Logical Reasoning of Large Language Models*.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2023. [Solving math word problems with process-based and outcome-based feedback](#).
- Florian Valade. 2026. [Accelerating large language model inference with self-supervised early exits](#). *Preprint*, arXiv:2407.21082.
- Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. [How does bert answer questions?: A layer-wise analysis of transformer representations](#). In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM ’19*, page 1823–1832. ACM.
- Abraham Wald and Jacob Wolfowitz. 1948. [Optimum character of the sequential probability ratio test](#). *Annals of Mathematical Statistics*, 19:326–339.
- 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 V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.

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*.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yifan Su, Yunxin Liu, Ya-Qin Zhang, and Yuanchun Li. 2026. [Budgetthinker: Empowering budget-aware LLM reasoning with control tokens](#).

Jonathan Williams, Olga Russakovsky, and Esin Tureci. 2026. [Prioritize the process, not just the outcome: Rewarding latent thought trajectories improves reasoning in looped language models](#). In *Forty-third International Conference on Machine Learning*.

Junda Wu, Yuxin Xiong, Xintong Li, Sheldon Yu, Zhengmian Hu, Tong Yu, Rui Wang, Xiang Chen, Jingbo Shang, and Julian McAuley. 2026. [CTRLS: Chain-of-thought reasoning via latent state transition](#). In *The 29th International Conference on Artificial Intelligence and Statistics*.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. [Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought](#). *Preprint*, arXiv:2305.11499.

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*.

Sangmin Yoo, Srikanth Malla, Chiho Choi, Wei D. Lu, and Joon Hee Choi. 2026. [Adept: Adaptive dynamic early-exit process for transformers](#). *Preprint*, arXiv:2601.03700.

Sheldon Yu, Yuxin Xiong, Junda Wu, Xintong Li, Tong Yu, Xiang Chen, Ritwik Sinha, Jingbo Shang, and Julian McAuley. 2025. [Explainable chain-of-thought reasoning: An empirical analysis on state-aware reasoning dynamics](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16660–16667, Suzhou, China. Association for Computational Linguistics.

A Derivation of the State-Aware Thresholding

In this section, we provide the derivation and interpretation of the state-aware stopping threshold introduced in Section 5.1.

A.1 Loss Definition

We define two types of decision errors under cluster c_t :

- **Premature stopping:** terminating reasoning before sufficient information is obtained.
- **Unnecessary continuation:** continuing reasoning after sufficient information is already available.

The corresponding losses are defined as:

$$L_I(c_t) = \lambda(A(K) - A(c_t)), \quad (2)$$

$$L_{II}(c_t) = \alpha, \quad (3)$$

where $A(c_t)$ denotes truncation accuracy at cluster c_t and $A(K)$ denotes final accuracy. The first term measures the expected utility lost due to premature stopping, which is larger in early clusters. The second term models a constant per-step computational cost.

A.2 Bayes Risk Formulation

At reasoning step t , let c_t denote the current cluster assignment and let $p_t \in [0, 1]$ denote the confidence score produced by the judge model. The stopping policy compares p_t against a cluster-specific threshold θ_k , and stop if $p_t \geq \theta_{c_t}$.

Two types of decision errors are possible:

- Premature stopping (Type I): stopping when the current reasoning trajectory is insufficient to produce the correct answer.
- Unnecessary continuation (Type II): continuing reasoning despite the current trajectory already being sufficient.

Let $L_I(c_t)$ and $L_{II}(c_t)$ denote the corresponding losses at cluster c_t . The expected Bayes risk under threshold θ is:

$$R(\theta | c_t) = L_I(c_t) \cdot P(p_t \geq \theta, a(\tau_t) \neq a^* | c_t) + L_{II}(c_t) \cdot P(p_t < \theta, a(\tau_t) = a^* | c_t).$$

Assuming the judge confidence is approximately calibrated, we interpret

$$P(a(\tau_t) = a^* | p_t) = p_t$$

which implies

$$P(a(\tau_t) \neq a^* | p_t) = 1 - p_t$$

Let $f(p | c_t)$ denote the density of judge confidence scores conditioned on cluster c_t . Substituting the calibrated probabilities into the Bayes risk yields:

$$R(\theta | c_t) = L_I(c_t) \int_{\theta}^1 (1 - p) f(p | c_t) dp + L_{II}(c_t) \int_0^{\theta} p f(p | c_t) dp$$

The first term corresponds to the expected loss from stopping incorrectly, while the second term captures the expected loss from unnecessary continuation.

A.3 Optimal Threshold Derivation

To derive the optimal threshold, we differentiate $R(\theta | c_t)$ with respect to θ . Applying Leibniz's rule gives:

$$\frac{\partial R}{\partial \theta} = -L_I(c_t)(1 - \theta)f(\theta | c_t) + L_{II}(c_t)\theta f(\theta | c_t)$$

Setting the derivative equal to zero:

$$-L_I(c_t)(1 - \theta)f(\theta | c_t) + L_{II}(c_t)\theta f(\theta | c_t) = 0$$

Assuming $f(\theta | c_t) > 0$, dividing both sides by $f(\theta | c_t)$ yields:

$$-L_I(c_t)(1 - \theta) + L_{II}(c_t)\theta = 0$$

Therefore, the Bayes-optimal threshold is:

$$\theta_k^* = \frac{L_I(c_t)}{L_I(c_t) + L_{II}(c_t)}$$

To verify optimality, note that:

$$\frac{\partial^2 R}{\partial \theta^2} = (L_I(c_t) + L_{II}(c_t))f(\theta | c_t) + \left(-L_I(c_t)(1 - \theta) + L_{II}(c_t)\theta\right) \frac{\partial f(\theta | c_t)}{\partial \theta}$$

At the stationary point $\theta = \theta_k^*$, the second term vanishes by the first-order optimality condition, giving

$$\frac{\partial^2 R}{\partial \theta^2} \Big|_{\theta=\theta_k^*} = (L_I(c_t) + L_{II}(c_t))f(\theta_k^* | c_t) > 0$$

Algorithm 2 Spectral Embedding Construction and State Clustering

Require: CoT trajectories $\{s_{1:T_i}^{(i)}\}_{i=1}^N$, Number of clusters K , spectral dimension r

- 1: **Initialize** embedding set $\mathcal{E} \leftarrow \emptyset$
- 2: **for** each trajectory i **do**
- 3: $G \leftarrow 0$
- 4: **for** each reasoning step $t = 1 \dots T_i$ **do**
- 5: Extract last-layer hidden state:

$$H_t \in \mathbb{R}^{m_t \times d}$$

- 6: Compute step Gram matrix:

$$\tilde{G}_t = \frac{1}{m_t} H_t^\top H_t$$

- 7: Update cumulative Gram matrix:

$$G \leftarrow G + \tilde{G}_t$$

- 8: Compute top- r eigenvalues of G :

$$e_t = \text{Top-}r(\lambda(G))$$

- 9: Add embedding:

$$\mathcal{E} \leftarrow \mathcal{E} \cup \{e_t\}$$

- 10: **end for**

- 11: **end for**

- 12: **Cluster embeddings using k-means:**

$$\{\mu_k\}_{k=1}^K, \phi(\cdot) = \text{kmeans}(\mathcal{E})$$

- 13: **Output:** cluster centers $\{\mu_k\}$ and assignment function ϕ
-

whenever $f(\theta_k^* \mid c_t) > 0$. Therefore, θ_k^* is a local minimizer of the Bayes risk.

We do not use Equation 1 to directly determine threshold values. Instead, the result motivates the qualitative expectation that states with larger remaining utility should require higher stopping confidence.

B Implementation Details

B.1 Clustering Configuration

Algorithm 2 describes the construction of spectral embeddings and their clustering into discrete states. Algorithm 3 then orders the clusters according to the average step index within each cluster.

Algorithm 3 Temporal Ordering of Clusters

Require: Clustered embeddings $\mathcal{E}$, assignment function ϕ , cluster count K

Ensure: Ordered cluster centers $\{\mu_k\}_{k=1}^K$

- 1: **Compute cluster-wise temporal statistics:**

- 2: **for** $k = 1 \dots K$ **do**

- 3: $\mathcal{E}_k \leftarrow \{e_t \in \mathcal{E} \mid \phi(e_t) = k\}$

- 4: $\tau_k \leftarrow \frac{1}{|\mathcal{E}_k|} \sum_{e_t \in \mathcal{E}_k} t(e_t)$

- 5: **end for**

- 6: **Compute temporal permutation**

- 7: $\pi \leftarrow \text{argsort}(\{\tau_k\}_{k=1}^K)$

- 8: **Re-index cluster centers:**

- 9: **for** $k = 1 \dots K$ **do**

- 10: $\mu_k \leftarrow \mu_{\pi(k)}$

- 11: **end for**

- 12: **Output:** reordered cluster centers $\{\mu_k\}_{k=1}^K$ and assignment function ϕ
-

B.2 Threshold Calibration

Let $k \in \{1, \dots, K\}$ denote the temporally ordered cluster index. For each cluster k , we estimate a cluster-specific stopping threshold θ_k using a held-out calibration set. Candidate thresholds are evaluated by simulating the online stopping procedure and measuring downstream task accuracy. The threshold yielding the best calibration performance is selected: $\theta_k = \arg \max_{\theta} \text{Acc}_k(\theta)$. Consistent with the temporal interpretation of clusters, the estimated thresholds are non-increasing with cluster index for most models, reflecting the intuition that later reasoning states require less evidence before termination.

For each model, we conduct a grid search and select the stopping thresholds on a held-out calibration set of 100 examples drawn from the AQUA training set, which is disjoint from the AQUA test set. The calibrated thresholds are then frozen and transferred unchanged to GSM8K, MATH500, and LogiQA. Note this calibration set is distinct from the 100-example GSM8K reference corpus used for spectral clustering in Section 5.

Table 3 compares performance on the held-out calibration set and the AQUA test set using the same thresholds. As expected, performance is broadly comparable across the two sets, with the held-out set outperforming the unseen test set in most cases. The results do not suggest substantial overfitting or underfitting.

We evaluate the calibration of the judge confidence

Model	Accuracy (%)		Avg. Tokens	
	Held-out	Test	Held-out	Test
Phi-4-mini-reasoning	91	87	1082	1070
Qwen3-8B	90	84.3	944	946
Qwen3-14B	84	86.6	1089	1056
QwQ-32B	94	90.2	1322	1331
Average	89.8	87.0	1109	1101

Table 3: Comparison of accuracy and average token usage on the held-out calibration set and the AQUA test set using the same stopping thresholds.

scores using Expected Calibration Error (ECE) with 10 equal-frequency bins over all Qwen3-8B evaluation samples. The resulting ECE is 0.10, indicating reasonable agreement between the judge’s confidence estimates and its empirical accuracy. Importantly, STARS does not rely solely on the absolute calibration of the judge confidence. The confidence-based criterion is combined with two additional state-aware signals, namely stagnation detection and terminal-state detection. Thus, moderate calibration errors in the judge confidence are unlikely to solely determine the stopping behavior.

B.3 Implementation Settings

Table 4 lists general implementation settings.

Table 5 lists model-specific implementation settings. In this table, Column 2 presents the stopping thresholds $c_1, \dots, c_7$ corresponding to the seven ordered clusters produced by Algorithm 3. Column 3 presents the choices of two methods to compute the cumulative Gram matrix.

1. Cumulative Sum. We construct the cumulative Gram matrix as defined in Section 3.1.

2. Linearly Weighted Accumulation (LWA). We alternatively construct the Gram representation using an incrementally increasing weighting of step-wise Gram matrices:

$$G_t = \frac{t-1}{t} (G_{t-1} + \tilde{G}_t), \quad G_1 = \tilde{G}_1$$

This yields

$$G_T = \frac{1}{T} (\tilde{G}_1 + \tilde{G}_2 + 2\tilde{G}_3 + \dots + (T-1)\tilde{G}_T)$$

Thereby assigning progressively larger weights to later response steps while retaining the initial question representation with unit weight.

Column 4 in Table 5 presents a model-specific factor, by which we multiply the completion confi-

dence at detected state-transition boundaries before applying the stopping criterion.

Our baselines were tuned in our evaluated setup when needed. For example, when evaluating DEER, we adapted the ratio of reasoning tokens to conclusion tokens in our evaluation setup, as the original ratio would otherwise leave the model with insufficient conclusion token budget to finish the response, resulting in worse performance.

B.4 Prompts

System prompt for the LLM judge.

You are a strict binary classifier. Determine whether the response already contains the final answer to the question.

Output rules:

- Output exactly one word.
- The word must be either YES or NO.
- Do not output anything else.

Meaning:

YES = the response contains the final answer.

NO = the response is still reasoning and the final answer has not appeared.

Example 1

Question: What is $6 + 7 - 1$?

Response: First compute $6 + 7 = 13$. Then $13 - 1 = 12$. Therefore the answer is 12.

Answer: YES

Example 2

Question: What is $6 + 7$?

Response: First compute $6 + 7 = 13$.

Answer: NO

Question: {question}

Response: {response}

Answer (YES or NO):

Description	Choice	Comment
Top eigenvalues	$r = 64$	Following Yu et al. (2025), selecting top spectral components for stability.
Step delimiters	" $\backslash n \backslash n$ "	Following Yang et al. (2026) for segmentation consistency.
Dwell length	$\delta = 5$	Estimated from transition statistics; mean dwell ≈ 4 , using conservative threshold.
Number of clusters	$K = 7$	Chosen via silhouette score on a held-out subset.
LLM judge	Qwen2.5-1.5B-Instruct	Lightweight judge with calibrated confidence; low inference cost (Ho et al., 2026).
GSM8K	1319 examples	Full testing dataset
Math-500	500 examples	Full dataset
AQuA	254 examples	Full testing dataset
LogiQA	651 examples	Full testing dataset

Table 4: General implementation settings.

Model	Stopping Thresholds $[c_1, \dots, c_7]$	Gram Matrix	Boundary boost
Phi-4-mini-reasoning	[0.98, 0.95, 0.90, 0.85, 0.70, 0.60, 0.50]	Cumulative Sum	1.00
Qwen3-8B	[0.75, 0.75, 0.75, 0.75, 0.75, 0.75, 0.75]	LWA	4.00
Qwen3-14B	[0.98, 0.90, 0.85, 0.80, 0.70, 0.60, 0.50]	Cumulative Sum	1.15
QwQ-32B	[0.98, 0.90, 0.90, 0.90, 0.70, 0.60, 0.50]	Cumulative Sum	1.05

Table 5: Model-specific implementation settings.

System prompt for Algorithm 1.

Please reason step by step. After reasoning, output ONLY one final answer enclosed in a single `\boxed{}`.

Never use more than one box. If there are multiple answers, include all of them inside the same box separated by commas (e.g., put 3, 5, 7 in `\boxed{3, 5, 7}`).

All mathematical expressions must be written in valid LaTeX (e.g. `\boxed{\sqrt{13}}`, `\boxed{\begin{pmatrix} -2 \\ -14 \\ -7 \end{pmatrix}}`, etc.).

C Additional Results

C.1 Results on Phi-4-mini-reasoning

Table 6 shows the additional results on Phi-4-mini-reasoning to evaluate the robustness of our approach across different model families.

C.2 Transition Heatmaps

Transition heatmaps for Qwen3-8B and QwQ-32B in Figure 6 show similar patterns as for Qwen3-14B in Figure 2.

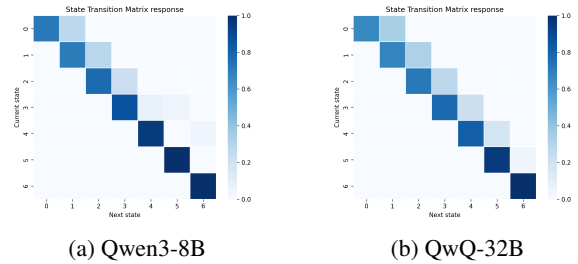


Figure 6: The horizontal axis denotes the next cluster state, while the vertical axis denotes the current cluster state. Color intensity represents transition probability. The heatmaps reveal increasingly strong self-loop behavior in later-stage clusters sorted in Algorithm 3.

C.3 Results on QwQ-32B over AIME

We further evaluate STARS on AIME using QwQ-32B, the strongest backbone considered in our experiments. STARS achieves 16.7% accuracy with only 2,441 generated tokens, matching the best baseline accuracy while substantially reducing generation length. In comparison, Standard CoT achieves 10.0% accuracy with 4,041 tokens, while EAT, TALE, and DEER achieve 6.7%, 16.7%, and 16.7% accuracy with 4,016, 3,814, and 3,890 tokens, respectively. Thus, STARS matches the accuracy of the strongest baselines while reducing token usage by approximately 40% relative to Standard CoT.

Method	MATH						LOGIC		Overall	
	GSM8K		MATH-500		AQuA		LogiQA			
	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓	Acc↑	Tok↓
<i>Phi-4-mini-reasoning</i>										
Standard CoT	89.5	1119	74.4	2440	80.7	1868	42.1	2411	74.6	1740
EAT	86.7	473	74.0	994	80.7	1026	41.6	989	73.0	744
TALE	84.8	1049	63.6	1881	76.8	1676	30.6	1939	67.2	1473
DEER	89.8	584	77.4	1297	82.3	975	49.8	1313	77.3	926
STARS	91.6	642	76.0	996	87.0	1070	55.9	1490	79.8	950

Table 6: Experimental results with Phi-4-mini-reasoning over four datasets. “Acc” denotes accuracy and “Tok” denotes token count. ↑ indicates that higher values are better, while ↓ indicates that lower values are better. Best results are in bold.

D Sensitivity to Reference Corpus

To further examine the robustness of STARS to the choice of reference corpus, we construct an alternative reference corpus using 150 training examples from AQuA, instead of the original 100 training examples from GSM8K. Thus, both the size and domain of the reference corpus are changed, while the evaluation protocol and model remain unchanged. The reference corpus is used only to construct the unsupervised spectral state representation and does not contain any test examples or test labels.

We evaluate the resulting clusters and stopping policy on GSM8K, MATH-500, AQuA, and LogiQA using Qwen3-14B. As shown in Table 7, the results remain close to those obtained with the original reference corpus. Overall accuracy changes only from 87.8% to 87.2%, while the average number of generated tokens decreases from 929 to 879. Performance remains similarly stable across individual datasets, with accuracy changes of at most 1.0 percentage point.

These results suggest that STARS is not highly sensitive to the specific reference corpus used to construct the spectral states, and that the discovered reasoning-state structure can transfer across substantially different reference-corpus choices.

E Judge Overhead

To quantify the computational overhead of the completion judge, we measure wall-clock runtime on the AQuA test set. As shown in Table 8, STARS reduces the total runtime from 6.67 hours to 4.23 hours while adding only 137 seconds of judge overhead. Each judge evaluation requires only a single

Dataset	Accuracy (%)		Avg. Tokens	
	Original	AQuA-150	Original	AQuA-150
GSM8K	95.9	95.5	579	552
MATH-500	88.0	87.0	1093	1029
AQuA	86.6	85.8	1056	957
LogiQA	71.9	71.1	1464	1396
Overall	87.8	87.2	929	879

Table 7: Robustness to the reference corpus. *Original* uses 100 GSM8K training examples to construct the reference clusters, while *AQuA-150* uses 150 AQuA training examples. All results are obtained with Qwen3-14B.

forward pass of Qwen2.5-1.5B-Instruct to read the next-token probabilities of “YES” and “NO”, rather than autoregressively generating a response. Consequently, the per-query judge time is only 0.047 seconds, accounting for less than 1% of the total STARS runtime. This small overhead is substantially outweighed by the reduction in generation time.

F Pareto Frontier

As shown in Figure 7, STARS consistently lies on or close to the accuracy–token Pareto frontier across the 12 dataset–model settings. On the two GSM8K settings where STARS trails the highest-accuracy baseline by 0.9 and 0.8 points (Qwen3-8B and QwQ-32B), it reduces token usage by 18% and 29%, respectively. The clearest accuracy–efficiency tradeoff occurs for Qwen3-8B on AQuA: DEER achieves 89.0% accuracy versus 84.3% for STARS, but requires 21% more tokens (1146 vs. 946). This illustrates that STARS does not universally maximize accuracy; rather, its stop-

	Standard CoT	STARS
Wall-clock time	6.67 h	4.23 h
Generation steps	8,718	2,909
Judge overhead	—	137 s
Judge time per query	—	0.047 s
Judge overhead / total runtime	—	0.9%

Table 8: Wall-clock runtime and completion-judge overhead on the AQuA test set conducted on NVIDIA H200 GPU. A dash (—) indicates that the corresponding metric is not applicable to standard CoT, which does not use a completion judge. STARS reduces runtime substantially while the completion judge accounts for less than 1% of total runtime.

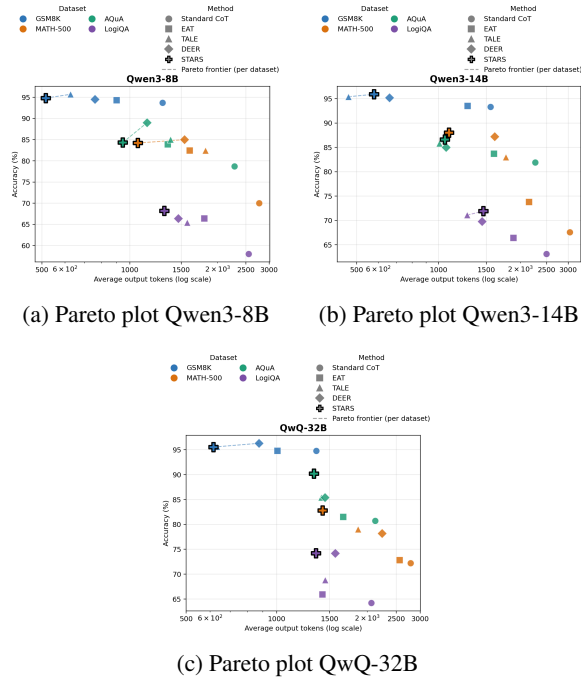


Figure 7: Pareto plots for three backbone models in Table 1.

ping criterion can trade a modest amount of accuracy for lower generation cost while maintaining a favorable accuracy–token tradeoff. We attribute the small GSM8K accuracy gaps to the relatively low difficulty of these problems for already-strong backbones. On such trajectories, STARS can terminate earlier than the highest-accuracy baseline in each setting (TALE on Qwen3-8B and DEER on QwQ-32B), trading a small amount of accuracy for substantial token savings.

G Comparison with Mean-Pooled Hidden States

To test whether spectral features are necessary beyond simpler hidden-state statistics, we replace

Dataset	Accuracy (%)		Avg. Tokens	
	Original	Mean Pooling	Original	Mean Pooling
GSM8K	94.8	94.6	516	586
MATH-500	84.2	85.4	1065	1154
AQuA	84.3	85.8	946	1062
LogiQA	68.2	67.4	1314	1362
Overall	85.5	85.5	848	920

Table 9: Comparison between STARS and a mean-pooled hidden-state representation on Qwen3-8B. The stopping procedure and calibration protocol are kept identical, with only the hidden-state representation replaced.

STARS’s spectral representation with mean pooling while keeping the stopping procedure and calibration protocol unchanged. Table 9 reports results on Qwen3-8B across four datasets. Mean pooling achieves the same overall accuracy as STARS (85.5%), and slightly higher accuracy on MATH-500 and AQuA. However, it consistently requires more tokens (3.7 to 13.6% more per dataset and 8.5% overall). On GSM8K and LogiQA, STARS achieves both higher accuracy and fewer tokens. These results suggest that while simple hidden-state aggregation provides a useful stopping signal, spectral features extract stopping-relevant information more efficiently, supporting STARS’s primary goal of reducing redundant CoT generation.

H Sensitivity to LLM Judge

Figure 8 compares the accuracy–token trade-off obtained by replacing the original Qwen2.5-1.5B-Instruct judge with Qwen2.5-3B-Instruct, while keeping the stopping thresholds fixed. The effect of the judge is primarily reflected in generation length rather than accuracy, but varies substantially across datasets. On GSM8K, where the original accuracy is already near saturation (94.8%), the larger judge increases generation by 456 tokens while improving accuracy by only 0.3 percentage points. In contrast, on AQuA, the larger judge increases generation by 262 tokens while improving accuracy by 5.1 percentage points, corresponding to a substantially more favorable accuracy–token trade-off. These results suggest that judge replacement can shift the operating point of STARS, with the magnitude of the efficiency impact depending on the available accuracy headroom.

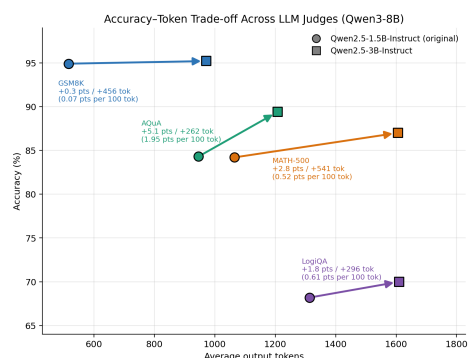


Figure 8: Accuracy–token trade-offs across LLM judges on Qwen3-8B. Arrows show the change induced by replacing the Qwen2.5-1.5B-Instruct judge with Qwen2.5-3B-Instruct under fixed stopping thresholds.

I Ethics Statement

Our work focuses on optimizing Chain-of-Thought reasoning via early stopping, which reduces computational overhead. However, a potential risk is that premature stopping could cut off necessary verification steps in the reasoning chain, leading to undetected hallucinations or errors if deployed in high-stakes domains without safety guardrails.