SyncThink: A Training-Free Strategy to Align Inference Termination with Reasoning Saturation

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) prompting improves reasoning but often produces long and redundant traces that substantially increase inference cost. We present **SyncThink**, a training-free and plug-and-play decoding method that reduces CoT overhead without modifying model weights. We find that answer tokens attend weakly to early reasoning and focus on `</think>`, indicating an information bottleneck. Building on this observation, SyncThink monitors the model's own reasoning-transition signal and terminates reasoning. Experiments on GSM8K, MMLU, GPQA, and BBH across three DeepSeek-R1 distilled models show that SyncThink achieves 62.00% average Top@1 accuracy using 656 generated tokens and 28.68s latency, compared to 61.22%, 2141 tokens, and 92.01s for full CoT decoding. On long-horizon tasks such as GPQA, SyncThink can further yield up to +8.1 absolute accuracy by preventing over-thinking.

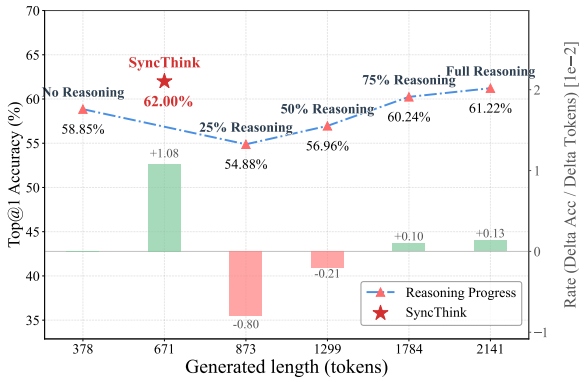


Figure 1: **Efficiency-Accuracy Trade-off.** The blue curve depicts the base model's capability boundary. **SyncThink** (red star) significantly surpasses this limit, occupying the **top-left Pareto frontier** with higher accuracy and reduced token usage. The bottom bars represent the **marginal reasoning efficiency** (ΔAcc/ΔTokens). Our method achieves the highest score (1.08), demonstrating that it effectively retains critical reasoning while pruning redundancy. More results across four datasets can be viewed in Appendix B.

1 Introduction

Large language models (LLMs) (Vaswani et al., 2017; Zhang et al., 2025b) achieve strong reasoning via Chain-of-Thought (CoT) prompting (Wei et al., 2022), but long CoT traces substantially increase inference cost. As Figure 1 shows, accuracy gains quickly saturate beyond a token budget threshold, yielding diminishing returns. Consistent with prior findings that models over-think (Chen et al., 2025), often reach correct answers early (Liu and Wang, 2025), and can perform well with concise drafts (Xu et al., 2025), this points to a mismatch between generated length and true reasoning demand, leaving many tokens redundant for the final prediction.

However, existing solutions remain limited. Training-based methods (Yao et al., 2023b; Bi et al., 2024; Ye et al., 2024; Zhang et al., 2024) require additional fine-tuning or specialized data, reducing deployability. Training-free heuristics such as Answer Convergence (Liu and Wang, 2025) and sampling-based stopping rules (Sui et al., 2025) rely on black-box consistency checks and repeated probing, often adding extra inference cost without leveraging the model's intrinsic signals.

We formalize this inefficiency as **Cognitive Lag**: the model has already reached a sufficient reasoning state, yet continues generating redundant tokens. From a **Universal Reasoning Bottleneck** view, attention/saliency analyses (figure 2) show that a transition token (e.g., `</think>`) acts as an **information bridge** that compresses the reasoning history into a decidable state. We therefore monitor reasoning by tracking the step-wise logit rank of this token (figure 3). Macro results on BBH further reveal that accuracy saturates in an "optimal truncation zone" (starting at ∼60% of the reasoning progress), well before the model emits its intrinsic termination signal.

To bridge this lag, we propose **SyncThink**, a training-free decoding framework motivated by the

Universal Reasoning Bottleneck. We view the shift from reasoning to answering as a semantic boundary, which is explicitly marked in reasoning-specialized models (e.g., `</think>` in DeepSeek-R1). SyncThink treats this transition token as a high-fidelity probe: by tracking its logit *rank* and *entropy*, we trigger *dynamic early stopping* when the model reaches information saturation.

Experiments show that SyncThink achieves a better efficiency–accuracy trade-off. In Figure 1, SyncThink reaches 62.00% Top@1 accuracy with $\sim$656 tokens, outperforming full CoT (61.22%, $\sim$2141 tokens) while reducing the token budget by 69.4% (3.21$\times$ speedup). We further quantify this gain with a **reasoning efficiency rate** (bottom bars), defined as the accuracy improvement per additional token over the non-reasoning baseline (ΔAcc/ΔTokens). Fixed truncation yields negligible or negative returns and full reasoning shows diminishing marginal utility (0.13), whereas SyncThink achieves the highest efficiency score (1.08), indicating it preserves high-value reasoning while removing redundant "over-thinking" tokens.

Our contributions are as follows:

- We define **Cognitive Lag** in CoT reasoning and show, via micro/macro analyses, that accuracy saturates well before the model's termination signal.
- We introduce a **Universal Reasoning Bottleneck** view and, supported by attention/saliency evidence (figures 2 to 3), establish `</think>` as a high-fidelity probe of the reasoning-to-answer transition.
- We propose **SyncThink**, a training-free decoding method that monitors intrinsic logit signals to dynamically truncate redundant reasoning, improving efficiency while preserving or improving accuracy across tasks.

2 Related Work

2.1 LLMs Reasoning

Chain-of-Thought (CoT) prompting (Wei et al., 2022) fundamentally enhances LLMs by decomposing complex tasks into intermediate reasoning steps. This paradigm has been further extended by strategies like Self-Consistency (Wang et al., 2022), which improves robustness via path aggregation, and Tree-of-Thoughts (Yao et al., 2023a), which introduces structured planning. Most recently, advanced models such as OpenAI's o1, Alibaba's QwQ (Team, 2025), and DeepSeek's R1 (Guo et al., 2025) integrate reflective capabilities—including trial-and-error and self-correction—directly into the reasoning process, moving beyond static prompting to dynamic inference (Shinn et al., 2023).

2.2 CoT Compression

Efficient CoT generation is primarily addressed through two paradigms: training-based compression and inference-time optimization (Qu et al., 2025a; Sui et al., 2025). Training approaches aim to minimize reasoning overhead by either pruning explicit tokens based on difficulty (Han et al., 2024; Zhang et al., 2025a) or mapping thoughts to latent states (Chen et al., 2024; Qu et al., 2025b). In contrast, inference-time methods like Sketch-of-Thought (Aytes et al., 2025; Xu et al., 2025) achieve efficiency by generating concise reasoning drafts, bypassing the need for model retraining.

Our SyncThink aligns with prior work on inference-time CoT optimization, but differs in that it is entirely training-free and model-agnostic, making it readily applicable to DeepSeek-R1 distilled models. Instead of learning to compress reasoning, SyncThink dynamically identifies the reasoning state from the model's logits and lightweightly truncates redundant reasoning once completion signal is detected, while a logit-driven completion mechanism restores output quality and formatting with minimal overhead.

3 Methodology

We propose **SyncThink**, a *training-free* framework designed to optimize CoT reasoning efficiency by eliminating redundant computation. To ground this approach in the model's intrinsic mechanics, our methodology proceeds through a rigorous analytical pipeline:

(1) **Mechanism Identification (§3.2):** Through attention and saliency analysis, we first validate the `</think>` token as a critical **Information Bottleneck**, establishing it as a high-fidelity probe for monitoring internal reasoning states.

(2) **Phenomenon Quantification (§3.3):** By tracking the temporal dynamics of this probe, we reveal the **Cognitive Lag**. Our micro-level phase analysis and macro-level accuracy statistics jointly demonstrate that answer saturation precedes the model's explicit termination.

(3) **Algorithm Design (§3.4):** Building on these insights, we formulate SyncThink, a heuristic

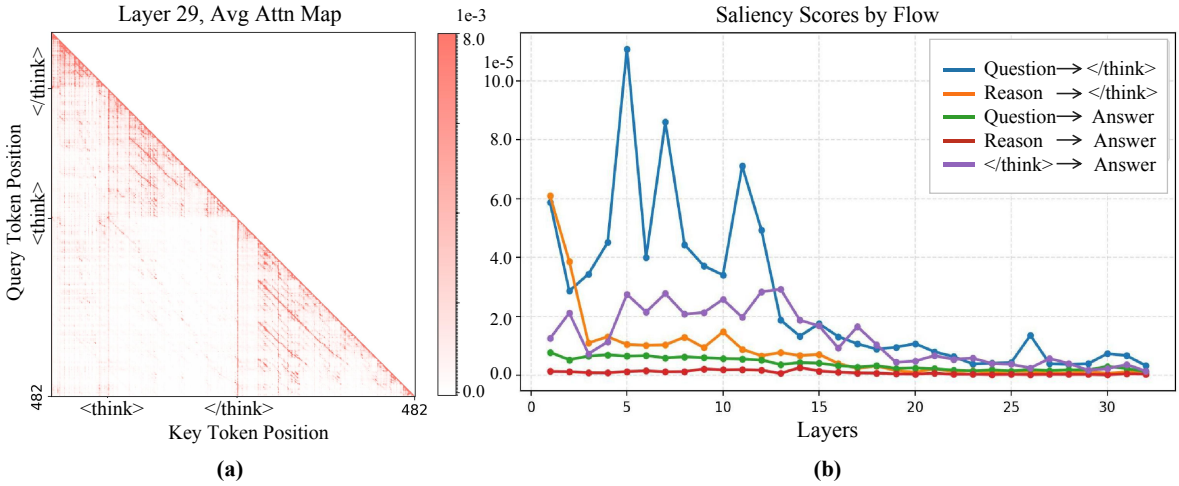


Figure 2: Empirical evidence of the **Universal Reasoning Bottleneck** on a GSM8K example. (a) At Layer 29, attention concentrates on the `</think>` boundary rather than reasoning tokens (more layers in Fig. 8). (b) Saliency analysis shows that information transfer from Reasoning to Answer is negligible (red line), while the dominant pathway is mediated by `</think>` (purple line), suggesting reasoning is compressed into this transition token.

strategy that leverages dynamic rank and entropy thresholds to intervene precisely at the point of information saturation.

3.1 Problem Formulation: The Efficiency-Accuracy Trade-off

Given a question q, an LLM generates a reasoning chain $r = (r_1, \ldots, r_L)$ followed by an answer a. While r generally enhances the conditional probability $P(a|q, r)$, the computational cost scales linearly with L. Recent observations in the literature (Chen et al., 2025; Liu and Wang, 2025) indicate that this performance gain exhibits *diminishing returns*: accuracy tends to saturate rapidly after an initial rise, implying that extending L beyond a critical point yields negligible gains.

Therefore, our objective is to find an optimal truncation point $\hat{t}$ that maximizes accuracy while minimizing cost:

$$\hat{t} = \arg\max_{t} \left(A(t) - \alpha \cdot \mathrm{Cost}(t) \right) \tag{1}$$

Since the true accuracy $A(t)$ is unobservable during inference, we seek an intrinsic proxy from the model's internal state to estimate reasoning sufficiency.

3.2 Attention Analysis: The Information Bottleneck

To identify an intrinsic proxy, we first examine internal attention maps, as self-attention governs information flow in Transformers. As shown in Figure 2, deep-layer attention allocates negligible mass to individual reasoning tokens and instead concentrates on the `</think>` boundary. This observation suggests that (i) the reasoning span contains substantial redundancy, and (ii) `</think>` serves as an **information bottleneck** where critical context is compressed and accumulated for answer generation.

Further, to validate these hypotheses, we employ Saliency Score analysis following Wang et al. (2023). This framework quantifies the contribution of specific attention regions by measuring the sensitivity of the loss function to masked perturbations via a first-order Taylor approximation:

$$\Delta L \approx \alpha \left\langle A_{h,l} \odot \frac{\partial L}{\partial A_{h,l}}, M \right\rangle_F . \tag{2}$$

This approximation reveals that saliency is essentially the element-wise product of the attention activations and their gradients within the masked region.

As illustrated in Figure 2, this analysis yields three key insights regarding the reasoning mechanism:

- **The `</think>` Token as an Information Bridge:** The information flow follows a "Reasoning → `</think>` → Answer" path. Direct saliency from reasoning tokens to the answer is negligible, confirming that `</think>` serves as a critical mediator that aggregates context before generation.
- **Early Aggregation and Logic Construction:** Saliency peaks in shallow layers (1–2) for context

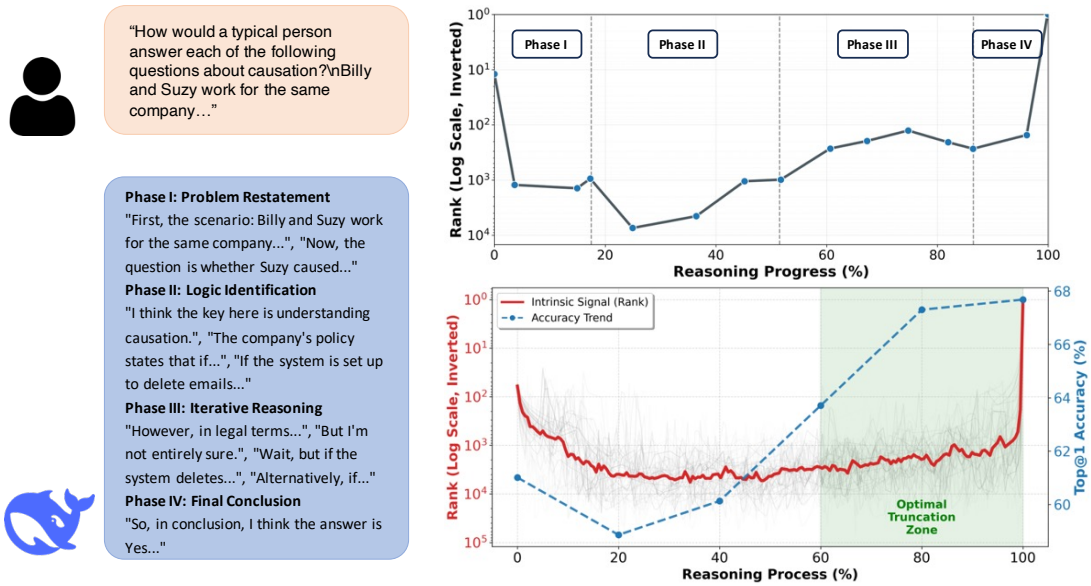


Figure 3: **The rationale behind SyncThink.** (Top) **Micro-Dynamics:** The </think> rank provides a real-time signal that tracks the four reasoning phases in §3.3. (Bottom) **Macro-Statistics:** On BBH, we observe a *cognitive lag*: truncation accuracy (blue) saturates within the shaded *optimal truncation zone* well before the model's intrinsic termination signal (red).

compression, while core logical decision-making is centralized in the middle layers (5–15). This indicates that the primary information processing occurs in the early-to-mid stages of the network.

- **Completion of Information Integration:** In deeper layers (> 20), saliency scores across all key paths diminish significantly (quiescence). This confirms that critical information migration to </think> is fully concluded, rendering further explicit retrieval of reasoning history redundant.

Our saliency analysis identifies </think> as a critical information bottleneck where reasoning content is distilled, while intermediate tokens exhibit rapid saliency decay. This phenomenon is not limited to specific instances and is analyzed in detail in Appendix C. Consequently, the model's primary reliance on this distilled signal supports our strategy of using it for dynamic early stopping.

3.3 The Temporal Dynamics of Reasoning Completeness

Building on the identification of *where* information concentrates, we now examine the temporal dynamics to determine *when* reasoning becomes sufficient. Specifically, we track the evolution of the </think> token's rank, defined as the rank of its logit among the full vocabulary at each decoding step. We use this step-wise logit rank as a lightweight proxy for the model's internal reasoning state. As shown in Figure 3 (Top), the rank trajectory maps to four distinct cognitive phases:

Phase I: Problem Restatement (Sharp Descent). The rank initially precipitates from the start to $> 10^4$. This marks the transition from instruction encoding to reasoning initiation, indicating the model's commitment to generating a long-form chain rather than an immediate answer.

Phase II: Logic Identification (Initial Recovery). A steady recovery to $\sim 10^3$ follows, corresponding to the *Orientation* stage. As the model structures the problem space and identifies a viable logical path, the uncertainty regarding termination decreases.

Phase III: Iterative Reasoning (Volatile Plateau). The rank oscillates between 10^2 and 10^3. These fluctuations reflect *Self-Reflection*: local peaks suggest the completion of sub-steps, while subsequent drops indicate the activation of verification or backtracking mechanisms during conflict resolution.

Phase IV: Final Conclusion (Linear Ascent). Upon resolving the internal debate, the trajectory shifts to a decisive, linear ascent to Top-1. Unlike the fluctuations in Phase III, this irreversible rise serves as a deterministic signal that the reasoning process is concluded.

This analysis highlights the limitation of fixed token budgets, which fail to distinguish between necessary formulation (Phase II) and potentially redundant exploration (Phase III). Our framework leverages these rank trends to detect latent termination signals, enabling dynamic truncation dur-

ing periods of "overthinking" to prevent redundant computation.

Macro-Statistics: The Cognitive Lag. We further aggregate reasoning trajectories across the BBH dataset to analyze macro-level trends (Figure 3, Bottom). Here, the red curve depicts the median rank of the `</think>` token overlaid on individual sample traces (grey), while the blue dashed line indicates the Top-1 accuracy achievable by truncating generation at the corresponding progress percentage.

This macro-view reveals a critical decoupling between model capability and generation policy, termed **"Cognitive Lag"**. Specifically, as reasoning progresses beyond 60%, the truncation accuracy (blue) enters a saturation plateau, recovering to near-optimal levels. Paradoxically, the model's intrinsic termination signal (red) remains deeply suppressed (rank $\sim 10^3$) throughout this phase, indicating a lack of confidence to terminate despite having effectively solved the problem. This misalignment drives the model to engage in prolonged, low-utility reasoning to pursue negligible performance gains.

Consequently, a dynamic intervention mechanism is required to decouple termination from the model's inertia, intervening precisely when information saturation is detected.

3.4 SyncThink: A Dynamic Reasoning Termination Method Based on the Logit

Our investigation reveals a mechanism of **Progressive Information Condensation**, where the model aggregates reasoning logic into evolving hidden states and employs the `</think>` token as a "summary embedding" of completeness. This dynamic reflects a **Mechanism Competition** between *Exploration* and *Conclusion* modes, where a rising rank of `</think>`—indicative of state saturation—signals the model's readiness to transition. Addressing the noise sensitivity caused by the magnitude dominance of raw logits, we demonstrate that the token **Rank** serves as an *intrinsic proxy for information sufficiency*.

Motivated by this, we introduce a heuristic dynamic truncation mechanism. When the rank of the terminator token ascends to a sufficiently high position, we manually inject `</think>` to terminate the reasoning process. We formulate this strategy as follows:

$$\mathbb{I}_{\text{stop}}(t) = \begin{cases} 1 & \text{if } \mathcal{R}_t(\texttt{</think>}) \leq \tau(t, \mathbf{p}_t) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, $\mathcal{R}_t(\texttt{</think>})$ denotes the rank of the terminator token within the current probability distribution $\mathbf{p}_t$ (0-indexed). The dynamic threshold $\tau(t, \mathbf{p}_t)$ is modulated by both temporal pacing and model uncertainty:

$$\tau(t, \mathbf{p}_t) = \lfloor \beta(t) \cdot \exp(-\lambda \cdot \mathcal{H}(\mathbf{p}_t)) \rfloor \tag{4}$$

The components of the threshold are defined as follows:

- **Time-Dependent Pacing $\beta(t)$:** We directly employ t as the time-scale control factor. As t increases, the threshold rises accordingly, aligning with the intuition that longer reasoning chains are increasingly prone to redundancy. Additionally, we impose an upper bound T_{max} to prevent the threshold from becoming excessively loose in extended sequences, thereby avoiding indiscriminate truncation.
- **Entropy-Aware Scaling $\mathcal{H}(\mathbf{p}_t)$:** We compute the Shannon entropy of the next-token distribution to adjust for model confidence:

$$\mathcal{H}(\mathbf{p}_t) = -\sum_{v \in V} p_t(v) \log p_t(v) \tag{5}$$

  where V is the vocabulary. High entropy (uncertainty) reduces the threshold via the term $e^{-\lambda \mathcal{H}}$, tightening the constraint to prevent premature truncation during exploration phases (Wang et al., 2025). Conversely, low entropy encourages commitment to a conclusion.

Once the condition $\mathbb{I}_{\text{stop}}(t) = 1$ is met, we manually inject the `</think>` token and proceed to the answer generation phase.

4 Experimental Setup

Datasets To comprehensively evaluate our proposed method across diverse reasoning and knowledge-intensive scenarios, we conduct experiments on the following four benchmark datasets:

- **GSM8K** (Cobbe et al., 2021): Evaluates multi-step arithmetic reasoning via grade-school math problems.
- **MMLU** (Hendrycks et al., 2020): Assesses general knowledge across 57 diverse domains, ranging from elementary to professional levels.

- **GPQA** (Rein et al., 2024): Tests high-level conceptual understanding with graduate-level scientific questions.
- **BBH** (Suzgun et al., 2022): Stress-tests models on hard-to-solve tasks involving symbolic and logical planning.

Backbone. We select models in the 7B–14B range, reflecting practical constraints while maintaining strong reasoning capabilities.

Qwen2.5-7B/14B (Yang et al., 2024): Leading open-source models with robust reasoning performance. *LLaMA3.1-8B* (Grattafiori et al., 2024): A balanced model optimized for efficient instruction following. Identical decoding settings are applied across all baselines for consistent comparison.

Baselines. We compare SyncThink against five baselines: (1) **Full CoT**: Generates complete reasoning traces, serving as the inference-time quality upper bound; (2) **No CoT**: Directly generates answers without reasoning; (3) **Fixed-Ratio Truncation**: A heuristic that stops reasoning at a predetermined length ratio; (4) **Answer Convergence** (Liu and Wang, 2025): A probing method that triggers early stopping when intermediate answer predictions stabilize across k consecutive segments. Additionally, on GSM8K, we include an **SFT Oracle/Upper Bound** (fine-tuned on gold reasoning traces) to benchmark the maximum achievable performance via supervised learning.

All baselines are evaluated with identical prompts and answer extraction, and we match the generation budget across methods whenever applicable, enabling an apples-to-apples comparison of accuracy and efficiency.

Metrics. We evaluate each method from both effectiveness and efficiency perspectives using three primary metrics: Top-1 accuracy (Top@1↑), wall-clock inference time (Time↓), and the number of generated tokens (Tokens↓). Top@1 accuracy is computed as the exact-match rate under a unified answer parser for each benchmark.

Decoding parameters. Unless otherwise specified, all results are obtained under **greedy decoding** (temperature $= 0$ and no stochastic sampling) with `max_new_tokens`$= 8192$ for all methods. Under this setting, generation is (near-)deterministic given the same model, tokenizer, prompts, and inputs; consequently, we report results from a **single deterministic run** per configuration. More specific details can be seen in Appendix A. **For our truncation strategy** in Eq. 4, we tune λ on the held-out validation set and fix it to $\lambda = 0.8$, which is then applied unchanged to all test benchmarks; we also set $\beta(t) = t$ (the decoding step) to gradually relax the early-exit constraint as generation proceeds.

5 Results and Analysis

5.1 Main Results

Table 1 presents the performance comparison across four benchmarks. As indicated by the average results, **SyncThink** achieves a superior trade-off between reasoning quality and inference efficiency. Notably, **SyncThink** yields the highest average Top@1 accuracy (**62.00%**), surpassing the **Full Reasoning** baseline (61.22%) while requiring only ∼30% of the token generation cost (655.80 vs. 2141.25 tokens). This suggests that our dynamic truncation mechanism effectively identifies the critical reasoning path without redundant computation.

5.2 Mitigating Over-Thinking via Information Synchronization

Models often succumb to "over-thinking," where valid reasoning degrades into hallucination due to delayed termination. **SyncThink** addresses this by synchronizing generation with internal information saturation.

Case Study: Pruning Harmful Redundancy. Figure 6 presents a representative GPQA example where *more* reasoning becomes *worse*. The Full CoT baseline reaches a correct intermediate state and is already poised to answer. However, instead of committing, it continues generating additional derivations and self-corrections (e.g., *"Wait, but I need to relate this to..."*), gradually drifting into an unnecessary re-formulation of the problem and ultimately outputting an incorrect option (C). In contrast, **SyncThink** monitors the logit dynamics of the reasoning-state transition token `</think>` and identifies the onset of this detrimental redundancy. As highlighted in the figure, SyncThink triggers an early exit at the *earlier* `</think>` boundary (upper marker), where the model's information has already saturated, and directly produces the correct answer (A), avoiding the later over-thinking branch that leads to error.

Adaptive Truncation. Our method also adapts to task complexity. On simple benchmarks (GSM8K),

Method	GSM8K			MMLU			GPQA			BBH			Avg Res		
	Top@1	Time	Tokens	Top@1	Time	Tokens	Top@1	Time	Tokens	Top@1	Time	Tokens	Top@1	Time	Tokens
DeepSeek-R1-Qwen2.5-7b															
Full Reasoning	87.57	18.79	440	61.15	73.00	1676	30.30	231.58	5444	65.86	44.65	1005	61.22	92.01	2141
No Reasoning	88.09	11.39	266	53.26	12.01	276	32.82	25.90	609	61.21	16.08	362	58.85	16.35	378
Reasoning Ratio($r = 0.25$)	85.52	13.45	317	49.85	28.41	674	24.75	78.89	1879	59.39	19.51	439	54.88	35.07	827
Reasoning Ratio($r = 0.5$)	85.97	15.22	359	52.48	42.61	1005	25.76	131.20	3170	63.64	29.35	660	56.96	54.60	1299
Reasoning Ratio($r = 0.75$)	86.43	17.05	417	57.55	58.75	1421	31.31	184.43	4458	65.67	37.38	841	60.24	74.40	1784
SFT Model	77.41	4.53	106	51.41	15.80	363	32.32	39.44	927	48.28	30.84	694	52.36	22.65	523
Answer Convergence	87.82	15.28	278	57.12	34.64	754	31.38	180.38	3919	62.36	27.21	503	59.67	64.38	1364
SyncThink	87.49	14.39	383	59.10	28.13	643	**38.38**	**47.73**	**1116**	63.03	24.46	541	62.00	28.68	671
DeepSeek-R1-Qwen2.5-14b															
Full Reasoning	93.33	35.24	493	81.60	105.67	1465	43.43	378.28	5300	85.25	92.34	1287	75.90	152.88	2136
No Reasoning	92.04	18.07	251	74.88	21.65	299	40.91	49.92	698	77.37	17.66	247	71.30	26.83	374
Reasoning Ratio($r = 0.25$)	91.15	22.59	319	66.52	44.68	630	35.48	140.23	2021	76.88	37.40	525	67.50	61.23	874
Reasoning Ratio($r = 0.5$)	91.62	27.09	379	70.03	63.90	888	36.92	217.19	3173	82.38	57.00	807	70.24	91.30	1312
Reasoning Ratio($r = 0.75$)	92.12	31.17	459	76.80	87.14	1271	44.88	300.80	4455	85.00	73.78	1070	74.70	123.22	1814
SFT Model	84.43	156.58	2191	64.23	329.49	4569	44.04	367.26	5145	70.59	341.63	4763	65.82	298.74	4167
Answer Convergence	92.23	33.49	381	75.33	70.29	744	41.91	188.38	3741	79.38	68.38	735	72.21	90.14	1400
SyncThink	93.03	31.59	442	77.99	61.59	852	**46.97**	**126.29**	**1764**	83.84	58.65	817	75.46	69.53	969
DeepSeek-R1-LLaMA3.1-8b															
Full Reasoning	78.47	23.66	498	62.80	99.38	2076	24.24	283.49	5978	73.33	69.03	1445	59.71	118.89	2499
No Reasoning	80.51	12.72	267	58.33	15.15	315	32.83	28.16	595	63.03	12.68	263	58.68	17.18	360
Reasoning Ratio($r = 0.25$)	76.63	15.52	340	51.20	36.58	768	19.80	97.74	2146	66.13	27.98	588	53.44	44.46	963
Reasoning Ratio($r = 0.5$)	77.04	18.49	403	53.90	58.99	1236	20.61	159.96	3455	70.86	40.87	850	55.60	69.58	1525
Reasoning Ratio($r = 0.75$)	77.45	21.16	486	59.10	79.42	1731	25.05	220.97	4885	73.12	54.95	1195	58.68	94.13	2049
SFT Model	63.99	3.99	84	46.45	10.53	220	24.75	23.51	496	49.29	17.12	358	46.12	13.79	289
Answer Convergence	78.18	15.33	385	59.13	54.93	538	28.98	174.58	3959	65.38	43.78	840	57.92	72.16	1431
SyncThink	78.32	19.85	419	59.98	41.91	872	**33.84**	**76.29**	**1602**	67.68	38.24	796	59.96	44.07	922

Table 1: Comparison of different methods across GSM8K, MMLU, GPQA, and BBH datasets on DeepSeek-R1-Qwen2.5 and LLaMA3.1 models. We report Top@1 accuracy, Time, and Tokens usage.

SyncThink applies conservative truncation, preserving linear reasoning paths. Conversely, on complex tasks (GPQA), it exhibits a higher truncation rate. This confirms that **SyncThink** selectively targets the "over-thinking" loops prevalent in difficult reasoning, rather than indiscriminately shortening all responses.

5.3 Universality of the Reasoning Bottleneck

A natural question is whether our reliance on the `</think>` signal is specific to DeepSeek-R1-style training. We argue that it reflects a more general *mechanistic* pattern in LLMs: attention heads often concentrate probability mass on a small set of "sink" tokens that stabilize internal state and aggregate long-range context. In CoT decoding, the shift from a divergent *reasoning phase* to a convergent *answering phase* requires such a semantic boundary, which effectively compresses the preceding rationale into a decidable state. DeepSeek-R1 makes this boundary explicit via `</think>`, while other models may realize the same function implicitly through discourse markers (e.g., *Therefore*, *Thus*) or role separators. Therefore, SyncThink leverages `</think>` as a concrete instance of a **Reasoning Attention Sink** (Xiao et al., 2023), and the core idea remains applicable whenever an equivalent transition token can be identified. We provide a more detailed discussion in Appendix C.

5.4 Efficiency and Pareto Trade-off

SyncThink effectively shifts the accuracy–efficiency Pareto frontier outward. On average, it matches **Full Reasoning** accuracy (62.00% vs. 61.22%) while reducing token consumption by $\sim$70% (655.80 vs. 2141.25). This contrasts with static heuristics that merely trade accuracy for speed, and **SFT baselines** which fail due to surface-level mimicry—often resulting in either aggressive over-truncation or degenerate infinite loops. The benefits are most pronounced on complex tasks like GPQA, where **SyncThink** (38.38%) outperforms both Full Reasoning (30.30%) and No Reasoning (32.82%) by mitigating error accumulation. Overall, **SyncThink** offers a superior operating point, translating token savings into practical latency and memory reductions.

Architectural Robustness. Table 1 confirms **SyncThink**'s consistency across diverse architectures (LLaMA and Qwen families, 7B–14B). Notably, it achieves a $\sim 3\times$ speedup on Qwen-7B with preserved accuracy. This validates the `</think>` logit gap as a robust, model-agnostic indicator for

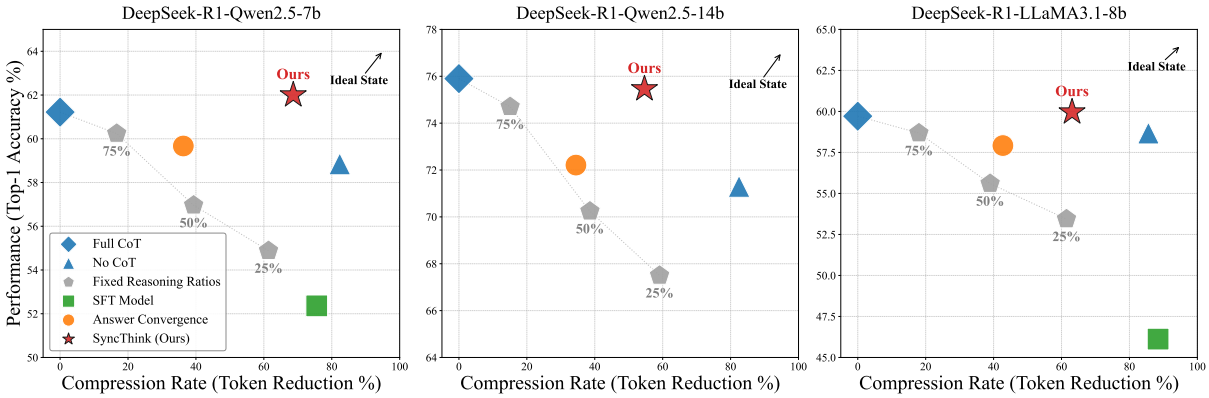


Figure 4: **Efficiency-Accuracy Trade-off.** SyncThink demonstrates superior Pareto efficiency compared to Full CoT, SFT, and static truncation baselines. The red star indicates our method achieves the best balance between token reduction and model performance across all evaluated models.

reasoning completeness.

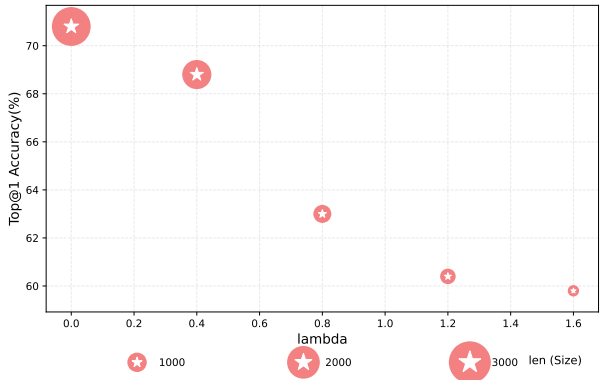


Figure 5: Ablation study on λ.

5.5 Parameter Analysis

To avoid overfitting to evaluation benchmarks, we tune the entropy weight λ on a held-out MATH500 (Lightman et al., 2023) validation set, sweeping λ and measuring Top@1 accuracy and average generation length (Figure 5).

Increasing λ monotonically shortens generations, confirming it controls truncation aggressiveness. Accuracy drops as λ grows, yielding a clear accuracy–length Pareto trade-off. Thus, λ provides a simple knob to select the desired efficiency–performance operating point.

5.6 Performance Across Diverse Tasks

SyncThink generalizes effectively across diverse domains, surpassing the **Full Reasoning** baseline with over 60% computational savings. Notably, this is achieved using a unified hyperparameter configuration without task-specific fine-tuning. On complex benchmarks like GPQA, our method successfully mitigates "over-thinking" by halting generation before hallucinations occur: for Qwen2.5-7B, SyncThink improves accuracy from 30.30% (Full Reasoning) to 38.38%. This stability confirms that SyncThink captures intrinsic reasoning patterns. **The hyperparameters thus serve not as sensitive tuning variables, but as a consistent lever for users to prioritize either extended reasoning for marginal gains or aggressive truncation for efficiency**.

6 Conclusion

We propose **SyncThink**, a training-free and model-agnostic decoding framework that synchronizes CoT generation with the model's intrinsic reasoning sufficiency signals. Our key insight is that these signals already indicate *where* and *when* reasoning becomes sufficient. Attention analyses reveal an **information bottleneck** at the reasoning boundary token (e.g., `</think>`), while temporal dynamics show the model can *sense* its reasoning state. Due to training-induced generation bias, the emitted termination is often delayed, causing **Cognitive Lag**—the model is already decidable yet continues producing redundant tokens. SyncThink addresses this misalignment with a lightweight decision rule that aligns decoding with information saturation, reducing unnecessary computation while preserving answer quality. Empirically, SyncThink matches **Full Reasoning** accuracy on average (62.00% vs. 61.22%) with ∼70% fewer tokens (655.80 vs. 2141.25), and improves GPQA to 38.38% (vs. 30.30% Full; 32.82% No).

7 Limitations

While our proposed method demonstrates effectiveness, we acknowledge two aspects regarding the scope of our current analysis:

- **Cross-Model Validation.** Our experiments mainly focus on models that expose an explicit reasoning-to-answer boundary (e.g., `</think>`). As discussed in Appendix C, we view `</think>` as a surface realization of a more general attention-dynamics phenomenon (e.g., sink-like attractor behavior) around phase transitions, rather than a model-specific artifact. Due to time constraints, we do not conduct a large-scale cross-model study in this submission; comprehensive validation across diverse model families.
- **Scale Verification:** Our current experiments have not yet extended to **larger-scale models** (e.g., 70B+ parameters). As model behavior and attention redundancy often evolve with scale, further empirical study is needed to confirm the consistency of the information bottleneck phenomenon and the optimal configuration of our method in massive-scale settings.

8 Ethical Considerations

We use publicly available datasets and model checkpoints under licenses permitting research use; license terms and restrictions are summarized in Section 4. All artifacts are used in accordance with their intended purposes specified by the original providers. We also use an LLM only for writing assistance and language polishing.

References

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

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning. *arXiv preprint arXiv:2412.09078*.

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. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

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

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, and 1 others. 2025a. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025b. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement

learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

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*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

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, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Hai Ye, Mingbao Lin, Hwee Tou Ng, and Shuicheng Yan. 2024. Multi-agent sampling: Scaling inference compute for data synthesis with tree search-based agentic collaboration. *arXiv preprint arXiv:2412.17061*.

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

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. *arXiv preprint arXiv:2401.02009*.

Xiaoying Zhang, Da Peng, Yipeng Zhang, Zonghao Guo, Chengyue Wu, Chi Chen, Wei Ke, Helen Meng, and Maosong Sun. 2025b. Will pre-training ever end? a first step toward next-generation foundation mllms via self-improving systematic cognition. *arXiv preprint arXiv:2503.12303*.

A Implementation Details

A.1 Inference Configuration

We conduct all inference experiments using the `transformers` library on a high-performance cluster node equipped with **8 × NVIDIA H800 GPUs**. To ensure reproducibility, we set the random seed to 5 across all runs. The specific hyperparameters for generation are listed below:

- **Decoding Strategy:** We utilize greedy decoding (`do_sample=False`) to eliminate randomness in the generation trajectory and ensure deterministic evaluation results.
- **Precision:** All models are loaded in `bfloat16` precision to balance computational efficiency and numerical stability.
- **Batch Size:** We set the inference batch size to 1 to simulate a real-world streaming latency scenario.
- **Max Generation Length:** The maximum number of new tokens is set to 8192 to accommodate the full reasoning chain of the baseline models.

A.2 Latency Measurement Protocol

To rigorously evaluate the efficiency of our proposed **SyncThink** method, we define the inference latency metric to include the full end-to-end processing time. Specifically, the reported time T_{total} is calculated as:

$$T_{total} = T_{gen} + T_{metric} + T_{eval} \tag{6}$$

where:

- T_{gen} denotes the raw model generation time.
- T_{metric} represents the computational overhead introduced by our method, including the real-time extraction and calculation of token logits and attention entropy.
- T_{eval} includes the time required for the final answer extraction and correctness verification.

This comprehensive measurement ensures that the reported speedup ($3.21\times$) reflects the net performance gain after accounting for all computational costs associated with our dynamic monitoring mechanism.

B Pareto Frontier Analysis of SyncThink

To examine the accuracy–efficiency trade-off, we sweep the truncation ratio of a fixed-ratio baseline and plot accuracy against the truncation ratio on four benchmarks. This sweep induces a set of candidate operating points, whose upper envelope forms an *empirical Pareto frontier* for uniform truncation policies. As shown in Figure 7, **SyncThink** consistently lies in the upper-left region relative to this frontier across all datasets, indicating a strictly better operating point: it achieves higher accuracy under the same truncation ratio, or equivalently reaches comparable accuracy with fewer reasoning tokens. Importantly, this gain cannot be replicated by simply selecting a different global truncation ratio. Instead, it stems from **SyncThink**'s instance-adaptive, logit-driven termination rule, which detects when additional reasoning becomes redundant and reallocates the reasoning budget more effectively than any single fixed-ratio strategy.

C The Universality of the Reasoning Bottleneck

In this section, we discuss the theoretical grounding of our method, arguing that the effectiveness of the `</think>` token is not an artifact of specific model training (e.g., DeepSeek-R1) but a manifestation of a universal mechanism in Large Language Models (LLMs).

The Attention Sink Hypothesis. Recent studies on efficient streaming LLMs have proposed the *Attention Sink* hypothesis (Xiao et al., 2023), which suggests that attention heads tend to allocate massive attention scores to specific "sink" tokens (often initial tokens or special separators) to maintain internal state stability and aggregate historical context. In the specific context of Chain-of-Thought (CoT) reasoning, the transition from a *reasoning phase* to an *answering phase* necessitates a semantic boundary—a token or sequence that compresses the preceding divergent rationale into a converged state for answer generation.

Function over Form. While DeepSeek-R1 explicates this boundary as `</think>`, we argue that other models employ implicit equivalents to fulfill the same functional role. For instance, standard base models often rely on linguistic markers such as "*Therefore*", "*So*", or "*Thus*" to signal this state shift. similarly, instruction-tuned models may utilize special tokens like `<|start_header_id|>`

or specific role separators. Consequently, the `</think>` token in our **SyncThink** framework essentially functions as a specialized *Reasoning Attention Sink*. It signals the phase shift from a high-entropy *exploration state* to a low-entropy *determination state*.

This theoretical perspective implies that **SyncThink** is generalizable to a broader range of CoT-capable models. The core methodology—detecting the saturation of reasoning information via a bottleneck token—remains valid, provided one identifies the model-specific transition token (the "sink"), which distinguishes itself only in surface form rather than functional utility.

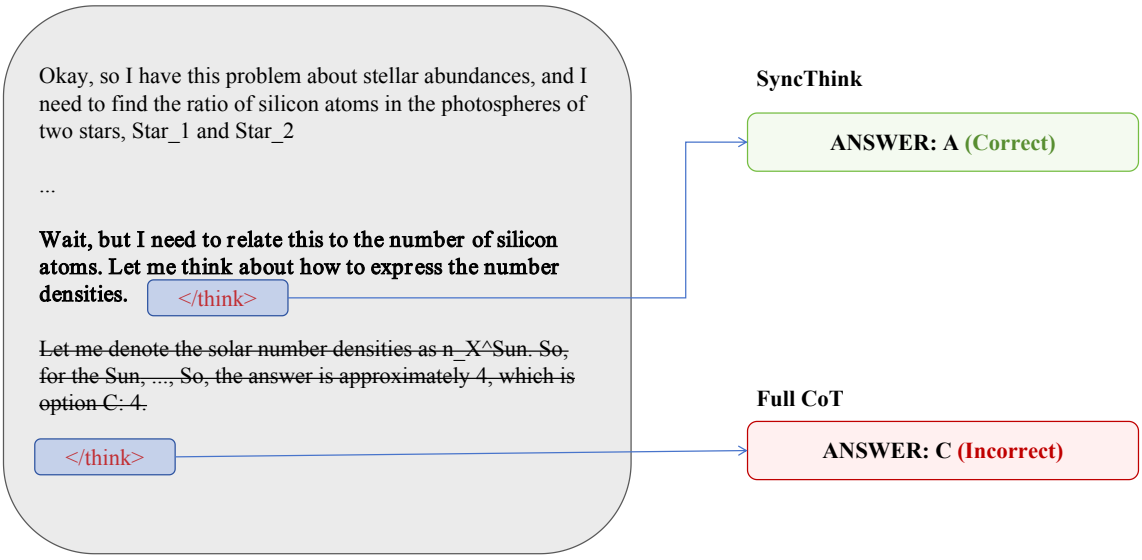
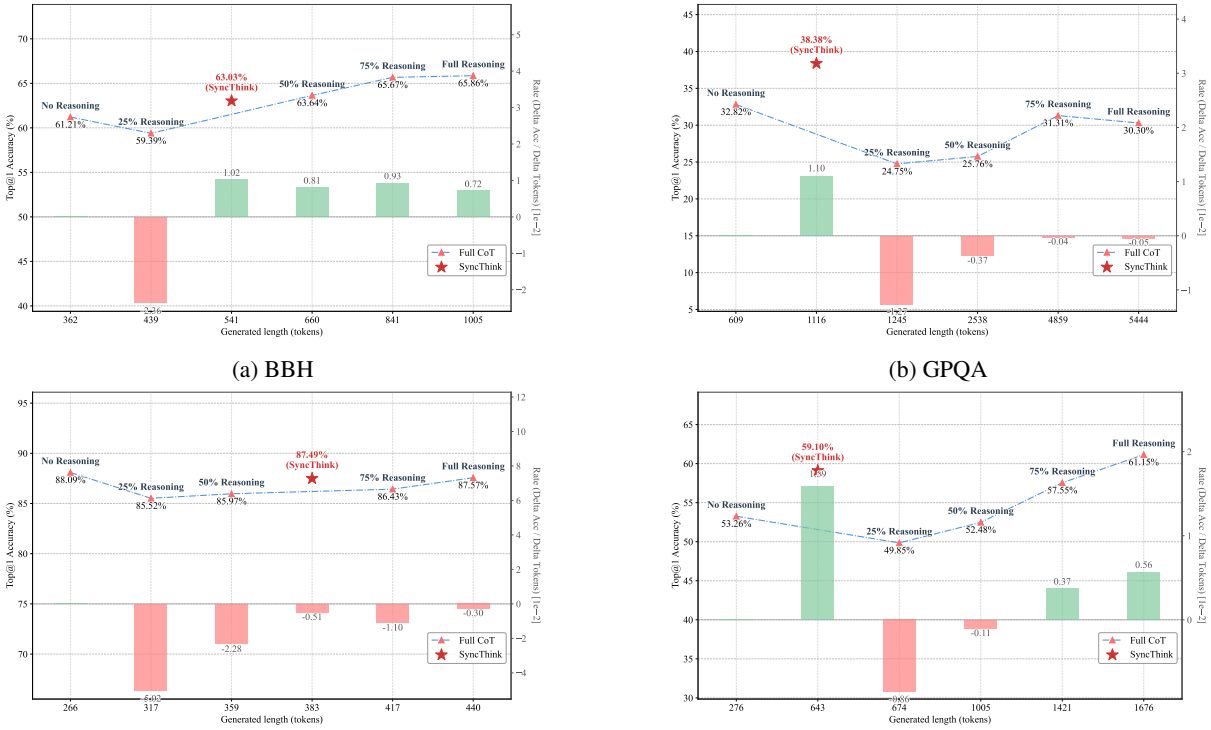


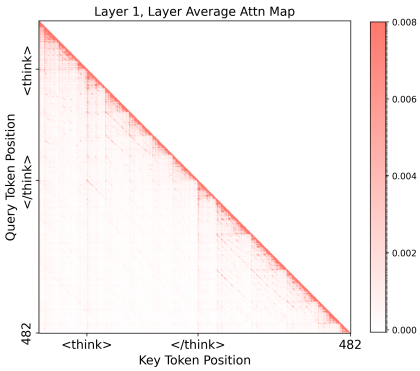
Figure 6: Case study on preventing model over-thinking. While the Full CoT path drifts into erroneous reasoning after a correct intermediate state, our SyncThink method detects the onset of this detrimental redundancy. By truncating the generation early, our method successfully avoids the subsequent error and retains the correct answer.



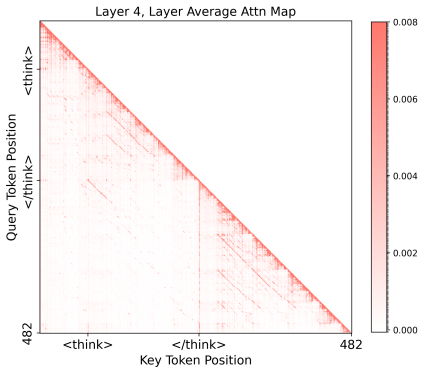
(a) BBH (b) GPQA

(c) GSM8K (d) MMLU

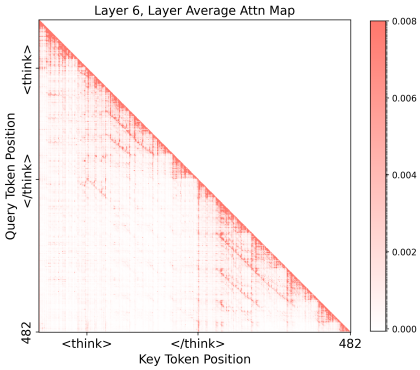
Figure 7: **Accuracy–length trade-off of SyncThink on DeepSeek-R1-Distill-Qwen-7B.** Blue curves denote fixed-ratio truncation baselines on full CoT traces, while the red star marks **SyncThink**. Across BBH, GPQA, GSM8K, and MMLU, **SyncThink** consistently lies in the upper-left region, achieving higher accuracy with fewer generated tokens, with the largest gain on GPQA.



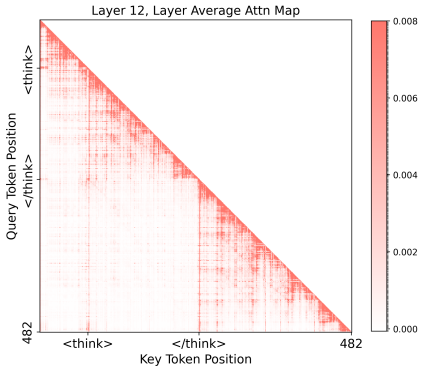
(a) Layer 1



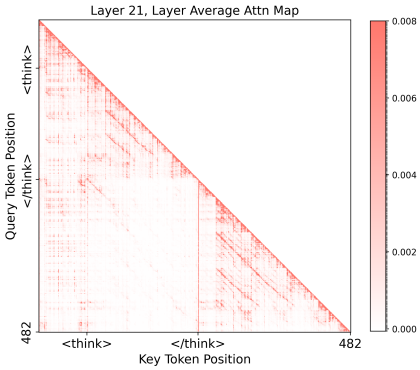
(b) Layer 4



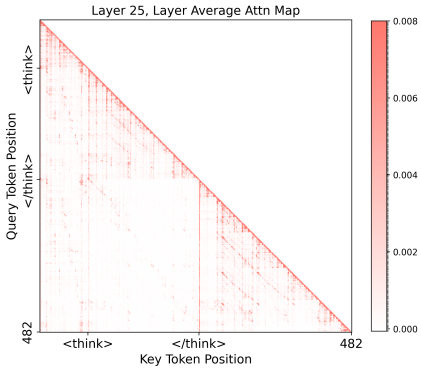
(c) Layer 6



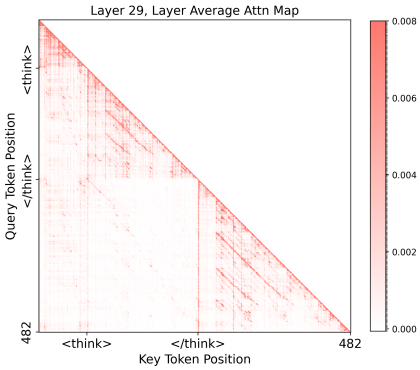
(d) Layer 12



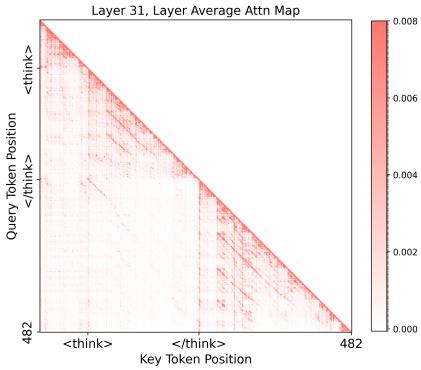
(e) Layer 21



(f) Layer 25



(g) Layer 29



(h) Layer 31

Figure 8: Attention heatmaps across different layers of DeepSeek-R1-Distill-LLaMA-8B on a GSM8K sample (Cobbe et al., 2021). Tokens within the <think>...</think> span receive uniform attention in early layers, but deeper layers gradually shift focus to the boundary tokens, indicating information migration and compression of reasoning content. Similar observations can be found in other models and datasets