State of Thought Enables Endogenous Reasoning

Zhiren Gong[1,2] **Zihao Zeng**[1,*] **Yikun Hou**[1,5] **Ming Xiao**[4]
Chau Yuen[3] **Wei Yang Bryan Lim**[1]
[1]College of Computing and Data Science, Nanyang Technological University, Singapore
[2]Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[4]Information Science & Engineering, KTH Royal Institute of Technology, Sweden
[5]Department of Mathematics and Mathematical Statistics, Umea University, Sweden

Code Project Page Tutorial

Abstract

Test-time compute has emerged as a major approach to improving the capabilities of Large Language Models (LLMs). However, existing test-time reasoning paradigms rely heavily on externally imposed control, either through fixed reasoning programs or through costly expansion in constrained search spaces, limiting both generalization and efficiency. We propose **State of Thought (SoT)**, a new reasoning paradigm that enables **endogenous reasoning** in LLMs, with the model's internal reasoning state governing how reasoning unfolds. Concretely, SoT extracts a compact *dynamics-geometric state* from the model's internal information transfer and uses a 582-parameter controller on frozen backbones to *selectively activate historical reasoning support* useful under the current reasoning state, framing reasoning as a state-conditioned process over evidence rather than an externally prescribed token chain. Across quantitative ($1.34\times$), general ($1.62\times$), symbolic-and-code ($1.76\times$), and long-context ($2.51\times$) reasoning on 3 LLMs and 16 datasets, SoT consistently improves mean-baseline accuracy while reducing generated tokens by 62.6% and end-to-end latency by 44.6%. Across 2 VLM scales and 3 reasoning tasks, it improves mean accuracy by 3.8 points over reasoning baselines, with 74.9% fewer completion tokens and 73.5% lower latency than search-based methods. Under constrained access, SoT retains 38.2%/36.5% mean accuracy gains in training-free/embedding-only settings, while trajectory-only judging reaches 78.1% agreement across 3 API models. Together, endogenous state-driven reasoning provides a generalizable and efficient alternative.

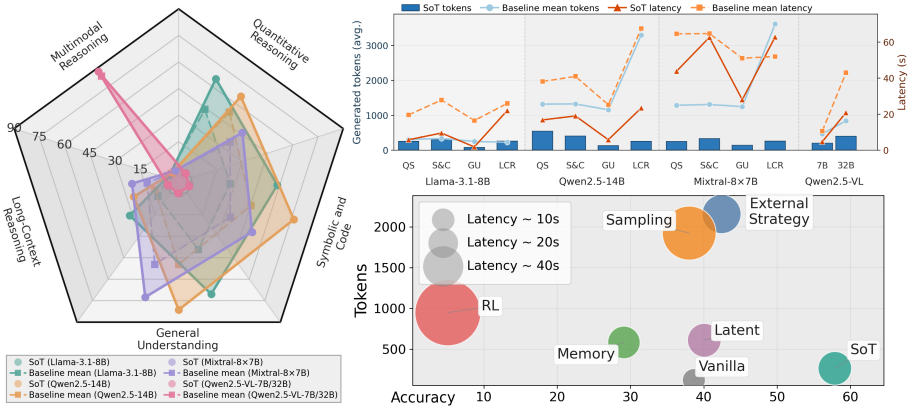


Figure 1: SoT Performance Across Models, Tasks, and Efficiency Metrics.

*Corresponding author: zihao.zeng@ntu.edu.sg

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

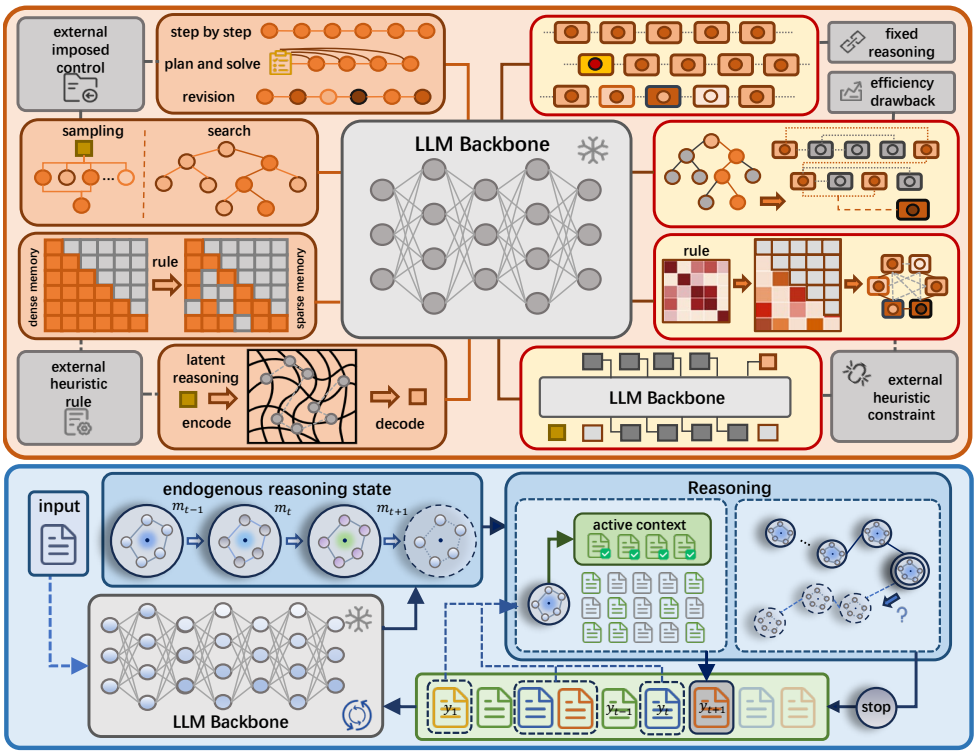


Figure 2: From External Control (Top) to State-of-Thought (Bottom).

1 Introduction

Reasoning in Large Language Models (LLMs) is fundamentally the process of organizing historical evidence context for subsequent decisions [Press et al., 2023, Trivedi et al., 2023, Yao et al., 2023b]. From this perspective, progress in test-time compute comes from better strategies for organizing the evidence context with additional computation, thereby improving reasoning capabilities [Brown et al., 2024, Snell et al., 2024]. One dominant line relies on externally structured reasoning programs, requiring the model to follow pre-specified reasoning formats such as step-by-step [Wei et al., 2022], planning [Wang et al., 2023a], or revision [Madaan et al., 2023]. Another line allocates more test-time search, exploring additional samples [Brown et al., 2024, Wang et al., 2023b], branches [Yao et al., 2023a], or trajectories [Snell et al., 2024] to accumulate broader candidate evidence for subsequent reasoning. Despite their empirical success in adapting to particular reasoning strategies, both lines impose reasoning control from outside the model (i.e., through pre-specified formats or search policies) and ignore its evolving internal state as a signal for what evidence is actually needed at subsequent decisions, limiting generalization across heterogeneous reasoning structures [Hassid et al., 2025, Gong et al., 2026b] and efficiency under constrained inference [Zhou et al., 2023].

Prior work has shown that LLMs internals exhibit structured representation geometry [Zou et al., 2023], trajectory-level regularities [Sun et al., 2026], and confidence dynamics [Tikhonov et al., 2026]. However, these signals are typically used descriptively, such as monitoring and reasoning-trajectory interpretation [Ballon et al., 2026, Damirchi et al., 2026, Feng et al., 2025, Gong et al., 2026a], or early stopping criteria [Hosseini et al., 2026], rather than to govern reasoning itself. Together, these observations point to the significance of the model's *endogenous state* (i.e., the internally evolving state induced by its own reasoning trajectory). In parallel, reasoning also depends critically on whether decisions are steadily grounded in suitable and decision-relevant historical evidence rather than generic or noisy history [Creswell et al., 2022, Press et al., 2023, Wang et al., 2024]. Existing approaches address this through context retrieval and compression [Jiang et al., 2024b, Li et al., 2023, Packer et al., 2023], or latent-space reasoning [Hao et al., 2025, Wang et al., 2025], both of which bypass endogenous state as a vital signal for organizing the evidence genuinely needed for subsequent steps. Appendix R provides further related discussion.

This motivates a reframing of reasoning itself: *effective reasoning is actually a dynamic process of organizing the right evidence context for the next decision under the current endogenous state, rather than following an externally prescribed token chain.* We introduce **State of Thought (SoT)**, a reasoning paradigm for *endogenous reasoning*. At each reasoning step, SoT reads out a compact dynamics-geometric state from the model's internal information transfer and uses a lightweight 582-parameter controller to selectively activate the historical evidence context most useful for the next decision. The active reasoning context is therefore not generated by pre-specified reasoning programs or broader searches, but by the model's own evolving state. In this way, SoT turns reasoning into a closed loop: endogenous state organizes active evidence, which conditions the next reasoning step, and the new step in turn updates the state.

We evaluate SoT at scale on 3 LLM backbones over 16 datasets and 2 VLM scales over 3 reasoning tasks. Across the 16-dataset LLM suite, SoT improves accuracy by 89.8%, 58.6%, and 46.0% over the mean baselines on Llama-3.1-8B, Qwen2.5-14B, and Mixtral-8x7B, while reducing tokens/latency by 19.7%/56.5%, 81.1%/62.2%, and 86.8%/15.1%, respectively. Across Qwen2.5-VL-7B/32B and 3 reasoning tasks, SoT improves mean accuracy by 3.8 points over reasoning baselines, leads in 5 of 6 model–task settings, and uses 74.9% fewer completion tokens with 73.5% lower latency than search-based methods. As exploratory extensions under constrained access, a training-free controller and an embedding-only controller still achieve 38.2% and 36.5% improvement over baselines on accuracy across models, while a trajectory-level post-hoc judge reaches 78.1% agreement on three API-accessed models. Together, these results suggest that changing the control variable of reasoning—from external token programs to endogenous state-conditioned evidence organization—offers a scalable and robust route to reasoning improvement.

2 State of Thought

2.1 Problem Formulation

We formulate test-time reasoning of a frozen autoregressive LLM f as a policy-governed multi-step trajectory generation started from a problem x presented by prompt p [Brown et al., 2024]. The model f dynamically generates the subsequent reasoning trajectory $y_{1:T} = (y_1, \ldots, y_T)$ leading to the final answer a:

$$\mathbb{P}_\pi(a, y_{1:T} \mid p) = \prod_{t=1}^{T} \mathbb{P}_\pi(y_t \mid p, y_{<t})\, \mathbb{P}_\pi(a \mid p, y_{1:T}), \tag{1}$$

with the policy π determining how evidence from prior steps is organized for subsequent decisions, while different reasoning paradigms differ in prompt construction and policy [Snell et al., 2024].

State of Thought seeks such a policy that the model's current endogenous state (Sec. 2.2) governs both evidence-context organization and reasoning continuation (Sec. 2.3), with no external reasoning instructions imposed on the prompt. This creates a direct bridge between *continuous* internal evolution and *discrete*, inspectable reasoning steps: the continuous state controls which explicit evidence is carried forward, without replacing language-space reasoning by an opaque latent chain.

2.2 Endogenous Reasoning State

Effective reasoning policy π should be governed not by external scripts but by a compact endogenous state that reflects the geometry and dynamics of ongoing reasoning [Park et al., 2025, 2024, Zou et al., 2023]. Accordingly, we read out a compact *dynamics-geometric state* $m_t \in \mathbb{R}^4$ from the model's internal information transfer of sentence-level reasoning unit y_t, with calculation details and design rationale in Appendix B:

$$m_t = \big[\delta_t,\ v_t,\ c_t,\ H_t\big], \tag{2}$$

where δ_t characterizes the internal organization distinguishing concentrated from diffuse local structure; v_t captures stepwise progress through magnitude; c_t captures directional consistency distinguishing stable continuation from deviation or redirection; and H_t measures the model's local predictive uncertainty. Together, these four quantities summarize the current reasoning endogenous control state and condition how the evidence context should be organized for subsequent decisions.

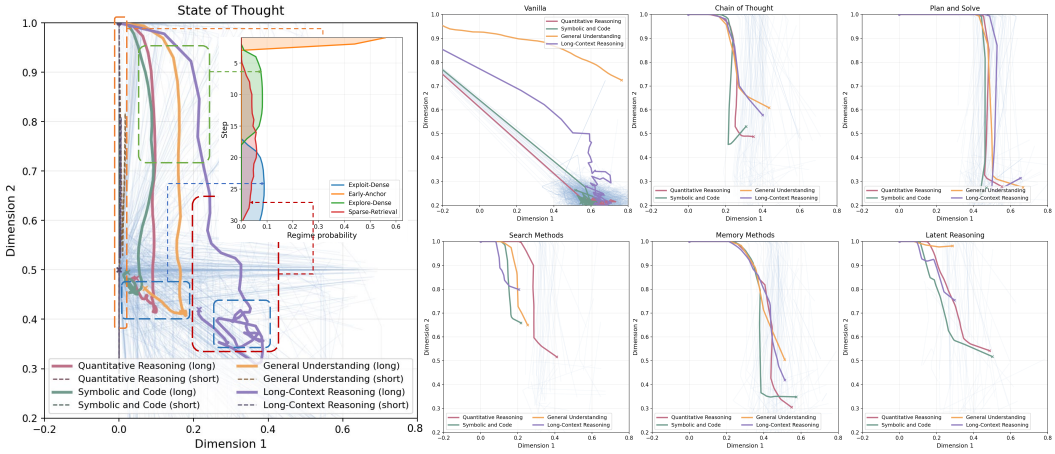


Figure 3: **Endogenous reasoning state trajectories across various reasoning paradigms** on Llama-3.1-8B with breakdown in Figure 8 and reasoning state definition in Figure 4. Different externally induced reasoning strategies occupy distinguishable regions in the dynamics-geometric state space, while SoT supports a broader and more adaptive state distribution.

Interpretability of endogenous reasoning state. To verify that m_t can capture reasoning-relevant internal structure rather than arbitrary hidden variation, we aggregate trajectories from multiple external reasoning paradigms in the constructed state space in Figure 3. SoT spans a broader and more adaptive state distribution, while others with externally structured reasoning programs occupy distinguishable regions, collapsing to fixed scripted patterns. This indicates that SoT is less constrained by externally imposed reasoning formats and more aligned with the model's endogenous state evolution, enabling stronger generalization across heterogeneous reasoning structures.

2.3 State-Conditioned Reasoning

SoT instantiates the reasoning policy π through evidence-organization operator $\mathcal{S}$ and stopping operator $\mathcal{T}$ as two state-conditioned operators with details in Appendix A.

State-conditioned evidence selection. Given the current state m_t and the historical reasoning trajectory $y_{<t}$, the evidence-organization operator $\mathcal{S}$ selects the reasoning-related sparse subset $\widetilde{y}_{<t} = \mathcal{S}(y_{<t}; m_t) \subset y_{<t}$ as the evidence context for subsequent reasoning to generate the next step

$$\mathbb{P}_\pi(y_t \mid p, y_{<t}) = \mathbb{P}(y_t \mid p, \widetilde{y}_{<t}), \tag{3}$$

In this way, SoT improves reasoning not by broadening search over more trajectories [Yao et al., 2023a], but by sharpening the evidence context carried into the next decision.

State-conditioned stopping. SoT also produces a continuation decision $z_t = \mathcal{T}(m_t) \in \{0, 1\}$, where $z_t = 0$ continues reasoning and $z_t = 1$ terminates reasoning and triggers final answer generation as

$$\mathbb{P}_\pi(a \mid p, y_{1:t}) = \mathbb{P}(a \mid p, \widetilde{y}_{<t}, y_t). \tag{4}$$

Stopping therefore emerges from the current reasoning endogenous state rather than from a fixed external schedule [Hosseini et al., 2026] on reasoning depth.

Overall, SoT treats evidence sparsification and stopping as two outputs of the same endogenous reasoning policy: one determines what evidence context remains active, and the other determines when sufficient support has been accumulated for answer generation.

Interpretability of state-conditioned reasoning. We aggregate reasoning trajectories, cluster endogenous states, and analyze the induced evidence-selection and stopping patterns in each state regime in Figure 4. We find that different state clusters consistently and stably correspond to distinct sparse activation profiles and stop probabilities, indicating that SoT organizes evidence context and continuation behavior in a state-dependent manner, supporting the core view that effective reasoning is governed by endogenous states rather than externally prescribed token chains. Cluster names summarize four-coordinate centroids and measured activation–stopping profiles, not task labels; Appendix O.2 details their interpretation.

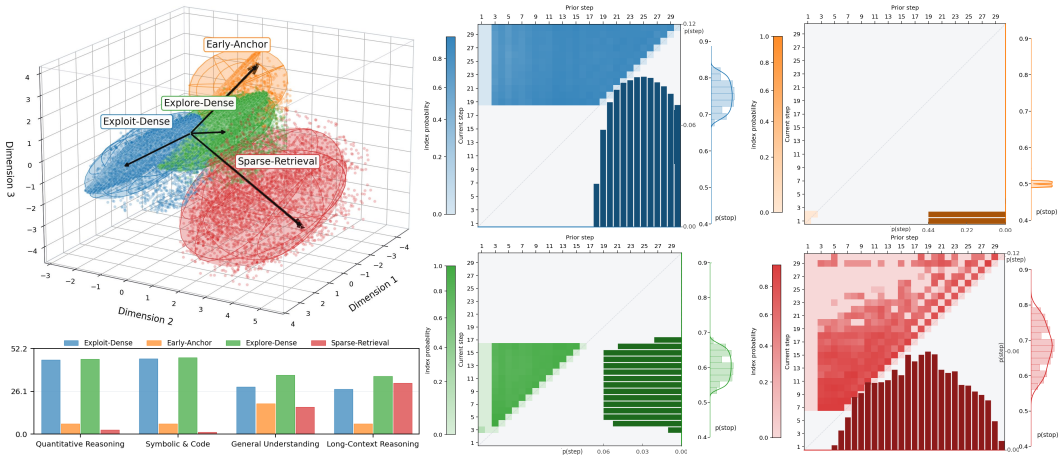


Figure 4: **State-conditioned reasoning in SoT** with breakdown in Figure 9. Different endogenous reasoning states induce different sparse activation patterns and different stop tendencies.

Lightweight training. SoT learns only the state-conditioned evidence-organization operator $\mathcal{S}$ and stopping operator $\mathcal{T}$, while the backbone remains frozen, with training details in Appendix C; Sec. 4.1 further removes this controller training. Supervision is constructed offline from multi-step reasoning trajectories: each endogenous state is paired with (i) preference-style signals indicating which historical evidence remains useful for subsequent reasoning, in the spirit of direct preference learning [Hong et al., 2024, Rafailov et al., 2023], and (ii) stop proxies indicating whether reasoning should continue, following findings that intermediate reasoning states can support learned early stopping [Liu and Wang, 2025]. This design is also consistent with context-selection and compression methods that learn lightweight selection modules from offline trajectories or contextual supervision rather than retraining the backbone model [Jiang et al., 2024b, Li et al., 2023]. The resulting controller has only 582 parameters (a 16×32 selection projection with a scalar output, plus a linear $4 \to 1$ stopping head); across 5 problem-held-out refits, its selector AUC is 0.779 ± 0.005. Thus, SoT learns a compact control interface rather than updating or duplicating backbone reasoning capacity.

3 Experiments

3.1 Experimental Setup

Backbone models. We evaluate SoT on three frozen *LLMs* with different scales and architectures (Llama-3.1-8B [Grattafiori et al., 2024], Qwen2.5-14B [Yang et al., 2024], and Mixtral-8x7B [Jiang et al., 2024a]) and transfer to two frozen VLM scales, Qwen2.5-VL-7B and Qwen2.5-VL-32B [Bai et al., 2025]. Full details are deferred to Appendix L.

Tasks and evaluation domains. For *LLMs*, we organize evaluation into *four reasoning domains* comprising *16 benchmarks*. **Quantitative Reasoning** (QS) includes GSM8K (GSM) [Cobbe et al., 2021], MATH [Hendrycks et al., 2021b], and DROP [Dua et al., 2019]. **General Understanding** (GU) includes CommonsenseQA (CSQA) [Talmor et al., 2019], StrategyQA (StrQA) [Geva et al., 2021], BoolQ [Clark et al., 2019], MMLU [Hendrycks et al., 2021a], and RACE [Lai et al., 2017]. **Symbolic and Code** (S&C) includes FOLIO (FOL) [Han et al., 2024], ProofWriter (PW) [Tafjord et al., 2021], BBH-Temporal (BBH) [Srivastava et al., 2023], HumanEval (HE) [Chen et al., 2021], and MBPP [Austin et al., 2021]. **Long-Context Reasoning** (LCR) includes HotpotQA (HQA) [Yang et al., 2018], NarrativeQA (NarQA) [Kočiskỳ et al., 2018], and LongBench MultiFieldQA (LB) [Bai et al., 2024]. For *VLM transfer*, we evaluate *three reasoning-oriented benchmarks* that require knowledge-grounded, diagrammatic, or multimodal chain-of-thought inference: A-OKVQA [Schwenk et al., 2022], AI2D [Kembhavi et al., 2016], and M^3CoT [Chen et al., 2024]. Full details are deferred to Appendix K.

Baselines. We compare SoT against *5 baseline families*. **Greedy:** Vanilla. **Reasoning paradigm:** Chain-of-Thought (CoT) [Wei et al., 2022], Plan-and-Solve (PS) [Wang et al., 2023a], Self-Refine (SR) [Madaan et al., 2023] with externally scripted programs, and Self-Consistency (SC) [Wang

Table 1: **Main results on Llama-3.1-8B (accuracy, %).**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Greedy	Vanilla	79.0	58.2	6.2	53.9	29.6	26.0	26.0	25.0	55.2	33.0
Reasoning Paradigm	CoT	80.8	64.2	1.7	55.5	4.9	24.5	70.0	25.0	51.6	31.2
	PS	71.2	40.0	1.4	43.4	4.9	15.2	55.0	10.0	22.0	17.6
	SR	75.8	44.8	15.6	50.4	32.5	31.0	46.0	25.0	36.4	32.9
	SC	83.6	54.0	1.6	53.2	16.8	28.5	52.0	10.0	35.2	27.3
	CB	48.0	36.5	19.8	37.1	40.4	35.8	34.0	30.0	49.2	38.6
	MCTS	55.2	33.5	13.2	37.5	36.0	31.5	30.0	35.0	49.2	36.7
Memory-Oriented	H2O	79.8	59.2	2.4	53.6	6.4	27.8	48.0	10.0	42.0	26.3
	SNAP	80.4	60.2	2.3	54.1	8.4	26.2	55.0	15.0	41.2	27.3
	STREAM	70.4	44.2	1.3	44.4	1.5	31.8	1.0	0.0	5.6	13.0
Latent	COCO	81.4	60.0	3.7	54.8	45.3	45.2	51.0	20.0	47.2	42.5
RL-Based	GRPO-SP	0.0	0.5	0.3	0.2	2.0	2.2	7.0	0.0	0.0	1.8
Ours	**SoT**	**81.8**	**64.8**	**40.5**	**65.8**	**49.8**	**64.8**	**78.0**	**59.1**	**46.8**	**58.4**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Greedy	Vanilla	48.0	68.8	59.2	45.6	50.7	54.2	7.0	20.1	32.7	16.2
Reasoning Paradigm	CoT	32.8	31.5	37.2	34.0	50.3	36.3	2.0	4.5	26.9	7.1
	PS	30.5	18.2	21.5	31.2	42.0	28.1	1.7	3.0	20.4	5.3
	SR	57.0	64.0	72.2	62.4	62.7	63.6	17.9	21.7	25.5	20.6
	SC	28.5	34.5	43.2	31.0	45.7	35.8	1.9	4.3	25.8	6.8
	CB	46.0	71.5	82.2	51.2	56.0	61.1	15.5	27.5	26.2	21.8
	MCTS	42.0	65.5	82.2	46.4	49.7	57.0	13.5	27.1	27.8	21.0
Memory-Oriented	H2O	33.5	30.8	39.2	35.0	19.7	32.4	2.0	2.0	0.0	1.7
	SNAP	33.5	31.0	38.5	34.6	16.7	31.8	2.0	2.0	0.0	1.7
	STREAM	34.8	26.2	38.8	22.4	0.0	25.6	2.5	0.0	0.0	1.1
Latent	COCO	40.5	63.5	70.0	42.8	43.3	52.0	4.4	10.8	34.6	11.9
RL-Based	GRPO-SP	10.5	3.8	9.8	8.2	12.0	8.7	0.6	1.1	5.3	1.6
Ours	**SoT**	**69.5**	**68.8**	**82.3**	**65.0**	**67.0**	**70.4**	**24.9**	**39.0**	**34.7**	**31.9**

Note. Other model results are reported in Table 12, 13. Cells colored with rankings as 1st , 2nd , and 3rd .

et al., 2023b], Best-of-N (BoN) [Snell et al., 2024], Constrained Beam (CB) [Hokamp and Liu, 2017], Monte Carlo Tree Search (MCTS) [Xie et al., 2024] with sampling or search methods. **Memory-oriented methods:** H_2O (H2O) [Zhang et al., 2023], SnapKV (SNAP) [Li et al., 2024], and StreamingLLM (STREAM) [Xiao et al., 2023]. **Latent reasoning:** COCONUT (COCO) [Hao et al., 2025]. **RL-based reasoning:** GRPO-inspired soft prompt (GRPO-SP) [Shao et al., 2024]. All learned components are fitted exclusively on training-side data, with evaluation examples excluded. Full details are deferred to Appendix M and N.

Evaluation metrics. We report task-standard accuracy metrics and efficiency cost. **Exact Match (EM)** is used for GSM8K, MATH, RACE, MMLU, CommonsenseQA, FOL, BBH, ProofWriter, BoolQ, StrategyQA, A-OKVQA, AI2D, and M^3CoT. **F1** is used for HotpotQA, DROP, NarrativeQA, and LongBench MultiFieldQA. **Pass@1** is used for HumanEval and MBPP. **Efficiency** is measured by generated tokens and end-to-end inference latency; hardware and timing details are provided in Appendix L.

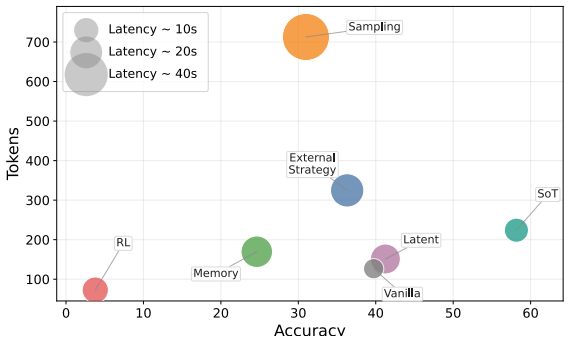


Figure 5: **Accuracy–efficiency trade-off** on Llama-3.1-8B with breakdown and other models in Figure 11.

3.2 Main Results

Overall performance. Table 1 reports the main results on Llama-3.1-8B across four reasoning domains. SoT achieves the best domain-level average in all domains. Compared with the strongest non-SoT baseline in each domain, it improves Quantitative Reasoning, Symbolic and Code, General Understanding, and Long-Context Reasoning by $+10.3/+15.9/+6.8/+10.1$ points, correspond-

Table 2: **Efficiency comparison on Llama-3.1-8B.**

Category	Method	QS		S&C		GU		LCR	
		Tok.	**Lat.**	**Tok.**	**Lat.**	**Tok.**	**Lat.**	**Tok.**	**Lat.**
Reasoning Paradigm	CoT	271.6	17.5	369.8	27.5	212.9	12.8	198.1	19.6
	PS	314.2	20.7	372.1	26.8	319.8	22.2	282.3	33.7
	SR	368.5	25.9	391.7	28.8	292.2	19.2	278.8	33.2
	SC	736.8	43.1	953.9	63.5	626.5	37.6	533.8	48.9
	CB	423.9	22.0	363.9	19.4	341.7	16.8	283.7	19.5
	MCTS	422.1	21.4	364.7	19.4	335.2	16.5	286.6	20.5
Memory-Oriented	H2O	226.0	16.7	264.4	34.6	185.6	15.8	164.6	21.5
	SNAP	226.0	16.6	264.4	34.8	185.6	16.1	164.6	20.9
	STREAM	92.6	12.5	64.2	24.0	93.4	9.8	103.1	10.3
Latent	COCO	171.5	11.5	214.6	20.4	87.2	6.7	131.5	30.3
RL-Based	GRPO-SP	73.1	7.5	69.7	7.1	71.4	7.5	76.2	26.9
Ours	**SoT**	**255.7**	**5.8**	**302.2**	**9.5**	**74.9**	**1.8**	**263.0**	**22.0**

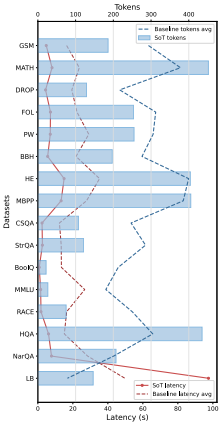


Figure 6: **Distribution.**

Note. Generated reasoning tokens (**Tok.**) and end-to-end latency (**Lat.**) are aggregated over four reasoning domains, with full breakdowns in Tables 14 and 15. Results for the other models appear in Tables 16, 17, 18, and 19.

ing to relative gains of $18.6\%/37.4\%/10.7\%/46.3\%$. It also obtains the best or tied-best result on 13 out of 16 datasets, with an average domain-level gain of 10.8 points over the strongest competing methods. This pattern suggests that reasoning gains need *not* come primarily from prescribing a stronger external reasoning program or from scaling search over more trajectories. A closed-loop controller driven by the model's own endogenous reasoning state can itself serve as an effective mechanism for improving test-time reasoning. We further demonstrate the stable performance gains of our method through quality sensitivity experiments on the training dataset in the Appendix P.5.

Generalization. A distinctive property of SoT is that the same controller can be applied across heterogeneous reasoning structures with stable performance. Compared with scripted reasoning, SoT avoids committing to a fixed external reasoning form; compared with sampling and search, it improves reasoning without relying mainly on broader trajectory expansion; compared with memory-oriented and latent reasoning baselines, it organizes historical evidence through the model's evolving endogenous state rather than generic context heuristics or hidden-space-only reasoning. Consistent gains across these regimes therefore indicate *mechanism-level generalization*: SoT transfers as a reasoning principle, not merely as a benchmark-specific strategy.

In general, these patterns suggest a more structural interpretation of the results. Within the family of externally prescribed token-chain reasoning methods, performance gains are typically obtained by strengthening scripts, increasing samples, or enlarging search. By contrast, SoT improves reasoning by changing the control variable itself, shifting from external token programs to endogenous state-conditioned evidence organization. This may explain why it can surpass strong token-chain baselines across heterogeneous regimes. Appendix F to J provide in-depth theoretical analysis.

3.3 Efficiency and Trade-off

Efficiency is central to inference-time reasoning because most performance gains come from expanding computation through longer trajectories, repeated sampling, or explicit search. Instead, SoT activates only the evidence context most suitable under the current endogenous state, so its advantage should appear not only in accuracy but also in efficiency.

Efficiency. Table 2 and Figure 6 report generated reasoning tokens and end-to-end latency on Llama-3.1-8B. Averaged over the four reasoning domains, SoT uses 223.9 tokens and 9.8s latency, compared with 472.7 tokens and 29.0s latency for sampling/search methods, reducing tokens by 52.6% and latency by 66.4%. The saving is especially clear in QS, S&C, and GU, where SoT attains strong accuracy with only 5.8s, 9.5s, and 1.8s latency, respectively. Compared with memory-oriented methods, however, SoT is not merely a more aggressive compression rule: generic pruning can reduce tokens, but often sacrifices accuracy because it does not condition retained evidence on the current reasoning state. Thus, SoT improves efficiency by removing unnecessary computation while preserving the support needed for the next decision.

Table 3: **Ablation results on Llama-3.1-8B,** with other models provided in Tables 20, 21.

Variant	QS			S&C			GU			LCR		
	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.
Full SoT	62.8	219.4	5.8	54.5	294.0	9.5	71.2	64.0	1.8	33.0	181.6	22.0
+ Threshold Tuning	62.8	219.4	5.8	54.3	294.0	9.5	71.2	64.0	1.8	31.8	229.0	22.0
w/o Evid. Org. $\mathcal{S}$	29.6	469.5	12.9	44.5	536.2	16.5	72.4	193.9	4.9	26.9	506.5	14.1
w/o Stopping $\mathcal{T}$	60.6	1769.6	80.7	57.1	1705.1	110.3	70.2	4148.5	99.5	31.6	2937.4	157.3
w/o Geometry δ_t	42.5	68.6	3.3	46.4	48.0	2.9	73.0	25.4	0.9	24.5	47.2	2.2
w/o Dynamics (v_t, c_t)	57.0	1039.2	29.1	46.2	846.7	25.7	71.2	1272.3	33.7	31.6	1000.3	30.8
w/o Uncertainty H_t	54.1	244.9	6.4	52.1	281.8	8.8	73.6	61.3	1.9	32.8	555.5	16.6
w/o Sent.-level Units	54.1	244.9	7.2	52.1	281.8	8.8	73.6	61.3	1.9	35.1	224.1	11.9

Table 4: **Vision–language reasoning across scales** (accuracy, %; completion tokens; latency, s).

Method	Qwen2.5-VL-7B									Qwen2.5-VL-32B								
	A-OKVQA			AI2D			M^3CoT			A-OKVQA			AI2D			M^3CoT		
	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.
CoT	81.0	137	3.0	82.7	189	3.9	80.7	178	4.4	84.0	267	12.8	87.3	321	17.0	87.3	329	16.0
PS	84.0	169	3.5	82.7	245	5.9	81.3	211	5.8	80.5	311	16.9	72.7	436	27.1	78.7	387	17.9
SR	78.0	213	5.1	80.0	264	6.1	83.3	254	6.4	84.5	414	22.2	89.3	465	26.1	84.0	469	24.3
SC	83.5	691	15.9	87.3	955	19.9	84.0	909	21.3	84.0	1319	64.0	86.0	1669	87.6	87.3	1618	78.2
BoN	84.5	691	14.9	**88.7**	961	21.7	86.7	919	21.0	85.0	1322	65.5	87.3	1680	88.8	85.3	1633	80.0
SoT	**87.5**	191	4.4	86.7	215	5.4	**87.3**	198	4.5	**85.5**	425	23.2	**90.0**	364	19.2	**88.0**	409	20.1

Accuracy–efficiency trade-off. Figure 5 further shows that SoT moves the accuracy–efficiency frontier rather than simply trading accuracy for shorter outputs. It reaches the highest accuracy while using far fewer tokens and lower latency than external strategies and sampling/search baselines. This pattern clarifies the core advantage of SoT: test-time compute is no longer spent on uniformly longer chains, larger candidate sets, or generic history retention, but is redirected toward state-conditioned evidence organization. The efficiency results therefore reinforce the main claim that endogenous reasoning state provides a better control signal for both reasoning quality and inference cost.

3.4 Ablation Studies

The ablations in Table 3 indicate that the components of SoT interact in a way that controls distinct reasoning failure modes rather than acting as independent optimizations. For instance, removing the evidence-organization operator S leads to a significant drop in performance, especially in QS (from 62.8 to 29.6) and LCR (from 33.0 to 26.9), highlighting the critical role of state-conditioned evidence selection. Conversely, removing the stopping operator T improves S&C (from 54.5 to 57.1), but this comes at the expense of drastically increased tokens (from 189.8 to 2640.2) and latency (from 9.8s to 112.0s), making the improvement unsustainable due to excessive computation. Ablating state components such as geometry δ_t, dynamics (v_t, c_t), and uncertainty H_t shows that they contribute complementary signals: removing geometry δ_t reduces performance (from 62.8 to 42.5 in QS), while removing dynamics (v_t, c_t) increases computational cost and leads to a drop in LCR accuracy (from 33.0 to 31.6). Removing uncertainty H_t results in only a minor performance change in most domains (from 54.5 to 52.1 in S&C), but still weakens the overall accuracy–cost balance. Finally, replacing the sentence-level reasoning units does not significantly improve performance, but suggests that the default sentence-level granularity provides stable control across domains. SoT's strength lies in the synergy between its components, rather than in any single one.

3.5 Transfer to Vision-Language Reasoning

Across these 6 model–reasoning-task settings, SoT averages 87.5% accuracy and is best in 5. On 7B AI2D, it is 2.0 points below BoN but uses $4.5\times$ fewer completion tokens. At 32B, it exceeds the strongest alternative by 0.5–0.7 points on all 3 tasks while using 3.1–$4.6\times$ fewer tokens than search-based methods. Across both scales, it also reduces end-to-end latency by 73.5% relative to SC and BoN. The gains across knowledge-grounded, diagrammatic, and multimodal chain-of-thought tasks show that state-conditioned evidence organization transfers beyond text-only reasoning and persists across scale without expanding the controller or relying on wider sampling.

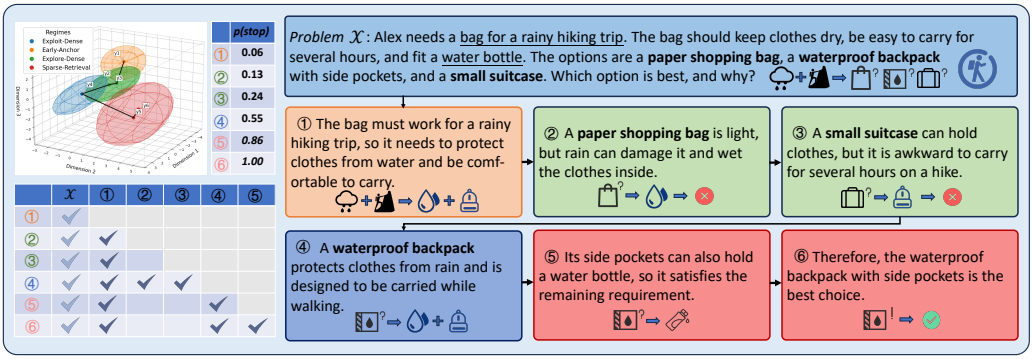


Figure 7: **A case study of SoT,** with more case studies in Appendix Q.

3.6 Case study

Figure 7 shows a representative example with activated historical evidence at each step, and the corresponding stop probability. SoT does not keep the full trajectory active, nor does it rely on a fixed external reasoning template. Instead, as constraints are progressively resolved, the endogenous state selectively retains the evidence still needed for the next decision, suppresses obsolete support, and raises stop readiness only when the remaining context becomes sufficient. This example therefore concretely illustrates the closed-loop view of SoT: state organizes support, support shapes the next step, and the updated step in turn changes the state.

Overall findings. Across accuracy, efficiency, ablations, scale transfer, and case evidence, SoT improves reasoning by reallocating existing inference computation rather than enlarging search or backbone capacity. Its endogenous state acts as a compact control interface that couples evidence retention with termination, translating continuous internal evolution into discrete, inspectable decisions. This shared mechanism explains why the same lightweight design advances the accuracy–efficiency frontier across heterogeneous text tasks and multimodal scales.

4 Exploratory Extensions of State of Thought

4.1 SoT under Limited Access

Training-Free SoT. We first ask *whether SoT requires learned control at all*. We instantiate the evidence-organization operator $\mathcal{S}$ and stopping operator $\mathcal{T}$ without training, using fixed rules derived from the same dynamics-geometric state. Historical evidence is selected by state compatibility with the current reasoning regime, while stopping is triggered when the trajectory becomes sufficiently stable and low-uncertainty. This yields a fully training-free SoT controller that preserves the same closed-loop formulation while removing learned control entirely.

SoT-Embed under Limited Access. We next ask *whether SoT depends on privileged access to internal information transfer*. We embed the model's multi-step segmented output with a fixed encoder and feed the embedding trajectory into the control loop in place of the internal state. This tests whether endogenous reasoning control remains viable when its observable interface is restricted to text.

SoT-Training-free evaluates whether the benefits of SoT come solely from the learned controller, while SoT-Embed tests whether it relies on white-box access. Table 5 summarizes that both variants maintain competitive performance, with macro-averaged accuracy lower relative to full SoT by approximately 17.5% for SoT-Training-free and 17.9% for SoT-Embed. In comparison, the Top-3 baseline methods show an average accuracy of 47.5% on those cells, with SoT variants achieving comparable or slightly better results in many tasks. These results suggest that SoT's core advantage lies in the principle of state-conditioned evidence organization, not in a specific implementation. Despite these reductions relative to full SoT, both variants remain highly competitive, often outperforming the Top-3 baselines on this Llama table. For further details, see Appendix D.

Table 5: **Boundary studies of SoT on Llama-3.1-8B,** with other models provided in Tables 22, 23.

	Quantitative Reasoning				Symbolic and Code					
	GSM8K	MATH	DROP	avg.	FOLIO	PW	BBH-T	HE	MBPP	avg.
Top-3 Baseline avg.	81.9	61.5	16.2	53.2	40.6	37.6	60.0	30.0	52.0	44.0
SoT-Training-free	81.8	60.5	34.6	59.0	43.4	30.2	77.0	53.0	50.8	50.1
SoT-Embedding	82.8	61.8	32.3	59.0	33.8	30.5	74.0	57.9	52.8	49.8

	General Understanding						Long-Context Reasoning			
	CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Top-3 Baseline avg.	50.3	68.6	78.9	53.3	56.5	61.5	15.6	25.4	31.7	24.2
SoT-Training-free	37.5	55.2	70.0	45.4	53.0	52.0	21.8	34.2	24.8	26.9
SoT-Embedding	37.5	56.8	71.2	41.8	52.0	51.9	28.5	30.3	24.8	27.9

4.2 SoT as a Trajectory-Level Judge

We further study *whether information visible only in the completed reasoning trace can support black-box quality assessment*, without online control or access to hidden states. Given sentence-segmented model outputs, we map each step to a standard sentence embedding and summarize the resulting sequence using compact trajectory statistics, with details in Appendix E. A small trainable predictor (SoT-Judge) then estimates whether the final answer matches reference correctness. The goal is a clean trajectory-only readout of reasoning health, parallel in spirit to SoT's use of trajectory organization, but reduced to post-hoc inputs that any API model exposes. We evaluate three closed or API-accessed models: GPT-5.4 [OpenAI, 2026], Claude Opus 4.7 [Anthropic, 2026], and Qwen-Max [Alibaba Cloud Model Studio, 2026].

Table 6: **Black-box judging with trajectory-based SoT** (agreement with reference correctness).

Model	Method	QR	GU	S&C	LCR	avg.
GPT-5.4	Self-Consistency	58.3	81.4	79.2	16.7	58.9
	Self-Verification	58.3	80.6	78.2	16.0	58.3
	LLM-as-a-Judge	58.3	82.0	79.2	16.3	59.0
	SoT-Judge	**86.0**	**70.0**	**69.8**	**81.0**	**76.7**
Claude Opus 4.7	Self-Consistency	56.3	83.6	75.2	5.0	55.0
	Self-Verification	59.4	62.7	80.3	9.3	52.9
	LLM-as-a-Judge	56.7	83.6	75.4	5.3	55.2
	SoT-Judge	**86.3**	**64.8**	**67.0**	**99.0**	**79.3**
Qwen-Max	Self-Consistency	57.9	84.0	80.0	14.5	59.1
	Self-Verification	29.3	86.5	77.9	11.6	51.3
	LLM-as-a-Judge	59.3	84.8	79.8	14.8	59.7
	SoT-Judge	**83.6**	**77.4**	**69.6**	**82.2**	**78.2**

We compare against three representative black-box judging strategies, including Self-Consistency [Wang et al., 2023b], Self-Verification [Weng et al., 2023], and LLM-as-a-Judge [Zheng et al., 2023], covering sample-based agreement, self-verification, and external judging. By contrast, SoT-Judge predicts correctness only from the embedded sentence trajectory, without extra rollouts, self-critique passes, or stronger external judge calls. Table 6, with breakdown in Table 24, shows that SoT-Judge improves the average agreement over the strongest baseline by $+17.7$, $+24.1$, and $+18.5$ points on GPT-5.4, Claude Opus 4.7, and Qwen-Max, respectively. The gain is not uniform across domains: on GU and S&C, trajectory-only judging is weaker than the strongest standard judges, while the largest improvements come from QR and especially LCR, where agreement of black-box baselines drops to 5.0–16.7% but SoT-Judge remains at 81.0–99.0%. This suggests that trajectory structure provides a useful correctness signal, especially when black-box judging becomes unreliable under sparse or long-range evidence.

5 Conclusion

We presented State of Thought (SoT), a reasoning paradigm that changes the control variable of test-time reasoning from external token programs to endogenous state-conditioned evidence organization. Rather than prescribing longer chains, larger search budgets, or fixed memory rules, SoT uses the model's evolving representational state to dynamically gate which evidence remains active and when reasoning should terminate. This closed-loop formulation yields measurable gains on the accuracy–efficiency frontier across text and multimodal benchmarks, and generalizes robustly to limited-access and trajectory-level evaluation settings. By using a compact continuous state to govern discrete, inspectable reasoning steps, SoT connects latent internal evolution with explicit reasoning without sacrificing either controllability or interpretability. Our results point to a broader principle: reasoning systems need not rely on increasingly elaborate external scaffolding — adaptivity and efficiency can emerge directly from the model's internal state geometry.

Acknowledgments and Disclosure of Funding

This research / project is supported by A*STAR under its Japan–Singapore Joint Call: JST–A*STAR 2024 (Project ID: R2416IR139). This research is also supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award MOE-T2EP20125-0005). Yikun Hou was supported by the Wallenberg AI, Autonomous Systems, and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

Competing Interests

The authors declare no competing interests.

References

Alibaba Cloud Model Studio. Model list. Alibaba Cloud Documentation, 2026. URL `https://help.aliyun.com/zh/model-studio/models`. Official model catalog documenting Qwen-Max.

Anthropic. Claude opus 4.7. Anthropic Documentation, 2026. URL `https://docs.anthropic.com/en/docs/about-claude/models/overview`. Official model documentation.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

Marthe Ballon, Brecht Verbeken, Vincent Ginis, and Andres Algaba. Probing the trajectories of reasoning traces in large language models. *arXiv preprint arXiv:2601.23163*, 2026.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which LLM reasoning steps matter? In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL `https://openreview.net/forum?id=VnSlfeRCaU`.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

Hamed Damirchi, Ignacio Meza De la Jara, Ehsan Abbasnejad, Afshar Shamsi, Zhen Zhang, and Javen Shi. Truth as a trajectory: What internal representations reveal about large language model reasoning. *arXiv preprint arXiv:2603.01326*, 2026.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of NAACL-HLT*, pages 2368–2378, 2019.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Zhiren Gong, He Lu, Tiantong Wang, Yichi Zhang, Yixin Wang, Zihao Zeng, Ming Xiao, Chau Yuen, and Wei Yang Bryan Lim. Conditional co-ablation: Recovering self-repair backups in transformer circuits, 2026a. URL `https://arxiv.org/abs/2607.01940`.

Zhiren Gong, Tiantong Wu, Jiaming Zhang, Fuyao Zhang, Che Wang, Yurong Hao, Yikun Hou, Foo Ping, Yilei Zhao, Fei Huang, Chau Yuen, and Wei Yang Bryan Lim. XDomainbench: Diagnosing reasoning collapse in high-dimensional scientific knowledge composition. In *Forty-third International Conference on Machine Learning*, 2026b. URL `https://openreview.net/forum?id=U8x5SYtT5b`.

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

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=Itxz7S4Ip3`.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 34:5327–5344, 2021b.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, 2017.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.626. URL `https://aclanthology.org/2024.emnlp-main.626/`.

Parsa Hosseini, Sumit Nawathe, Mahdi Salmani, Meisam Razaviyayn, and Soheil Feizi. Early stopping for large reasoning models via confidence dynamics. *arXiv preprint arXiv:2604.04930*, 2026.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, 2024b.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, 2016.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of EMNLP*, pages 785–794, 2017.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6342–6353, 2023.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Xin Liu and Lu Wang. 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, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.904. URL https://aclanthology.org/2025.emnlp-main.904/.

Zhichen Liu, Yongyuan Li, and Yang Xu. Think in sentences: Explicit sentence boundaries enhance language model's capabilities. *arXiv preprint arXiv:2604.10135*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

OpenAI. Introducing gpt-5.4. OpenAI, 2026. URL https://openai.com/index/introducing-gpt-5-4/. Official model announcement.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: Towards llms as operating systems. 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR, 2024.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. URL https://arxiv.org/abs/2305.18290.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision (ECCV)*, 2022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Llm reasoning as trajectories: Step-specific representation geometry and correctness signals. *arXiv preprint arXiv:2604.05655*, 2026.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.

Pavel Tikhonov, Ivan Oseledets, and Elena Tutubalina. Confidence leaps in llm reasoning: Early stopping and cross-model transfer. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 602–616, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, 2023.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, 2023a.

Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. System-1.5 reasoning: Traversal in language and latent spaces with dynamic shortcuts. *arXiv preprint arXiv:2505.18962*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, 2023.

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

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Dayiheng Fan, Fei Huang, Haoran Wei, Jie Xiong, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, pages 2369–2380, 2018.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023a.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2023.

Appendix Overview

This appendix provides the complete algorithmic details, theoretical analysis, experimental setup, extended results, and discussion for **State of Thought (SoT)**. Concretely, the appendix serves two complementary purposes: *(i)* to make the actual controller precise enough to be reproducible, and *(ii)* to make the underlying reasoning claim precise enough to be evaluable.

Part A: Algorithmic Details. This part expands the method into explicit components and aligns each component with the actual implementation used in our experiments.

- Appendix A: the complete SoT inference loop as state-conditioned reasoning.
- Appendix B: the endogenous reasoning state, its geometric meaning, and the controller-facing state interface.
- Appendix C: the controller fitting protocol from offline trajectories.
- Appendix D: limited-access SoT variants, including training-free SoT and SoT-Embed with sentence-embedding trajectory features.
- Appendix E: SoT-Judge as a trajectory-level black-box evaluator.

Part B: Theoretical Analysis. This part formalizes the central claim of SoT as a reasoning-control paradigm.

- Appendix F: why externally prescribed token-chain policy classes can be strictly weaker than state-conditioned control.
- Appendix G: why state-conditioned evidence organization is necessary on heterogeneous reasoning distributions.
- Appendix H: why sparse, state-aligned evidence activation can improve both decision quality and efficiency.
- Appendix I: why feedback evidence control dominates open-loop reasoning schedules under regime variation.
- Appendix J: why state-driven stopping is a rational component of bounded-cost reasoning.

Part C: Experimental Setup and Implementation Details. This part expands the evaluation protocol, dataset organization, baseline taxonomy, and hyperparameter structure.

- Appendix K: datasets, task types, reasoning families, and metric conventions.
- Appendix L: backbones, evaluation protocol, and inference accounting.
- Appendix M: detailed baseline taxonomy and comparison rationale.
- Appendix N: controller, search, and evaluation hyperparameters.

Part D: Additional Experimental Results and Analysis. This part collects interpretability analyses, additional results, and qualitative trajectory inspections that support the main paper's claims.

- Appendix O: extended state-space and cluster-based evidence for endogenous structure.
- Appendix P: additional tables, efficiency breakdowns, and supplementary comparisons.
- Appendix Q: qualitative trajectory-level case studies.

Part E: Related Work and Discussion. This part expands the paper's broader positioning and the main limitations and future opportunities opened by SoT.

- Appendix R: expanded related-work positioning.
- Appendix S: limitations and future work.

Part A: Algorithmic Details

A Implementation Details of State-Conditioned Reasoning

This appendix makes the inference-time realization of SoT explicit. As summarized in Algorithm 1, the framework is fully specified by three components: the endogenous state m_t, the evidence-organization operator $\mathcal{S}$, and the stopping operator $\mathcal{T}$.

Sentence-level inference policy. SoT executes reasoning at the sentence level, because the sentence is typically the smallest unit that still carries a locally coherent reasoning intention while preserving sufficient temporal resolution for stepwise control [Bogdan et al., 2025, Liu et al., 2026, Wang et al., 2023a]. Accordingly, at step t, the frozen backbone generates a sentence-level reasoning unit y_t, from which SoT reads out the endogenous state $m_t \in \mathbb{R}^4$ as defined in Eq. (2) and detailed in Appendix B. The running history is therefore represented by the trajectory $(y_1, m_1), \ldots, (y_t, m_t)$, and the controller operates directly on this evolving trajectory rather than on a handcrafted reasoning template.

At each step, SoT applies the two state-conditioned operators introduced in Sec. 2.3:

$$\widetilde{y}_{<t} = \mathcal{S}(y_{<t}; m_t), \qquad z_t = \mathcal{T}(m_t) \in \{0, 1\}, \tag{5}$$

where $\mathcal{S}$ determines which historical evidence remains active for the next decision, and $\mathcal{T}$ determines whether reasoning should continue ($z_t = 0$) or terminate ($z_t = 1$). Inference therefore follows

$$\mathbb{P}_\pi(y_t \mid p, y_{<t}) = \mathbb{P}(y_t \mid p, \widetilde{y}_{<t}), \qquad \mathbb{P}_\pi(a \mid p, y_{1:t}) = \mathbb{P}(a \mid p, \widetilde{y}_{<t}, y_t), \tag{6}$$

so that both evidence carrying and stopping become consequences of the current endogenous state.

State-conditioned evidence organization. In the implemented controller, $\mathcal{S}$ scores each historical reasoning step y_i through its paired state m_i relative to the current state m_t. For each $i < t$, we form the pairwise descriptor

$$d_{t,i} = \big[m_t;\, m_i;\, m_t - m_i;\, m_t \odot m_i\big] \in \mathbb{R}^{16}, \tag{7}$$

which compactly captures the current regime, the historical regime, their displacement, and their channelwise interaction. A lightweight evidence head ϕ_{sel} maps this descriptor to a scalar selection score

$$g_{t,i} = \phi_{\text{sel}}(d_{t,i}; \theta), \tag{8}$$

where θ denotes the controller parameters learned offline as described in Appendix C. Under the context budget B, accepted steps are then inserted in chronological order to form

$$\widetilde{y}_{<t} = \mathcal{S}(y_{<t}; m_t) \subseteq y_{<t}. \tag{9}$$

Hence $\widetilde{y}_{<t}$ contains exactly the historical reasoning units selected to remain active for the next decision. No extra recency-only or semantic-only heuristic is used at runtime, so evidence organization remains grounded in endogenous state rather than generic memory management.

Algorithm 1 State of Thought (SoT) inference

Require: Frozen backbone f, problem x, evidence operator $\mathcal{S}$, stopping operator $\mathcal{T}$, context budget B, maximum steps $T_{\max}$
Ensure: Reasoning trajectory $y_{1:T}$ and final answer a

```
Initialize prompt p ← x, history y_{<1} ← ∅, active evidence ỹ_{<1} ← ∅
for t = 1, 2, ..., T_max do
    Generate the next reasoning unit y_t ~ P(· | p, ỹ_{<t})
    Read out the endogenous state m_t from y_t
    Update active evidence by ỹ_{<t+1} ← S(y_{≤t}; m_t) under budget B
    Compute stopping decision z_t ← T(m_t)
    if z_t = 1 then
        Generate the final answer a ~ P(· | p, ỹ_{<t+1}, y_t) and break
    end if
end for
```

State-conditioned stopping. The stopping operator is implemented by a second lightweight head on the same state m_t:

$$o_t = \phi_{\text{stop}}(m_t; \phi), \qquad p_t^{\text{stop}} = \sigma(o_t), \qquad z_t = \mathbb{I}\big[p_t^{\text{stop}} > \tau\big], \tag{10}$$

where ϕ denotes the stop-head parameters, $\sigma(\cdot)$ is the sigmoid, and $\tau \in (0, 1)$ is the stop threshold learned. Stopping is therefore determined by the current reasoning regime rather than by a fixed external depth schedule.

Overall, SoT runs as a closed-loop state-conditioned controller:

$$m_t \;\to\; \widetilde{y}_{<t} = \mathcal{S}(y_{<t}; m_t) \;\to\; y_t \;\to\; m_{t+1}, \qquad z_t = \mathcal{T}(m_t). \tag{11}$$

Because the evidence carried at one step changes the next reasoning unit, and the new reasoning unit in turn changes the next endogenous state, the trajectory is path-dependent by construction. This is the operational sense in which SoT acts as a feedback controller for reasoning rather than as a fixed reasoning script or a generic history-compression rule.

Threshold robustness. The gate and stop thresholds are part of the fitted controller bundle and could in principle raise sensitivity concerns. For this reason, the main ablation study in Sec. 3.4 and Appendix P.3 additionally reports a calibrated variant that adjusts these thresholds post hoc. Its effect is consistently secondary to the full SoT mechanism, indicating that the observed gains are not driven by brittle threshold tuning alone.

B Endogenous Reasoning State

This section expands Sec. 2.2 and makes the state construction explicit. The goal is to expose the smallest controller state that still captures the regime information needed for online reasoning control.

B.1 State construction

Sentence center. Let $h_{t,i}^{(L)} \in \mathbb{R}^D$ denote the final-layer hidden state of token i in reasoning sentence y_t, and let n_t be the sentence length. We define the sentence center by

$$u_t = \frac{1}{n_t} \sum_{i=1}^{n_t} h_{t,i}^{(L)} \in \mathbb{R}^D. \tag{12}$$

This provides a coarse-grained anchor of the current reasoning unit in hidden space through simple sentence-level pooling.

Controller state. From the sentence-center trajectory, SoT constructs the raw controller signature

$$s_t = [\delta_t, v_t, c_t, H_t]^\top \in \mathbb{R}^4, \tag{13}$$

which combines geometry, dynamics, and uncertainty into a compact endogenous control surface.

The geometric coordinate is

$$\delta_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \|h_{t,i}^{(L)}\|_2^2 - \|u_t\|_2^2, \tag{14}$$

which measures within-sentence dispersion around the sentence center and thus characterizes whether the current reasoning unit is internally concentrated or diffuse.

The two dynamical coordinates are

$$v_t = \|u_t - u_{t-1}\|_2, \tag{15}$$

and

$$c_t = \frac{\langle u_t - u_{t-1},\, u_{t-1} - u_{t-2} \rangle}{\|u_t - u_{t-1}\|_2 \cdot \|u_{t-1} - u_{t-2}\|_2 + \varepsilon}, \tag{16}$$

where v_t measures the magnitude of state movement between adjacent reasoning units, while c_t measures cosine alignment between consecutive displacements, namely whether the current step

continues the recent trajectory direction or instead departs toward a new regime. Together, v_t and c_t separate movement from directionally stable movement.

The uncertainty coordinate is

$$H_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{H}(p_{t,i}), \tag{17}$$

where $\mathcal{H}(\cdot)$ denotes Shannon entropy and $p_{t,i}$ is the next-token predictive distribution at position i. It summarizes the model's local predictive uncertainty over the current reasoning unit.

These four coordinates are then normalized componentwise:

$$\hat{s}_t = \frac{s_t - \mu}{\sigma + \varepsilon}, \tag{18}$$

where $\mu, \sigma \in \mathbb{R}^4$ are the componentwise mean and standard deviation estimated from the offline training trajectories, and $\varepsilon > 0$ is a small constant for numerical stability. The controller-facing endogenous state is finally

$$m_t = \hat{s}_t \in \mathbb{R}^4. \tag{19}$$

For boundary steps, we set $v_1 = c_1 = 0$ and $c_2 = 0$.

The resulting state is intentionally minimal: δ_t captures geometry, (v_t, c_t) capture dynamics, and H_t captures uncertainty. It is therefore not a handcrafted reasoning program, but a compact endogenous interface that answers four controller-relevant questions: whether the current step is internally organized, whether the trajectory is moving, whether it remains directionally stable, and whether the model is locally uncertain.

B.2 Why four coordinates suffice for control

Control requirements. We define sufficiency with respect to SoT's control problem: the backbone supplies semantic reasoning, while the state retains the feedback variables needed to organize prior evidence and decide whether reasoning should continue. These two decisions require a snapshot of the current unit, a description of how the trajectory arrived there, and a readout of predictive readiness. The four coordinates provide this control basis with distinct, complementary observables.

Table 7: Complementary control roles of the four endogenous coordinates.

Coord.	Observable	Control question resolved
δ_t	Within-unit dispersion	Is the current representation internally concentrated or diffuse?
v_t	Step displacement	Is the trajectory advancing or nearly stationary?
c_t	Directional persistence	Is it continuing along the preceding transition or redirecting at comparable speed?
H_t	Predictive entropy	Is the model's local prediction decisive or unresolved?

The distinctions in Table 7 cannot be recovered from one another. In particular, v_t measures how far the state moves but cannot distinguish continuation from reversal, which is the role of c_t. Likewise, δ_t measures organization inside the current representation, whereas H_t measures uncertainty in the predictive distribution; one describes representation geometry and the other output readiness. Thus (δ_t, H_t) provide complementary state snapshots, while (v_t, c_t) resolve the magnitude and direction of temporal change.

From state to control. For evidence organization, SoT compares the current state m_t with each historical state m_j through $[m_t; m_j; m_t - m_j; m_t \odot m_j]$. The differences expose regime change and the products expose coordinate-wise compatibility, turning the four observables into a 16-dimensional relational descriptor. The stopping head reads m_t directly. The same compact state therefore supports two distinct actions: selecting history compatible with the trajectory's present regime and terminating once its geometry, dynamics, and uncertainty indicate sufficient progress.

Functional validation. We test whether each coordinate actually affects the deployed control law. On held-out selector decisions, one normalized coordinate is replaced by zero—its training-population mean—while the other three coordinates, candidate texts, backbone, controller weights, and context

budget are fixed. We then rerun the selector and record whether its top-ranked historical unit changes. This intervention measures controller behavior directly and requires no task label.

Table 8: Single-coordinate interventions on evidence selection.

Coordinate fixed	δ_t	v_t	c_t	H_t
Top-1 selection changed	26.5%	39.1%	56.2%	44.7%

Every coordinate changes a substantial fraction of selections, with direction and uncertainty producing the largest rerouting rates. A second intervention connects these decisions to generation outcomes. On 120 examples from each of GSM8K, DROP, and MATH, zeroing δ_t during generation lowers accuracy by 6.7–22.5 points on both Llama-3.1-8B and Qwen2.5-14B. Zeroing v_t, by contrast, makes trajectories 4–5$\times$ longer. The separation matches the intended roles: dispersion is strongly coupled to answer quality, whereas displacement supplies a principal halting signal. Together, the two intervention levels trace a coherent path from individual coordinates, through evidence selection, to accuracy and stopping.

Excluding shortcut explanations. We examine alternatives at both the representation and controller levels. First, token composition and sentence length explain at most $R^2 = 0.46$ of δ_t across eight backbones spanning 2B–72B. Over the same scale range, $c_t \approx -0.43$ and remains negative after projecting out the mean drift and top five shared directions. Hence dispersion is not reducible to surface composition, and directional redirection is not an artifact of a dominant shared direction. Second, a diagnostic decomposition adds the four-coordinate state to recency and semantic similarity when recovering the training-side relevance rule; the state contributes 0.044–0.138 held-out AUC on every tested backbone over 2B–47B. This decomposition identifies information beyond the two generic heuristics, while an end-task permutation supplies the independent compatibility test: we permute historical states across candidates while preserving each candidate's text, position, and the inference budget. Breaking only the state–unit correspondence reduces accuracy from 79.0 to 62.4 on GSM8K, 38.6 to 24.6 on DROP, and 60.0 to 38.5 on MATH. The gains therefore arise from matching evidence to the evolving endogenous regime rather than from recency or surface similarity alone.

Sufficiency and economy. The design argument and interventions support the same conclusion. The four-coordinate basis covers the controller's required snapshot and transition variables; the relational map composes them for evidence selection, and the direct readout composes them for stopping. This yields a 577-parameter selection head plus a 5-parameter stopping head—582 trainable parameters in total—without updating the backbone. Its sufficiency is borne out operationally across heterogeneous reasoning structures: a compact endogenous state can organize evidence and termination while the backbone continues to supply the semantic reasoning itself.

C Lightweight Training of State-Conditioned Operators

This section closes the loop between Sec. 2.3, the runtime parameterizations ϕ_{sel} and ϕ_{stop}, and the offline data used to fit them. The backbone LLM remains frozen throughout; learning attaches only to the lightweight operators that instantiate $\mathcal{S}$ and $\mathcal{T}$.

Offline trajectory pool. We first roll out the frozen backbone on training problems, segment each transcript into sentence-level reasoning units $y_1, \ldots, y_T$, and compute the corresponding states m_t as in Sec. B. The same trajectory collection is also used once to estimate the normalization statistics (μ, σ) in Eq. (18), so training and deployment share a single state geometry. For VLM transfer, the same train–evaluation separation is applied to multimodal offline trajectories: the image–text backbone remains frozen, evaluation examples are excluded, and fitting is confined to the lightweight controller bundle.

Each step t is stored together with its trajectory identity, step index, endogenous state m_t, final outcome (e.g., correctness), and the paired historical set $\{(y_j, m_j)\}_{j<t}$. This pairing is essential because evidence selection depends on (m_t, m_j), not on past text alone. From lightweight offline replay on these trajectories, we derive two forms of supervision. For evidence organization, replay compresses whether a historical unit y_j is both state-compatible and incrementally useful for downstream reasoning, yielding a soft or hard retention target $\hat{b}_{t,j} \in [0, 1]$. For stopping, replay yields a

target $z_t^* \in [0, 1]$ that reflects sufficiency of accumulated support: on correct trajectories, stop mass appears only once the trace is mature and locally confident; on incorrect trajectories, it is deferred until the trace is clearly late and stagnating. This prevents the stop head from degenerating into a trivial "stop whenever uncertain" rule.

Note on trajectory quality and generalization: Trajectory quality affects how sharply the controller can be supervised, but SoT is not trained to mimic any fixed reasoning trace. It only learns compact decisions about state-conditioned evidence utility and stop readiness. The trajectory pool and the evaluation suite are not identical, and SoT still transfers to datasets outside the trajectory-collection set, which suggests that the learned control signal is not reducible to benchmark-specific trajectory memorization. This point is further strengthened by the training-free variant in Sec. 4.1, where the same state-conditioned principle remains effective even without learned controller fitting. We additionally report a dedicated sensitivity study to trajectory-pool quality in Appendix P.5.

Training the evidence-organization operator. For each pair (t, j) with $j < t$, let $d_{t,j}$ be the descriptor in Eq. (7). The trainable evidence score is

$$g_{t,j} = \phi_{\text{sel}}(d_{t,j}; \theta), \tag{20}$$

where θ collects the parameters of the lightweight selection head. The evidence operator is fit by

$$\mathcal{L}_{\text{sel}} = \sum_t \sum_{j<t} w_{t,j}\, \ell_{\text{BCE}}\big(\sigma(g_{t,j}), \hat{b}_{t,j}\big) + \lambda_{\text{sp}} \mathcal{R}_{\text{sp}} + \lambda_2 \|\theta\|_2^2, \tag{21}$$

where $w_{t,j}$ reweights rare positive supervision when needed, $\mathcal{R}_{\text{sp}}$ penalizes excessive batch-level gate mass to prevent trivial dense retention, and λ_2 regularizes the controller parameters. Supervision is concentrated on later steps of sufficiently long traces, where evidence selection materially changes the carried context rather than merely preserving the most recent sentence.

Training the stopping operator. The stopping head uses the same interface as at inference:

$$o_t = \phi_{\text{stop}}(m_t; \phi), \qquad p_t^{\text{stop}} = \sigma(o_t), \tag{22}$$

where ϕ denotes the stop-head parameters. Given the replay-derived stop target z_t^*, the stopping loss is

$$\mathcal{L}_{\text{stop}} = \sum_t \ell_{\text{BCE}}\big(p_t^{\text{stop}}, z_t^*\big). \tag{23}$$

This target encodes evidence sufficiency rather than compliance with any external reasoning template.

Overall objective. The final training objective is

$$\mathcal{L}_{\text{SoT}} = \mathcal{L}_{\text{sel}} + \lambda_{\text{stop}} \mathcal{L}_{\text{stop}}. \tag{24}$$

Only the controller parameters (θ, ϕ) are updated, while all backbone weights remain fixed. Training therefore identifies a lightweight control law on the endogenous interface $(m_t, \{m_j\}_{j<t})$: it neither rewrites the model's internal representations nor imitates a fixed multi-step reasoning script. The selector and stopping heads contain only 582 trainable parameters in total. They are fit for 12 warm-up epochs followed by 400 lightweight refinement steps over offline trajectories; no backbone gradients, evaluation examples, or test-time rewards are used. Across 5 problem-held-out refits, selector AUC is 0.779 ± 0.005, indicating that the small control interface is stable across trajectory partitions rather than dependent on a favorable fit. The trajectory-quality analysis in Appendix P.5 further shows where weaker supervision primarily increases computation, while the training-free results establish that the state-conditioned principle does not depend on learned fitting alone.

D SoT under Limited Access and Limited Training

This section expands Section 4.1. The goal of these variants is not to introduce new SoT formulations, but to probe the boundary of the SoT principle under weaker implementation conditions. Concretely, we ask whether state-conditioned evidence organization remains useful when either (i) the controller is not learned, or (ii) privileged hidden-state access is unavailable.

Training-free SoT. Training-free SoT preserves the same inference loop, the same sentence-level reasoning units y_t, and the same endogenous state interface m_t, but removes learned controller parameters from both $\mathcal{S}$ and $\mathcal{T}$. Instead of using the learned evidence operator in Appendix A, it scores each historical step y_j by a fixed state-compatibility rule:

$$s_{t,j}^{\mathrm{tf}} = \alpha_1 \langle m_t, m_j \rangle - \alpha_2 \|m_t - m_j\|_2 + \alpha_3 c_j - \alpha_4 H_j, \tag{25}$$

where $\langle m_t, m_j \rangle$ encourages compatibility between the current and historical reasoning regimes, $\|m_t - m_j\|_2$ penalizes mismatched states, c_j favors directionally stable historical steps, and H_j downweights locally uncertain ones. Under the same context budget B, the deterministic evidence operator keeps the top-scoring historical steps:

$$\widetilde{y}_{<t}^{\mathrm{tf}} = \mathcal{S}_{\mathrm{tf}}(y_{<t}; m_t) \subseteq y_{<t}. \tag{26}$$

Stopping is likewise made deterministic. Rather than using the learned stop head, Training-free SoT terminates only when the current state remains in a stable low-uncertainty regime for r_{stop} consecutive steps:

$$z_t^{\mathrm{tf}} = \mathcal{T}_{\mathrm{tf}}(m_{t-r_{\mathrm{stop}}+1:t}) \in \{0, 1\}. \tag{27}$$

In practice, this means that stopping requires jointly low H_t, sufficiently small state movement v_t, and positive directional consistency c_t over a short trailing window, so that the controller does not terminate on transient confidence alone. This variant therefore isolates whether the *state-conditioned reasoning principle* already helps before any controller fitting.

SoT-Embed (sentence-embedding trajectory). SoT-Embed targets the same limited-access setting as in the main text: internal hidden states are not exposed to the controller. We adopt sentence-level units $y_1, \ldots, y_T$ that are mapped by a fixed sentence encoder to vectors

$$u_t = \mathrm{Enc}(y_t) \in \mathbb{R}^d, \tag{28}$$

and the matrix $U \in \mathbb{R}^{T \times d}$ stacks the trajectory. Control inputs are built *directly* from the embedding trajectory: the controller conditions on u_t and on compact trajectory summaries derived from U, such as a low-dimensional subspace fit on the training distribution (PCA coordinates per step). Those features feed the same evidence and stopping interface as full SoT, so the loop remains state-conditioned, but the state signal is read only from public text and embeddings.

What these variants isolate. The two variants probe different boundary conditions. Training-free SoT asks whether SoT already has value as a fixed control principle before learning. SoT-Embed asks whether the paradigm fundamentally depends on privileged internal access when the control interface is replaced by embedding-based trajectory features. Together, they separate *endogenous reasoning as a principle* from *the strongest open-weight implementation of that principle*.

E SoT-Judge

This section expands Section 4.2. SoT-Judge transfers the trajectory perspective of SoT from online reasoning control to post-hoc black-box evaluation. The central question is whether the *observable* organization of a finished trace—without hidden states or extra generator calls—still carries enough signal to predict reference correctness.

Trajectory from text only. Given a completed rollout from a closed or API-accessed model, we segment the generated text into sentence-level units $y_1, \ldots, y_T$ and map them with a fixed sentence encoder to vectors $u_1, \ldots, u_T \in \mathbb{R}^d$. No architectural hooks are required: the judge consumes the same surface trace a human or downstream tool would see.

Compact trajectory readout. Let $U \in \mathbb{R}^{T \times d}$ stack the embeddings. On the training split we fit a standard low-dimensional subspace (PCA) and represent each step by its coordinates in that subspace. We then pool the sequence into a short feature vector φ_{traj} using elementary summaries along time (e.g., coordinate-wise means and standard deviations) plus a few optional length statistics of the segmented text. Auxiliary trajectory summaries can be appended for ablations, but the default recipe stays deliberately small: one fixed embedder, one subspace fit, and pooled statistics.

Judge model. The judge is a lightweight predictor J_ψ trained only on reference correctness labels:

$$\hat{r} = J_\psi(\varphi_{\mathrm{traj}}) \,, \tag{29}$$

where $\hat{r} \in [0, 1]$ scores how likely the final answer is to be correct. Thus SoT-JUDGE reads *trajectory health* from the embedded sentence path, not from sample agreement, self-critique, or an external judge prompt.

What SoT-Judge tests. The value of SoT-JUDGE is both practical and conceptual. Practically, it provides a lightweight black-box alternative to majority-vote style selection or stronger-model judging. Conceptually, it tests whether the SoT view survives the loss of online control: even when the model's internals are hidden and intervention is impossible, reasoning may still be assessed through the organization of its trajectory rather than only through output agreement or a stronger judge prompt. In this sense, SoT-JUDGE is not a separate reasoning paradigm, but a boundary-case extension of the same SoT principle to post-hoc evaluation.

Part B: Theoretical Analysis

This part formalizes the theoretical logic underlying SoT as a reasoning-control paradigm. Here, we show a sharp structural point: once reasoning control is restricted to an external token-chain policy class, the attainable optimum is constrained by the information available to that class, whereas endogenous state-conditioned control can access a strictly richer decision interface on heterogeneous reasoning distributions.

The argument proceeds in five steps. Appendix F establishes policy-class separation as the theoretical motivation. Appendix G shows why evidence organization should depend on the realized reasoning state. Appendix H shows why the selected support should generally be sparse rather than transcript-complete. Appendix I shows why the mechanism should be closed loop rather than one-shot or fixed in advance. Finally, Appendix J shows why stopping is not an auxiliary trick but part of the same bounded-cost feedback control problem.

F Policy-Class Separation

We first formalize the theoretical starting point of the paper: the empirical gains of SoT suggest that reasoning improvements may saturate within a shared family of externally prescribed token-chain policies, whereas state-conditioned evidence organization opens a distinct control axis. The goal of this section is not to claim that every external method is universally bounded by a single numeric ceiling, but to show that once the control variable is restricted to *external token-chain policy classes*, the attainable optimum can be strictly below that of a state-conditioned policy class.

Setup. Let p denote the input problem, $y_{1:T}$ the reasoning trajectory, and a the final answer. For each step t, let m_t be the endogenous controller state. We compare two policy classes.

External token-chain policies choose the next reasoning step from the prompt, observed token history, and an externally specified control variable $c \in \mathcal{C}$:

$$\mathbb{P}_\pi(y_t \mid p, y_{<t}, c). \tag{30}$$

Here c may represent a prompting template, a planning script, a search budget, a branching rule, or any other externally imposed strategy variable, but it is fixed independently of the realized endogenous state m_t.

State-conditioned policies instead choose the next-step support through the current endogenous state:

$$\mathbb{P}_\pi(y_t \mid p, y_{<t}, m_t). \tag{31}$$

This is the policy class instantiated by SoT through the state-conditioned operators.

One-step control utility. For a realized step t, let u_t be a bounded one-step utility, which may be instantiated as expected correctness gain, negative step loss, or any equivalent bounded progress signal. For any admissible control variable x_t, define

$$J_t(x_t) = \mathbb{E}[u_t \mid p, y_{<t}, x_t]\,. \tag{32}$$

When $x_t = c$, this is the utility under an external token-chain policy; when $x_t = m_t$, it is the utility under a state-conditioned policy.

Assumption F.1 (State-relevant heterogeneity). There exists a positive-probability set of realized prefixes $(p, y_{<t})$ for which two states m, m' can arise such that the Bayes-optimal next-step support differs:

$$\arg\max_{\widetilde{y}_{<t}} \mathbb{E}[u_t \mid p, y_{<t}, m_t = m, \widetilde{y}_{<t}] \neq \arg\max_{\widetilde{y}_{<t}} \mathbb{E}[u_t \mid p, y_{<t}, m_t = m', \widetilde{y}_{<t}]\,. \tag{33}$$

In words, even under the same problem and token prefix, different endogenous reasoning regimes can require different evidence support for the next decision.

Assumption F.2 (External-state mismatch). There exists a positive-probability set of realized prefixes $(p, y_{<t})$ such that no externally specified control variable $c \in \mathcal{C}$ uniquely identifies the realized state-relevant optimum in (33). Equivalently, within that set, at least two realizations with different optimal evidence support are indistinguishable to the external policy class (30).

Proposition F.1 (Strict policy-class separation). Under Assumptions F.1 and F.2, the optimal expected one-step utility achievable by the external token-chain policy class is strictly below that achievable by the state-conditioned policy class:

$$\sup_{\pi \in \Pi_{\text{ext}}} \mathbb{E}[u_t] \; < \; \sup_{\pi \in \Pi_{\text{state}}} \mathbb{E}[u_t], \tag{34}$$

where Π_{ext} denotes the class in (30) and Π_{state} the class in (31).

Proof. Fix a realized prefix $(p, y_{<t})$ in the positive-probability set from Assumptions F.1 and F.2. By Assumption F.1, there exist two possible realized states $m \neq m'$ under this same prefix for which the Bayes-optimal evidence support differs. By Assumption F.2, the external class cannot distinguish these two cases through any externally specified variable c. Hence any policy in Π_{ext} must assign the same control decision to both realizations once $(p, y_{<t}, c)$ is fixed. Therefore it cannot be simultaneously optimal for both state realizations. As a result, on this positive-probability set, every external token-chain policy incurs a strictly positive conditional utility gap relative to the Bayes-optimal state-aware decision.

By contrast, the state-conditioned class Π_{state} conditions directly on m_t, so it can choose the Bayes-optimal support separately for the realization $m_t = m$ and for the realization $m_t = m'$. Hence Π_{state} contains a policy that attains the Bayes-optimal conditional utility on that set.

Since the set has positive probability and the utility gap is strict on that set, the gap remains strict after averaging over the data-generating distribution. This yields (34). □

Implication. Proposition F.1 gives the theoretical motivation for the paper. It does not say that any particular baseline is weak in isolation, nor that external token-chain reasoning can never improve performance. Rather, it shows that when reasoning control is restricted to an external policy class, the attainable optimum is constrained by the information available to that class. If the next reasoning decision depends on the realized endogenous state, then changing prompts, adding scripts, or enlarging search may still leave the controller inside the same external class. SoT improves along an orthogonal axis: it changes the control variable itself, from externally prescribed token programs to endogenous state-conditioned evidence organization.

The remaining sections characterize the structure of that advantage: state conditioning, sparsity, feedback, and stopping.

G State-Conditioned Evidence Organization

This section formalizes the first theoretical claim of SoT: when the usefulness of historical reasoning evidence varies across the model's current reasoning regime, a fixed evidence strategy that ignores the endogenous state is generally suboptimal, whereas a state-conditioned operator $\mathcal{S}(y_{<t}; m_t)$ can be optimal if m_t is sufficient for control.

Setup. At reasoning step t, let $y_{<t}$ denote the available reasoning history, and let $\widetilde{y}_{<t} \subseteq y_{<t}$ be the subset retained for the next decision. To make explicit that different hidden reasoning regimes may require different support, introduce a latent regime variable ω_t. For any candidate subset $\widetilde{y}_{<t} \subseteq y_{<t}$, define the one-step conditional utility

$$V_t(\widetilde{y}_{<t}; y_{<t}, \omega_t) := \mathbb{E}[u_t \mid y_{<t},\ \omega_t,\ \widetilde{y}_{<t}]\,, \tag{35}$$

where u_t is any bounded one-step utility for the next reasoning decision, such as negative next-step loss, expected correctness gain, or any equivalent bounded surrogate. The regime-optimal retained subset is

$$\widetilde{y}^{\star}_{<t}(y_{<t}, \omega_t) \in \arg\max_{\widetilde{y}_{<t} \subseteq y_{<t}} V_t(\widetilde{y}_{<t}; y_{<t}, \omega_t). \tag{36}$$

Assumption G.1 (Heterogeneous evidence demand). There exist a realized history $\bar{y}$, two distinct regimes $\omega \neq \omega'$, and positive-probability events $\{y_{<t} = \bar{y}, \omega_t = \omega\}$ and $\{y_{<t} = \bar{y}, \omega_t = \omega'\}$, such that the corresponding optimizers in Eq. (36) are unique and distinct:

$$\widetilde{y}^{\star}(\bar{y}, \omega) \neq \widetilde{y}^{\star}(\bar{y}, \omega'). \tag{37}$$

Assumption G.2 (State sufficiency for evidence control). There exists a measurable map ψ such that, almost surely,

$$\widetilde{y}^{\star}_{<t}(y_{<t}, \omega_t) = \psi(m_t, y_{<t}). \tag{38}$$

This assumption does not require m_t to reconstruct the full internal state of the backbone. It only requires m_t to be sufficient for choosing the evidence subset relevant to the next decision.

Proposition G.1. Under Assumptions G.1–G.2, any fixed evidence operator $\mathcal{S}_{\text{fix}} : y_{<t} \mapsto \widetilde{y}_{<t}$ that ignores m_t is strictly suboptimal on the heterogeneous distribution over $(y_{<t}, \omega_t)$. By contrast, the class of state-conditioned operators $(m_t, y_{<t}) \mapsto \widetilde{y}_{<t}$ contains a Bayes-optimal evidence selector.

Proof. Consider the realized history $\bar{y}$ from Assumption G.1. A fixed evidence operator must output a single subset $\mathcal{S}_{\text{fix}}(\bar{y})$, independent of the realized regime. Because the optimizers $\widetilde{y}^{\star}(\bar{y}, \omega)$ and $\widetilde{y}^{\star}(\bar{y}, \omega')$ are unique and distinct, $\mathcal{S}_{\text{fix}}(\bar{y})$ cannot equal both. Hence at least one of the following strict inequalities must hold:

$$V_t\big(\mathcal{S}_{\text{fix}}(\bar{y}); \bar{y}, \omega\big) < V_t\big(\widetilde{y}^{\star}(\bar{y}, \omega); \bar{y}, \omega\big), \tag{39}$$

or

$$V_t\big(\mathcal{S}_{\text{fix}}(\bar{y}); \bar{y}, \omega'\big) < V_t\big(\widetilde{y}^{\star}(\bar{y}, \omega'); \bar{y}, \omega'\big). \tag{40}$$

Since both regime-history events occur with positive probability, averaging over the data-generating distribution yields a strictly positive expected utility gap between any fixed operator and the Bayes-optimal rule.

Now consider the state-conditioned operator class. By Assumption G.2, the map $\psi(m_t, y_{<t})$ recovers the optimal subset $\widetilde{y}^{\star}_{<t}(y_{<t}, \omega_t)$ almost surely. Therefore the class $(m_t, y_{<t}) \mapsto \widetilde{y}_{<t}$ contains a Bayes-optimal selector. □

Implication. Proposition G.1 gives the basic control-theoretic justification for SoT. If the evidence required for the next reasoning step changes with the current reasoning regime, then no single fixed evidence strategy can be uniformly optimal. The control variable must therefore depend on the current endogenous state. In this sense, SoT is not merely replacing one reasoning template with another; it is changing the problem from *fixed strategy execution* to *state-conditioned evidence organization*.

H Sparse Evidence Organization

This section formalizes the second theoretical claim of SoT: once evidence organization is conditioned on the current reasoning regime, the optimal operator should in general be sparse rather than full-history preserving. The reason is not only computational. Historical evidence can be simultaneously helpful, neutral, or actively interfering for the next decision, and carrying the full history may therefore reduce both step utility and efficiency.

Setup. We continue the one-step formulation from Appendix G. At step t, let $y_{<t}$ be the available reasoning history, let ω_t denote the latent reasoning regime, and let $\widetilde{y}_{<t} \subseteq y_{<t}$ be the subset retained for the next decision. For any retained subset, the one-step conditional utility is

$$V_t(\widetilde{y}_{<t}; y_{<t}, \omega_t) := \mathbb{E}[u_t \mid y_{<t},\ \omega_t,\ \widetilde{y}_{<t}]\,. \tag{41}$$

Assumption H.1 (Helpful and interfering evidence). For each realized pair $(y_{<t}, \omega_t)$, the history admits a disjoint decomposition

$$y_{<t} = y^{+}_{<t}(\omega_t) \,\dot{\cup}\, y^{-}_{<t}(\omega_t), \tag{42}$$

where $y^{+}_{<t}(\omega_t)$ contains evidence that is helpful for the next decision under regime ω_t, and $y^{-}_{<t}(\omega_t)$ contains evidence that is irrelevant, misleading, or distracting under that regime.

Assumption H.2 (Utility decomposition). There exist functions R_t, I_t, and a constant $\lambda > 0$ such that

$$V_t(\widetilde{y}_{<t}; y_{<t}, \omega_t) = R_t\big(\widetilde{y}_{<t} \cap y^{+}_{<t}(\omega_t);\ y_{<t}, \omega_t\big) - I_t\big(\widetilde{y}_{<t} \cap y^{-}_{<t}(\omega_t);\ y_{<t}, \omega_t\big) - \lambda|\widetilde{y}_{<t}|, \tag{43}$$

where R_t is monotone nondecreasing in its set argument and I_t is monotone nondecreasing in its set argument. Thus, retaining more helpful evidence cannot reduce reward, retaining more interfering evidence cannot reduce interference, and carrying more context incurs a positive efficiency cost.

Proposition H.1. Fix a realized pair $(y_{<t}, \omega_t)$. Let the full-history operator retain

$$\widetilde{y}^{\text{full}}_{<t} = y_{<t}, \tag{44}$$

and let a sparse operator retain some $\widetilde{y}^{\text{sp}}_{<t} \subseteq y_{<t}$ such that

$$\widetilde{y}^{\text{sp}}_{<t} \cap y^{+}_{<t}(\omega_t) = y^{+}_{<t}(\omega_t), \tag{45}$$

and

$$\widetilde{y}^{\text{sp}}_{<t} \cap y^{-}_{<t}(\omega_t) \subsetneq y^{-}_{<t}(\omega_t). \tag{46}$$

Then

$$V_t(\widetilde{y}^{\text{sp}}_{<t}; y_{<t}, \omega_t) > V_t(\widetilde{y}^{\text{full}}_{<t}; y_{<t}, \omega_t). \tag{47}$$

Proof. Under the full-history operator, Eq. (43) gives

$$V_t(\widetilde{y}^{\text{full}}_{<t}; y_{<t}, \omega_t) = R_t\big(y^{+}_{<t}(\omega_t); y_{<t}, \omega_t\big) - I_t\big(y^{-}_{<t}(\omega_t); y_{<t}, \omega_t\big) - \lambda|y_{<t}|. \tag{48}$$

For the sparse operator, condition (45) implies that the helpful-evidence term is unchanged:

$$R_t\big(\widetilde{y}^{\text{sp}}_{<t} \cap y^{+}_{<t}(\omega_t); y_{<t}, \omega_t\big) = R_t\big(y^{+}_{<t}(\omega_t); y_{<t}, \omega_t\big). \tag{49}$$

Condition (46) implies that fewer interfering units are retained, so by monotonicity of I_t,

$$I_t\big(\widetilde{y}^{\text{sp}}_{<t} \cap y^{-}_{<t}(\omega_t); y_{<t}, \omega_t\big) \le I_t\big(y^{-}_{<t}(\omega_t); y_{<t}, \omega_t\big). \tag{50}$$

Moreover,

$$|\widetilde{y}^{\text{sp}}_{<t}| < |y_{<t}|, \tag{51}$$

because at least one interfering unit is removed while all helpful units are preserved. Substituting Eqs. (49)–(51) into Eq. (43) shows that the reward term is unchanged, the interference term does not increase, and the cost term strictly decreases by $\lambda\big(|y_{<t}| - |\widetilde{y}^{\text{sp}}_{<t}|\big) > 0$. Hence

$$V_t(\widetilde{y}^{\text{sp}}_{<t}; y_{<t}, \omega_t) - V_t(\widetilde{y}^{\text{full}}_{<t}; y_{<t}, \omega_t) \ge \lambda\big(|y_{<t}| - |\widetilde{y}^{\text{sp}}_{<t}|\big) > 0, \tag{52}$$

which proves Eq. (47). □

Implication. Proposition H.1 clarifies why SoT should be sparse. Once evidence usefulness is regime-dependent, carrying the full history is generally not neutral: it preserves helpful support, but it also preserves regime-mismatched interference and incurs additional cost. A state-aligned sparse operator improves the next reasoning decision not by generic compression, but by preserving the evidence that remains decision-relevant under the current endogenous state while removing distracting history. This is precisely why SoT can improve accuracy and efficiency jointly rather than trading one for the other.

I Closed-Loop Reasoning Control

This section formalizes the third theoretical claim of SoT: once evidence organization is state-conditioned and sparse, it should also be *closed loop*. That is, the reasoning controller should adapt its evidence-organization and continuation decisions to the realized endogenous state m_t, rather than committing in advance to a fixed schedule.

Setup. At reasoning step t, let the control action be

$$a_t := \big(\widetilde{y}_{<t}, z_t\big), \tag{53}$$

where $\widetilde{y}_{<t} \subseteq y_{<t}$ is the retained historical evidence and $z_t \in \{0, 1\}$ is the continuation decision, with $z_t = 0$ meaning continue and $z_t = 1$ meaning stop. The endogenous state evolves according to the controlled stochastic dynamics

$$m_{t+1} = F_t(m_t, a_t, \varepsilon_t), \tag{54}$$

where ε_t collects the randomness induced by token generation and environment variation. Let $r_t(m_t, a_t)$ denote the one-step reward, which may incorporate reasoning progress, answer quality, interference reduction, and compute cost. For a finite horizon T, define the value of a policy π from state m at step t as

$$V^{\pi}_t(m) := \mathbb{E}^{\pi}\left[\sum_{\tau=t}^{T} r_\tau(m_\tau, a_\tau) \,\middle|\, m_t = m\right]. \tag{55}$$

Open-loop and closed-loop policies. An *open-loop* policy fixes its actions in advance as a deterministic schedule depending only on the step index:

$$a_t^{\text{open}} = \bar{a}_t, \tag{56}$$

where $\bar{a}_t$ does not depend on the realized state m_t. A *closed-loop* policy instead adapts to the realized endogenous state:

$$a_t^{\text{cl}} = \pi_t(m_t). \tag{57}$$

Thus, open-loop control corresponds to a fixed external reasoning schedule, while closed-loop control conditions evidence organization and stopping on how reasoning has actually unfolded.

Proposition I.1. Let

$$V_t^{\text{cl}}(m) := \sup_{\pi \in \Pi_{\text{cl}}} V_t^{\pi}(m), \qquad V_t^{\text{open}}(m) := \sup_{\pi \in \Pi_{\text{open}}} V_t^{\pi}(m), \tag{58}$$

where Π_{cl} and Π_{open} denote the closed-loop and open-loop policy classes defined above. Then for every step t and every state m,

$$V_t^{\text{cl}}(m) \geq V_t^{\text{open}}(m). \tag{59}$$

Moreover, if there exists a positive-probability set of realized states A such that the continuation-optimal action at step t is not constant over A, then the inequality is strict in expectation over that set.

Proof. Every open-loop policy is a special case of a closed-loop policy. Indeed, for any fixed schedule $\{\bar{a}_\tau\}_{\tau=t}^{T} \in \Pi_{\text{open}}$, define the state-independent closed-loop policy

$$\pi_\tau(m) \equiv \bar{a}_\tau \qquad \text{for all } m \text{ and } \tau \geq t. \tag{60}$$

This policy belongs to Π_{cl} and induces exactly the same controlled process and therefore the same value as the original open-loop schedule. Hence

$$\Pi_{\text{open}} \subseteq \Pi_{\text{cl}}, \tag{61}$$

which immediately implies Eq. (59).

To show when the inequality is strict, consider the optimal state-action value at step t,

$$Q_t(m, a) := r_t(m, a) + \mathbb{E}\left[V_{t+1}^{\text{cl}}(m_{t+1}) \,\middle|\, m_t = m,\ a_t = a\right]. \tag{62}$$

Suppose there exists a measurable set A with positive probability such that for some two states $m, m' \in A$,

$$\arg\max_a Q_t(m, a) \neq \arg\max_a Q_t(m', a). \tag{63}$$

Then no single open-loop action $\bar{a}_t$ can be optimal for all realized states in A. A closed-loop policy can instead choose, at each realized state $m_t \in A$, an action in $\arg\max_a Q_t(m_t, a)$, while matching the open-loop schedule outside A. This yields a strictly larger $Q_t(m_t, a_t)$ on a positive-probability subset of A, and no smaller value elsewhere. Therefore the expected value under the best closed-loop policy is strictly larger than that under the best open-loop policy. □

Implication. Proposition I.1 explains why SoT should be closed loop rather than one-shot. When the best evidence support or continuation decision varies with the realized endogenous state, any fixed external reasoning schedule is necessarily unable to adapt to all realized reasoning regimes. Closed-loop control strictly enlarges the feasible policy class and becomes strictly better whenever different realized states require different actions. This is the formal reason SoT is not merely another pre-specified reasoning program: it treats reasoning as adaptive control over evidence organization and stopping under the evolving endogenous state.

J State-Driven Stopping

This section formalizes the final theoretical claim of SoT: stopping is not an auxiliary heuristic, but part of the same state-conditioned control problem as evidence organization. Once reasoning incurs cost, the controller should decide from the current endogenous state m_t not only what evidence to retain, but also whether another reasoning step is worth taking.

Setup. At step t, let $z_t \in \{0, 1\}$ denote the stopping decision, where $z_t = 0$ means continue reasoning and $z_t = 1$ means stop and emit the final answer a. Let $V_t^\star(m)$ denote the optimal bounded-cost value starting from state m at step t. We decompose the value of the two admissible actions as

$$V_t^\star(m) = \max\left\{V_t^{\text{stop}}(m),\ V_t^{\text{cont}}(m)\right\}, \tag{64}$$

where $V_t^{\text{stop}}(m)$ is the value of stopping immediately and answering from the current active context, and

$$V_t^{\text{cont}}(m) = \sup_\pi \mathbb{E}^\pi\left[r_t^{\text{cont}} + V_{t+1}^\star(m_{t+1}) \,\middle|\, m_t = m,\ z_t = 0\right] \tag{65}$$

is the optimal value of continuing for one more step and then behaving optimally thereafter. Here r_t^{cont} already includes the marginal computation cost of taking another reasoning step.

Continuation advantage. Define the continuation advantage

$$\Delta_t(m) := V_t^{\text{cont}}(m) - V_t^{\text{stop}}(m). \tag{66}$$

The sign of $\Delta_t(m)$ determines whether an additional step is worthwhile under the current endogenous state: continuation is beneficial when $\Delta_t(m) > 0$, and stopping is optimal when $\Delta_t(m) \le 0$.

Proposition J.1. Assume that m_t is sufficient for the stopping decision in the sense that both V_t^{stop} and V_t^{cont} are measurable functions of m_t. Then an optimal bounded-cost stopping rule is state-conditioned and can be written as

$$z_t^\star(m_t) = \mathbf{1}\{\Delta_t(m_t) \le 0\}. \tag{67}$$

Equivalently, the stop region

$$\Omega_{\text{stop}} := \{m \in \mathbb{R}^4 : \Delta_t(m) \le 0\} \tag{68}$$

is optimal: the controller should stop exactly on Ω_{stop} and continue on its complement.

Proof. By Eq. (64), the optimal action at state m is whichever attains the larger of $V_t^{\text{stop}}(m)$ and $V_t^{\text{cont}}(m)$. If $\Delta_t(m) \le 0$, then by Eq. (66),

$$V_t^{\text{cont}}(m) \le V_t^{\text{stop}}(m), \tag{69}$$

so stopping weakly dominates continuation and is therefore optimal. If $\Delta_t(m) > 0$, then

$$V_t^{\text{cont}}(m) > V_t^{\text{stop}}(m), \tag{70}$$

so continuation is strictly optimal. Hence the optimal decision is exactly

$$z_t^\star(m) = 1 \iff \Delta_t(m) \le 0, \tag{71}$$

which proves Eq. (67). Because V_t^{stop} and V_t^{cont} are measurable in m_t, the resulting stopping rule is a measurable state-conditioned policy. □

Corollary J.1. Any stopping rule that depends only on a fixed external depth budget, a pre-specified schedule, or the step index t, but not on the realized endogenous state m_t, is generally suboptimal whenever $\Delta_t(m)$ changes sign over a positive-probability set of realized states.

Proof. If $\Delta_t(m)$ changes sign across realized states, then there exist states m and m' with positive probability such that stopping is optimal at m but continuation is optimal at m'. Any state-agnostic rule must assign the same decision to both states and therefore errs on at least one positive-probability subset. By Proposition I.1, the state-conditioned rule in Eq. (67) attains the optimal action on both. Thus the state-agnostic rule is strictly suboptimal in expectation. □

Implication. Proposition J.1 shows that stopping is the bounded-cost counterpart of evidence organization. Both are decisions about whether additional reasoning support is worth carrying forward under the current endogenous state. Corollary J.1 further explains why SoT does not use a fixed reasoning depth or an external stopping schedule: when the marginal value of another step varies across realized reasoning regimes, rational stopping must itself be state-conditioned. This is why the stop controller is part of the same closed-loop reasoning mechanism rather than an afterthought.

Part C: Experimental Setup and Implementation

K Dataset

K.1 Category organization

The LLM datasets are grouped into four categories:

- **Quantitative Reasoning (QS):** GSM8K [Cobbe et al., 2021] (grade-school arithmetic word problems), MATH [Hendrycks et al., 2021b] (competition-level mathematical reasoning), and DROP [Dua et al., 2019] (discrete reading comprehension with numerical operations).
- **General Understanding (GU):** CommonsenseQA [Talmor et al., 2019] (commonsense multiple-choice QA), StrategyQA [Geva et al., 2021] (implicit multi-hop yes/no reasoning), BoolQ [Clark et al., 2019] (passage-grounded boolean QA), MMLU [Hendrycks et al., 2021a] (multi-domain professional and academic knowledge), and RACE [Lai et al., 2017] (exam-style reading comprehension).
- **Symbolic and Code (S&C):** FOLIO [Han et al., 2024] (first-order logical entailment), ProofWriter [Tafjord et al., 2021] (multi-step rule-based deduction), BBH-Temporal [Srivastava et al., 2023] (temporal symbolic reasoning), HumanEval [Chen et al., 2021] (function-level code generation), and MBPP [Austin et al., 2021] (program synthesis from natural-language specifications).
- **Long-Context Reasoning (LCR):** HotpotQA [Yang et al., 2018] (multi-hop evidence aggregation), NarrativeQA [Kočiskỳ et al., 2018] (long-form story understanding), and LongBench MultiFieldQA [Bai et al., 2024] (long-context question answering across heterogeneous documents).

For multimodal transfer, we use A-OKVQA [Schwenk et al., 2022] for knowledge-intensive visual question answering, AI2D [Kembhavi et al., 2016] for science-diagram reasoning, and M^3CoT [Chen et al., 2024] for multi-domain, multi-step multimodal reasoning.

K.2 Metrics

Table 9 summarizes the dataset-level organization, task type, split size, and primary metric used in our experiments. The headline "accuracy" in the main text is a unified label for the *task-standard final metric* of each benchmark.

We use the following primary metrics:

Table 9: **Dataset-level organization, split size, and primary metric.**

Category	Dataset	Task type	Train	Test	Metric
Quantitative Reasoning (QS)	GSM8K	Math QA	—	500	EM
	MATH	Competition math	—	400	EM
	DROP	Discrete reasoning QA	—	300	F1
General Understanding (GU)	CommonsenseQA (CSQA)	Multiple choice QA	52	400	EM
	StrategyQA (StrQA)	Binary QA	52	400	EM
	BoolQ	Boolean QA	—	400	EM
	MMLU	Multi-domain MCQA	52	500	EM
	RACE	Reading comprehension	—	300	EM
Symbolic and Code (S&C)	FOLIO	Logical entailment	—	203	EM
	ProofWriter (PW)	Deductive QA	52	400	EM
	BBH-Temporal (BBH-T)	Temporal reasoning	52	100	EM
	HumanEval (HE)	Code generation	—	164	P@1
	MBPP	Code generation	52	250	P@1
Long-Context Reasoning (LCR)	HotpotQA (HQA)	Multi-hop QA	52	300	F1
	NarrativeQA (NarQA)	Narrative QA	—	250	F1
	LongBench MultiFieldQA (LB)	Long-Context QA	—	110	F1
Vision–language multimodal (VLM)	A-OKVQA	Knowledge-intensive VQA	—	200	EM
	AI2D	Science-diagram reasoning	—	150	EM
	M^3CoT	Multi-step multimodal reasoning	—	150	EM

Note: **EM** = exact match, **F1** = token-level F1, **P@1** = pass@1.

- **EM** for classification-style, short-answer, logical-label, and visual recognition tasks;
- **F1** for long-form QA and evidence-aggregation tasks;
- **P@1** for code-generation benchmarks.

For efficiency, we report generated tokens and end-to-end latency for all models:

- **Tokens**: the number of generated reasoning tokens per example, averaged over the evaluation set;
- **Latency**: the end-to-end wall-clock inference time under the same hardware and decoding.

L Model Configuration and Evaluation Protocol

Backbone configuration. For text-only reasoning, we evaluate frozen Llama-3.1-8B [Grattafiori et al., 2024], Qwen2.5-14B [Yang et al., 2024], and Mixtral-8×7B [Jiang et al., 2024a]. For multimodal transfer, we evaluate frozen Qwen2.5-VL-7B and Qwen2.5-VL-32B backbones [Bai et al., 2025]; the 32B block uses BF16 weights throughout. In all cases, backbone weights remain fixed throughout; any learned component is confined to the lightweight SoT controller introduced in Section 2.3.

Unified evaluation protocol. All methods are evaluated from the same benchmark splits under a shared protocol that fixes decoding settings, inference budgets, answer extraction, and metric computation. For each run, the evaluation subset is fixed before inference, with the same decoding caps and comparable search or sampling budgets when applicable; completed-example counts are retained with the evaluation artifacts and used for sample-weighted aggregates. Task outputs are then passed through the same dataset-specific answer extraction, normalization, and scoring functions. Accordingly, SoT changes only the internal organization of evidence during reasoning; it does not benefit from a different evaluator, parser, or metric definition relative to the baselines. Text-backbone latency is measured on H200 GPUs. VLM latency is measured on one dedicated 80GB A100.

Consistency across baseline families. The comparison is controlled at three levels. First, all methods use the same frozen backbone within each experiment. Second, all methods follow the same prompt–generation–parsing pipeline except for the reasoning mechanism specific to that baseline family. Third, efficiency statistics are collected under the same runtime interface. Generated tokens are counted as the total reasoning tokens produced per example, including internal generations for multi-pass methods, and latency is measured as end-to-end wall-clock inference time under the same hardware and decoding setup. This ensures that both quality and efficiency comparisons are mechanism-level rather than implementation-level.

Training-side consistency for methods with learned components. Whenever a method introduces a learned component beyond the frozen backbone, it is fit only on a pre-specified training-side corpus, with evaluation questions excluded in advance. SoT follows the same rule: its evidence-selection and stopping operators are trained only from offline reasoning trajectories derived from the frozen backbone, without access to evaluation instances. This keeps train–evaluation separation uniform across all methods requiring lightweight fitting and prevents gains from arising from unequal data exposure.

Prompting and post-processing. For baseline families that share a common reasoning prompt style, SoT uses the same user-level prompt format; differences are confined to the internal control policy rather than surface prompt advantages. Final predictions for all methods are extracted by the same dataset-specific post-processing rules, and the primary metrics reported in Table 9 are always computed by the same evaluation routines.

M Baseline Methods

M.1 Direct generation

Vanilla. Vanilla performs direct greedy answering without explicit intermediate reasoning control. It serves as the minimal frozen-backbone reference.

M.2 Reasoning paradigms

CoT. Chain-of-Thought (CoT) prompting elicits reasoning through an externally specified step-by-step textual scaffold [Wei et al., 2022]. It is the canonical baseline for scripted single-trajectory reasoning.

PS. Plan-and-Solve (PS) separates reasoning into an explicit planning stage followed by execution [Wang et al., 2023a]. Compared with CoT, it imposes a more structured external decomposition before answering.

SR. Self-Refine (SR) uses an externally scripted draft–feedback–revision loop to iteratively improve an initial response [Madaan et al., 2023]. Its improvement comes from explicit textual self-correction rather than endogenous control of carried evidence.

SC. Self-Consistency (SC) samples multiple reasoning traces and selects the final answer by agreement across sampled chains [Wang et al., 2023b]. It represents sample-based test-time scaling through breadth rather than state-conditioned control.

BoN. Best-of-N (BoN) samples five candidate solutions and uses a fixed zero-temperature selector pass to choose the final answer from those candidates [Snell et al., 2024]. It therefore tests candidate-level selection beyond SC's answer-agreement rule; all candidate and selector completions are included in its reported cost.

CB. Constrained Beam (CB) allocates extra inference budget through beam-style structured decoding under explicit search constraints [Hokamp and Liu, 2017]. It provides a search-oriented comparison point for whether better reasoning must come from broader trajectory exploration.

MCTS. Monte Carlo Tree Search (MCTS) performs explicit search over reasoning branches using rollout-based lookahead and selection [Xie et al., 2024]. It is a strong representative of search-heavy reasoning methods that trade additional compute for broader exploration.

M.3 Memory-oriented methods

H_2O. H_2O prunes attention history by retaining tokens estimated to be important for generation [Zhang et al., 2023], reducing effective context usage.

SNAP. SnapKV accelerates long-context inference by compressing KV cache content through observation of stable attention patterns [Li et al., 2024]. Its objective is efficient context management rather than reasoning-specific evidence control.

STREAM. StreamingLLM maintains stable long-context decoding by preserving a small set of attention anchors together with a streaming context window [Xiao et al., 2023]. It is an important efficiency baseline because it also avoids full-history accumulation.

M.4 Latent and RL-based reasoning

COCO. COCONUT performs reasoning partly in continuous latent space rather than purely through explicit textual chains [Hao et al., 2025]. It is the closest latent-reasoning comparison, but differs from SoT in that it shifts the reasoning substrate itself rather than using endogenous state to organize evidence.

GRPO-SP. GRPO-SP is a lightweight RL-based comparison inspired by grouped relative policy optimization [Shao et al., 2024]. It freezes the backbone and applies grouped reward-normalized policy-gradient updates only to a four-token continuous prompt, testing whether a compact reward-trained inference prefix can recover reasoning gains without backbone updates.

Comparison logic. These baselines probe complementary alternatives to SoT. Vanilla tests whether explicit reasoning control is necessary. CoT, PS, and SR test externally prescribed single-trajectory programs. SC, BoN, CB, and MCTS test whether gains instead come from larger test-time search.

Table 10: **Deployment-side inference hyperparameters of SoT.** We report only externally specified quantities used at test time.

Parameter	Value	Notes
Context budget	1536	Active evidence budget
Maximum reasoning steps	32	Safety cap only
Observation window	4096	Visible context span
Final answer budget	512	Answer generation cap
Stop threshold	0.5	Default stop rule
Gate threshold	0.5	Default keep rule
Score mixing coefficient	0.5	Score fusion only
Score temperature	0.8	Ranking smoothing only

Table 11: **Inference hyperparameters of baseline methods.**

Baseline	Parameter	Value
Vanilla	Maximum new tokens	512
	Temperature	0
CoT	Maximum new tokens	768
	Temperature	0
PS	Planning tokens	96
	Solving tokens	320
	Temperature	0
SR	Draft tokens	256
	Critique tokens	96
	Revision tokens	256
	Temperature	0
SC	Number of samples	3 (LLM); 5 (VLM)
	Tokens per sample	384
	Temperature	0.7
BoN	Number of candidates	5
	Candidate token cap	768
	Candidate temperature	0.7
	Selector token cap	64
	Selector temperature	0
CB	Search depth	3
	Beam width	3
	Branches per beam	2
	Step tokens	56
	Scoring tokens	12
	Answer tokens	256
	Expansion temp.	0.4
	Tree token budget	3200
MCTS	Search iterations	6
	Expansion branches	2
	Step tokens	56
	Scoring tokens	12
	Answer tokens	256
	Exploration const.	1.2
	Expansion temp.	0.5
	Tree token budget	3200
H2O	Sink tokens	4
	Recent tokens	256
	Context budget	512
	Maximum new tokens	512
	Temperature	0
SNAP	Sink tokens	4
	Recent tokens	256
	Context budget	512
	Maximum new tokens	512
	Temperature	0
STREAM	Sink tokens	4
	Recent tokens	256
	Context budget	512
	Maximum new tokens	512
	Temperature	0
COCO	Latent steps	3
	Maximum new tokens	512
	Temperature	0
GRPO-SP	Maximum new tokens	768
	Temperature	0

H_2O, SNAP, and STREAM test whether generic context reduction is sufficient. COCO and GRPO-SP test latent-space or optimization-based alternatives to inference-time state-conditioned control. Together, they provide the comparison surface needed to show that SoT is not reducible to prompting, search, memory compression, latent reasoning, or RL-style behavioral optimization.

Implementation and fairness. Names follow the implemented algorithmic roles: SC selects by answer agreement, BoN uses a fixed selector over five sampled candidates, CB and MCTS perform structured search, and GRPO-SP is the grouped-advantage soft-prompt variant defined above. Each family uses one pre-specified configuration across datasets; sample counts, temperatures, and search budgets are reported in Table 11. All generated stages count toward efficiency, and all methods share the same task-specific answer extraction and metric.

N Hyperparameter Configuration

Table 10 reports the deployment-side hyperparameters of SoT, while Table 11 summarizes the corresponding hyperparameters for all baselines.

The two score-smoothing parameters used by SoT—the attention mixing coefficient and attention temperature—are deployment-side stabilizers rather than additional reasoning modules: they only

smooth or fuse ranking signals before thresholding, and do not change the state interface, control form, or backbone generation budget. The maximum reasoning-step budget is a deployment-side safeguard only. Empirically, nearly all trajectories terminate through the learned stop policy before the hard cap is reached (about 99.6% in our runs), so the reported efficiency gains are not driven by forced truncation.

For fairness, all methods are evaluated under the same backbone, data split, decoding interface, and scoring pipeline. Each method is assigned only the minimal family-specific hyperparameters required for execution. In particular, SoT does not receive additional model capacity, longer unrestricted generation, or a privileged prompt path relative to the baselines. Accordingly, any improvement should be attributed to its state-conditioned control mechanism rather than to favorable hyperparameter budgeting. Supplementary ablations also help verify that the observed gains are not driven by fragile score calibration or deployment-side smoothing choices, but by the full state-conditioned control mechanism.

Part D: Additional Experimental Results and Analysis

O Interpretable Evidence for State-Conditioned Reasoning

This section extends the interpretability evidence in Secs. 2.2 and 2.3, together with SoT as a trajectory-level judge in Sec. 4.2. Our goal is to show that SoT is not merely a strong inference-time heuristic: its gains are accompanied by stable and interpretable state-space structure, regime-dependent evidence organization, and measurable differences between successful and unsuccessful reasoning trajectories.

O.1 State-space trajectories across reasoning paradigms

We first extend the state-space visualization in Sec. 2.2 by covering all four task categories and a broader set of reasoning paradigms. For each method, we project the trajectory of m_t into a shared two-dimensional space and visualize the stepwise evolution across representative examples. The comparison is designed to answer two questions: whether different external reasoning paradigms induce distinct trajectory regions or motion patterns, and whether SoT occupies a broader and more adaptive portion of the state space rather than collapsing to a fixed scripted mode.

Interpretation. As shown in Figure 8, this analysis does not claim that each paradigm maps to a discrete symbolic state label. Instead, it shows that the compact state m_t carries stable geometric structure: different paradigms trace distinguishable trajectory regions, while SoT occupies a broader and more adaptive regime rather than collapsing to a fixed externally scripted pattern. The key insight is that SoT's gain is associated with controllable state coverage—reaching multiple useful regimes when needed—rather than with a single pre-committed reasoning template.

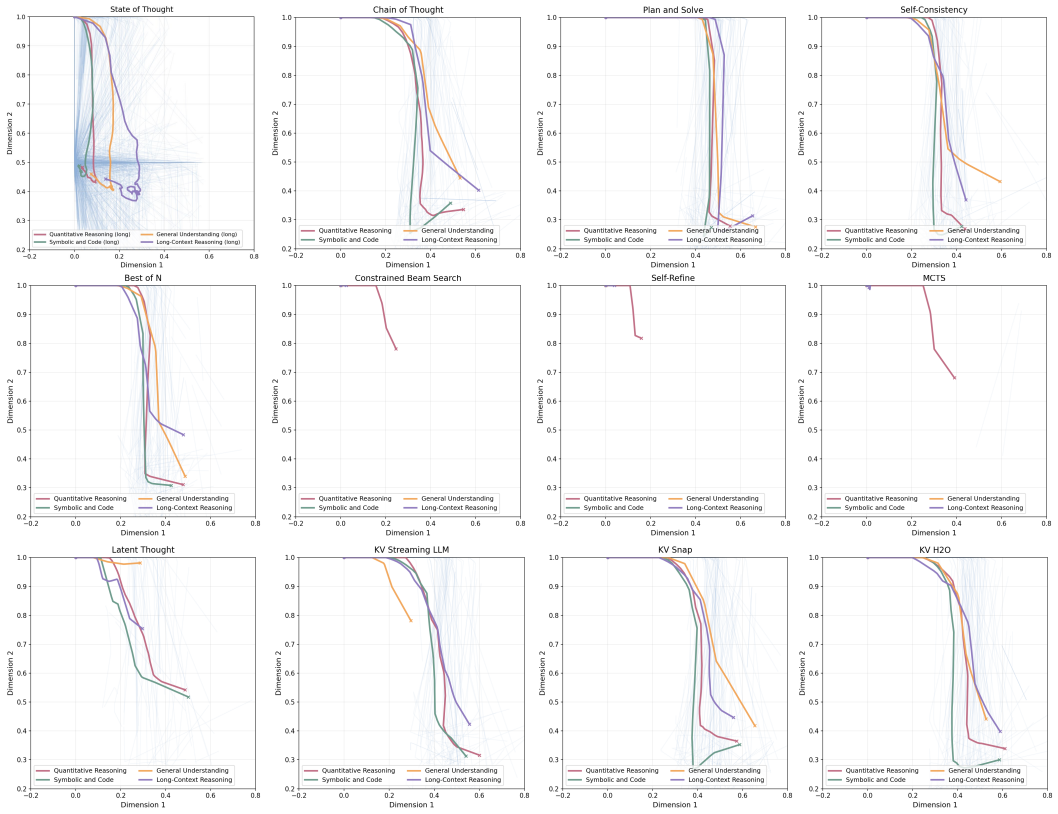


Figure 8: **Extended state-space trajectories across reasoning paradigms.** This figure supplements the main-text trajectory view with additional paradigms and task categories.

O.2 State clusters, sparse memory, and stopping

We next extend the operator-level analysis of Sec. 2.3. After aggregating SoT states across tasks, we cluster the observed m_t and summarize, for each cluster, (i) its occupancy across task categories, (ii) the average sparsity pattern of $\mathcal{S}(y_{<t}; m_t)$, and (iii) the stop tendency induced by $\mathcal{T}(m_t)$. This links the geometry of the state space to concrete control behavior.

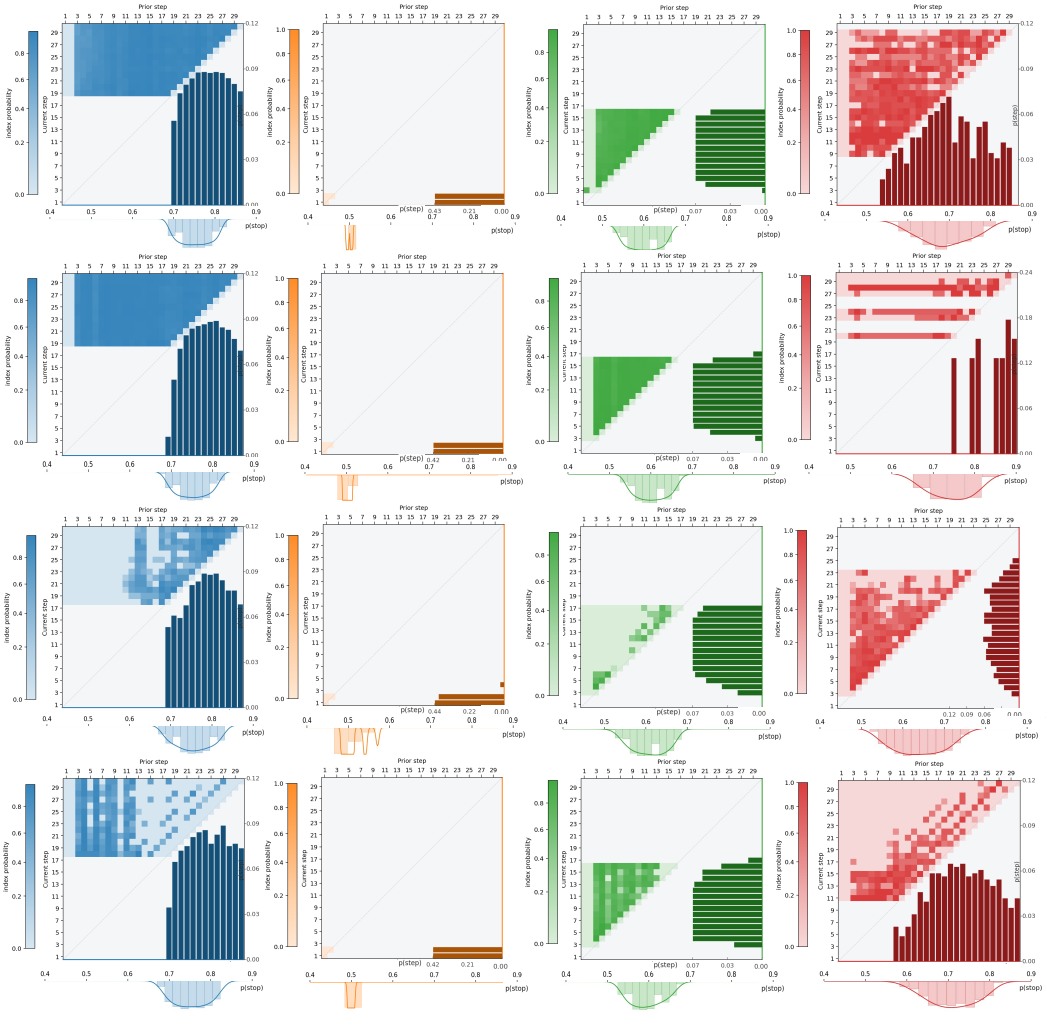


Figure 9: **Clustered state regimes and operator profiles.**

Interpretation. Figure 9 tests whether clustered states correspond to distinct control behavior, not just visual separability. The observed cluster-specific sparsity and stop profiles indicate regime-dependent evidence organization and stopping, rather than generic recency bias, static compression, or a fixed-depth schedule. This suggests that SoT's policy is conditionally compositional: the state first selects *what* evidence remains active and then implicitly calibrates *when* to terminate. The four labels—*exploit-dense*, *explore-dense*, *early-anchor*, and *sparse-retrieval*—are assigned only after clustering. We inspect each normalized centroid (δ, v, c, H) together with three measured properties: activation density, the temporal location of retained units, and stop probability. "Dense" and "sparse" therefore describe the activation matrix; "early-anchor" denotes concentration on early units; and the explore–exploit distinction additionally requires the centroid dynamics and stop profile to indicate continued movement or consolidation. These labels never enter fitting or evaluation. The evidence for distinct regimes is the operator profiles themselves, not the names attached to them.

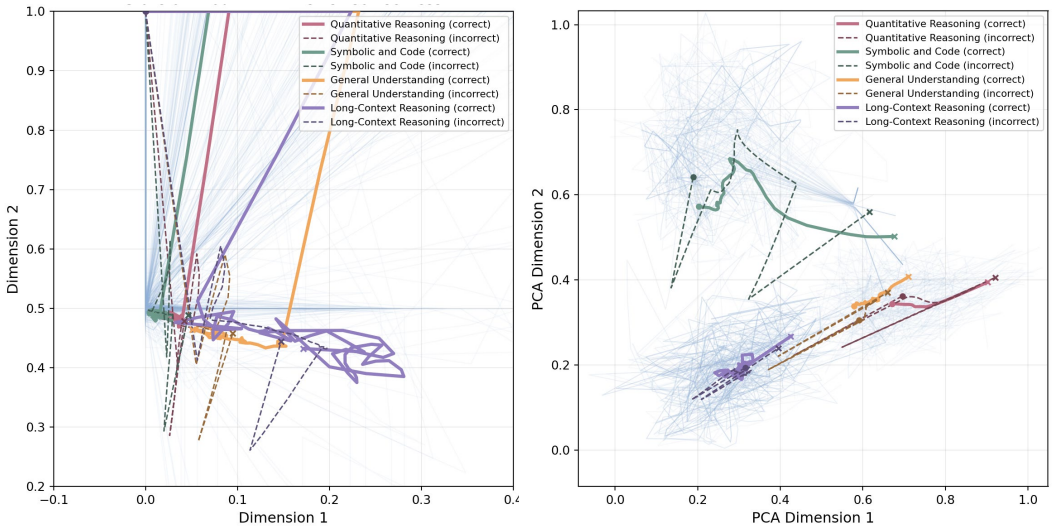


Figure 10: **Correct and incorrect trajectory regimes under full-state and embedding-trajectory SoT.** This figure compares successful and unsuccessful reasoning trajectories across all four task categories, under both the internal-state SoT and SoT-Embed.

O.3 Correct versus incorrect trajectories under internal and embedding-trajectory SoT

Finally, we compare successful and unsuccessful trajectories under both the full SoT state and the SoT-Embed variant from Sec. 2.2 and Sec. 4.2. For each of the four task categories, we aggregate trajectories by correctness and visualize the resulting state-space distributions. We perform the same analysis twice: once using the internal state derived from internal information transfer, and once using features derived from the sentence-embedding trajectory (Appendix D). This allows us to test whether the separation between healthy and unhealthy reasoning trajectories is preserved even when privileged internal access is removed.

Interpretation. Figure 10 serves two roles. For full SoT, it links gains to trajectory quality by showing whether correct runs occupy more stable and favorable state regimes than incorrect runs. For SoT-Embed, it tests whether this correct–incorrect separation persists under limited access; the preserved gap supports that SoT is a transferable trajectory-level principle rather than a privileged internal-state trick. The practical insight is that trajectory-state separability can function as a lightweight health signal for reliability monitoring, even when white-box internals are unavailable.

P Additional Results

This section provides supplementary result tables and figures omitted from the main text for space. Specifically, we provide backbone-specific main-result tables, token and latency breakdowns, and expanded boundary studies for the limited-access variants.

P.1 Additional backbone results

Table 12 and Table 13 report the full per-dataset accuracy results for the two LLM backbones not shown in the main text. Two patterns are notable. On Qwen2.5-14B, SoT achieves the best domain averages in QS (66.1), S&C (76.8), and GU (81.4); the GU gain is especially clear, with all five GU datasets reaching the highest scores in the table (CSQA 80.5, StrQA 75.0, BoolQ 86.8, MMLU 80.0, RACE 86.0). On Mixtral-8x7B, SoT is near-best in QS (46.3, close to 46.4) and leads clearly in S&C (51.7), GU (72.6), and LCR (23.5), indicating that the same state-conditioned control remains effective across substantially different backbone characteristics. Taken together, these cross-backbone results support a mechanism-level claim: SoT's advantage is not tied to one architecture or one dataset family, but to a transferable control principle that improves diverse reasoning regimes under heterogeneous model priors.

Table 12: **Main results on Qwen2.5-14B (accuracy, %).**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	94.4	77.5	1.9	65.6	0.0	55.5	84.0	75.0	66.0	49.9
	PS	88.0	51.5	2.1	54.4	0.5	65.5	92.0	50.0	47.2	49.6
	SR	92.0	68.2	17.6	65.5	61.1	83.4	79.0	60.0	50.4	68.1
	SC	92.8	61.0	1.9	59.5	0.5	59.5	87.0	55.0	48.8	47.2
	CB	60.0	41.2	30.8	46.4	62.1	85.0	51.0	50.0	60.8	69.8
	MCTS	66.2	50.0	19.3	49.1	63.1	88.5	53.0	60.0	63.6	72.6
Memory-Oriented	H2O	92.6	67.0	2.6	61.6	0.5	14.8	81.0	72.8	45.2	33.4
	SNAP	92.8	66.5	2.7	61.5	2.0	13.0	83.0	73.5	44.0	33.1
	STREAM	78.0	45.8	1.8	48.2	5.9	19.5	3.0	68.4	3.2	19.1
Latent	COCO	93.6	74.8	2.8	64.6	2.5	40.5	75.0	71.2	60.0	45.6
RL-Based	GRPO-SP	0.6	5.0	0.0	1.9	3.4	3.0	5.0	58.7	0.0	10.8
Ours	**SoT**	**91.8**	**72.2**	**15.1**	**66.1**	**66.5**	**87.8**	**91.0**	**75.6**	**62.8**	**76.8**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	42.8	51.2	61.5	39.8	36.8	46.6	2.0	4.6	18.4	5.7
	PS	58.0	58.4	42.0	48.6	40.2	49.9	2.5	5.6	23.9	7.3
	SR	75.0	71.6	81.8	74.0	63.4	73.7	16.3	25.0	30.8	22.0
	SC	40.2	55.8	61.8	38.6	48.6	48.5	2.0	4.4	22.5	6.3
	CB	71.0	70.2	84.5	67.8	63.8	71.7	25.8	33.5	26.3	28.8
	MCTS	71.2	69.4	85.8	65.8	62.6	71.1	17.0	29.0	28.5	23.5
Memory-Oriented	H2O	41.8	47.2	61.5	37.0	24.7	43.1	2.0	2.0	0.0	1.7
	SNAP	41.8	47.2	61.0	37.4	27.0	43.4	2.0	2.0	0.0	1.7
	STREAM	39.8	46.2	52.8	33.2	0.0	36.1	2.4	0.0	0.0	1.1
Latent	COCO	49.0	45.5	61.8	53.6	65.7	54.5	2.3	7.4	25.3	8.0
RL-Based	GRPO-SP	10.0	5.8	5.2	8.2	12.7	8.2	0.1	0.1	0.3	0.1
Ours	**SoT**	**80.5**	**75.0**	**86.8**	**80.0**	**86.0**	**81.4**	**24.2**	**25.4**	**27.3**	**25.2**

Table 13: **Main results on Mixtral-8x7B (accuracy, %).**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	64.8	41.5	9.9	43.3	23.2	43.0	47.0	20.0	40.4	38.1
	PS	60.6	35.8	10.6	39.8	12.8	38.2	52.0	25.0	42.4	35.1
	SR	49.2	37.8	9.7	35.5	27.6	54.5	51.0	15.0	32.0	41.9
	SC	69.8	43.0	9.8	45.9	25.6	45.2	56.0	20.0	37.2	39.7
	CB	50.0	30.2	10.1	33.4	48.3	60.5	39.0	10.0	34.4	48.0
	MCTS	45.6	26.8	8.0	29.9	50.7	57.2	39.0	20.0	38.4	48.4
Memory-Oriented	H2O	65.2	43.0	9.4	43.9	23.6	43.8	52.0	15.0	31.2	36.6
	SNAP	64.8	43.2	9.1	43.7	23.6	43.8	55.0	15.0	30.8	36.8
	STREAM	59.0	38.5	1.1	37.7	8.9	39.0	0.0	0.0	2.8	18.6
Latent	COCO	60.8	38.0	9.8	40.5	29.1	29.0	55.0	5.0	18.0	28.4
RL-Based	GRPO-SP	1.2	3.8	0.0	1.8	4.4	3.8	8.0	0.0	0.0	3.3
Ours	**SoT**	**69.2**	**42.0**	**14.0**	**46.3**	**43.8**	**67.5**	**55.0**	**33.5**	**43.2**	**51.7**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	46.5	53.5	72.8	49.8	47.7	54.1	14.5	15.3	37.8	18.7
	PS	51.5	38.5	46.8	49.6	47.3	46.9	11.1	17.4	33.2	17.2
	SR	65.5	62.0	80.2	63.2	61.0	66.5	9.9	12.2	33.0	14.6
	SC	44.5	57.2	75.8	55.0	50.0	56.8	14.5	15.3	38.1	18.8
	CB	67.2	69.5	82.0	58.6	49.3	65.8	13.0	16.6	37.4	18.4
	MCTS	60.0	68.2	81.0	53.6	54.0	63.4	9.5	17.3	36.4	16.9
Memory-Oriented	H2O	45.8	51.7	73.5	52.0	11.7	49.0	13.1	6.6	0.0	8.5
	SNAP	45.8	51.7	73.8	52.4	12.0	49.1	13.1	6.7	0.0	8.5
	STREAM	47.0	50.2	30.2	39.4	0.0	35.4	15.2	0.0	0.0	6.9
Latent	COCO	48.0	41.2	57.5	48.0	48.3	48.6	15.2	20.4	17.4	17.5
RL-Based	GRPO-SP	11.8	8.2	7.2	11.6	13.3	10.3	0.1	0.2	0.6	0.2
Ours	**SoT**	**71.0**	**67.8**	**81.5**	**68.4**	**76.0**	**72.6**	**17.2**	**22.2**	**43.7**	**23.5**

P.2 Extended efficiency and trade-off results

Here we provide full per-dataset efficiency breakdowns for Llama-3.1-8B (Tables 14, 15), Qwen2.5-14B (Tables 16, 17), and Mixtral-8x7B (Tables 18, 19). Two consistent trends emerge: (i) search-heavy methods incur very large token/latency overhead, especially on long-context tasks (e.g., Qwen LCR token average 4901.5 for SC vs. 254.0 for SoT; Mixtral LCR token average 7546.2 for CB vs. 260.6 for SoT), and (ii) SoT remains in a low-cost regime across backbones, with particularly strong

Llama efficiency in GU (74.9 tokens, 1.8s) and QS (255.7 tokens, 5.8s). This indicates that SoT improves the accuracy–efficiency frontier through selective state-conditioned computation rather than brute-force trajectory expansion, so its efficiency gains reflect better control allocation, not merely shorter outputs by default.

Table 14: **Generated reasoning tokens on Llama-3.1-8B.**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	230.1	381.5	194.1	271.6	342.9	332.0	264.4	490.1	415.4	369.8
	PS	311.0	358.9	259.9	314.2	369.9	354.4	346.0	404.6	391.4	372.1
	SR	346.5	441.6	307.6	368.5	337.8	330.4	336.4	517.6	473.0	391.7
	SC	678.4	928.6	578.2	736.8	913.0	875.4	798.1	1135.3	1056.2	953.9
	CB	441.8	521.0	264.5	423.9	325.8	324.6	290.0	424.8	447.3	363.9
	MCTS	462.2	482.3	275.0	422.1	327.0	326.9	286.7	435.1	440.7	364.7
Memory-Oriented	H2O	222.4	293.8	141.6	226.0	264.6	249.5	223.6	267.9	302.1	264.4
	SNAP	222.4	293.8	141.6	226.0	264.6	249.5	223.6	267.9	302.1	264.4
	STREAM	104.8	122.8	32.1	92.6	66.3	66.0	6.1	54.4	89.3	64.2
Latent	COCO	133.4	260.0	117.0	171.5	159.4	186.2	192.0	318.9	245.4	214.6
RL-Based	GRPO-SP	71.5	73.9	74.8	73.1	63.9	65.3	69.2	79.0	75.5	69.7
Ours	**SoT**	**186.1**	**452.0**	**129.1**	**255.7**	**253.7**	**254.9**	**197.2**	**404.0**	**405.2**	**302.2**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	194.7	221.7	164.8	279.4	178.8	212.9	275.4	165.9	60.7	198.1
	PS	283.7	344.4	310.1	353.0	292.8	319.8	316.1	310.6	125.8	282.3
	SR	278.3	304.0	260.9	326.8	278.8	292.2	326.9	284.1	135.3	278.8
	SC	568.3	660.8	488.6	790.3	569.0	626.5	671.0	541.9	141.3	533.8
	CB	392.8	426.7	298.7	364.2	180.3	341.7	443.4	196.4	46.7	283.7
	MCTS	375.6	423.3	301.9	353.5	178.0	335.2	446.1	196.6	56.2	286.6
Memory-Oriented	H2O	190.3	219.6	157.0	254.0	57.8	185.6	237.4	126.3	53.0	164.6
	SNAP	190.3	219.6	157.0	254.0	57.8	185.6	237.4	126.3	53.0	164.6
	STREAM	117.5	149.5	60.7	90.5	35.0	93.4	148.7	79.0	33.4	103.1
Latent	COCO	56.1	99.5	72.4	120.8	76.0	87.2	175.8	104.9	71.0	131.5
RL-Based	GRPO-SP	70.0	69.6	70.7	71.1	76.9	71.4	71.9	77.6	84.9	76.2
Ours	**SoT**	**107.5**	**121.2**	**21.8**	**97.6**	**26.4**	**74.9**	**435.2**	**207.1**	**146.6**	**263.0**

Table 15: **End-to-end latency on Llama-3.1-8B.**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	12.1	25.6	15.7	17.5	24.0	23.9	18.9	41.3	30.4	27.5
	PS	18.7	22.1	22.3	20.7	26.2	24.4	27.7	32.3	27.3	26.8
	SR	21.6	30.3	27.2	25.9	22.6	21.6	26.2	45.1	35.5	28.8
	SC	34.8	51.8	45.5	43.1	58.9	55.7	57.4	85.1	68.1	63.5
	CB	20.9	25.6	19.2	22.0	16.5	16.2	17.9	26.3	23.0	19.4
	MCTS	21.9	22.1	19.8	21.4	16.6	16.5	17.6	26.6	22.1	19.4
Memory-Oriented	H2O	13.3	21.9	15.5	16.7	23.4	46.2	20.5	38.1	28.4	34.6
	SNAP	13.3	22.0	15.1	16.6	23.2	47.2	20.6	37.2	28.4	34.8
	STREAM	11.0	16.4	9.9	12.5	18.1	33.9	11.4	22.3	19.2	24.0
Latent	COCO	6.7	16.0	13.4	11.5	10.2	28.0	14.0	24.9	16.0	20.4
RL-Based	GRPO-SP	6.6	6.8	9.9	7.5	6.2	6.5	7.7	8.7	7.4	7.1
Ours	**SoT**	**4.8**	**8.1**	**4.4**	**5.8**	**7.2**	**7.1**	**5.6**	**15.0**	**13.3**	**9.5**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	9.3	9.8	9.6	16.7	19.1	12.8	15.2	18.5	33.8	19.6
	PS	15.8	18.7	21.8	22.9	34.8	22.2	16.6	38.0	70.7	33.7
	SR	14.7	15.1	17.0	20.0	32.1	19.2	17.0	33.7	76.3	33.2
	SC	26.0	28.6	27.8	44.9	66.0	37.6	29.4	60.9	74.9	48.9
	CB	16.6	16.4	16.1	17.1	18.0	16.8	17.4	19.2	26.2	19.5
	MCTS	15.9	16.3	16.3	16.7	17.7	16.5	17.2	19.2	32.6	20.5
Memory-Oriented	H2O	9.9	10.6	10.4	17.0	35.7	15.8	13.4	35.6	11.5	21.5
	SNAP	9.9	10.6	10.4	18.4	35.6	16.1	13.3	34.3	11.4	20.9
	STREAM	8.4	11.0	7.2	15.3	4.5	9.8	11.2	13.4	1.0	10.3
Latent	COCO	2.5	3.9	4.1	6.9	18.9	6.7	9.6	27.5	93.3	30.3
RL-Based	GRPO-SP	6.6	5.8	7.3	6.8	12.4	7.5	6.1	12.7	115.6	26.9
Ours	**SoT**	**2.4**	**2.6**	**0.8**	**1.7**	**1.6**	**1.8**	**6.1**	**7.8**	**97.5**	**22.0**

Table 16: **Generated reasoning tokens on Qwen2.5-14B.**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	399.4	521.3	544.5	476.3	503.3	416.7	456.0	644.4	597.8	490.0
	PS	653.4	698.4	1021.3	760.4	639.7	633.2	896.3	935.2	829.9	718.3
	SR	1326.7	1454.9	1804.9	1489.0	1496.6	1528.6	1601.1	1891.5	1717.5	1625.0
	SC	1188.4	1366.3	1646.3	1362.2	1430.4	1227.5	1433.2	1798.6	1614.9	1402.2
	CB	3868.3	3866.0	4187.8	3947.4	3846.2	3823.3	4166.9	3991.8	4045.2	3923.9
	MCTS	3899.3	3884.1	4184.7	3965.6	3896.8	3822.2	4170.8	4040.7	3998.4	3923.3
Memory-Oriented	H2O	390.5	446.4	490.7	434.2	447.9	398.1	453.6	509.5	488.4	440.5
	SNAP	390.5	446.4	490.7	434.2	447.9	398.1	453.6	509.5	488.4	440.5
	STREAM	260.7	262.7	371.5	289.1	263.1	256.1	272.2	279.4	267.2	262.8
Latent	COCO	371.8	466.9	520.8	440.7	473.8	314.2	441.7	597.3	579.5	436.6
RL-Based	GRPO-SP	861.6	839.9	1065.7	905.4	920.6	893.4	966.5	902.2	888.4	905.4
Ours	**SoT**	**885.0**	**392.7**	**184.6**	545.8	**338.4**	**233.7**	**310.4**	**569.2**	**678.2**	408.4

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	263.1	251.8	314.9	401.0	612.3	358.1	322.1	729.9	6597.5	529.6
	PS	490.2	472.6	686.0	660.5	638.2	590.6	507.3	1386.6	14498.2	3172.2
	SR	985.7	938.4	1139.9	1361.6	702.1	1058.5	1156.6	2363.8	21847.2	5062.3
	SC	810.6	771.2	951.1	1206.6	685.4	911.0	990.2	2208.3	21690.8	4901.7
	CB	3706.4	3592.7	3862.3	3802.0	3688.4	3736.0	3814.4	4637.4	21703.1	7107.6
	MCTS	3844.0	3728.3	3915.2	3853.4	3715.6	3818.2	3724.2	4612.7	21714.8	7059.2
Memory-Oriented	H2O	264.0	263.9	313.3	389.6	572.2	351.5	321.7	624.7	7141.2	1573.1
	SNAP	264.0	263.9	313.3	389.6	572.2	351.5	321.7	624.7	7141.2	1573.1
	STREAM	238.4	231.8	264.0	259.1	550.5	294.2	257.2	565.4	7141.2	1521.3
Latent	COCO	242.7	207.4	300.5	345.7	652.5	334.4	311.1	671.2	7246.6	1603.4
RL-Based	GRPO-SP	859.0	823.0	945.2	882.9	1264.8	935.9	819.9	1280.9	7806.1	2158.9
Ours	**SoT**	**149.2**	**126.4**	**45.1**	**199.4**	**127.8**	**133.2**	**229.2**	**175.1**	**500.9**	**254.0**

Table 17: **End-to-end latency on Qwen2.5-14B.**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	25.0	47.4	26.6	32.9	39.2	25.8	23.0	60.7	56.7	36.9
	PS	27.9	36.7	32.2	31.9	22.4	19.6	38.0	50.1	46.1	29.5
	SR	37.8	56.3	43.5	45.4	41.1	49.8	39.6	87.1	80.4	59.6
	SC	73.0	100.4	81.7	84.3	107.6	72.8	79.9	149.6	129.4	96.9
	CB	40.6	43.1	39.0	41.0	35.0	33.0	40.1	49.8	52.6	39.5
	MCTS	43.7	46.8	41.5	44.2	34.8	30.5	40.0	50.6	52.8	38.5
Memory-Oriented	H2O	16.2	25.4	15.0	18.9	23.5	16.3	15.7	33.8	30.0	21.8
	SNAP	16.2	25.5	14.9	18.9	23.5	16.1	15.7	34.6	30.9	21.9
	STREAM	14.5	21.7	11.9	16.2	19.2	14.6	13.5	28.1	27.0	19.0
Latent	COCO	12.9	21.2	12.3	15.5	19.5	7.3	12.3	27.9	29.0	16.5
RL-Based	GRPO-SP	66.3	64.0	85.5	70.3	71.7	68.9	76.0	68.9	68.4	70.1
Ours	**SoT**	**21.1**	**16.8**	**9.5**	16.8	**14.8**	**10.2**	**14.3**	**27.4**	**32.8**	19.0

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	10.1	11.4	9.2	25.3	21.8	15.7	18.1	35.2	254.1	26.7
	PS	11.9	13.2	16.8	25.1	23.6	18.2	20.1	38.3	262.8	67.5
	SR	18.9	20.6	18.9	35.4	30.2	25.1	32.0	57.2	366.3	97.2
	SC	32.2	33.8	27.5	72.4	65.8	46.7	56.4	110.1	555.9	160.0
	CB	27.2	28.4	27.3	34.9	29.7	29.8	39.2	42.0	162.4	60.8
	MCTS	32.6	34.1	31.9	35.9	30.5	33.3	40.1	43.3	172.1	63.3
Memory-Oriented	H2O	7.0	8.8	5.9	15.8	10.7	9.9	12.6	14.8	7.4	12.6
	SNAP	6.9	8.7	5.9	15.8	11.7	10.0	12.6	15.3	7.4	12.8
	STREAM	6.3	8.3	5.7	15.3	3.6	8.4	11.2	12.8	5.4	10.8
Latent	COCO	5.3	5.0	4.4	10.5	10.6	7.1	10.7	11.7	126.2	30.3
RL-Based	GRPO-SP	65.7	62.2	74.1	67.9	105.5	73.2	61.9	107.8	787.9	200.3
Ours	**SoT**	**6.1**	**5.0**	**2.2**	**8.2**	**7.0**	**5.8**	**10.2**	**10.2**	**89.6**	**23.4**

Table 18: **Generated reasoning tokens on Mixtral-8x7B.**

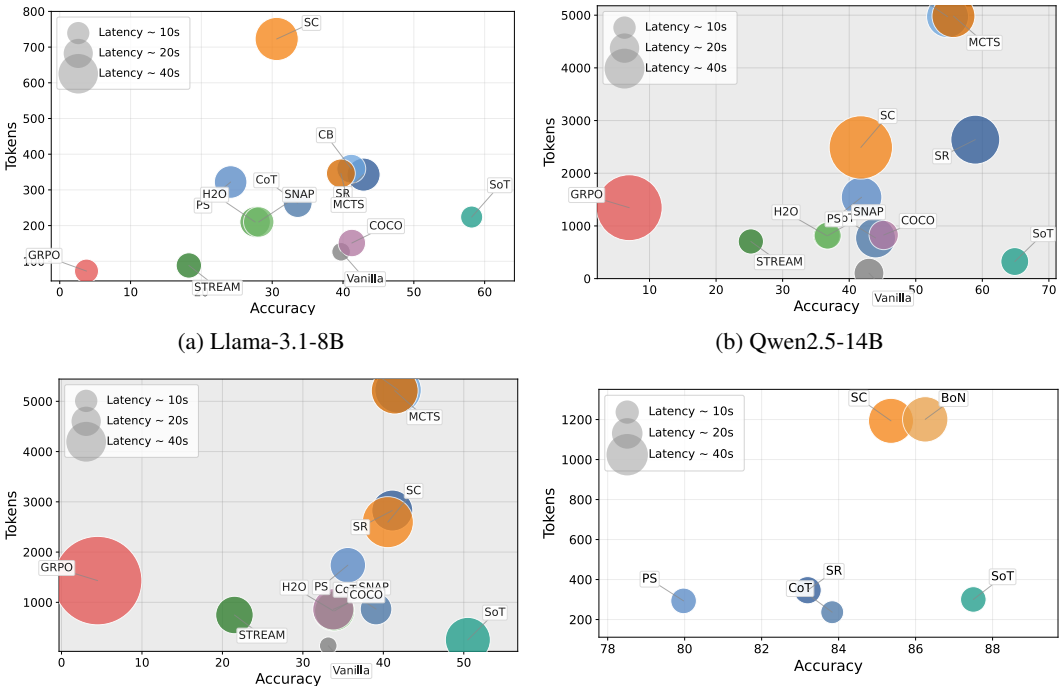
Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	348.8	439.0	534.7	425.3	394.8	353.3	471.7	617.2	574.4	436.3
	PS	665.9	703.1	1094.4	785.4	801.5	728.8	944.9	1000.4	868.1	807.6
	SR	1222.4	1336.4	1779.2	1399.6	1403.3	1283.2	1782.4	1970.0	1709.8	1483.3
	SC	1038.6	1205.6	1612.7	1237.8	1168.6	1070.2	1408.3	1817.2	1588.2	1273.9
	CB	3872.7	3857.2	4205.3	3950.7	3913.1	3867.3	4130.2	4221.4	4093.7	3969.3
	MCTS	3901.8	3917.4	4191.3	3979.4	3991.9	3915.2	4148.5	4170.1	4070.7	4000.4
Memory-Oriented	H2O	339.4	392.5	485.2	393.5	378.2	354.1	443.5	500.1	480.9	403.9
	SNAP	339.4	392.5	485.2	393.5	378.2	354.1	443.5	500.1	480.9	403.9
	STREAM	256.9	259.2	386.4	290.0	261.4	254.0	279.1	288.6	268.8	262.7
Latent	COCO	295.1	327.0	498.1	356.5	313.1	343.1	422.9	591.0	553.2	404.2
RL-Based	GRPO-SP	892.3	882.3	1138.4	950.5	954.7	943.2	1045.8	967.9	952.0	958.9
Ours	**SoT**	**208.0**	**361.9**	**163.7**	**248.2**	**192.0**	**155.6**	**184.4**	**605.5**	**597.6**	**329.8**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	249.9	210.3	303.7	328.0	678.5	336.6	251.4	726.3	7933.4	1711.6
	PS	564.3	520.0	776.7	698.3	1504.5	772.5	504.2	1532.9	16024.9	3480.7
	SR	1028.7	930.4	1183.0	1336.8	2407.1	1323.7	1021.2	2314.3	24064.0	5351.5
	SC	759.4	642.2	914.4	999.0	2068.0	1023.1	765.5	2173.8	23802.6	5138.5
	CB	3859.3	3821.0	3966.2	3903.2	4678.4	4006.8	3831.2	4804.9	23908.3	7546.2
	MCTS	3884.2	3752.5	3961.4	3924.7	4651.3	3998.5	3721.6	4781.4	23933.4	7491.7
Memory-Oriented	H2O	248.2	208.3	300.8	323.5	597.7	322.0	249.8	645.0	7872.4	1670.0
	SNAP	248.2	208.3	300.8	323.5	597.7	322.0	249.8	645.0	7872.4	1670.0
	STREAM	223.8	198.1	265.1	247.1	582.8	286.6	231.2	602.8	7872.4	1645.5
Latent	COCO	260.0	268.5	365.4	329.5	728.5	370.4	231.2	713.4	8116.8	1728.1
RL-Based	GRPO-SP	881.4	832.7	968.3	919.0	1323.5	964.7	835.6	1341.9	8574.6	2317.2
Ours	**SoT**	**135.0**	**123.8**	**95.7**	**182.5**	**180.2**	**143.6**	**185.1**	**281.3**	**419.8**	**260.6**

Table 19: **End-to-end latency on Mixtral-8x7B.**

Category	Method	Quantitative Reasoning				Symbolic and Code					
		GSM	MATH	DROP	avg.	FOL	PW	BBH	HE	MBPP	avg.
Reasoning Paradigm	CoT	26.7	40.5	18.0	29.1	28.6	19.3	27.0	49.7	46.7	29.7
	PS	31.2	41.1	23.8	32.6	36.3	25.1	37.1	46.7	42.7	33.6
	SR	44.6	52.2	29.9	43.5	51.7	39.0	54.1	72.8	64.1	50.4
	SC	76.4	130.4	56.5	89.4	70.8	66.0	72.4	132.9	123.9	83.9
	CB	63.2	65.4	35.5	57.0	56.1	53.8	52.1	71.5	72.0	59.1
	MCTS	66.3	70.5	39.9	61.1	60.6	55.7	57.1	70.7	71.2	61.2
Memory-Oriented	H2O	42.8	64.0	36.7	48.4	47.5	36.2	38.3	96.7	89.6	53.7
	SNAP	43.6	65.3	35.8	48.9	42.3	36.5	38.4	96.4	88.5	52.5
	STREAM	38.6	56.9	26.8	41.7	35.6	31.8	33.3	85.8	81.3	46.6
Latent	COCO	33.5	42.1	26.3	34.6	23.4	34.6	30.7	82.6	79.4	44.4
RL-Based	GRPO-SP	190.2	278.6	206.3	223.7	191.5	203.6	204.7	186.2	186.4	196.4
Ours	**SoT**	**37.7**	**60.1**	**32.1**	**43.8**	**38.3**	**30.3**	**35.3**	**115.1**	**110.2**	**62.6**

Category	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Reasoning Paradigm	CoT	15.6	15.1	11.6	21.3	15.8	16.2	19.3	15.2	15.1	17.1
	PS	22.2	25.6	27.2	30.7	34.3	27.8	22.5	23.3	31.3	24.3
	SR	33.7	35.4	27.8	44.3	37.4	36.1	36.2	26.4	40.0	33.1
	SC	46.2	46.4	30.8	63.9	43.4	47.2	59.3	44.8	33.8	49.5
	CB	97.1	66.4	49.3	56.4	37.1	62.2	66.7	28.6	23.0	45.0
	MCTS	65.3	72.4	55.6	61.1	40.3	60.0	68.2	31.1	25.3	47.0
Memory-Oriented	H2O	26.3	25.6	16.2	35.4	30.0	27.0	34.8	35.6	4.3	30.0
	SNAP	26.3	26.1	16.3	36.0	31.3	27.4	33.2	35.3	3.9	29.1
	STREAM	25.2	24.4	14.7	32.4	15.1	23.2	32.9	27.5	2.5	25.8
Latent	COCO	28.8	40.4	31.8	34.4	40.2	34.8	30.7	29.2	118.6	44.8
RL-Based	GRPO-SP	184.1	185.1	197.3	198.7	242.5	199.4	172.2	232.8	357.4	226.0
Ours	**SoT**	**26.8**	**23.0**	**19.4**	**34.0**	**37.4**	**28.0**	**35.3**	**52.7**	**160.8**	**62.8**



(a) Llama-3.1-8B

(b) Qwen2.5-14B

(c) Mixtral-8x7B

(d) Qwen2.5-VL-7B/32B

Figure 11: **Extended accuracy–efficiency trade-off plots across backbones.**

Table 20: **Additional ablation results on Qwen2.5-14B.**

Variant	QS			S&C			GU			LCR		
	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.
Full SoT	66.1	545.8	16.8	76.8	408.4	19.0	81.3	133.2	5.8	25.2	254.0	23.4
+ Threshold Tuning	66.1	545.8	16.8	76.8	408.4	19.0	81.3	133.2	5.8	25.2	254.0	23.4
w/o Evid. Org. $\mathcal{S}$	30.0	532.1	21.5	28.4	556.1	21.7	72.4	234.7	9.4	26.5	489.4	14.3
w/o Stopping $\mathcal{T}$	76.0	1401.9	37.4	32.4	2326.2	59.6	70.2	4175.0	100.9	31.7	2815.8	72.8
w/o Geometry δ_t	56.5	77.9	3.9	23.1	45.8	2.5	73.0	25.4	0.9	21.2	48.2	2.2
w/o Dynamics (v_t, c_t)	70.0	1220.9	32.1	30.8	910.0	26.9	70.6	1170.5	31.8	31.2	869.0	27.8
w/o Uncertainty H_t	62.5	306.0	7.6	30.6	255.6	7.6	73.0	61.3	1.9	33.6	190.2	6.6
w/o Sent.-level Units	62.5	306.0	7.6	30.6	255.6	7.6	73.0	61.3	1.9	33.6	190.2	6.6

Figure 11 shows a consistent cross-backbone pattern: methods that gain accuracy mainly through broader sampling/search move upward to high-token and high-latency regions, while SoT stays on or near the Pareto frontier. On text backbones, SoT combines top-tier accuracy with substantially lower compute than search-heavy alternatives, indicating that its gains come from better control allocation rather than budget expansion. The same trend also appears in multimodal transfer, supporting the view that state-conditioned evidence organization is a mechanism-level efficiency advantage rather than a backbone-specific tuning effect.

P.3 Additional ablation results

To test whether the main ablation pattern is specific to Llama-3.1-8B or reflects a more stable mechanism, we report additional ablations on Qwen2.5-14B, Mixtral-8×7B in Table 20 and Table 21. The cross-backbone pattern is consistent with the main-text ablation conclusion: removing evidence organization ($\mathcal{S}$) causes clear accuracy degradation (e.g., Qwen QS $66.1 \rightarrow 30.0$, S&C $76.8 \rightarrow 28.4$; Mixtral QS $46.3 \rightarrow 31.3$, LCR $23.5 \rightarrow 3.0$). Removing stopping ($\mathcal{T}$) can occasionally increase one domain score, but it breaks efficiency by orders of magnitude (e.g., Qwen GU tokens $133.2 \rightarrow 4175.0$). Meanwhile, threshold tuning is negligible on text backbones (identical to Full SoT on Qwen/Mixtral), and state-component removals typically trade short outputs for weaker accuracy, supporting the same mechanism-level claim as the main text: SoT gains come from the full closed-loop control, not from isolated calibration or trivial truncation.

Table 21: **Additional ablation results on Mixtral-8×7B.**

Variant	QS			S&C			GU			LCR		
	acc.	**tok.**	**lat.**	**acc.**	**tok.**	**lat.**	**acc.**	**tok.**	**lat.**	**acc.**	**tok.**	**lat.**
Full SoT	46.3	248.2	43.8	51.7	329.8	62.6	72.5	143.6	28.0	23.5	260.6	62.8
+ Threshold Tuning	46.3	248.2	43.8	51.7	329.8	62.6	72.5	143.6	28.0	23.5	260.6	62.8
w/o Evid. Org. $\mathcal{S}$	31.3	995.0	70.4	36.0	1002.6	65.8	63.2	836.8	51.0	3.0	921.1	90.7
w/o Stopping $\mathcal{T}$	31.3	995.0	64.0	36.0	1002.6	65.4	63.2	836.8	50.7	3.0	921.1	90.6
w/o Geometry δ_t	34.0	58.1	11.8	30.7	56.9	13.3	57.6	50.8	7.6	3.3	54.0	13.8
w/o Dynamics (v_t, c_t)	31.3	995.0	64.0	36.0	1002.6	65.4	63.2	836.8	50.7	3.0	921.1	90.6
w/o Uncertainty H_t	31.3	995.0	64.0	36.0	1002.6	65.4	63.2	836.8	50.7	3.0	921.1	90.6
w/o Sent.-level Units	31.3	995.0	64.0	36.0	1002.6	65.4	63.2	836.8	50.7	3.0	921.1	90.6

P.4 Limited-access variants and SoT-Judge

We further expand the exploratory results in Section 4 with Tables 22 and 23.

On Qwen2.5-14B, Training-free SoT reaches an LCR average of 26.2 versus 25.8 for the Top-3 baselines, while SoT-Embed remains competitive at 23.1. On Mixtral-8x7B, both variants exceed the Top-3 averages in quantitative, symbolic/code, and long-context reasoning, reaching 26.4 and 27.9 versus 23.7 on LCR. The consistent HQA/NarQA gains across both backbones show that trajectory-level state summaries retain useful long-range structure when either controller fitting or internal-state access is removed.

Table 22: **Boundary studies of SoT on Qwen2.5-14B (accuracy, %).**

	Quantitative Reasoning				**Symbolic and Code**					
	GSM8K	MATH	DROP	avg.	FOLIO	PW	BBH-T	HE	MBPP	avg.
Top-3 Baseline avg.	93.6	73.5	22.6	63.2	62.1	85.6	87.7	73.8	63.5	74.5
SoT-Training-free	81.8	60.8	34.6	59.1	49.5	30.2	77.0	53.0	50.4	52.0
SoT-Embedding	82.6	58.3	34.8	58.6	51.2	59.5	71.0	48.8	49.6	56.0

	General Understanding						**Long-Context Reasoning**			
	CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Top-3 Baseline avg.	72.4	70.4	84.0	69.2	64.3	72.1	19.7	29.2	28.5	25.8
SoT-Training-free	59.3	55.2	70.0	45.4	52.7	56.5	22.9	34.2	21.6	26.2
SoT-Embedding	57.0	43.0	64.8	42.8	48.7	51.3	21.2	29.8	18.2	23.1

Table 23: **Boundary studies of SoT on Mixtral-8x7B (accuracy, %).**

	Quantitative Reasoning				**Symbolic and Code**					
	GSM8K	MATH	DROP	avg.	FOLIO	PW	BBH-T	HE	MBPP	avg.
Top-3 Baseline avg.	66.6	43.1	10.2	40.0	42.7	57.4	55.3	21.7	40.4	43.5
SoT-Training-free	75.2	52.2	34.4	53.9	31.3	50.5	63.0	56.1	49.2	50.0
SoT-Embedding	76.0	52.0	36.7	54.9	30.8	50.5	63.0	55.5	49.2	49.8

	General Understanding						**Long-Context Reasoning**			
	CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
Top-3 Baseline avg.	64.2	66.6	81.1	58.9	55.0	65.2	15.0	18.4	37.8	23.7
SoT-Training-free	49.5	54.5	66.5	50.8	48.3	53.9	24.3	34.3	20.5	26.4
SoT-Embedding	49.3	44.3	52.5	51.2	47.5	49.0	24.3	34.0	25.5	27.9

Table 24: **Black-box judging with trajectory-based SoT (dataset breakdown, %).**

Model	Method	Quantitative Reasoning				Symbolic and Code					
		GSM8K	MATH	DROP	avg.	FOLIO	PW	BBH-T	HE	MBPP	avg.
GPT-5.4	Self-Consistency	92.0	78.0	5.0	58.3	69.0	77.0	86.0	87.0	77.0	79.2
	Self-Verification	92.0	78.0	5.0	58.3	62.0	79.0	85.0	87.0	78.0	78.2
	LLM-as-a-Judge	92.0	79.0	4.0	58.3	70.0	76.0	86.0	86.0	78.0	79.2
	SoT-Judge	**85.0**	**79.0**	**94.0**	86.0	**50.0**	**60.0**	**77.0**	**86.0**	**76.0**	69.8
Claude 4.7	Self-Consistency	90.0	78.0	1.0	56.3	65.0	73.0	58.0	94.0	86.0	75.2
	Self-Verification	94.1	80.0	4.0	59.4	71.4	80.0	57.1	98.4	94.4	80.3
	LLM-as-a-Judge	90.0	78.0	2.0	56.7	66.0	74.0	57.0	94.0	86.0	75.4
	SoT-Judge	**90.0**	**78.0**	**91.0**	86.3	**50.0**	**78.0**	**23.0**	**97.0**	**87.0**	67.0
Qwen-Max	Self-Consistency	88.0	82.8	3.0	57.9	76.0	86.0	81.0	83.8	73.0	80.0
	Self-Verification	46.7	35.3	6.0	29.3	80.6	82.9	74.5	81.5	70.0	77.9
	LLM-as-a-Judge	92.0	82.8	3.0	59.3	74.0	86.0	81.0	83.8	74.0	79.8
	SoT-Judge	**77.0**	**75.9**	**98.0**	83.6	**68.0**	**64.0**	**74.0**	**75.0**	**67.0**	69.6

Model	Method	General Understanding						Long-Context Reasoning			
		CSQA	StrQA	BoolQ	MMLU	RACE	avg.	HQA	NarQA	LB	avg.
GPT-5.4	Self-Consistency	85.0	82.0	94.0	76.0	70.0	81.4	7.0	9.0	34.0	16.7
	Self-Verification	84.0	81.0	93.0	75.0	70.0	80.6	7.0	9.0	32.0	16.0
	LLM-as-a-Judge	86.0	85.0	94.0	75.0	70.0	82.0	7.0	9.0	33.0	16.3
	SoT-Judge	**85.0**	**54.0**	**67.0**	**74.0**	**70.0**	70.0	**98.0**	**87.0**	**58.0**	81.0
Claude 4.7	Self-Consistency	83.0	86.0	95.0	76.0	78.0	83.6	11.0	1.0	3.0	5.0
	Self-Verification	46.5	87.2	94.0	23.3	62.7	62.7	23.0	1.0	4.0	9.3
	LLM-as-a-Judge	82.0	85.0	95.0	77.0	79.0	83.6	11.0	1.0	4.0	5.3
	SoT-Judge	**81.0**	**49.0**	**53.0**	**67.0**	**74.0**	64.8	**100.0**	**99.0**	**98.0**	99.0
Qwen-Max	Self-Consistency	82.0	82.0	91.0	77.0	88.0	84.0	11.0	1.0	31.5	14.5
	Self-Verification	86.7	86.2	93.8	80.6	85.0	86.5	1.9	1.4	31.5	11.6
	LLM-as-a-Judge	84.0	82.0	93.0	77.0	88.0	84.8	12.0	1.0	31.5	14.8
	SoT-Judge	**80.0**	**74.0**	**75.0**	**72.0**	**86.0**	77.4	**95.0**	**100.0**	**51.7**	82.2

We report full judge behavior in Table 24. The same insight as the main section is reinforced: SoT-Judge's largest gains concentrate in trajectory-structure-sensitive regimes (especially LCR, e.g., GPT-5.4 81.0 vs. $\leq$ 16.7 for baselines, Claude 4.7 99.0 vs. $\leq$ 9.3, Qwen-Max 82.2 vs. $\leq$ 14.8), while trading some GU/S&C agreement for much stronger QR/LCR agreement, suggesting that trajectory-state signals are most informative where long-horizon coherence, not local answer matching, is the primary bottleneck.

P.5 Sensitivity to trajectory quality

We test whether SoT remains informative when the offline trajectory pool is collected or curated differently, with Llama-3.1-8B results in Table 25 (default HQ pool vs. weaker variants and training-free SoT). The table shows a consistent split: weaker pools largely leave QS/S&C accuracy near plateau but disproportionately hurt GU/LCR while greatly inflating tokens and latency relative to HQ; training-free SoT sits in the same heavy-trajectory regime with domain-mixed accuracy shifts. Overall, curation affects whether supervision yields a cheap, reliable trajectory-level control signal more than it acts as a uniform rescaling of backbone capability.

Table 25: **Sensitivity to trajectory-pool quality on Llama-3.1-8B**.

Trajectory pool	QS			S&C			GU			LCR		
	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.	acc.	tok.	lat.
Full HQ pool	70.8	247.9	6.3	57.1	324.2	11.1	75.5	58.0	1.6	37.3	121.1	97.5
No filter	71.3	931.3	30.0	57.7	1080.8	84.1	49.2	886.0	29.5	21.5	900.9	91.5
Weak filter	71.3	931.3	24.3	57.7	1080.8	34.5	49.2	886.0	17.7	21.3	900.9	90.7
No pipeline	71.3	931.3	26.1	57.7	1080.8	53.4	49.2	886.0	49.3	21.5	900.9	89.3
Training-free	72.3	917.4	26.0	58.3	1078.9	35.0	53.8	906.6	19.7	28.1	872.9	95.8

Q Case Studies

Q.1 Case 1: Quantitative Reasoning (QS)

Figure 12 follows a staged numeric workflow (per-store accounting, then comparison). Intermediate steps reuse only the quantities that remain algebraically active in the next operation, and $p(\text{stop})$ ramps once the competing subtotals are both fixed: the mechanism reads as *derivation support* that is pruned as subgoals close, not as uniform context replay.

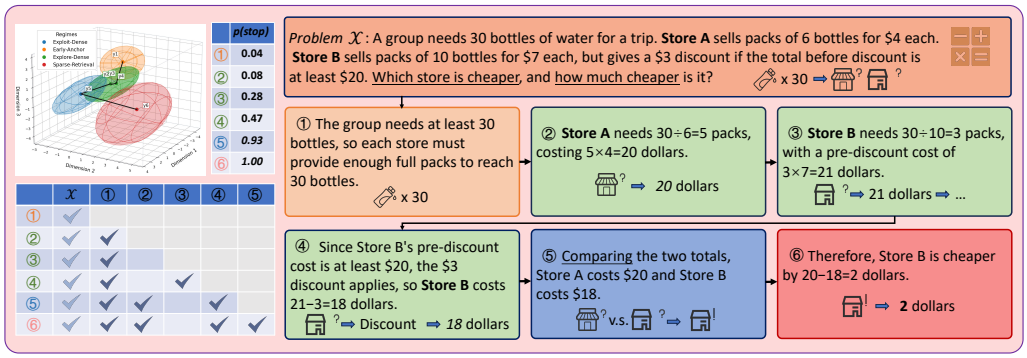


Figure 12: Case study on Quantitative Reasoning (QS).

Q.2 Case 2: General Understanding (GU)

Figure 13 is a commonsense elimination problem: late steps depend on the prompt and the locally decisive intermediates while dropping earlier scaffolding—a pattern that matters when verbal distractors would otherwise pollute a full-history transcript.

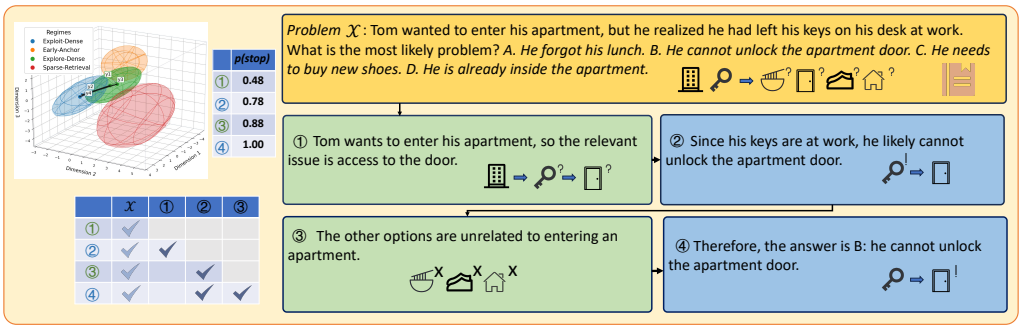


Figure 13: Case study on General Understanding (GU).

Q.3 Case 3: Symbolic and Code (S&C)

Figure 14 chains explicit rules forward but the evidence matrix is non-monotone—later steps re-anchor on conclusions and skip obsolete rule instantiations—so correctness is tied to *structural* carry (what still entails the next judgment), not to preserving the full forward chain verbatim.

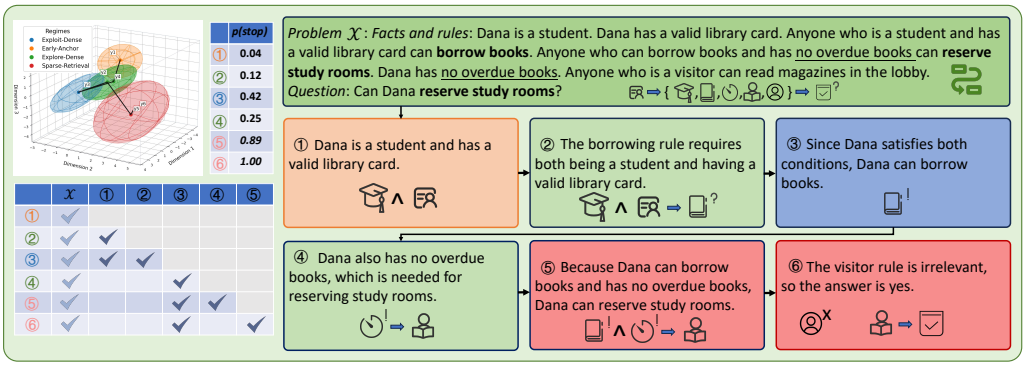


Figure 14: Case study on Symbolic and Code (S&C).

Q.4 Case 4: Long-Context Reasoning (LCR)

Figure 15 integrates constraints spread across a long narrative; the trajectory lingers in exploration-like regimes while facts are harvested, then compresses into a sparse-retrieval phase that fuses only the clauses still relevant to the contrastive *why* question—i.e., compositional support rather than verbatim long-context carry.

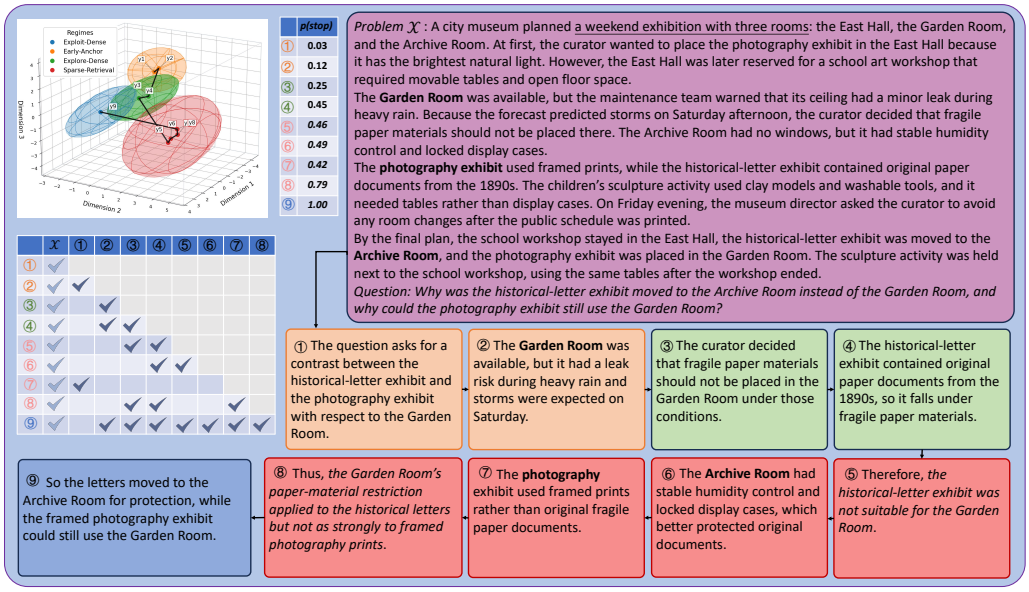


Figure 15: Case study on Long-Context Reasoning (LCR).

Q.5 Case 5: Visual-language reasoning

Figure 16 (Qwen3-VL-8B) shows the same interface on chart QA: panel-wise extraction, numeric gap reasoning, rising $p(\text{stop})$ through the computation, and sparse dependencies—evidence that SoT remains a trajectory-level organizer when evidence is grounded in pixels rather than text alone.

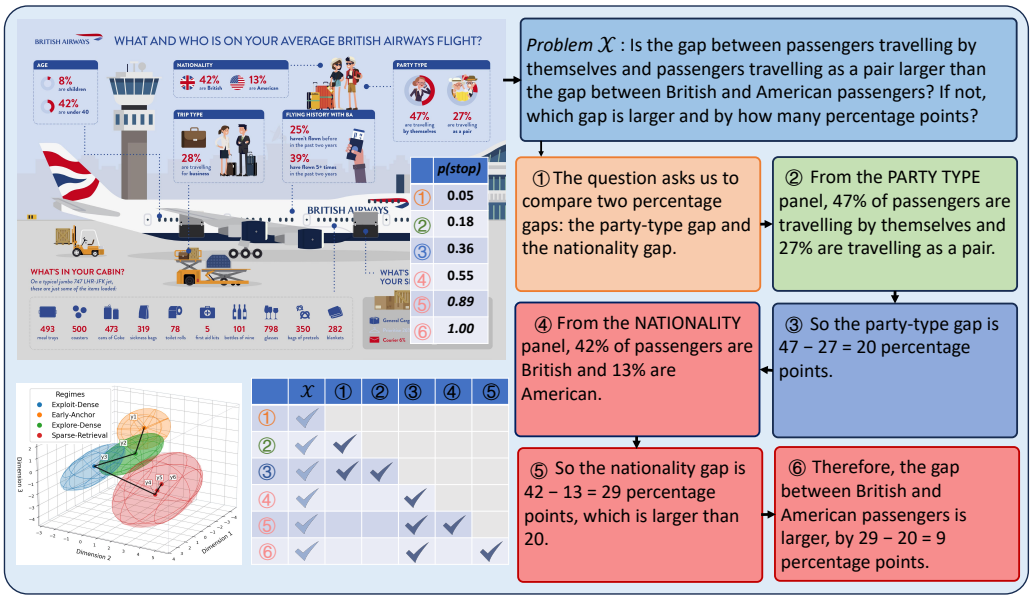


Figure 16: Case study on visual-language reasoning.

Part E: Related Work and Discussion

R Related Work

This section expands the compressed related-work discussion from the main paper. SoT sits at the intersection of four research directions, but is not reducible to any one of them.

R.1 Externally scripted reasoning

One major line of work improves reasoning by imposing explicit inference-time structure: step-by-step prompting [Wei et al., 2022], self-consistency [Wang et al., 2023b], plan-and-solve decomposition [Wang et al., 2023a], self-refinement [Madaan et al., 2023], and tree-style search [Yao et al., 2023a]. These methods substantially advance empirical reasoning performance, but they share a common modeling assumption: reasoning improves when we prescribe a better *external* program for how the model should think.

SoT differs at the level of problem formulation. It does not ask for a better externally specified chain. It asks whether the model's current endogenous regime can decide what evidence should remain active for the next decision. In this sense, SoT shifts the focus from *external reasoning choreography* to *endogenous reasoning control*.

R.2 Internal reasoning trajectories

Another relevant line studies hidden-state geometry, representation dynamics, and correctness-related trajectory structure in language models [Ballon et al., 2026, Damirchi et al., 2026, Park et al., 2025, 2024, Sun et al., 2026]. Related work also shows that internal signals can support monitoring and early-exit decisions [Feng et al., 2025, Hosseini et al., 2026, Tikhonov et al., 2026]. These works motivate the view that reasoning-relevant structure exists inside the model.

However, most of this literature is primarily descriptive: it interprets hidden trajectories, monitors confidence, or probes latent variables. SoT draws on similar geometric-dynamic structure but uses it differently. Rather than analyzing the state after the fact, SoT engages it online as the control variable that determines evidence organization and continuation.

R.3 Context augmentation and compression

Retrieval-augmented and context-management methods also intersect with SoT. Examples include retrieval-interleaved reasoning [Trivedi et al., 2023, Wang et al., 2024], context compression [Jiang et al., 2024b, Li et al., 2023], and KV-cache or memory-selection methods such as H_2O, SnapKV, and StreamingLLM [Li et al., 2024, Xiao et al., 2023, Zhang et al., 2023]. These methods are close to SoT in execution form because they do not always carry the entire history forward.

The key distinction lies in *what drives selection*. Most prior methods rely on semantic relevance, recency, generic compression objectives, or infrastructure efficiency. SoT instead conditions historical activation on the model's current reasoning regime. This turns memory selection from a general context-management tool into a component of reasoning control itself.

R.4 Latent reasoning and test-time control

Recent work also explores reasoning in latent space or with alternative test-time control [Hao et al., 2025, Hassid et al., 2025, Wang et al., 2025]. Some methods push more reasoning inside hidden computation; others show that shorter or differently allocated reasoning traces can be beneficial. These directions are highly relevant because they challenge the naive idea that better reasoning must always mean longer explicit token chains.

SoT is complementary to this line. It does not replace explicit reasoning with a fully latent process, nor does it merely shorten chains. Instead, it treats explicit sentence-level reasoning as a trajectory whose supporting context should be routed by a compact endogenous controller. This makes SoT a bridge between explicit-token and latent-control views of reasoning.

R.5 Positioning summary

Overall, SoT should not be read as another CoT variant, another search baseline, or another memory-pruning heuristic. Its central contribution is to formulate reasoning-time control as a closed loop in which endogenous state governs sparse evidence activation and stopping. That is a different object of study from externally scripted reasoning, descriptive hidden-state analysis, or generic context compression.

S Limitations and Future Work

This paper focuses on a concrete instantiation of SoT across multiple text backbones and two scales of one multimodal backbone family; several scope boundaries naturally remain. From a broader-impact perspective, improving reasoning under fixed inference budgets could make capable models more usable in cost- or latency-sensitive settings (e.g., education, healthcare-related assistance, research tooling), where efficiency and reliability both matter. As with any method that can improve fluent reasoning, deployment should follow domain-appropriate safeguards (e.g., policy constraints, human oversight in high-stakes use, and monitoring for misuse).

S.1 Current limitations

- **Performance depends on the state interface.** SoT routes memory and stopping through the extracted endogenous state; if that interface is a poor summary of the regime relevant to the next decision, control quality can degrade.
- **The current state is intentionally compact.** Our implemented controller uses $m_t \in \mathbb{R}^4$ for clarity and portability, but this choice may under-specify unusually long, highly branching, or richly grounded trajectories where additional structure could help.
- **Mechanistic evidence is supportive but not uniquely identifying.** Our ablations and interventions are consistent with state-conditioned routing playing a central role, yet—as in most systems that couple representation, memory, and generation—alternative narratives cannot be ruled out in every corner case without heavier instrumentation.
- **Empirical coverage matches the paper's claims, not every deployment regime.** We report results across a broad multi-task suite, but stronger proprietary models, different tool-use/agent stacks, and safety-critical deployments are outside our evaluated scope.
- **Limited-access variants are informative but not exhaustive.** Training-free SoT, embedding-based features, and SoT-JUDGE probe the same interface under restricted supervision or observability; they are evaluated at a smaller footprint than the full supervised controller and are best read as exploratory complements.

S.2 Future work

- **Richer but still compact states.** A natural direction is to add state dimensions or structured summaries only when they measurably improve control (e.g., branching, retrieval confidence, multimodal grounding), without bloating the feedback loop.
- **Tighter identification of what the controller uses.** Additional counterfactual replay, routing interventions, and closed-loop stress tests could further clarify which components of the interface matter most across task families.
- **Adaptive memory units.** Sentence-level units are easy to audit, but variable segmentation or hybrid symbolic/latent memories may better match heterogeneous reasoning traces.
- **Unified white-box and black-box trajectory control.** SoT-Embed and SoT-JUDGE suggest a continuum between internals-aware controllers and trajectory-only signals; a more systematic comparison of that continuum is an interesting next step.
- **Broader multimodal and interactive settings.** Environments with dynamic evidence—long multimodal contexts, tool use, and interactive decisions—are a natural fit for state-conditioned organization, but will require task suites and training protocols beyond this paper.

Final perspective. We intend SoT as a precise formulation of reasoning-time control—not a claim that any single architecture is final. The underlying point is empirical and conceptual: reasoning benefits from treating context activation and continuation as a feedback process coupled to the model's evolving internal regime, rather than only as prompt programming over a growing transcript.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contribution (endogenous state-conditioned reasoning) and summarize both quality and efficiency gains, which are supported by the experimental and analysis sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are explicitly discussed in Appendix Section S, including scope constraints, potential confounds, and open directions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper includes explicit theoretical assumptions and formal statements, with detailed proofs and supporting derivations provided in the appendix theory sections.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide model settings, dataset organization, metrics, baseline definitions, evaluation protocol, and hyperparameter tables in the main text and appendix to support reproducibility of the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, controller checkpoints, final result records, and end-to-end reproduction instructions are publicly available at `https://github.com/GongZhiren/State-of-Thought`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and inference details, data splits, metrics, and method-specific hyperparameters are documented in the main setup section and Appendix Sections K, L, and N.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report explicit error bars or formal significance tests; however, the conclusions are supported by large-scale and consistent trends across many datasets, model backbones, and baseline families. In addition, many settings use deterministic decoding (temperature $= 0$; see the hyperparameter section), which reduces run-to-run randomness in reported outputs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix L specifies H200 GPUs for text-backbone timing and one dedicated 80GB A100 for VLM timing; the main and appendix efficiency tables report end-to-end latency and generated-token costs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the study follows the NeurIPS Code of Ethics and involves no human-subject intervention, sensitive personal data collection, or deceptive experimental practice.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both potential positive value and potential misuse risks, together with mitigation directions, in Appendix Section S.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release new high-risk models or datasets as part of this submission, so release-time safeguard mechanisms are not applicable at submission.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: Existing assets are cited, but explicit license names and terms of use for each dataset/model are not yet fully enumerated in the manuscript.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code and lightweight controller checkpoints are documented in the public repository, including installation, quick-start, checkpoint use, training, evaluation, result provenance, and licensing information.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or experiments with human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human-subject research was conducted, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs/VLMs are core experimental components and are explicitly described in the setup and protocol sections, including model families, evaluation settings, and usage context.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.