Neural Chain-of-Thought Search: Searching the Optimal Reasoning Path to Enhance Large Language Models

Guoming Ling[1], **Zhongzhan Huang**[1], **Yupei Lin**[1], **Junxin Li**[1,2], **Shanshan Zhong**[1], **Hefeng Wu**[1], **Liang Lin**[*1]
[1]Sun Yat-sen University, [2]South China Normal University

Abstract

Chain-of-Thought reasoning has significantly enhanced the problem-solving capabilities of Large Language Models. Unfortunately, current models generate reasoning steps sequentially without foresight, often becoming trapped in suboptimal reasoning paths with redundant steps. In contrast, we introduce Neural Chain-of-Thought Search (NCoTS), a framework that reformulates reasoning as a dynamic search for the optimal thinking strategy. By quantitatively characterizing the solution space, we reveal the existence of sparse superior reasoning paths that are simultaneously more accurate and concise than standard outputs. Our method actively navigates towards these paths by evaluating candidate reasoning operators using a dual-factor heuristic that optimizes for both correctness and computational cost. Consequently, NCoTS achieves a Pareto improvement across diverse reasoning benchmarks, boosting accuracy by over 3.5% while reducing generation length by over 22%. Our code and data are available on Github.

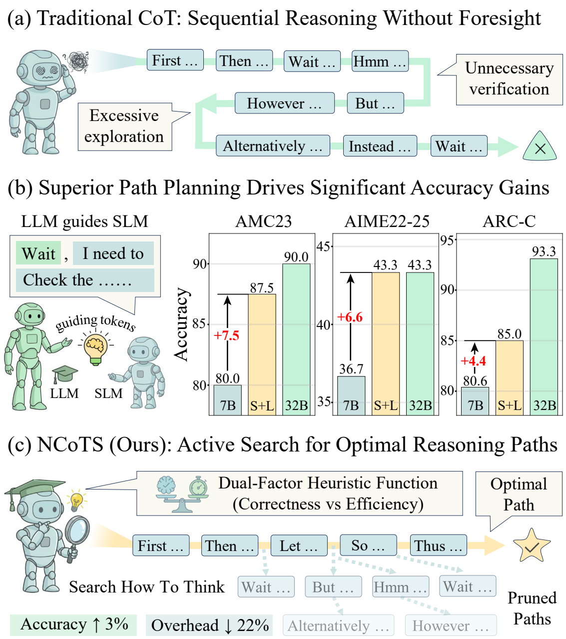

Figure 1: Motivation and Overview of our NCoTS. (a) Planning Bottleneck in Traditional CoT. (b) Importance of Path Planning. Sparse guiding tokens from a strong teacher significantly boost performance, confirming that path planning is the key bottleneck. (c) The NCoTS framework. Our method reformulates reasoning as a search process, employing a dual-factor heuristic to actively discover paths that are both accurate and concise.

1 Introduction

Large Language Models (LLMs) have evolved into specialized Large Reasoning Models (LRMs) (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Chen et al., 2025b; Li et al., 2025i; Xu et al., 2025a) that excel at complex tasks through Chain-of-Thought (CoT) reasoning (Wei et al., 2023; Kojima et al., 2022). These models achieved state-of-the-art performance on math, logic, and programming benchmarks (Zhang et al., 2025d; Snell et al., 2025). However, recent research indicates that Large Reasoning Models suffer from a strategic bottleneck at reasoning path planning (Shojaee et al., 2025; Liu et al., 2025e; Sui et al., 2025; An et al., 2025b; Jiang et al., 2025a). They frequently fail to foresee the optimal reasoning direction, causing them to drift into inefficient patterns (Kang et al., 2025). For instance, they may frequently output reflective tokens like "Wait" or "Hmm", triggering unnecessary verification steps or getting stuck in excessive branch exploration(Wang et al., 2025a; Jiang et al., 2025a; Yang et al., 2025b). This behavior suggests a lack of foresight in navigating the reasoning path.

We investigate this bottleneck through a hybrid guidance experiment (Detailed in Appendix A.1). We employed a larger model to generate only the initial token at each reasoning step for a smaller model. These guiding tokens accounted for only 2.9% of the total output but yielded an average accuracy gain of 6.2% across benchmarks (Fig. 1). This result confirms that the core limitation of rea-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 22900–22933
July 2-7, 2026

soning models lies in their inability to strategically navigate reasoning paths at critical decision points.

Based on these insights, we propose treating reasoning generation as a dynamic search problem. To validate the potential of this paradigm, we quantitatively characterize the reasoning solution space in Section 3.3. This analysis reveals the existence of superior reasoning paths that achieve higher accuracy and lower generation length than standard model outputs. These optimal paths are sparse and difficult to locate via standard sampling, which necessitates a targeted search mechanism to identify them efficiently. To this end, we introduce Neural Chain-of-Thought Search (NCoTS) in Section 2. This framework models reasoning as a search for the optimal sequence of reasoning operators. At each decision point, the model evaluates potential directions using a dual-factor heuristic that estimates both correctness and efficiency. As demonstrated in Section 3, our method actively discovers superior reasoning paths that outperform baselines in both accuracy and efficiency with negligible overhead. We provide a deeper analysis of the proposed framework in Section 4. We show the related works in Appendix C, and summarize the contributions of this paper as follows:

1. We identify the reasoning path planning bottleneck in current reasoning models. Our hybrid guidance experiment reveals that correcting sparse thinking tokens, comprising only 2.9% of the output, yields an average accuracy gain of 6.2%.

2. We provide the first quantitative analysis of the reasoning solution space, confirming the existence of superior paths that simultaneously achieve higher accuracy and reduced generation cost.

3. We propose NCoTS, a framework that actively searches how to think to discover superior reasoning paths. NCoTS consistently achieves the highest efficiency metric across all experimental settings, improving average accuracy by over 3.5% while reducing generation length by over 22%.

2 Method

We propose Neural Chain-of-Thought Search, a framework that reformulates the generative reasoning process as a dynamic search for the optimal reasoning path. To simultaneously maximize performance and minimize reasoning length, our method explicitly navigates the solution space by evaluating *how to think* at critical decision points.

2.1 Preliminary

We formalize Chain-of-Thought reasoning as a sequential decision process. Let x denote the input query. The reasoning chain y consists of a sequence of T discrete steps $y = (s_1, s_2, \ldots, s_T)$. Each step s_t constitutes a complete semantic unit such as a deduction or calculation. Following prior work (Yang et al., 2025d), we mark the completion of a step with a specific delimiter token `"\n\n"`. (See Appendix E.1 for empirical evidence).

We identify the locations of these delimiters as Decision Points. At a given decision point t, the model tends to output a thinking token (Qian et al., 2025) to indicate the logical direction of the subsequent step s_{t+1}. For instance, the model might generate "wait" to initiate reflection or "alternatively" to explore other possibilities. We formulate these thinking tokens as Reasoning Operators o_t drawn from a action space $\mathcal{O}$. We define $\mathcal{O}$ as a finite and small set of thinking tokens which allows for efficient enumeration: $\mathcal{O} = \{$"Wait", "So", "Then", ...$\}$. The sequence of operators $\alpha = (o_1, o_2, \ldots, o_T)$ defines the high-level structure which we term the Reasoning Architecture. Our objective is to find the optimal architecture α^* for a query that maximizes accuracy while minimizing the total sequence length.

2.2 Overview: Search How to Think

Intuition. Existing large reasoning models typically execute reasoning sequentially. Upon completing a step, they immediately generate the subsequent step, often lacking high-level planning. Specifically, the model commits to a specific line of reasoning without evaluating the most effective direction. This lack of foresight may trap models in suboptimal paths, leading to redundant verification loops or verbose derivations.

Proposed Mechanism. To address this, we introduce a mechanism to *search how to think*. Fig. 2 illustrates the overview of our framework, which comprises four phases. (1) Pause Generation: The standard generation halts immediately upon detecting a step delimiter. (2) Lookahead Simulation: The model simulates potential reasoning directions by projecting all candidate operators from the set $\mathcal{O}$ into the future context. (3) Heuristic Evaluation: A dual-factor heuristic function assesses each direction by estimating its success probability and computational cost. (4) Strategic Selection: The model samples the optimal operator based on these

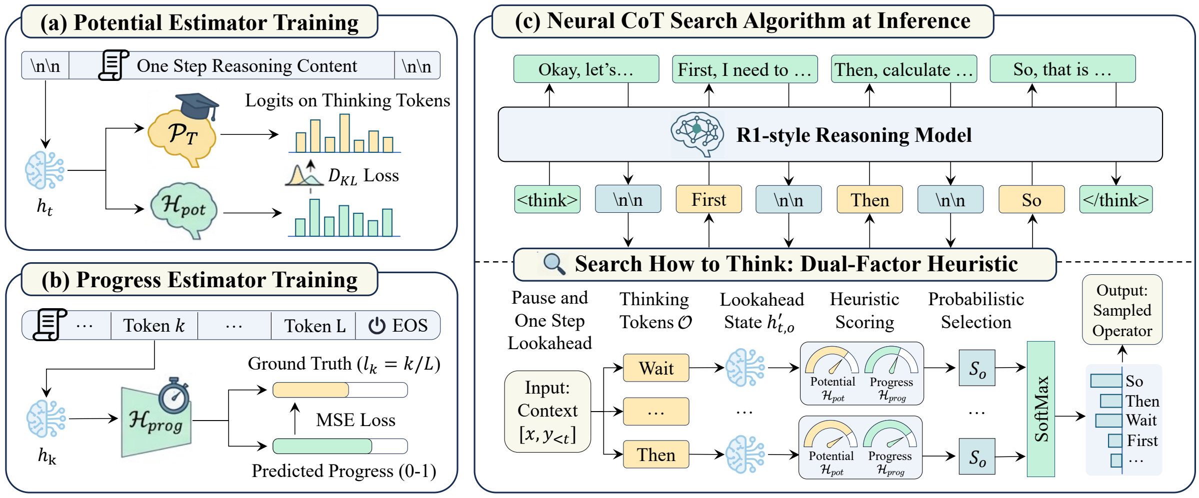

Figure 2: Overview of the Neural Chain-of-Thought Search (NCoTS) Framework. (a) The Path Potential Estimator employs policy distillation from a teacher model to capture high level planning capabilities. (b) The Reasoning Progress Estimator learns to predict the normalized solution progress via token level dense supervision. (c) The search algorithm during inference. The model pauses at decision points to *search how to think*. It performs a one step lookahead and evaluates candidate thinking tokens using a dual-factor heuristic function.

estimates and resumes generation. This active decision process prunes inefficient branches before they consume computational resources.

2.3 Dual-Factor Heuristic Function

We employ a composite heuristic function $\mathcal{H}(h_t, o)$ to evaluate the efficacy of applying operator o at the current hidden state h_t. This function comprises two specialized estimators designed to quantify the quality and efficiency of the reasoning path.

Path Potential Estimator. The first component is the Path Potential Estimator $\mathcal{H}_{\text{pot}}$. It predicts the probability that a specific reasoning direction will lead to a correct solution. We implement this estimator as a linear projection layer taking the final hidden state as input to output logits over the operator set $\mathcal{O}$. As demonstrated in Section 1, Larger Models possess stronger capabilities in high-level planning. Therefore, we train this estimator via policy distillation from a fixed Teacher LRM. We treat the teacher's probability distribution over $\mathcal{O}$ as the expert policy P_T. The estimator is optimized by minimizing the Kullback-Leibler divergence:

$$\mathcal{L}_{\text{pot}} = \mathbb{E}_{h_t \sim \mathcal{D}} \left[D_{\text{KL}} \Big(P_T(h_t) \parallel \mathcal{H}_{\text{pot}}(h_t) \Big) \right]. \quad (1)$$

This estimator effectively transfers the strategic planning capabilities of the teacher into the search process, serving as the compass for correctness.

Reasoning Progress Estimator. The second component is the Reasoning Progress Estimator $\mathcal{H}_{\text{prog}}$. It estimates the efficiency of a reasoning path. We implement this estimator as a linear regression head that maps the hidden state to a scalar value representing normalized progress. Similar to recent works on reasoning monitoring (Eisenstadt et al., 2025), this estimator predicts the completion ratio of the solution given the current state. We train this estimator on a token-level dense supervision task. For each training query, we collect multiple complete reasoning paths. Specifically, for every token at index k within a completed path of total length L, we construct a training pair (h_k, l_k). Here, h_k denotes the hidden state and $l_k = k/L$ represents the ground truth normalized progress, indicating the portion of the solution completed. The estimator $\mathcal{H}_{\text{prog}}$ projects h_k to a scalar value, trained by minimizing the Mean Squared Error:

$$\mathcal{L}_{\text{prog}} = \mathbb{E}_{(h_k, l_k) \sim \mathcal{D}} \left[\left\| \mathcal{H}_{\text{prog}}(h_k) - l_k \right\|^2 \right]. \quad (2)$$

By maximizing this estimated progress, the search algorithm favors operators that significantly advance the reasoning state toward the solution, effectively penalizing verbose or circular steps.

2.4 Search Algorithm

We optimize the reasoning path by actively *searching how to think* during inference. This strategy evaluates potential reasoning directions at decision points to identify the optimal path.

One Step Lookahead. At decision point t, marked by the delimiter "\n\n", we proactively explore the potential future space. Let $y_{<t}$ denote

the current reasoning path. For each candidate operator $o \in \mathcal{O}$, we simulate the next step by appending o to the KV cache of model $\mathcal{M}$:

$$\mathbf{h}'_{t,o} = \mathcal{M}\big([x, y_{<t}, o]\big), \quad \forall o \in \mathcal{O}. \tag{3}$$

This yields the lookahead hidden state $\mathbf{h}'_{t,o}$. Given that the thinking token governs the thinking mode as detailed in Section 4, this lightweight lookahead captures the semantic trajectory of the branch without the overhead of full step generation.

Heuristic Scoring. Once the lookahead states are generated, we assign a composite score $S(o)$ to each branch by aggregating the outputs of the dual-factor heuristics. The score integrates both the potential for accuracy and the efficiency of progress:

$$S(o) = \underbrace{\mathcal{H}_{\text{potential}}(h_t, o)}_{\text{Success Potential}} + \lambda \cdot \underbrace{\mathcal{H}_{\text{progress}}(h'_{t,o})}_{\text{Efficiency Progress}}. \tag{4}$$

Here, λ is a hyperparameter that governs the emphasis on conciseness. A higher λ encourages the model to select more concise reasoning paths.

Probabilistic Selection. To ensure diversity and avoid local optima, we convert these scores into a probabilistic search policy P_{search} using Softmax function with a temperature parameter τ:

$$P_{\text{search}}(o|h_t) = \frac{\exp\left(S(o)/\tau\right)}{\sum_{o' \in \mathcal{O}} \exp\left(S(o')/\tau\right)}. \tag{5}$$

The final operator is selected by sampling $o^* \sim P_{\text{search}}$. This procedure ensures that the selected reasoning direction is both strategically sound and computationally efficient.

3 Experiments

In this section, we empirically validate the proposed framework. We first characterize the reasoning solution space, confirming the existence of superior paths that achieve higher accuracy and lower length than standard generation. We then demonstrate that our method actively locates these paths, consistently achieving the highest efficiency metrics (η) across all experimental settings.

3.1 Experimental Setup

Datasets. We evaluate the performance of our method across four diverse benchmarks. The selected benchmarks include AMC23, ARC-C (Clark et al., 2018), GPQA (Rein et al., 2023), and GSM8K (Cobbe et al., 2021) which collectively cover symbolic deductive reasoning, commonsense reasoning, expert knowledge reasoning and multi-step arithmetic reasoning. We provide details of these benchmarks in Appendix A.3.

Models. To broadly explore the characteristics of the solution space, our analysis in Section 3.3 employs multiple models of varying sizes and architectures: DeepSeek-R1-Distill-Qwen-{1.5B, 7B, 14B, 32B}, and DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI et al., 2025; Qwen et al., 2025; Dubey et al., 2024). For the evaluation of our search method (Section 3.4), we employ two configurations: a small pair, which uses {SLM=Qwen-1.5B, LLM=Qwen-7B}, and a large pair, which uses {SLM=Qwen-7B, LLM=Qwen-32B}.

Baselines. We compare the proposed framework against six baselines. We use Mean and Original to represent the performance of standard sampling. The evaluation also includes recent strategies for optimizing reasoning efficiency such as NoWait (Wang et al., 2025a), AdaptThink (Zhang et al., 2025a), ThinkPrune (Hou et al., 2025) and Laser (Liu et al., 2025c). See Appendix A.4 for details.

Metrics. We report task-specific Accuracy (A) and the average token count (L). To quantify the trade-off between performance gains and computational cost, we adopt a composite Efficiency Metric (η), inspired by previous works on efficient reasoning (An et al., 2025b; Qu et al., 2025a). This metric places a quadratic emphasis on accuracy, as computational savings are secondary to correctness:

$$\eta = \underbrace{\left(\frac{\mathbb{E}_{y\sim\pi^*}[A(y)]}{\mathbb{E}_{y_0\sim\pi}[A(y_0)]}\right)^2}_{\text{Performance Gain}} \cdot \underbrace{\frac{\mathbb{E}_{y_0\sim\pi}[L(y_0)]}{\mathbb{E}_{y\sim\pi^*}[L(y)]}}_{\text{Computational Savings}}. \tag{6}$$

Here, π^* denotes our search-augmented policy and π represents the original model. $A(\cdot)$ measures solution correctness and $L(\cdot)$ denotes sequence length. A value of $\eta > 1$ indicates that the method improves the reasoning density and provides more correct reasoning per unit of computation.

3.2 Implementation Details

Characterization of Solution Space. We conduct a randomized search experiment to characterize the architectural search space $\mathcal{A}$ and empirically map the performance boundaries of the model. For each query, we generate multiple independent reasoning paths by intervening at every step delimiter

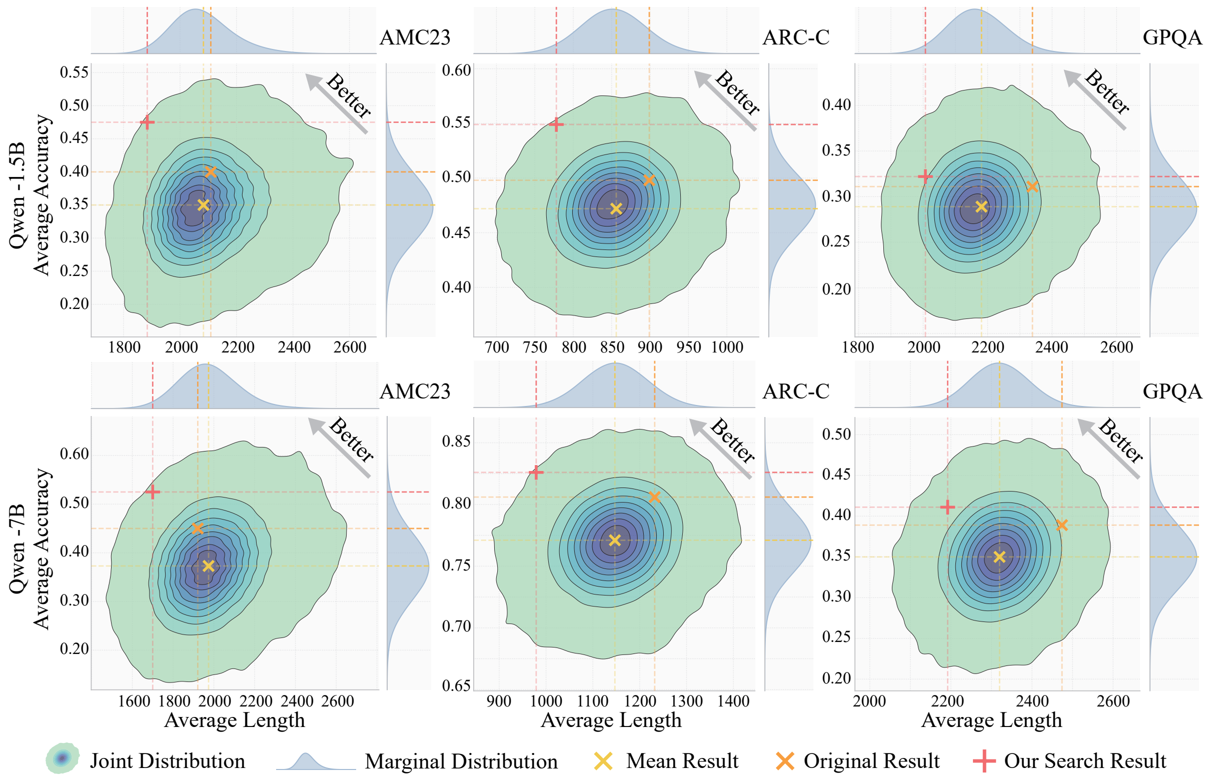

Figure 3: Visualization of the reasoning solution space. The region to the upper-left of the Original result indicates the existence of superior solutions. This confirms that paths with higher accuracy and lower length are attainable, validating the feasibility of our search framework. The red cross mark represents our method, demonstrating that our strategy successfully discovers these superior paths that optimize both accuracy and conciseness.

to sample the reasoning operator o_t from the uniform distribution over $\mathcal{O}$. We then aggregate these paths to construct density heatmaps on the Accuracy versus Length plane. This visualization reveals the distribution of potential search strategies and the theoretical limits of the model. Appendix A.2 provides more details regarding the experimental setup and searching mechanism.

Training. We initialize the potential estimator using weights from the pre-trained language head of the student model. Specifically, we extract the rows of its embedding matrix corresponding to the thinking tokens in $\mathcal{O}$ to preserve the model's initial semantic priors. The progress estimator is initialized randomly. Training is performed on a composite dataset comprising LogicQA (Liu et al., 2020), Math500 (Hendrycks et al., 2021), AIME22-25 (Balunović et al., 2025), and HumanEval (Chen et al., 2021). For the potential estimator, we employ a distillation objective. Given a query, the model generates steps until a decision point is reached. We then compute logits for the thinking tokens using the fixed Teacher LLM and minimize the KL divergence between the Teacher's distribution and the estimator's output. The progress estimator is trained via Mean Squared Error to predict the complete ratio of solution, as detailed in Section 2.3.

Testing. For all methods, we set the temperature to 0.6, top-p to 0.95, and the global maximum token limit to 4096. For our search method, we impose a maximum limit of 50 reasoning steps. During inference, we set the balancing hyperparameter $\lambda = 1$ and sample the reasoning operators based on the composite score $S(o)$, following the search policy detailed in Section 2.4. The prompt used is: `Please reason step by step, and put your final answer within \boxed{}`, following previous works (Chen et al., 2025d; Yang et al., 2025d; Cheng et al., 2025a).

3.3 The Reasoning Solution Space

Fig. 3 presents the density heatmaps of Average Length versus Average Accuracy derived from our random search characterization (See Appendix B for more results). This visualization reveals four insights into the nature of CoT reasoning:

(1) Operator Choice Drives High Variance. The reasoning path is highly sensitive to the choice of reasoning operators. Selecting different operators leads to vastly different outcomes in both accu-

Method	AMC23			ARC-C			GPQA			GSM8K			Average		
	Acc↑	Length↓	η↑	Acc↑	Length↓	η↑	Acc↑	Length↓	η↑	Acc↑	Length↓	η↑	ΔAcc↑	ΔLength↓	η↑
DeepSeek-R1-Distill-Qwen-1.5B															
Mean	35.0	2083	0.775	47.2	856	0.943	28.9	2181	0.926	80.9	1846	0.997	-3.0	-4.4%	0.910
Original	40.0	2109	1.000	49.8	899	1.000	31.1	2339	1.000	83.0	1938	1.000	+0.0	+0.0%	1.000
NoWait	40.0	1967	1.072	50.8	812	1.152	28.9	1992	1.014	84.1	1211	1.641	+0.0	-17.2%	1.220
AdaptThink	42.5	1926	1.236	50.5	897	1.031	32.2	2429	1.032	85.4	1109	1.848	+1.7	-12.0%	1.287
ThinkPrune	45.0	1803	1.480	51.5	768	1.252	31.1	1900	1.231	84.9	1191	1.702	+2.1	-21.6%	1.416
Laser	40.0	1902	1.109	50.8	870	1.075	27.8	2121	0.881	82.3	1064	1.792	-0.7	-16.9%	1.214
Ours	47.5	1884	**1.578**	54.9	778	**1.405**	32.2	2007	**1.249**	85.4	954	**2.148**	**+4.0**	**-22.3%**	**1.595**
DeepSeek-R1-Distill-Qwen-7B															
Mean	37.3	1974	0.669	77.1	1147	0.983	35.0	2320	0.864	88.1	1649	0.976	-4.3	-3.2%	0.873
Original	45.0	1921	1.000	80.6	1232	1.000	38.9	2474	1.000	90.3	1690	1.000	+0.0	+0.0%	1.000
NoWait	50.0	1894	1.252	80.3	1082	1.130	38.9	2248	1.101	91.3	1147	1.506	+1.4	-13.7%	1.247
AdaptThink	47.5	1910	1.121	82.9	1088	1.198	42.2	2393	1.218	92.6	1086	1.635	+2.6	-12.8%	1.293
Laser	50.0	1650	1.437	79.9	944	1.283	38.9	2279	1.086	93.0	968	1.852	+1.8	-22.0%	1.414
Ours	52.5	1700	**1.538**	82.6	979	**1.322**	41.1	2192	**1.261**	92.6	899	**1.976**	**+3.5**	**-22.6%**	**1.524**

Table 1: Main results comparing the proposed Neural CoT Search against baselines on AMC23, ARC-C, GPQA, and GSM8K benchmarks. The table reports task-specific Accuracy (Acc), Average Generation Length (Length), and the Efficiency Metric (η). Our method consistently achieves the highest η across all settings, demonstrating simultaneous improvements in accuracy and efficiency. Best results are highlighted in bold.

racy and length. This structural divergence confirms that the high-level planning of the reasoning path is a critical determinant of the final solution quality. (2) Suboptimality of Standard Decoding. The Original baseline consistently outperforms the random Mean baseline but remains far from the theoretical performance boundary. This gap suggests that the model's standard generation strategy fails to exploit the full intrinsic potential of the model. (3) Existence of Superior Paths. The heatmaps reveal a region in the upper-left quadrant with higher accuracy and lower length than the original baseline. These Pareto-superior solutions are empirical proof that it is feasible to simultaneously optimize correctness and cost. (4) Sparsity of Superior Solutions. The region containing these superior paths is extremely sparse compared to the dense clusters of suboptimal paths. This sparsity explains why standard sampling fails to yield consistent improvements. The probability of randomly encountering a superior path is negligible, necessitating a targeted search approach.

3.4 Efficacy of the Proposed Search Strategy

Table 1 compares our method against baselines across DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B. While many existing baselines struggle to balance the trade-off between performance and cost, our method simultaneously enhances accuracy and reduces computational cost. On the 1.5B model, we achieve a 4.0% accuracy gain and a 22.3% reduction in token usage. On the 7B model, NCoTS improves average accuracy by 3.5% and decreases generation length by 22.6%. Notably, on GSM8K with the 1.5B model, our approach reduces the generation length by over 50% while achieving an accuracy gain of 2.4%. Moreover, on GSM8K with the 7B model, accuracy improves by 2.3% with a length reduction of 47%, and on AMC23 accuracy improves substantially by 7.5% with length reduced by 12%. Our method consistently achieves the highest efficiency metric η across all settings, yielding an average η of 1.595 for the 1.5B model and 1.524 for the 7B model. This confirms that our search strategy maximizes reasoning density and effectively prunes redundant steps to deliver more correct reasoning per unit of computation.

Furthermore, we observe a distinct correlation between the nature of the task and the magnitude of efficiency gains. The method excels in reasoning-intensive tasks. On GSM8K and AMC23, it achieves the highest efficiency scores between 1.5 and 2.1, as the search mechanism effectively navigates complex reasoning branches. In hybrid tasks like ARC-C, which require a blend of common sense and reasoning, gains remain substantial with η ranging from 1.3 to 1.4. On knowledge-intensive tasks such as GPQA, efficiency gains are the lowest at approximately 1.2. This is expected behavior, as performance in these domains relies more on factual retrieval than strategic planning, yet the consistent improvement across all benchmarks validates the generalizability of our framework.

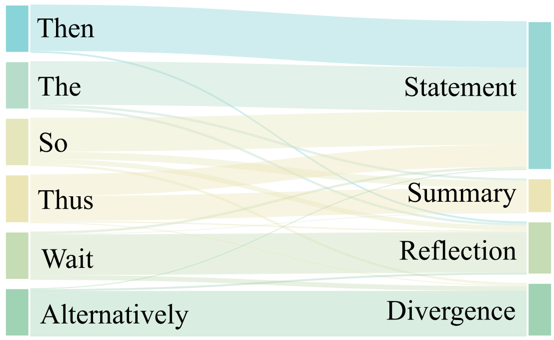

Figure 4: Correlation between thinking tokens and thinking modes. This Sankey diagram illustrates the strong influence of the chosen operator (thinking token) on the functional purpose of the subsequent reasoning step.

4 Further Discussion

In this Section, We conduct a more comprehensive analysis of the proposed search framework. For more analysis, please refer to Appendix D and E.

(1) How does the thinking token affect the corresponding reasoning step?

In section 2, we use thinking tokens from the operator set $\mathcal{O}$ to steer the reasoning direction at each decision point. To illustrate the influence of these tokens, we analyzed a large corpus of reasoning paths generated by DeepSeek-R1-Distill-Qwen-1.5B on the AMC23 benchmark. We extracted each (operator, step) pair and employed DeepSeek-V3 to classify the functional purpose of the step s_i into one of four modes: Statement, Summary, Reflection, or Divergence (see Appendix D for prompt, methodology and more results). As shown in Fig. 4, our analysis reveals a strong correspondence between the chosen operator and the resulting thinking mode. For instance, the "Wait" operator consistently precedes Reflection steps, whereas "Then" strongly correlates with Statement steps.

Psychological studies suggest that human System 2 reasoning involves multiple distinct modes of thinking, such as stating, summarizing, reflecting, and exploring (Evans, 2008; Moshman, 2014). People dynamically switch between them during complex reasoning. We argue that for LLMs to solve complex problems, they also require this ability to dynamically shift their thinking mode. A key insight of our work is that thinking tokens are not just superficial prefixes, they function as a control mechanism to select the thinking mode for next step. Leveraging this insight, our method dynamically guides the model's thinking modes, thereby steering the reasoning path toward a better solution.

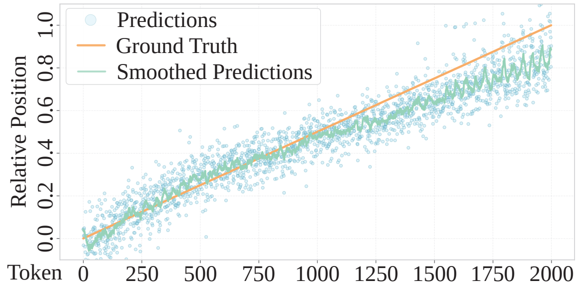

Figure 5: A comparison of the estimated progress against the ground truth progress. The exponentially smoothed estimator output closely aligns with the ground truth progress $y = x/L$.

(2) Does the reasoning progress estimator predict the progress accurately?

We introduce the reasoning progress estimator $\mathcal{H}_{\text{prog}}$ in Section 2.3, grounded in recent evidence that the hidden states of reasoning models implicitly encode the progress of the solution (Eisenstadt et al., 2025). Figure 5 plots the estimator's predictions against ground-truth normalized positions. The exponentially smoothed prediction trajectory aligns well with the true progress, demonstrating that a lightweight regression estimator effectively extracts this signal and estimate the remaining computational cost. The visible variance in the scatter plot reflects semantic sensitivity rather than stochastic noise. As noted in (Eisenstadt et al., 2025), reflective tokens (e.g., "Wait", "Hmm") induce drops in predicted progress, correctly signaling reasoning expansion, while decisive operators (e.g., "Therefore") indicate proximity to the solution. In our search framework, we prioritize the capability to distinguish efficiency over exact progress prediction. The reasoning progress estimator need only preserve the correct preference ordering by assigning higher values to efficient operators (e.g., $v_{\text{“Then”}} > v_{\text{“Wait”}}$). This ensures that the search algorithm correctly prioritizes more efficient branches without necessitating precise estimation of the absolute length.

(3) Is the path potential estimator or the reasoning progress estimator necessary?

To validate our dual-factor heuristic design, we conducted an ablation study by removing the potential and progress estimators respectively. Table 2 reports the results on the English Math Competition subset of OlympiadBench (He et al., 2024). The results demonstrate that each component contributes to the search process in a unique and indispensable way. The configuration without the

OlympiadBench			
Method	Acc ↑	Len ↓	η ↑
Original	51.3	8765	1.000
Ours w/o $\mathcal{H}_{potential}$	48.7	6334	1.247
Ours w/o $\mathcal{H}_{progress}$	53.0	5562	1.682
Ours	52.2	4187	2.167

Table 2: Ablation study of the dual-factor heuristic. The results verify that both potential and progress estimators are essential for balancing correctness and conciseness.

OlympiadBench			
Method	Acc ↑	Len ↓	η ↑
Original	51.3	8765	1.000
AdaptThink	52.2	5267	1.723
Ours	52.2	4187	2.167
Ours + AdaptThink	54.8	3691	2.708

Table 3: Compatibility analysis with AdaptThink. The results demonstrate that our method complements existing strategies to achieve additive efficiency gains.

progress estimator achieves high accuracy but fails to maximize efficiency, as the potential estimator prioritizes correctness without incentives to prune valid but redundant steps. Conversely, removing the potential estimator leads to a collapse in performance. It is worth noting that while this setting reduces length compared to the original baseline, it is less efficient than our full method. This occurs because the progress estimator, lacking semantic guidance, tends to select operators that disrupt the logical flow, causing the model to generate incoherent, compensatory text in an attempt to recover. Therefore, our framework relies on the synergy between the two estimators. The potential estimator leverages distilled strategic priors to identify paths with high success probability. The progress estimator proactively steers the search toward the most compact reasoning paths. This combination steers the search toward reasoning paths that are simultaneously correct and concise.

(4) Is the search paradigm we proposed compatible with other methods?

Our proposed search paradigm is compatible with existing methods. Since our approach operates at the decoding stage by intervening in the selection of reasoning operators, it functions as a plug-and-play module that is orthogonal to model architecture modifications or sample-level routing strategies. To demonstrate this compatibility, we analyze the integration of our method with AdaptThink (Zhang et al., 2025a). AdaptThink represents a class of long/short thinking strategies that dynamically determine the inference budget based on the difficulty of the input query. The synergy is clear: AdaptThink optimizes the macro-level resource allocation (deciding when to reason), while our method optimizes the micro-level reasoning path (steering how to reason). As shown in Table 3, the composite method achieves additive efficiency gains, proving that our search effectively complements budget-adaptive baselines.

(5) How is the cost and latency of the dual-factor heuristic function?

The overhead introduced by our dual-factor heuristic function is negligible in terms of both memory and latency. Regarding parameter efficiency, for the 1.5B model (hidden dimension $d = 1536$), the potential estimator ($d \to |\mathcal{O}|$) and progress estimator ($d \to 1$) collectively introduce approximately 2.6×10^4 parameters. This represents a mere 0.0017% increase, incurring negligible memory overhead. Inference latency is mitigated by sparse activation and parallel lookahead. The search mechanism activates strictly at critical decision points, which comprise only 3% of total tokens, allowing the model to execute standard decoding for the remaining 97%. When activated, candidate branches share an identical prefix, enabling us to compute lookahead steps in a single parallel batch via KV caching. Ultimately, this minor cost is surpassed by efficiency gains; our method reduces average generation length by over 22% across all benchmarks. This substantial decrease in generation length yields a net reduction in aggregate computational operations.

5 Conclusion

In this paper, we introduce NCoTS, a framework that searches for optimal reasoning paths by dynamically steering the thinking modes at decision points. By explicitly optimizing for correctness and conciseness with a dual-factor heuristic, NCoTS achieves a Pareto improvement, boosting accuracy by over 3.5% while reducing generation length by 22%. Our findings demonstrate that the bottleneck of efficient reasoning lies in the myopia of next-token prediction; resolving this requires equipping models with the foresight to plan *how to think*.

Limitations

We propose a search mechanism guided by a defined operator set. However, our current set is primarily optimized for English STEM reasoning and does not account for other languages or creative tasks. Fortunately, the framework allows for straightforward extension to multilingual or creative domains by recalibrating these thinking tokens. Additionally, while our potential estimator relies on teacher supervision which theoretically bounds the planning capability, future works could employ reinforcement learning to enable self-improved exploration beyond the teacher's distribution. Furthermore, our reliance on static newline delimiters effectively captures major pauses but may be too rigid for non-standard formats, suggesting a need for dynamic entropy-based triggers in future works. Moreover, we employ a local lookahead strategy rather than a global search mechanism like MCTS. Although this limits long-horizon planning in extremely complex scenarios, it represents a deliberate trade-off to simultaneously optimize correctness and conciseness, thereby achieving efficiency gains without incurring the heavy computational overhead of exhaustive search.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No. 62325605, 62272494 and 62536010.

This work was partially assisted by AI tools during its development. Specifically, Claude Sonnet 4.5 was used to support code implementation, and Gemini 3.0 Pro was used to assist with writing refinement and language polishing. All scientific contributions, experimental designs, and intellectual content remain solely the work of the authors.

References

Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. 2025. First finish search: Efficient test-time scaling in large language models. *Preprint*, arXiv:2505.18149.

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. Optimalthinkingbench: Evaluating over and underthinking in llms. *arXiv preprint arXiv:2508.13141*.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, and Shuang Zhou. 2025a. Amo-bench: Large language models still struggle in high school math competitions. *Preprint*, arXiv:2510.26768.

Sohyun An, Ruochen Wang, Tianyi Zhou, and Cho-Jui Hsieh. 2025b. Don't think longer, think wisely: Optimizing thinking dynamics for large reasoning models. *Preprint*, arXiv:2505.21765.

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *Preprint*, arXiv:2502.04463.

Seyedarmin Azizi, Erfan Baghaei Potraghloo, and Massoud Pedram. 2025. Activation steering for chain-of-thought compression. *Preprint*, arXiv:2507.04742.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

Dilyara Baymurzina, Eugene Golikov, and Mikhail Burtsev. 2022. A review of neural architecture search. *Neurocomputing*, 474:82–93.

Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. 2018. Understanding and simplifying one-shot architecture search. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 550–559. PMLR.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Aidan O'Mahony, Onur Mutlu, and Torsten Hoefler. 2025. Demystifying chains, trees, and graphs of thoughts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):10967–10989.

Han Cai, Ligeng Zhu, and Song Han. 2019. Proxylessnas: Direct neural architecture search on target task and hardware. *Preprint*, arXiv:1812.00332.

Mert Cemri, Nived Rajaraman, Rishabh Tiwari, Xiaoxuan Liu, Kurt Keutzer, Ion Stoica, Kannan Ramchandran, Ahmad Beirami, and Ziteng Sun. 2025. SPECS: Faster test-time scaling through speculative drafts. *Preprint*, arXiv:2506.15733.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Qiguang Chen, Dengyun Peng, Jinhao Liu, HuiKang Su, Jiannan Guan, Libo Qin, and Wanxiang Che. 2025a. Aware first, think less: Dynamic boundary self-awareness drives extreme reasoning efficiency in large language models. *Preprint*, arXiv:2508.11582.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025b. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Qiguang Chen, Libo Qin, Jiaqi Wang, Jinxuan Zhou, and Wanxiang Che. 2024. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. *Preprint*, arXiv:2410.05695.

Xi Chen and Min Zeng. 2025. Prototype conditioned generative replay for continual learning in nlp. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12754–12770.

Zhuokun Chen, Zeren Chen, Jiahao He, Lu Sheng, Mingkui Tan, Jianfei Cai, and Bohan Zhuang. 2025c. R-stitch: Dynamic trajectory stitching for efficient reasoning. *Preprint*, arXiv:2507.17307.

Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. 2025d. Verithinker: Learning to verify makes reasoning model efficient. *Preprint*, arXiv:2505.17941.

Xiang Cheng, Chengyan Pan, Minjun Zhao, Deyang Li, Fangchao Liu, Xinyu Zhang, Xiao Zhang, and Yong Liu. 2025a. Revisiting chain-of-thought prompting: Zero-shot can be stronger than few-shot. *Preprint*, arXiv:2506.14641.

Xiaoxue Cheng, Junyi Li, Zhenduo Zhang, Xinyu Tang, Wayne Xin Zhao, Xinyu Kong, and Zhiqiang Zhang. 2025b. Incentivizing dual process thinking for efficient large language model reasoning. *Preprint*, arXiv:2505.16315.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. 2025c. Optimizing length compression in large reasoning models. *Preprint*, arXiv:2506.14755.

Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. 2016. Data cleaning: Overview and emerging challenges. *Proceedings of the 2016 International Conference on Management of Data*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

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

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *Preprint*, arXiv:2502.13260.

Muzhi Dai, Shixuan Liu, and Qingyi Si. 2025a. Stable reinforcement learning for efficient reasoning. *Preprint*, arXiv:2505.18086.

Muzhi Dai, Chenxu Yang, and Qingyi Si. 2025b. S-grpo: Early exit via reinforcement learning in reasoning models. *Preprint*, arXiv:2505.07686.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Bowen Ding, Yuhan Chen, Futing Wang, Lingfeng Ming, and Tao Lin. 2025a. Do thinking tokens help or trap? towards more efficient large reasoning model. *Preprint*, arXiv:2506.23840.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. 2025b. Best-route: Adaptive llm routing with test-time optimal compute. *Preprint*, arXiv:2506.22716.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2024. Everything of thoughts: Defying the law of penrose triangle for thought generation. *Preprint*, arXiv:2311.04254.

Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and Dacheng Tao. 2025c. Dynamic parallel tree search for efficient llm reasoning. *Preprint*, arXiv:2502.16235.

Yuanhao Ding, Esteban Garces Arias, Meimingwei Li, Julian Rodemann, Matthias Aßenmacher, Danlu Chen, Gaojuan Fan, Christian Heumann, and Chongsheng Zhang. 2025d. Guard: Glocal uncertainty-aware robust decoding for effective and efficient open-ended text generation. *Preprint*, arXiv:2508.20757.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Razvan-Gabriel Dumitru, Darius Peteleaza, Vikas Yadav, and Liangming Pan. 2025. Conciserl: Conciseness-guided reinforcement learning for efficient reasoning models. *Preprint*, arXiv:2505.17250.

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

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural architecture search: A survey. *Preprint*, arXiv:1808.05377.

Jonathan St BT Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annu. Rev. Psychol.*, 59(1):255–278.

Siqi Fan, Bowen Qin, Peng Han, Shuo Shang, Yequan Wang, and Aixin Sun. 2025. The price of a second thought: On the evaluation of reasoning efficiency in large language models. *Preprint*, arXiv:2505.22017.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: Llm learns when to think. *Preprint*, arXiv:2505.13379.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning. *Preprint*, arXiv:2504.05185.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024. Alphazero-like tree-search can guide large language model decoding and training. *Preprint*, arXiv:2309.17179.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and Hao Zhang. 2025a. Efficiently scaling llm reasoning with certaindex. *Preprint*, arXiv:2412.20993.

Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025b. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Jiaxuan Gao, Shu Yan, Qixin Tan, Lu Yang, Shusheng Xu, Wei Fu, Zhiyu Mei, Kaifeng Lyu, and Yi Wu. 2025. How far are we from optimal reasoning efficiency? *Preprint*, arXiv:2506.07104.

Amirhosein Ghasemabadi, Keith G. Mills, Baochun Li, and Di Niu. 2025. Guided by gut: Efficient test-time scaling with reinforced intrinsic confidence. *Preprint*, arXiv:2505.20325.

Olga Golovneva, Sean O'Brien, Ramakanth Pasunuru, Tianlu Wang, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Pathfinder: Guided search over multi-step reasoning paths. *Preprint*, arXiv:2312.05180.

Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. 2025. Train long, think short: Curriculum learning for efficient reasoning. *Preprint*, arXiv:2508.08940.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. *Preprint*, arXiv:2412.18547.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *Preprint*, arXiv:2305.14992.

Masoud Hashemi, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudhan, Jishnu Sethumadhavan Nair, Aman Tiwari, and Vikas Yadav. 2025. Dnr bench: Benchmarking over-reasoning in reasoning llms. *Preprint*, arXiv:2503.15793.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *Preprint*, arXiv:2402.14008.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.

Xingyang He, Xiao Ling, and Jie Liu. 2025. Smartthinker: Learning to compress and preserve reasoning by step-level length control. *Preprint*, arXiv:2507.04348.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Jialiang Hong, Taihang Zhen, Kai Chen, Jiaheng Liu, Wenpeng Zhu, Jing Huo, Yang Gao, Depeng Wang, Haitao Wan, Xi Yang, Boyan Wang, and Fanyu Meng. 2025. Reconsidering overthinking: Penalizing internal and external redundancy in cot reasoning. *Preprint*, arXiv:2508.02178.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *Preprint*, arXiv:2504.01296.

Pengbo Hu, Ji Qi, Xingyu Li, Hong Li, Xinqi Wang, Bing Quan, Ruiyu Wang, and Yi Zhou. 2023. Tree-of-mixed-thought: Combining fast and slow thinking for multi-hop visual reasoning. *Preprint*, arXiv:2308.09658.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2025a. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *Preprint*, arXiv:2508.05337.

Junquan Huang, Haotian Wu, Yubo Gao, Yibo Yan, Junyan Zhang, Yonghua Hei, Song Dai, Jie Zhang, Puay Siew Tan, and Xuming Hu. 2025b. Effireason-bench: A unified benchmark for evaluating and advancing efficient reasoning in large language models. *arXiv preprint arXiv:2511.10201*.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025c. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*.

Zhongzhan Huang, Guoming Ling, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025d. Minilong-bench: The low-cost long context understanding benchmark for large language models. *arXiv preprint arXiv:2505.19959*.

Zhongzhan Huang, Wenqi Shao, Xinjiang Wang, Liang Lin, and Ping Luo. 2021. Rethinking the pruning criteria for convolutional neural network. *Advances in Neural Information Processing Systems*, 34:16305–16318.

Zhongzhan Huang, Pan Zhou, Shuicheng Yan, and Liang Lin. 2023. Scalelong: Towards more stable training of diffusion model via scaling network long skip connection. *Advances in Neural Information Processing Systems*, 36:70376–70401.

Joonwon Jang, Jaehee Kim, Wonbin Kweon, Seonghyeon Lee, and Hwanjo Yu. 2025. Verbosity-aware rationale reduction: Effective reduction of redundant rationale via principled criteria. *Preprint*, arXiv:2412.21006.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Wei Bi, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. 2025a. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6501–6525.

Guochao Jiang, Guofeng Quan, Zepeng Ding, Ziqin Luo, Dixuan Wang, and Zheng Hu. 2025b. Flash-think: An early exit method for efficient reasoning. *Preprint*, arXiv:2505.13949.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025c. Think only when you need with large hybrid-reasoning models. *Preprint*, arXiv:2505.14631.

Yuxuan Jiang, Dawei Li, and Frank Ferraro. 2025d. Drp: Distilled reasoning pruning with skill-aware step decomposition for efficient large reasoning models. *Preprint*, arXiv:2505.13975.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. *Preprint*, arXiv:2401.04925.

Zhensheng Jin, Xinze Li, Yifan Ji, Chunyi Peng, Zhenghao Liu, Qi Shi, Yukun Yan, Shuo Wang, Furong Peng, and Ge Yu. 2025. Recut: Balancing reasoning length and accuracy in llms via stepwise trails and preference optimization. *Preprint*, arXiv:2506.10822.

Liwei Kang, Yue Deng, Yao Xiao, Zhanfeng Mo, Wee Sun Lee, and Lidong Bing. 2025. First try matters: Revisiting the role of reflection in reasoning models. *arXiv preprint arXiv:2510.08308*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2024. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *Preprint*, arXiv:2412.11664.

Henrik Klagges, Robert Dahlke, Fabian Klemm, Benjamin Merkel, Daniel Klingmann, David A. Reiss, and Dan Zecha. 2025. Assembly of experts: Linear-time construction of the chimera llm variants with emergent and adaptable behaviors. *Preprint*, arXiv:2506.14794.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2025. Tree search for language model agents. *Preprint*, arXiv:2407.01476.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhenglun Kong, Yize Li, Fanhu Zeng, Lei Xin, Shvat Messica, Xue Lin, Pu Zhao, Manolis Kellis, Hao Tang, and Marinka Zitnik. 2026. Token reduction should go beyond efficiency in generative models – from vision, language to multimodality. *Preprint*, arXiv:2505.18227.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025a. How well do llms compress their own chain-of-thought? a token complexity approach. *Preprint*, arXiv:2503.01141.

Byeongchan Lee, Jonghoon Lee, Dongyoung Kim, Jaehyung Kim, Kyungjoon Park, Dongjun Lee, and Jinwoo Shin. 2025b. Efficient llm collaboration via planning. *Preprint*, arXiv:2506.11578.

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. 2025a. Steering llm thinking with budget guidance. *Preprint*, arXiv:2506.13752.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Preprint*, arXiv:1603.06560.

Peiji Li, Kai Lv, Yunfan Shao, Yichuan Ma, Linyang Li, Xiaoqing Zheng, Xipeng Qiu, and Qipeng Guo. 2025b. Fastmcts: A simple sampling strategy for data synthesis. *Preprint*, arXiv:2502.11476.

Ruosen Li, Ziming Luo, Quan Zhang, Ruochen Li, Ben Zhou, Ali Payani, and Xinya Du. 2025c. Aalc: Large language model efficient reasoning via adaptive accuracy-length control. *Preprint*, arXiv:2506.20160.

Xinzhe Li. 2025. A survey on llm test-time compute via search: Tasks, llm profiling, search algorithms, and relevant frameworks. *Preprint*, arXiv:2501.10069.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *Preprint*, arXiv:2401.10480.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025d. Small models struggle to learn from strong reasoners. *Preprint*, arXiv:2502.12143.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2025e. Compressing chain-of-thought in llms via step entropy. *Preprint*, arXiv:2508.03346.

Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. 2025f. Selfbudgeter: Adaptive token allocation for efficient llm reasoning. *Preprint*, arXiv:2505.11274.

Zhiyuan Li, Yi Chang, and Yuan Wu. 2025g. Thinkbench: Evaluating thinking efficiency and chain-of-thought quality of large reasoning models. *arXiv preprint arXiv:2505.22113*.

Zhong-Zhi Li, Xiao Liang, Zihao Tang, Lei Ji, Peijie Wang, Haotian Xu, Xing W, Haizhen Huang, Weiwei Deng, Yeyun Gong, Zhijiang Guo, Xiao Liu, Fei Yin, and Cheng-Lin Liu. 2025h. Tl;dr: Too long, do reweighting for efficient llm reasoning compression. *Preprint*, arXiv:2506.02678.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025i. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Baohao Liao, Hanze Dong, Yuhui Xu, Doyen Sahoo, Christof Monz, Junnan Li, and Caiming Xiong. 2025a. Fractured chain-of-thought reasoning. *Preprint*, arXiv:2505.12992.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025b. Reward-guided speculative decoding for efficient llm reasoning. *Preprint*, arXiv:2501.19324.

Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. 2025a. Plan and budget: Effective and efficient test-time scaling on large language model reasoning. *Preprint*, arXiv:2505.16122.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E. Gonzalez. 2025b. Sleep-time compute: Beyond inference scaling at test-time. *Preprint*, arXiv:2504.13171.

Weizhe Lin, Xing Li, Zhiyuan Yang, Xiaojin Fu, Hui-Ling Zhen, Yaoyuan Wang, Xianzhi Yu, Wulong Liu, Xiaosong Li, and Mingxuan Yuan. 2025c. Trimr: Verifier-based training-free thinking compression for efficient test-time scaling. *Preprint*, arXiv:2505.17155.

Yupei Lin, Xiaoyu Xian, Yukai Shi, and Liang Lin. 2024. Mirrordiffusion: Stabilizing diffusion process in zero-shot image translation by prompts redescription and beyond. *IEEE Signal Processing Letters*, 31:306–310.

George Ling, Shanshan Zhong, and Richard Huang. 2026. Agent skills: A data-driven analysis of claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*.

Zehui Ling, Deshu Chen, Hongwei Zhang, Yifeng Jiao, Xin Guo, and Yuan Cheng. 2025. Fast on the easy, deep on the hard: Efficient reasoning via powered length penalty. *Preprint*, arXiv:2506.10446.

Hanbing Liu, Lang Cao, Yuanyi Ren, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. 2025a. Bingo: Boosting efficient reasoning of llms via dynamic and significance-based reinforcement learning. *Preprint*, arXiv:2506.08125.

Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. 2018. Hierarchical representations for efficient architecture search. *Preprint*, arXiv:1711.00436.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. Darts: Differentiable architecture search. *Preprint*, arXiv:1806.09055.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *Preprint*, arXiv:2007.08124.

Junlin Liu, Shengnan An, Shuang Zhou, Dan Ma, Shixiong Luo, Ying Xie, Yuan Zhang, Wenling Yuan, Yifan Zhou, Xiaoyu Li, Ziwen Wang, Xuezhi Cao, and Xunliang Cai. 2026. General365: Benchmarking general reasoning in large language models across diverse and challenging tasks. *Preprint*, arXiv:2604.11778.

Kaiyuan Liu, Chen Shen, Zhanwei Zhang, Junjie Liu, Xiaosong Yuan, and Jieping ye. 2025b. Efficient reasoning through suppression of self-affirmation reflections in large reasoning models. *Preprint*, arXiv:2506.12353.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *Preprint*, arXiv:2411.01855.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025c. Learn to reason efficiently with adaptive length-based reward shaping. *Preprint*, arXiv:2505.15612.

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

Yifan Liu, Wenkuan Zhao, Shanshan Zhong, Jinghui Qin, Mingfu Liang, Zhongzhan Huang, and Wushao Wen. 2025d. Associam: A benchmark for evaluating association thinking while circumventing ambiguity. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5219.

Yue Liu, Jiaying Wu, Yufei He, Ruihan Gong, Jun Xia, Liang Li, Hongcheng Gao, Hongyu Chen, Baolong Bi, Jiaheng Zhang, Zhiqi Huang, Bryan Hooi, Stan Z. Li, and Keqin Li. 2025e. Efficient inference for large reasoning models: A survey. *Preprint*, arXiv:2503.23077.

Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. 2025f. Thought manipulation: External thought can be efficient for large reasoning models. *Preprint*, arXiv:2504.13626.

Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, Chuheng Zhang, Wei Shen, and Zhouhan Lin. 2025g. Adaptivestep: Automatically dividing reasoning step through model confidence. *Preprint*, arXiv:2502.13943.

Jieyi Long. 2023. Large language model guided tree-of-thought. *Preprint*, arXiv:2305.08291.

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

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, Han Wang, and Can Huang. 2025a. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *Preprint*, arXiv:2505.15154.

Ximing Lu, Seungju Han, David Acuna, Hyunwoo Kim, Jaehun Jung, Shrimai Prabhumoye, Niklas Muennighoff, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. 2025b. Retro-search: Exploring untaken paths for deeper and efficient reasoning. *Preprint*, arXiv:2504.04383.

Yahao Lu, Yupei Lin, Han Wu, Xiaoyu Xian, Yukai Shi, and Liang Lin. 2024. Sirst-5k: Exploring massive negatives synthesis with self-supervised learning for robust infrared small target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–11.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Hoang Anh Duy Le, Shaochen Zhong, Hongyi Liu, Jiayi Yuan, Yang Sui, Vladimir Braverman, Vipin Chaudhary, and Xia Hu. 2025a. Autol2s: Auto long-short reasoning for efficient large language models. *Preprint*, arXiv:2505.22662.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025b. Ada-r1: Hybrid-cot via bi-level adaptive reasoning optimization. *Preprint*, arXiv:2504.21659.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025c. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

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

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025b. Cot-valve: Length-compressible chain-of-thought tuning. *Preprint*, arXiv:2502.09601.

Silin Meng, Yiwei Wang, Cheng-Fu Yang, Nanyun Peng, and Kai-Wei Chang. 2025. Llm-a*: Large language model enhanced incremental heuristic search on path planning. *Preprint*, arXiv:2407.02511.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

David Moshman. 2014. *Epistemic cognition and development: The psychology of justification and truth.* Psychology Press.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *Preprint*, arXiv:2502.20122.

Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. Attnpo: Attention-guided process supervision for efficient reasoning. *Preprint*, arXiv:2602.09953.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2024. Skeleton-of-thought: Prompting llms for efficient parallel generation. *Preprint*, arXiv:2307.15337.

Yansong Ning, Wei Li, Jun Fang, Naiqiang Tan, and Hao Liu. 2025. Not all thoughts are generated equal: Efficient llm reasoning via multi-turn reinforcement learning. *Preprint*, arXiv:2505.11827.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Rui Pan, Yinwei Dai, Zhihao Zhang, Gabriele Oliaro, Zhihao Jia, and Ravi Netravali. 2025. Specreason: Fast and accurate inference-time compute via speculative reasoning. *Preprint*, arXiv:2504.07891.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. 2018. Efficient neural architecture search via parameter sharing. *Preprint*, arXiv:1802.03268.

Soham Poddar, Paramita Koley, Janardan Misra, Sanjay Podder, Navveen Balani, Niloy Ganguly, and Saptarshi Ghosh. 2025. Brevity is the soul of sustainability: Characterizing llm response lengths. *Preprint*, arXiv:2506.08686.

Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. 2025. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models. *arXiv preprint arXiv:2504.13367*.

Saif Punjwani and Larry Heck. 2025. Weight-of-thought reasoning: Exploring neural network weights for enhanced llm reasoning. *Preprint*, arXiv:2504.10646.

Penghui Qi, Zichen Liu, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Optimizing anytime reasoning via budget relative policy optimization. *Preprint*, arXiv:2505.13438.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. 2025. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in llm reasoning. *Preprint*, arXiv:2506.02867.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. *Preprint*, arXiv:2505.04881.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. 2025a. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *Preprint*, arXiv:2503.21614.

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. *Preprint*, arXiv:1802.01548.

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

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE.

Kusha Sareen, Morgane M Moss, Alessandro Sordoni, Rishabh Agarwal, and Arian Hosseini. 2025. Putting the value back in rl: Better test-time scaling by unifying llm reasoners with verifiers. *Preprint*, arXiv:2505.04842.

Jianshu She, Zhuohao Li, Zhemin Huang, Qi Li, Peiran Xu, Haonan Li, and Qirong Ho. 2025. Hawkeye:efficient reasoning with model collaboration. *Preprint*, arXiv:2504.00424.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.

Zhenqian Shen, Yongqi Zhang, Lanning Wei, Huan Zhao, and Quanming Yao. 2024. Automated machine learning: From principles to practices. *Preprint*, arXiv:1810.13306.

Yukai Shi, Yupei Lin, Pengxu Wei, Xiaoyu Xian, Tianshui Chen, and Liang Lin. 2024. Diff-mosaic: augmenting realistic representations in infrared small target detection via diffusion prior. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–11.

Zijing Shi, Meng Fang, and Ling Chen. 2025. Monte carlo planning with large language model for text-based game agents. *Preprint*, arXiv:2504.16855.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *Preprint*, arXiv:2506.06941.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. 2025. Sample more to think less: Group filtered policy optimization for concise reasoning. *Preprint*, arXiv:2508.09726.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Preprint*, arXiv:1206.2944.

Jiwon Song, Dongwon Jo, Yulhwa Kim, and Jae-Joon Kim. 2025a. Reasoning path compression: Compressing generation trajectories for efficient llm reasoning. *Preprint*, arXiv:2505.13866.

Mingyang Song and Mao Zheng. 2025. Walk before you run! concise llm reasoning via reinforcement learning. *Preprint*, arXiv:2505.21178.

Woomin Song, Saket Dingliwal, Sai Muralidhar Jayanthi, Bhavana Ganesh, Jinwoo Shin, Aram Galstyan, and Sravan Babu Bodapati. 2025b. Accelerated test-time scaling with model-free speculative sampling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, page 30611–30624. Association for Computational Linguistics.

Gaurav Srivastava, Aafiya Hussain, Sriram Srinivasan, and Xuan Wang. 2025. Llmthinkbench: Towards basic math reasoning and overthinking in large language models. *arXiv e-prints*, pages arXiv–2507.

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

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *Preprint*, arXiv:2410.20290.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. 2019. Mnasnet: Platform-aware neural architecture search for mobile. *Preprint*, arXiv:1807.11626.

Mingxing Tan and Quoc V. Le. 2020. Efficientnet: Rethinking model scaling for convolutional neural networks. *Preprint*, arXiv:1905.11946.

Siao Tang, Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2025. Concisehint: Boosting efficient reasoning via continuous concise hints during generation. *Preprint*, arXiv:2506.18810.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, page 20090–20111. Association for Computational Linguistics.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 77 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *Preprint*, arXiv:2505.10832.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2025. Reasoning aware self-consistency: Leveraging reasoning paths for efficient llm sampling. *Preprint*, arXiv:2408.17017.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024. Q*: Improving multi-step reasoning for llms with deliberative planning. *Preprint*, arXiv:2406.14283.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025a. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. *Preprint*, arXiv:2506.08343.

Jikai Wang, Juntao Li, Jianye Hou, Bowen Yan, Lijun Wu, and Min Zhang. 2025b. Efficient reasoning for llms through speculative chain-of-thought. *Preprint*, arXiv:2504.19095.

Junlin Wang, Shang Zhu, Jon Saad-Falcon, Ben Athiwaratkun, Qingyang Wu, Jue Wang, Shuaiwen Leon Song, Ce Zhang, Bhuwan Dhingra, and James Zou. 2025c. Think deep, think fast: Investigating efficiency of verifier-free inference-time-scaling methods. *Preprint*, arXiv:2504.14047.

Kaiwen Wang, Jin Peng Zhou, Jonathan Chang, Zhaolin Gao, Nathan Kallus, Kianté Brantley, and Wen Sun. 2025d. Value-guided search for efficient chain-of-thought reasoning. *Preprint*, arXiv:2505.17373.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025e. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025f. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. *Preprint*, arXiv:2408.13457.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Yueqi Zhang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025g. Every rollout counts: Optimal resource allocation for efficient test-time scaling. *Preprint*, arXiv:2506.15707.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Yibo Wang, Haotian Luo, Huanjin Yao, Tiansheng Huang, Haiying He, Rui Liu, Naiqiang Tan, Jiaxing Huang, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025h. R1-compress: Long chain-of-thought compression via chunk compression and search. *Preprint*, arXiv:2505.16838.

Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. 2025i. Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding. *Preprint*, arXiv:2503.01422.

Zhihai Wang, Jie Wang, Jilai Pan, Xilin Xia, Huiling Zhen, Mingxuan Yuan, Jianye Hao, and Feng Wu. 2025j. Accelerating large language model reasoning via speculative search. *Preprint*, arXiv:2505.02865.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Yu-Hang Wu, Yu-Jie Xiong, Hao Zhang, Jia-Chen Zhang, and Zheng Zhou. 2025a. Sugar-coated poison: Benign generation unlocks llm jailbreaking. *Preprint*, arXiv:2504.05652.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025b. When more is less: Understanding chain-of-thought length in llms. *Preprint*, arXiv:2502.07266.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *Preprint*, arXiv:2502.12067.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. 2025. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *Preprint*, arXiv:2506.05256.

Yang Xiao, Jiashuo Wang, Ruifeng Yuan, Chunpu Xu, Kaishuai Xu, Wenjie Li, and Pengfei Liu. 2025. Limopro: Reasoning refinement for efficient and effective test-time scaling. *Preprint*, arXiv:2505.19187.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. *Preprint*, arXiv:2305.00633.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025a. Towards large reasoning models: A survey of reinforced reasoning with large language models. *Preprint*, arXiv:2501.09686.

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

Xiaoang Xu, Shuo Wang, Xu Han, Zhenghao Liu, Huijia Wu, Peipei Li, Zhiyuan Liu, Maosong Sun, and Zhaofeng He. 2025c. A*-thought: Efficient reasoning via bidirectional compression for low-resource settings. *Preprint*, arXiv:2505.24550.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. 2025d. Scalable chain of thoughts via elastic reasoning. *Preprint*, arXiv:2505.05315.

Hang Yan, Fangzhi Xu, Rongman Xu, Yifei Li, Jian Zhang, Haoran Luo, Xiaobao Wu, Luu Anh Tuan, Haiteng Zhao, Qika Lin, and Jun Liu. 2025a. Mur: Momentum uncertainty guided reasoning for large language models. *Preprint*, arXiv:2507.14958.

Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. 2025b. Inftythink: Breaking the length limits of long-context reasoning in large language models. *Preprint*, arXiv:2503.06692.

Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. 2025a. Test-time prompt intervention. *Preprint*, arXiv:2508.02511.

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

Junjie Yang, Ke Lin, and Xing Yu. 2025c. Think when you need: Self-adaptive chain-of-thought learning. *Preprint*, arXiv:2504.03234.

Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025d. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *Preprint*, arXiv:2504.12329.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025e. Towards thinking-optimal scaling of test-time compute for llm reasoning. *Preprint*, arXiv:2502.18080.

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

Yao Yao, Zuchao Li, and Hai Zhao. 2024. GoT: Effective graph-of-thought reasoning in language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2901–2921, Mexico City, Mexico. Association for Computational Linguistics.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *Preprint*, arXiv:2502.03373.

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. 2025a. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *Preprint*, arXiv:2505.03469.

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

Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and Mengyue Yang. 2025b. Causal sufficiency and necessity improves chain-of-thought reasoning. *Preprint*, arXiv:2506.09853.

Ye Yu, Yaoning Yu, and Haohan Wang. 2025c. Premise: Scalable and strategic prompt optimization for efficient mathematical reasoning in large models. *Preprint*, arXiv:2506.10716.

Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025d. Z1: Efficient test-time scaling with code. *Preprint*, arXiv:2504.00810.

Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, Hao Ma, and Han Fang. 2025e. Think smarter not harder: Adaptive reasoning with inference aware optimization. *Preprint*, arXiv:2501.17974.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. 2025a. Efficient rl training for reasoning models via length-aware optimization. *Preprint*, arXiv:2505.12284.

Hang Yuan, Bin Yu, Haotian Li, Shijun Yang, Christina Dan Wang, Zhou Yu, Xueyin Xu, Weizhen Qi, and Kai Chen. 2025b. Not all tokens are what you need in thinking. *Preprint*, arXiv:2505.17827.

Haohan Yuan, Sukhwa Hong, and Haopeng Zhang. 2026. Strucsum: Graph-structured reasoning for long document extractive summarization with llms. *Preprint*, arXiv:2505.22950.

Chuhuai Yue, Chengqi Dong, Yinan Gao, Hang He, Jiajun Chai, Guojun Yin, and Wei Lin. 2025. Promoting efficient reasoning with verifiable stepwise reward. *Preprint*, arXiv:2508.10293.

Wenhao Zeng, Yaoning Wang, Chao Hu, Yuling Shi, Chengcheng Wan, Hongyu Zhang, and Xiaodong Gu. 2025. Pruning the unsurprising: Efficient code reasoning via first-token surprisal. *Preprint*, arXiv:2508.05988.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Preprint*, arXiv:2406.03816.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025a. Adaptthink: Reasoning models can learn when to think. *Preprint*, arXiv:2505.13417.

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

Junyu Zhang, Runpei Dong, Han Wang, Xuying Ning, Haoran Geng, Peihao Li, Xialin He, Yutong Bai, Jitendra Malik, Saurabh Gupta, and Huan Zhang. 2025c. Alphaone: Reasoning models thinking slow and fast at test time. *Preprint*, arXiv:2505.24863.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025d. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Shengjia Zhang, Junjie Wu, Jiawei Chen, Changwang Zhang, Xingyu Lou, Wangchunshu Zhou, Sheng Zhou, Can Wang, and Jun Wang. 2025e. Othink-r1: Intrinsic fast/slow thinking mode switching for over-reasoning mitigation. *Preprint*, arXiv:2506.02397.

Wenyuan Zhang, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. 2025f. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*.

Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. 2025g. When to continue thinking: Adaptive thinking mode switching for efficient reasoning. *Preprint*, arXiv:2505.15400.

Xinglang Zhang, Yunyao Zhang, ZeLiang Chen, Junqing Yu, Wei Yang, and Zikai Song. 2026a. Logical phase transitions: Understanding collapse in llm logical reasoning. *Preprint*, arXiv:2601.02902.

Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew Chi-Chih Yao. 2025h. Cumulative reasoning with large language models. *Preprint*, arXiv:2308.04371.

Yunyao Zhang, Xinglang Zhang, Junxi Sheng, Wenbing Li, Junqing Yu, Yi-Ping Phoebe Chen, Wei Yang, and Zikai Song. 2026b. Semantic-aware logical reasoning via a semiotic framework. *Preprint*, arXiv:2509.24765.

Ziying Zhang, Yaqing Wang, and Quanming Yao. 2025i. Searching meta reasoning skeleton to guide llm reasoning. *Preprint*, arXiv:2510.04116.

Kai Zhao, Yanjun Zhao, Jiaming Song, Shien He, Lusheng Zhang, Qiang Zhang, and Tianjiao Li. 2025a. Saber: Switchable and balanced training for efficient llm reasoning. *Preprint*, arXiv:2508.10026.

Shangziqi Zhao, Jiahao Yuan, Guisong Yang, and Usman Naseem. 2025b. Can pruning improve reasoning? revisiting long-cot compression with capability in mind for better reasoning. *Preprint*, arXiv:2505.14582.

Weixiang Zhao, Jiahe Guo, Yang Deng, Xingyu Sui, Yulin Hu, Yanyan Zhao, Wanxiang Che, Bing Qin, Tat-Seng Chua, and Ting Liu. 2025c. Exploring and exploiting the inherent efficiency within large reasoning models for self-guided efficiency enhancement. *Preprint*, arXiv:2506.15647.

Wenkuan Zhao, Shanshan Zhong, Yifan Liu, Wushao Wen, Jinghui Qin, Mingfu Liang, and Zhongzhan Huang. 2025d. Dvib: Towards robust multimodal recommender systems via variational information bottleneck distillation. In *Proceedings of the ACM on Web Conference 2025*, pages 2549–2561.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. *Preprint*, arXiv:2305.14078.

Shanshan Zhong, Zhongzhan Huang, Shanghua Gao, Wushao Wen, Liang Lin, Marinka Zitnik, and Pan Zhou. 2024a. Let's think outside the box: Exploring leap-of-thought in large language models with creative humor generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13246–13257.

Shanshan Zhong, Zhongzhan Huang, Daifeng Li, Wushao Wen, Jinghui Qin, and Liang Lin. 2024b. Mirror gradient: Towards robust multimodal recommender systems via exploring flat local minima. In *Proceedings of the ACM Web Conference 2024*, pages 3700–3711.

Shanshan Zhong, Zhongzhan Huang, Weushao Wen, Jinghui Qin, and Liang Lin. 2023. Sur-adapter: Enhancing text-to-image pre-trained diffusion models with large language models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 567–578.

Shanshan Zhong, Jinghui Qin, Zhongzhan Huang, and Daifeng Li. 2022. Cem: Machine-human chatting handoff via causal-enhance module. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3242–3253.

Huang Zhongzhan, Zhong Shanshan, Zhou Pan, Gao Shanghua, Zitnik Marinka, and Lin Liang. 2025. A causality-aware paradigm for evaluating creativity of multimodal large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language agent tree search unifies reasoning acting and planning in language models. *Preprint*, arXiv:2310.04406.

Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. 2026. Look inward to explore outward: Learning temperature policy from llm internal states via hierarchical rl. *Preprint*, arXiv:2602.13035.

Jiace Zhu, Yuanzhe Huang, Yingtao Shen, Jie Zhao, and An Zou. 2025. Path-consistency with prefix enhancement for efficient inference in llms. *Preprint*, arXiv:2409.01281.

Wei Zhu, Jian Zhang, Lixing Yu, Kun Yue, and Zhiwen Tang. 2026. Dissecting failure dynamics in large language model reasoning. *Preprint*, arXiv:2604.14528.

Ren Zhuang, Ben Wang, and Shuifa Sun. 2025. Accelerating chain-of-thought reasoning: When goal-gradient importance meets dynamic skipping. *Preprint*, arXiv:2505.08392.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. 2023. Toolchain*: Efficient action space navigation in large language models with a* search. *Preprint*, arXiv:2310.13227.

Barret Zoph and Quoc V. Le. 2017. Neural architecture search with reinforcement learning. *Preprint*, arXiv:1611.01578.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2018. Learning transferable architectures for scalable image recognition. *Preprint*, arXiv:1707.07012.

Contents

A Implementation Details

A.1 Details of the hybrid guidance experiment

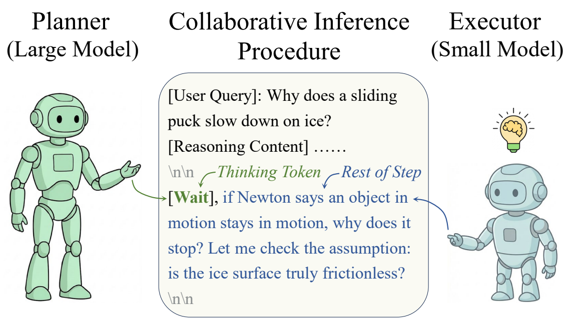

Figure 6: Illustration of the Collaborative Inference Procedure. At each reasoning step delimiter (\n\n), the larger Planner model intervenes to generate a single, strategic Thinking Token (e.g., [Wait]). This token directs the reasoning path, while the smaller Executor model generates the detailed remainder of the step.

To empirically validate the hypothesis that Small Reasoning Models (SRMs) suffer primarily from a lack of high-level planning rather than token-level execution, we designed a Hybrid Guidance framework. This framework decouples strategic planning from detailed execution by employing a collaborative generation process between a stronger Planner model and a weaker Executor model.

A.1.1 Experimental Setup

We utilize DeepSeek-R1-Distill-Qwen-32B as the strategic planner ($\mathcal{M}_{\text{plan}}$) and DeepSeek-R1-Distill-Qwen-7B as the executor ($\mathcal{M}_{\text{exec}}$). The reasoning process is treated as a sequence of discrete steps, delimited by the token sequence `"\n\n"`.

A.1.2 Collaborative Inference Procedure

As shown in Fig.6, the generation follows an iterative handover mechanism. The process begins with the input query x. At the start of each reasoning step (after a `"\n\n"` delimiter, as introduced in Section 2.1), the context is passed to the planner $\mathcal{M}_{\text{plan}}$. The planner is constrained to generate exactly one single token. This token serves as the directional guide (e.g., a logical connective or a reflective marker). Once this guiding token is generated, it is appended to the context. Control is then transferred to the executor $\mathcal{M}_{\text{exec}}$, which generates the remainder of the reasoning step until it predicts the next step delimiter `"\n\n"`. This cycle repeats until the final answer is derived or the maximum token length is reached.

Token	Freq (%)	Token	Freq (%)
So	12.15	I	1.98
Wait	8.75	Hmm	1.53
Let	5.68	Then	1.13
But	4.89	Thus	0.85
Option	4.36	If	0.57
First	3.54	Since	0.53
Therefore	3.27	Given	0.48
Now	3.06	Simplify	0.46
Alternatively	2.82	Alright	0.41
The	2.37	Yes	0.40

Table 4: Frequency Distribution of Guiding Tokens generated by the Planner $\mathcal{M}_{\text{plan}}$. The distribution is heavily dominated by logical connectives (e.g., "So", "Wait"), demonstrating that the Planner provides structural guidance to navigate the reasoning path rather than specific content.

Performance. Despite the minimal intervention, where guiding tokens account for only **2.9%** of the total generated tokens, the hybrid approach yields a substantial performance improvement. As shown in Fig.1), the 7B model achieves an average accuracy gain of **6.2%** across benchmarks when guided by the 32B model's guiding tokens. These results suggest that SRMs possess sufficient capability for granular execution but struggle to independently navigate complex reasoning paths.

A.1.3 Analysis of Guiding Tokens

To examine whether $\mathcal{M}_{\text{plan}}$ injects factual knowledge or structural guidance, we analyzed the frequency distribution of the tokens generated by $\mathcal{M}_{\text{plan}}$ across the test set. As illustrated in Table 4, the vast majority of generated tokens are logical connectives (defined as Thinking Tokens and Reasoning Operators in our method), such as "Wait", "So", and "Alternatively". Content-heavy nouns or entities are rarely generated during this phase. This confirms that the larger model primarily contributes to the reasoning process by steering the logical flow and correcting the reasoning path rather than providing direct factual answers.

A.2 Details of the random search experiment

To investigate the feasibility and potential of our proposed search-based framework, we designed a randomized search experiment. The primary objective of this experiment is to probe the boundaries of the solution space and determine whether there exist superior paths, which are defined as reasoning paths that achieve higher accuracy and lower computational cost than those produced by the model's standard generation policy.

A.2.1 Randomized Intervention Procedure

We utilize a stochastic intervention mechanism to explore diverse reasoning paths. Let $\mathcal{M}$ denote the language model and x the input query. During the generation process, we monitor the stream for the step delimiter token sequence `"\n\n"`, which marks the completion of a reasoning step s_{t-1}. At this decision point, we suspend the standard sampling process. Instead of selecting the subsequent token from the model's predicted distribution, we uniformly sample a reasoning operator o_t from a fixed set $\mathcal{O}$. Specifically, We define this set as $\mathcal{O} = \{$"The", "Thus", "Therefore", "So", "Then", "Let", "Wait", "Alternatively"$\}$. This operator is forced into the context context as the prefix for step s_t. The model $\mathcal{M}$ then resumes generation conditioned on this intervention. This cycle repeats until the model outputs a final answer or reaches a horizon of $T_{max} = 50$ steps.

A.2.2 Construction of the Solution Space

We characterize the solution space through high-volume sampling. For each query x_i in the evaluation dataset of size N, we generate $K = 16$ independent reasoning paths via the intervention procedure described above. We visualize the resulting performance distribution (Fig. 3) using Monte Carlo aggregation. A single data point in the density heatmap corresponds to a coordinate pair $(\bar{L}, \bar{A})$, representing the average length and average accuracy over the full dataset. To generate one such point, we traverse all N queries and randomly select exactly one path from the K available candidates for each query. We then calculate the mean length $\bar{L}$ and mean accuracy $\bar{A}$ for this specific combination of selected paths. By repeating this sampling process for a large number of iterations (e.g., 10^6 times), we obtain a dense distribution of coordinates. This distribution effectively estimates the probability density of the model's performance across the entire feasible solution space. The region in the solution space where paths exhibit both higher accuracy and lower length than the original baseline confirms the existence of the Superior Paths and validates the motivation for our Neural Chain-of-Thought Search.

A.3 Details of the Benchmarks considered

We selected five benchmarks to empirically encompass the spectrum of reasoning capabilities: symbolic deductive reasoning, commonsense reasoning, expert knowledge reasoning, multi-step arithmetic reasoning and Olympiad-level mathematical reasoning. This diversity ensures that our observed efficiency gains are substantive and extend beyond any single problem type.

AMC23. Derived from the 2023 American Mathematics Competitions, this dataset represents a significant step up in difficulty compared to standard arithmetic benchmarks. Unlike grade-school problems, AMC23 requires rigorous multi-step logical deduction and the application of complex mathematical theorems. We use this benchmark to test the model's ability to maintain coherent long-chain reasoning without degenerating into circular logic, a common failure mode in harder deductive tasks.

ARC-C (Clark et al., 2018). The Abstraction and Reasoning Challenge (Challenge Set) evaluates a model's ability to infer abstract rules from few-shot examples. While originally a visual grid-based task, we use the text-encoded version to test the capacity to recognize patterns and generalize to unseen problems. This benchmark is relevant for analyzing thinking tokens, as it demands a search process to hypothesize and verify transformation rules, distinguishing it from pure retrieval tasks.

GPQA (Rein et al., 2023). The Graduate-Level Google-Proof Q&A benchmark consists of difficult multiple-choice questions in biology, physics, and chemistry. Validated by domain experts who hold or are pursuing PhDs, these questions are designed to be resistant to simple web search. We include GPQA to evaluate the "knowledge-intensive" reasoning regime. Here, the efficiency bottleneck is often not the length of the deduction, but the accuracy of the fact retrieval and the avoidance of "hallucinated reasoning," where models generate verbose justifications for incorrect premises.

GSM8K (Cobbe et al., 2021). This widely-used benchmark consists of 8.5k high-quality grade school math word problems that require 2 to 8 steps to solve. While less challenging than AMC23, its arithmetic operations allows us to measure the efficacy of our method in pruning redundant verification steps in well-defined solution spaces.

OlympiadBench (He et al., 2024). As a comprehensive dataset sourced from international Olympiad-level mathematics and physics competitions, this benchmark presents a formidable challenge to current reasoning models. Unlike the routine application of formulas in GSM8K, OlympiadBench demands creative problem-solving strategies and extended logical derivations. We employ it to evaluate the efficacy of our search mechanism in high complexity regimes, specifically testing its capability to navigate the deep reasoning trees required for creative problem solving.

A.4 Details of the Baselines considered

Mean and Original. The Original baseline represents the standard sampling (temperature=0.6, top-p=0.95) from the base model without intervention. The Mean baseline, as described in Section 3.3, represents the expected performance of a random search strategy where operators are sampled uniformly from the set $\mathcal{O}$ at decision points. This comparison isolates the specific contribution of our learned policy network versus a blind search.

NoWait (Wang et al., 2025a). This training-free decoding strategy operates on the hypothesis that reflective tokens often signal hesitation or redundant loops. NoWait explicitly suppresses the generation of self-reflection tokens (e.g., "Wait", "Hmm") during the decoding process. We include this baseline to demonstrate that naive truncation of reasoning paths often degrades accuracy, whereas our method preserves correctness.

AdaptThink (Zhang et al., 2025a). AdaptThink is a reinforcement learning-based approach that focuses on the extensive margin. It trains the model to adaptively select between "thinking" (long-CoT) and "no-thinking" (direct answer) modes based on the estimated difficulty of the input query. Unlike our fine-grained operator search which structures the internal steps of the reasoning chain, AdaptThink makes a binary, high-level decision on whether to engage the reasoning engine at all.

ThinkPrune (Hou et al., 2025). ThinkPrune addresses the efficiency-accuracy trade-off by incorporating a strict token budget into the reward function during RL training. It penalizes generation length linearly or non-linearly to force the model to compress its reasoning. This baseline serves as a direct comparison for our length-penalty reward component, validating whether our Dual-Factor Heuristic Function offers superior control compared to scalar reward shaping alone.

Laser (Liu et al., 2025c). Length-bAsed StEp Reward shaping (LASER) is a technique that optimizes the trade-off between performance and efficiency using adaptive length-based incentives. It

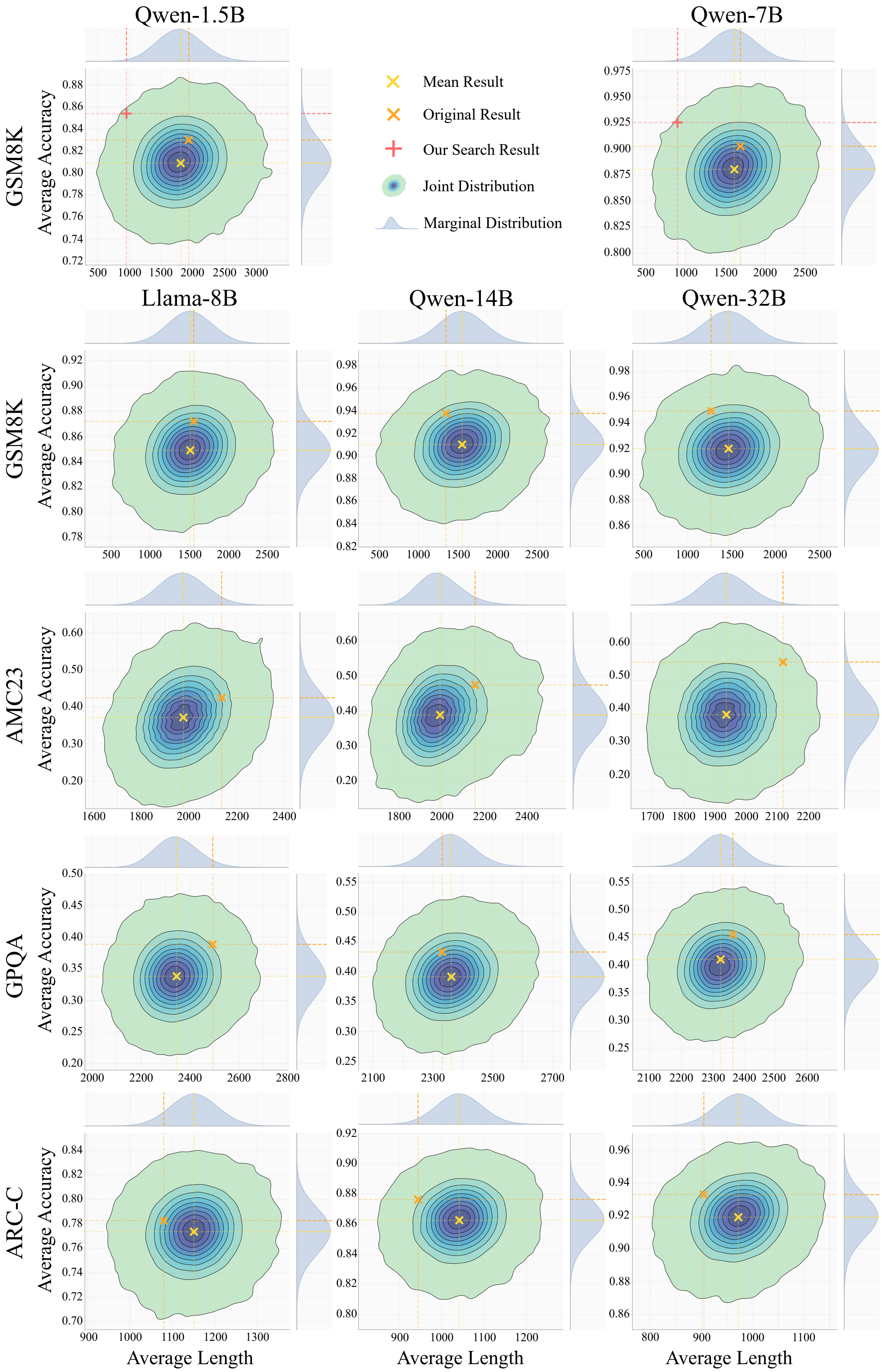

Figure 7: Reasoning solution space visualization across diverse models and benchmarks.

employs a step-function reward scheme that dynamically adjusts the penalty for length based on the current training stage and problem difficulty. We include LASER as a representative of reward-shaping approaches to efficient reasoning.

B Additional Experimental Results

B.1 Visualizations of solution spaces for more models and more benchmarks

Experimental Settings. We examine the generalizability of the solution space characteristics by extending the random search analysis to a broader experimental settings. We conduct comprehensive experiments across five models featuring varying parameter scales and architectures: DeepSeek-R1-Distill-Qwen-{1.5B, 7B, 14B, 32B} and DeepSeek-R1-Distill-Llama-8B. Furthermore, to ensure robustness across different reasoning modalities, we evaluate these models on four distinct benchmarks: AMC23, ARC-C (Clark et al., 2018), GPQA (Rein et al., 2023), and GSM8K (Cobbe et al., 2021). Fig 7 illustrates the density heatmaps for all combinations of models and datasets, generated using the Monte Carlo aggregation method detailed in Appendix A.2.

Consistency of Insights. The comprehensive evaluations consistently demonstrate the four fundamental insights discussed in Section 3.3: (1) The choice of reasoning operators induces high variance in output quality. (2) Standard decoding strategies consistently result in suboptimal paths relative to the potential maximum. (3) Superior paths that are simultaneous more accurate and efficient than the original output exist across all models and tasks. (4) These superior paths are distributed sparsely within the solution space.

Impact of Model Scale. Beyond these confirmations, we observe a inverse correlation between model scale and the density of improved solutions. Comparing heatmaps reveals that the solution space area superior to the original baseline contracts as model size increases. For instance, on the GSM8K benchmark, the density of superior paths for the 1.5B model is 9.12%, whereas this value drops to 1.30% for the 32B model. This phenomenon suggests that larger models possess stronger intrinsic planning capabilities. Their default generation policies align more closely with optimal reasoning paths, narrowing the margin for improvement accessible through random exploration.

C Related Work

C.1 Efficient Reasoning

Efficient reasoning has emerged as a critical research direction to mitigate the computational overhead and overthinking phenomenon (Sui et al., 2025; Kong et al., 2026) observed in Large Reasoning Models (LRMs) like DeepSeek-R1 (DeepSeek-AI et al., 2025) and OpenAI o1 (OpenAI et al., 2024). We categorize existing efficient reasoning approaches into four main paradigms: Reinforcement Learning (RL) with length reward design, Supervised Fine-Tuning (SFT) with variable-length data, dynamic reasoning paradigms during inference, and prompt-guided efficiency.

C.1.1 RL with Length Reward Design

Reinforcement learning has been widely adopted to enhance reasoning capabilities, yet standard accuracy-based rewards often lead to verbose chains of thought. To address this, recent works incorporate length-based penalties directly into the reward function to encourage conciseness without sacrificing performance. Kimi k1.5 (Team et al., 2025) integrates a length penalty into its policy optimization (a variant of online policy mirror descent) to facilitate effective model merging and control long CoT activations. 01-Pruner (Luo et al., 2025c) introduces a Length-Harmonizing Reward combined with a PPO-style loss, optimizing the ratio of CoT lengths between a reference model and the student to shorten reasoning while maintaining accuracy constraints. Similarly, L1 (Aggarwal and Welleck, 2025) modifies training data with length constraints (e.g., "Think for N tokens") before applying policy optimization. Demystifying Long CoT (Yeo et al., 2025) proposes a Cosine Reward based on a Dirichlet function and an exceed length penalty to stabilize performance and control length growth during RL. DAST (Shen et al., 2025) employs SimPO (Meng et al., 2024) with a constructed length-preference dataset based on a token-length budget, while Arora et al. (Arora and Zanette, 2025) utilize length-based rewards conditioned on correctness, assigning higher scores to shorter, correct answers. AttnPO (Nie et al., 2026) further exploits the model's own attention heads to provide process-level supervision, distinguishing essential from redundant reasoning steps without additional overhead. Given the rapid expansion of research in this direction, we summarize other significant contributions in Table 5 and Table 6.

Short Name	Venue	Year
Demystifying Long (Yeo et al., 2025)	ICML	2025
ASRR (Zhang et al., 2025g)	EMNLP	2025
ConciseRL (Dumitru et al., 2025)	EMNLP	2025
AdaptThink (Zhang et al., 2025a)	EMNLP	2025
BRPO (Qi et al., 2025)	NeurIPS	2025
ACPO (Cheng et al., 2025b)	NeurIPS	2025
HGPO (Jiang et al., 2025c)	NeurIPS	2025
S-GRPO (Dai et al., 2025b)	NeurIPS	2025
DeGRPO (Fang et al., 2025)	NeurIPS	2025
REO-RL (Gao et al., 2025)	NeurIPS	2025
LIMOPro (Xiao et al., 2025)	NeurIPS	2025
AutoThink (Tu et al., 2025)	NeurIPS	2025
Arora et al. (Arora and Zanette, 2025)	NeurIPS	2025
80/20 rule (Wang et al., 2025e)	NeurIPS	2025

Table 5: Summary of peer-reviewed Conference Papers addressing Efficient Reasoning through RL with Length Reward Design.

Short Name	Venue	Year
L1 (Aggarwal and Welleck, 2025)	arXiv	2025
MRT (Qu et al., 2025b)	arXiv	2025
DTO (An et al., 2025b)	arXiv	2025
ALP (Xiang et al., 2025)	arXiv	2025
PLP (Ling et al., 2025)	arXiv	2025
FCS (Hong et al., 2025)	arXiv	2025
DAST (Shen et al., 2025)	arXiv	2025
AALC (Li et al., 2025c)	arXiv	2025
SCPO (He et al., 2025)	arXiv	2025
GFPO (Shrivastava et al., 2025)	arXiv	2025
VSRM (Yue et al., 2025)	arXiv	2025
Bingo (Liu et al., 2025a)	arXiv	2025
TL;DR (Li et al., 2025h)	arXiv	2025
LASER (Liu et al., 2025c)	arXiv	2025
LC-RL (Cheng et al., 2025c)	arXiv	2025
SABER (Zhao et al., 2025a)	arXiv	2025
L-GRPO (Song and Zheng, 2025)	arXiv	2025
GRPO-X (Dai et al., 2025a)	arXiv	2025
DuP-PO (Ding et al., 2025a)	arXiv	2025
DR.SAF (Chen et al., 2025a)	arXiv	2025
AdaCoT (Lou et al., 2025)	arXiv	2025
HAWKEYE (She et al., 2025)	arXiv	2025
Short-RL (Yuan et al., 2025a)	arXiv	2025
01-Pruner (Luo et al., 2025c)	arXiv	2025
LongShort (Ning et al., 2025)	arXiv	2025
ThinkPrune (Hou et al., 2025)	arXiv	2025
SelfBudgeter (Li et al., 2025f)	arXiv	2025
Self-adaptive (Yang et al., 2025c)	arXiv	2025
Concise Reasoning (Fatemi et al., 2025)	arXiv	2025
Elastic Reasoning (Xu et al., 2025d)	arXiv	2025
CurriculumGRPO (Hammoud et al., 2025)	arXiv	2025

Table 6: Summary of recent arXiv Preprints addressing Efficient Reasoning through RL with Length Reward Design.

C.1.2 SFT with Variable-Length CoT Data

Fine-tuning LLMs on curated variable-length CoT datasets is another effective strategy to distill efficient reasoning capabilities. These methods generally fall into two categories: post-reasoning compression and during-reasoning compression. In post-reasoning compression, Distilling System 2 into System 1 (Yu et al., 2024) removes the reasoning process entirely to distill direct answer generation. C3oT (Kang et al., 2024) utilizes GPT-4 as a compressor to reduce reasoning length while retaining key information. TokenSkip (Xia et al., 2025) reduces tokens based on semantic importance estimation. In during-reasoning compression, Learn to Skip (Liu et al., 2024) adopts a human-like step-skipping method, first manually creating concise solutions and then training the model to intrinsically skip steps. Token-Budget (Han et al., 2025) employs a binary search to find optimal token budgets and trains the model to follow these constraints. Self-Training (Munkhbat et al., 2025) uses Best-of-N sampling to select the shortest correct reasoning path as training data. CoT-Valve (Ma et al., 2025b) progressively mixes parameters of long-reasoning and non-reasoning models to generate variable-length training data (Zhu et al., 2026). To provide a structured overview of the rapidly evolving landscape, we summarize relevant peer-reviewed conference papers in Table 7 and recent arXiv preprints in Table 8.

C.1.3 Inference Time Dynamic Reasoning

Dynamic reasoning aims to optimize the inference process without extensive retraining, often by selecting efficient reasoning paths or terminating early. Reward-Guided Efficient Reasoning: Speculative Rejection (Sun et al., 2024) optimizes Best-of-N decoding by using a reward model to periodically reject unpromising sequences, reducing computational overhead. RSD (Liao et al., 2025b) employs a Process Reward Model (PRM) to selectively accept high-quality outputs from a draft model. Confidence and Certainty-Based Adaptive Reasoning: DPTS (Ding et al., 2025c) optimizes tree search by dynamically adjusting node expansion based on confidence. FastMCTS (Li et al., 2025b) prioritizes high-confidence traces in an MCTS-inspired framework. Certaindex (Fu et al., 2025a) and Dynasor (Fu et al., 2025b) allocate compute based on a statistical measure of reasoning progress. Length-filtered Vote (Wu et al., 2025b) filters out excessively short or long paths before majority voting. CISC (Taubenfeld et al., 2025) utilizes confidence scores to implement early stopping in sampling. Consistency-Based Reasoning: ST-BoN (Wang et al., 2025i) leverages the consistency of latent embeddings to truncate in-

Short Name	Venue	Year
Stepwise (Cui et al., 2025)	ACL	2025
CoT-Valve (Ma et al., 2025b)	ACL	2025
Token-Budget (Han et al., 2025)	ACL	2025
Self-Training (Munkhbat et al., 2025)	ACL	2025
C3oT (Kang et al., 2024)	AAAI	2025
ReCUT (Jin et al., 2025)	EMNLP	2025
ConCISE (Qiao et al., 2025)	EMNLP	2025
TokenSkip (Xia et al., 2025)	EMNLP	2025
Learn to Skip (Liu et al., 2024)	NeurIPS	2024
PNS (Yu et al., 2025b)	NeurIPS	2025
Ada-R1 (Luo et al., 2025b)	NeurIPS	2025
A-Thought* (Xu et al., 2025c)	NeurIPS	2025
VeriThinker (Chen et al., 2025d)	NeurIPS	2025

Table 7: Summary of peer-reviewed Conference Papers addressing Efficient Reasoning through SFT with Variable-Length CoT Data.

Short Name	Venue	Year
Distilling System 2 (Yu et al., 2024)	arXiv	2024
Z1 (Yu et al., 2025d)	arXiv	2025
DRP (Jiang et al., 2025d)	arXiv	2025
CTS (Yuan et al., 2025b)	arXiv	2025
ASAP (Zeng et al., 2025)	arXiv	2025
AutoL2S (Luo et al., 2025a)	arXiv	2025
Verbosity (Jang et al., 2025)	arXiv	2025
OThink-R1 (Zhang et al., 2025e)	arXiv	2025
R1-Compress (Wang et al., 2025h)	arXiv	2025
StepEntropy (Li et al., 2025e)	arXiv	2025
Prune-on-Logic (Zhao et al., 2025b)	arXiv	2025
LS-Mixture SFT (Yu et al., 2025a)	arXiv	2025
Assembly of Experts (Klagges et al., 2025)	arXiv	2025

Table 8: Summary of recent arXiv Preprints addressing Efficient Reasoning through SFT with Variable-Length CoT Data.

ferior samples early, serving as a proxy for answer correctness. Summarization-Based Reasoning: LightThinker (Zhang et al., 2025b) trains models to compress intermediate thoughts into "gist tokens," while InftyThink (Yan et al., 2025b) iteratively summarizes thoughts to enable unbounded reasoning depth within context limits. Given the rapid proliferation of strategies in this field, we provide a comprehensive summary of other significant contributions, categorized into peer-reviewed conference papers and recent preprints, in Table 9 and Table 10, respectively.

C.1.4 Prompt-Guided Efficient Reasoning

Explicit prompting offers a lightweight mechanism to enforce efficiency. Token-Budget (Han et al., 2025) (TALE-EP) estimates a minimal token requirement and explicitly prompts the model to adhere to it. Chain of Draft (CoD) (Xu et al., 2025b) encourages the model to write only a minimum

Short Name	Venue	Year
DPTS (Ding et al., 2025c)	ACL	2025
CISC (Taubenfeld et al., 2025)	ACL	2025
FastMCTS (Li et al., 2025b)	ACL	2025
ESC (Li et al., 2024)	ICLR	2024
RSD (Liao et al., 2025b)	ICML	2025
SpecSearch (Wang et al., 2025j)	ICML	2025
Best-Route (Ding et al., 2025b)	ICML	2025
AdaptiveStep (Liu et al., 2025g)	ICML	2025
Adaptive Reasoning (Yu et al., 2025e)	ICML	2025
DSC (Wang et al., 2025f)	NAACL	2025
RASC (Wan et al., 2025)	NAACL	2025
STAND (Song et al., 2025b)	EMNLP	2025
AlphaOne (Zhang et al., 2025c)	EMNLP	2025
LightThinker (Zhang et al., 2025b)	EMNLP	2025
Answer Convergence (Liu and Wang, 2025)	EMNLP	2025
Dynasor (Fu et al., 2025b)	ICLR'WS	2025
Speculative Rejection (Sun et al., 2024)	NeurIPS	2024
VGS (Wang et al., 2025d)	NeurIPS	2025
RPC (Song et al., 2025a)	NeurIPS	2025
DORA (Wang et al., 2025g)	NeurIPS	2025
TOPS (Yang et al., 2025e)	NeurIPS	2025
ST-BoN (Wang et al., 2025i)	NeurIPS	2025
ThinkLess (Fang et al., 2025)	NeurIPS	2025
SpecReason (Pan et al., 2025)	NeurIPS	2025

Table 9: Summary of peer-reviewed Conference Papers addressing Efficient Reasoning through Inference Time Dynamic Reasoning. WS denotes workshop.

draft (e.g., limiting steps to 5 words), finding that this preserves accuracy while reducing verbosity. Token Complexity (Lee et al., 2025a) analyzes the trade-off between prompt-based compression and accuracy. Concise CoT (CCoT) (Renze and Guven, 2024) simply prompts models to "be concise," while MARP (Chen et al., 2024) limits single-step computations to refine reasoning boundaries. Other works investigating prompt-based efficiency include Brevity (Poddar et al., 2025), PREMISE (Yu et al., 2025c), GUARD (Ding et al., 2025d) and ConciseHint (Tang et al., 2025).

C.1.5 Related Benchmarks and Evaluations

The development of efficient reasoning methods is supported by parallel works on specialized benchmarks. These benchmarks primarily focus on two interconnected aspects: identifying the *overthinking* pathology in LRMs and establishing standardized metrics for assessing the trade-off between reasoning quality and computational cost. For problem diagnosis, benchmarks like (Hashemi et al., 2025) and (Zhang et al., 2025f) are designed to trigger and measure excessive verbosity on trivial or intuitive tasks, revealing a deep-seated reasoning bias. Similarly, (Srivastava et al., 2025) provides fine-grained analysis of overthinking patterns in

Short Name	Venue	Year
GG (Ghasemabadi et al., 2025)	arXiv	2025
DO (Hassid et al., 2025)	arXiv	2025
FFS (Agarwal et al., 2025)	arXiv	2025
CAR (Lu et al., 2025a)	arXiv	2025
ASC (Azizi et al., 2025)	arXiv	2025
MUR (Yan et al., 2025a)	arXiv	2025
SCoT (Wang et al., 2025b)	arXiv	2025
GoGI (Zhuang et al., 2025)	arXiv	2025
TTPI (Yang et al., 2025a)	arXiv	2025
CGRS (Huang et al., 2025a)	arXiv	2025
SPECS (Cemri et al., 2025)	arXiv	2025
PathC (Zhu et al., 2025)	arXiv	2025
TrimR (Lin et al., 2025c)	arXiv	2025
NOWAIT (Wang et al., 2025a)	arXiv	2025
FracCoT (Liao et al., 2025a)	arXiv	2025
CoThink (Fan et al., 2025)	arXiv	2025
R-Stitch (Chen et al., 2025c)	arXiv	2025
ValueFree (Sareen et al., 2025)	arXiv	2025
Certaindex (Fu et al., 2025a)	arXiv	2025
InftyThink (Yan et al., 2025b)	arXiv	2025
FlashThink (Jiang et al., 2025b)	arXiv	2025
NoThinking (Ma et al., 2025a)	arXiv	2025
Self-Guided (Zhao et al., 2025c)	arXiv	2025
ThoughtMani (Liu et al., 2025f)	arXiv	2025
Retro-Search (Lu et al., 2025b)	arXiv	2025
ThinkDeepFast (Wang et al., 2025c)	arXiv	2025
Collaborative (Lee et al., 2025b)	arXiv	2025
BudgetGuidance (Li et al., 2025a)	arXiv	2025
Plan and Budget (Lin et al., 2025a)	arXiv	2025
Self-Affirmation (Liu et al., 2025b)	arXiv	2025
Sleep-time Compute (Lin et al., 2025b)	arXiv	2025
Length-filtered Vote (Wu et al., 2025b)	arXiv	2025
Searching Skeleton (Zhang et al., 2025i)	arXiv	2025
Speculative Thinking (Yang et al., 2025d)	arXiv	2025

Table 10: Summary of recent arXiv Preprints addressing Efficient Reasoning through Inference Time Dynamic Reasoning.

basic math, while Zhang et al. (2026a) identify abrupt performance collapse beyond critical complexity thresholds. To evaluate mitigation strategies and model calibration, benchmarks such as (Pu et al., 2025) and (Li et al., 2025g) introduce metrics like token efficiency and CoT precision/recall. Moving towards a holistic evaluation, unified frameworks like (Aggarwal et al., 2025) and (Huang et al., 2025b) formalize the dual challenge of preventing waste on easy tasks while ensuring sufficient thought for hard ones, using composite scores like the E3-Score. Beyond these, parallel works also encompass benchmarks for evaluating long context understanding (Huang et al., 2025d), routing mechanisms in LLMs (Huang et al., 2025c), creative thinking (Zhongzhan et al., 2025; Zhong et al., 2024a), agentic tasks (Ling et al., 2026), challenging math competitions (An et al., 2025a), and general reasoning across diverse tasks (Liu et al., 2026). These benchmarks collectively provide the ground truth for developing and comparing the RL, SFT, dynamic, and prompt-guided methods discussed in prior subsections.

C.1.6 Connection to Our Work

Distinct from RL (Team et al., 2025; Luo et al., 2025c; Yeo et al., 2025; Zhou et al., 2026) and SFT (Kang et al., 2024; Ma et al., 2025b; Xia et al., 2025; Yu et al., 2024) approaches that enforce efficiency via static training objectives, our method avoids inducing a fixed length bias. We instead formulate efficiency as a dynamic search objective. This decoupling enables adaptive compute allocation; the model expands reasoning for complex queries and prunes redundancy for simpler ones. We therefore prevent the performance degradation frequently observed with forced conciseness (Li et al., 2025d; Jin et al., 2024). Since we intervene at inference time, our strategy remains orthogonal to these training-based optimizations. Our framework aligns with dynamic reasoning paradigms (Sun et al., 2024; Ding et al., 2025c; Li et al., 2025b) utilizing test-time compute, yet introduces a structural shift in the search space. While prior work relies on token-level search (Sun et al., 2024) or heuristic early stopping (Fu et al., 2025a), we reformulate CoT generation as a dynamic search over discrete reasoning operators. By abstracting tokens into operators, we resolve the high-level planning bottleneck. This renders the search strategic and computationally feasible compared to unstructured sampling (Wang et al., 2025i).

C.2 Test-Time Compute via Search

The paradigm of scaling test-time compute has emerged as a critical frontier in enhancing LLM reasoning. This field can be broadly categorized through two complementary lenses: the topology of reasoning (the structural connection of thoughts) and the search algorithms (the control policies for traversing these structures).

C.2.1 Structured Reasoning Topologies

This line of research focuses on the structural representation of intermediate reasoning steps, moving from linear sequences to complex non-linear structures (Besta et al., 2025).

Chain-based Structures The seminal Chain-of-Thought (CoT) (Wei et al., 2023) prompting demonstrated that eliciting intermediate reasoning steps significantly boosts performance on complex tasks.

This linear topology models reasoning as a sequential path graph. Extensions such as CoT-SC (Self-Consistency) (Wang et al., 2023) introduce a "Tree of Chains" topology by sampling multiple independent reasoning paths and aggregating the final answer via majority voting. While effective, chain-based methods suffer from error propagation in long-horizon tasks, as they lack mechanisms to explore alternative branches once a step is generated.

Tree-based Structures To enable exploration and backtracking, Tree of Thoughts (ToT) (Yao et al., 2023) and (Long, 2023) generalize CoT by modeling reasoning as a tree, where nodes represent partial solutions or "thoughts". This allows the model to explore multiple reasoning branches at each step. Variants such as Thought Decomposition (Xie et al., 2023) and Tree-of-Mixed-Thought (Hu et al., 2023) further refine this by varying the granularity of tree nodes. Other works like Skeleton-of-Thought (Ning et al., 2024) utilize a parallel tree structure (or 1-level tree) to accelerate generation by expanding independent points simultaneously. While tree topologies allow for local exploration, they often require manually defining the branching factor and depth.

Graph-based Structures Graph of Thoughts (GoT) (Besta et al., 2024) and (Yao et al., 2024) further extend reasoning topologies to arbitrary directed acyclic graphs (DAGs). These frameworks introduce aggregation operations, allowing information from multiple independent reasoning paths to be combined into a synergistic solution. Similarly, Cumulative Reasoning (Zhang et al., 2025h) and Everything of Thoughts (XoT) (Ding et al., 2024) utilize graph structures to model complex dependencies where a thought may depend on multiple non-consecutive precursors. While powerful, graph-based methods incur significant computational overhead due to the complexity of managing arbitrary dependencies. Other graph-based reasoning works include Weight-of-Thought (Punjwani and Heck, 2025), LogicAgent (Zhang et al., 2026b) and StrucSum (Yuan et al., 2026).

C.2.2 Search Algorithms and Planning

Parallel to structural definitions, significant research focuses on the algorithmic procedures used to traverse the reasoning space. These approaches typically formulate the reasoning task as an MDP defined by states, actions, and rewards (Li, 2025).

Uninformed and Heuristic Search Early attempts applied standard search algorithms to LLM decoding. Beam Search, as utilized in Beam-LLM (Xie et al., 2023) and PathFinder (Golovneva et al., 2023), maintains the top-k most promising partial candidates at each step. To guide the search more effectively, heuristic methods like Best-First Search (Koh et al., 2025) and A* Search have been adapted. For instance, LLM-A* (Meng et al., 2025) and ToolChain* (Zhuang et al., 2023) integrate cost-to-go heuristics (often estimating the distance to the goal) to prioritize the expansion of promising nodes. Q^* (Wang et al., 2024) further approximates optimal Q-values to guide the search using A*-like heuristics. These methods rely heavily on the quality of the heuristic function, which is often difficult to define for open-ended reasoning tasks.

Monte Carlo Tree Search (MCTS) MCTS has become a dominant paradigm for solving complex reasoning tasks due to its ability to balance exploration and exploitation. Frameworks such as RAP (Hao et al., 2023), LATS (Zhou et al., 2024), and LLM-MCTS (Zhao et al., 2023) employ MCTS to simulate future outcomes (rollouts) and back-propagate value estimates to the current state. Recent advancements like rStar (Qi et al., 2024) and MC-DML (Shi et al., 2025) introduce specialized selection policies and self-consistency-based evaluations to enhance MCTS in reasoning domains. Furthermore, AlphaZero-inspired approaches like TS-LLM (Feng et al., 2024) and ReST-MCTS* (Zhang et al., 2024) integrate MCTS with model training, using the search results to iteratively fine-tune the policy and value networks.

C.2.3 Connection to Our Work

While existing search-based inference methods demonstrate strong performance, they typically rely on heavy sampling, expensive rollouts (as in MCTS), or rigid topological constraints (as in ToT). Our proposed Neural Chain-of-Thought Search differs by internalizing the search process. Instead of managing an external search tree, we treat the discrete thinking tokens as the action space in a Neural Architecture Search (NAS) formulation. This allows our model to dynamically learn a lightweight policy that steers the reasoning topology on-the-fly, achieving the benefits of structured search with significantly lower inference latency than traditional MCTS or massive parallel sampling.

C.3 AutoML

Automated Machine Learning (AutoML) aims to automate the end-to-end process of applying machine learning to real-world problems, thereby reducing the reliance on human expertise and manual trial-and-error (He et al., 2021). The scope of AutoML is broad, covering various stages of the deep learning pipeline including data preparation, feature engineering, hyperparameter optimization (HPO), and model generation (Shen et al., 2024). In the realm of data preparation and feature engineering, techniques have been developed for automated data cleaning, synthesis, and feature selection to maximize the predictive power of raw data (Chu et al., 2016). However, the most computationally intensive aspect of AutoML lies in model selection and optimization. Traditional approaches focused heavily on Hyperparameter Optimization (HPO) to tune static parameters such as learning rates or batch sizes using methods like Grid Search, Random Search (Bergstra and Bengio, 2012), Bayesian Optimization (Snoek et al., 2012), and bandit-based strategies like Hyperband (Li et al., 2018). With the advent of deep learning, the focus of AutoML has progressively shifted from tuning hyperparameters of fixed models to the automatic discovery of the model structure itself, leading to the emergence of Neural Architecture Search (NAS).

C.3.1 Neural Architecture Search

Neural Architecture Search (NAS) is a prominent subfield of AutoML dedicated to automating the design of neural network topologies, which has successfully produced architectures surpassing manually designed counterparts in tasks like image classification and object detection (Elsken et al., 2019). A standard NAS framework is typically categorized into three dimensions: search space, search strategy, and performance estimation strategy (Elsken et al., 2019). The search space defines the set of representable architectures, evolving from simple chain-structured sequences to complex cell-based search spaces (Zoph et al., 2018) and hierarchical representations (Liu et al., 2018). Regarding search strategies, early works utilized Reinforcement Learning, where a controller RNN samples architectures and is trained via policy gradient to maximize validation accuracy (Zoph and Le, 2017). Evolutionary Algorithms (EA) have also proven effective by evolving a population of architectures through mutation and crossover operations (Real et al., 2019). To mitigate the prohibitive computational costs of training each candidate from scratch, recent research has pivoted towards efficiency. This includes differentiable search methods like DARTS (Liu et al., 2019) which relax the discrete search space to allow gradient-based optimization, and One-Shot methods (Bender et al., 2018) that utilize weight sharing within a supernet (Pham et al., 2018). Furthermore, resource-aware NAS has gained traction, where objective functions are modified to penalize computational costs such as FLOPs or latency (Tan et al., 2019), explicitly balancing performance with efficiency (Cai et al., 2019). Similar principles of architecture optimization and automatic design have been explored in diverse domains, including diffusion models (Huang et al., 2023; Zhong et al., 2023; Lin et al., 2024), convolutional network pruning (Huang et al., 2021), and specialized vision tasks (Shi et al., 2024; Wu et al., 2025a; Lu et al., 2024).

C.3.2 Connection to Our Work

Our proposed Neural Chain-of-Thought Search draws significant inspiration from the formulations and methodologies of NAS, yet adapts them to the novel domain of linguistic reasoning. Analogous to how NAS searches for an optimal sequence of layers or operations to process an image (Baymurzina et al., 2022), our framework searches for an optimal sequence of thinking tokens (reasoning operators) to process a complex query. We explicitly define a discrete search space of reasoning operators and employ a policy network to navigate this space, mirroring the controller-based paradigms seen in RL-based NAS (Zoph and Le, 2017). Furthermore, our dual-factor heuristic function, which penalizes generation length to encourage efficient reasoning, directly parallels the multi-objective optimization found in resource-aware NAS methods like MnasNet (Tan et al., 2019) or EfficientNet (Tan and Le, 2020). However, a critical distinction lies in the granularity and dynamism of the search. While traditional NAS typically outputs a static architecture that is fixed for all dataset instances (Elsken et al., 2019), our method performs a dynamic, instance-wise search where the "architecture" of the reasoning path is constructed on-the-fly conditioned on the specific input query. Additionally, unlike NAS which often requires expensive retraining of the searched architecture, our method optimizes the reasoning topology during inference time, leveraging the pre-trained capabilities of the underlying Large Language Model.

Prompt1

You are an expert reasoning analyst.
The following is one reasoning step extracted from a long chain-of-thought.
Classify its thinking pattern into exactly one of:
- "statement" (presenting facts, calculations, or direct reasoning)
- "reflection" (self-check, doubt, or correction)
- "summary" (concluding or asserting finality)
- "divergence" (exploring alternative paths or possibilities)

Step: """{Thus, the number of digits is $\boxed{18}$. }"""
Return only: {"label": "<one of statement | reflection | summary | divergence>"}

Figure 8: The definition-based prompt template for classifying thinking modes based on static definitions.

Prompt2

You are an expert reasoning analyst.
The following is one reasoning step extracted from a long chain-of-thought.
Analyze the functional role of this step within the problem-solving flow and classify it into one of:
- "statement" (advancing the flow by deriving new information, calculating, or deducing next steps)
- "reflection" (pausing the flow to critique, verify, or validate the correctness of previous steps)
- "summary" (closing the flow by synthesizing results or explicitly stating the final conclusion)
- "divergence" (branching the flow, shifting strategy, or proposing alternative hypotheses)

Step: """{Thus, the number of digits is $\boxed{18}$. }"""
Return only: {"label": "<one of statement | reflection | summary | divergence>"}

Figure 9: The function-based prompt template for analyzing the role of reasoning steps within the reasoning flow.

Our approach also contrasts with other automated ML techniques applied to different problem domains, such as multimodal recommendation systems (Zhao et al., 2025d; Zhong et al., 2024b), multimodal association evaluation (Liu et al., 2025d), dialogue systems (Zhong et al., 2022), and continual learning in NLP (Chen and Zeng, 2025), which focus on optimizing specific application pipelines.

D Characterization of Thinking Tokens

D.1 Experimental Setup

To explore the relationship between thinking tokens and think patterns, we analyze traces from two Small Reasoning Models: DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Llama-8B. We evaluate on AMC23 (standard competition math) and AIME24 (complex reasoning chains). For each model-dataset pair, we generate full reasoning traces, segmenting them into discrete steps via the `"\n\n"` delimiter. We employ DeepSeek-V3 and GPT-4o to annotate the thinking mode of each step. To ensure robustness, we utilize two prompt strategies: (1) definition-based, classifying steps against rigorous academic definitions, and (2) function-based, assessing the step's role in the problem-solving flow (see Fig. 8 and 9).

D.2 Analysis of Results

We observe a deterministic correspondence between specific initial tokens and the subsequent reasoning trajectory. The token "Wait" serves as a trigger for self-correction, initiating Reflection steps with a probability exceeding 90%. Similarly, "Alternatively" functions as a dedicated branch indicator, leading to Divergence steps in over 95% of cases. In contrast, "Thus" acts as a transitional operator, distributing its probability mass nearly equally between deductive Statements and conclusive Summaries. This distribution suggests that thinking tokens function not merely as syntactic connectors but as semantic control signals that modulate the generation logic. We posit that distinct initial tokens activate specific, latent thinking modes inherent to the LRM. These modes are not explicitly

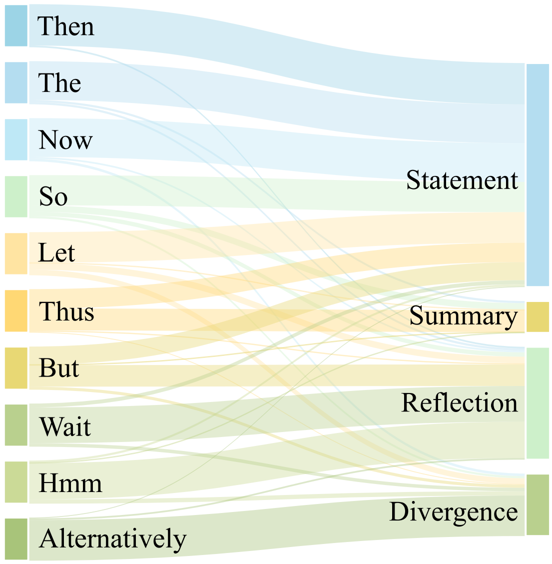

Figure 10: Correlation between thinking tokens and thinking modes for DeepSeek-R1-Distill-Qwen-1.5B on the AMC23 dataset. The reasoning steps were classified by DeepSeek-V3 using the definition-based prompt strategy (Prompt 1).

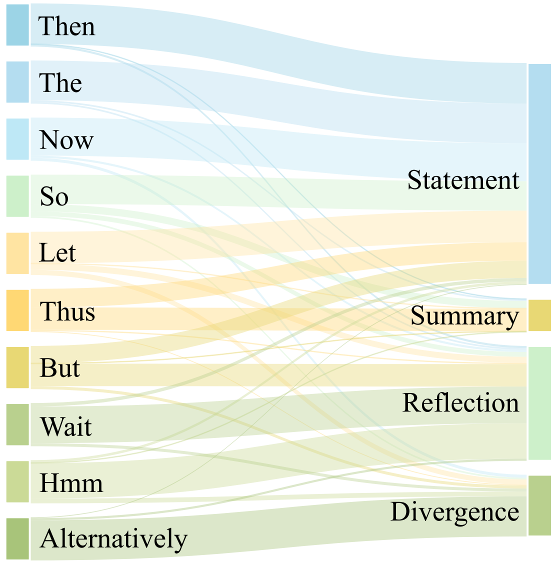

Figure 11: Correlation between thinking tokens and thinking modes for DeepSeek-R1-Distill-Llama-8B on the AIME24 dataset. The reasoning steps were classified by GPT-4o using the function-based prompt strategy (Prompt 2).

defined but emerge as clustered behaviors within the model's high-dimensional representation space. Our Neural Chain-of-Thought Search exploits this structure. By discretely selecting the optimal thinking token at each decision point, the algorithm effectively performs dynamic cognitive switching. This mechanism allows the system to navigate the solution space by actively engaging the most appropriate latent reasoning mode for the current context, thereby maximizing solution efficiency.

E Justification of Design Choices

E.1 Why do we use `\n\n` as delimiter

E.1.1 Statistical Evidence

Our choice of the double newline (`"\n\n"`) as the delimiter for reasoning steps is grounded in the empirical analysis of Large Reasoning Models' generation patterns. Recent work (Yang et al., 2025d) studies the distribution of tokens preceding "reasoning-supportive" keywords, which are terms that explicitly signal a shift in reasoning mode, including "Wait" (reflection), "Alternatively" (branching), and "Hmm" (hesitation). Their analysis on the MATH500 dataset reveals a strong conditional dependence: these pivot tokens are overwhelmingly preceded by the `"\n\n"` delimiter. As shown in Table 11, for the token "Wait", approximately 90% of occurrences follow a double newline. Similarly, "Alternatively" follows this delimiter pattern with a probability of over 92%. This statistical dominance indicates that `"\n\n"` is not merely a syntactic formatter but a latent signal where the model naturally pauses to determine the trajectory of the subsequent thought process.

E.1.2 Functional Role as Discourse Marker

Beyond statistical correlation, the `"\n\n"` token functions as a critical discourse marker in the latent space of LRMs. The segment immediately following a double newline typically determines the functional category of the next step. Analysis categorizes these post-delimiter segments into distinct modes: Affirmation (continuing the current logic), Reflection (backtracking or verifying), or Statement (deriving new formulas). For instance, when a model generates `"\n\n"`, it enters a "decision state" where it must implicitly choose whether to proceed or reflect. In standard autoregressive decoding, this choice is made probabilistically based on the preceding context. However, smaller models often fail at this juncture, producing repetitive "Statement" steps when a "Reflection" is required, or entering verification loops unnecessarily.

E.1.3 Implications for our work

These findings validate our formulation of the search space defined in Section 2.1. By designating `"\n\n"` as the decision point d_t, we align our search intervention with the model's intrinsic cognitive structure. Rather than imposing arbitrary boundaries, we intervene exactly at the moment the

Keyword	Previous Token	Probability
	`"x\n\n"`	0.950
"alternatively"	" "	0.050
	Other	< 0.001
	" "	0.690
"hmm"	`"x\n\n"`	0.297
	Other	0.013
	`"x\n\n"`	0.808
"wait"	" "	0.182
	Other	0.100

Table 11: Proportion of top preceding tokens for reasoning-supportive words in Deepseek-Distilled Qwen-2.5-32B on MATH500. Data adapted from previous work (Yang et al., 2025d). The results show that the double newline delimiter dominates the distribution. `"x\n\n"` denotes all tokens containing "\n\n", including "\n\n", ").\n\n", "?\n\n", " \n\n", "].\n\n", "\n\n", ")\n\n", "]\n\n", "?)\n\n".

model naturally pauses to select a thinking mode. Our method effectively externalizes this implicit decision-making process. By injecting explicit operators (e.g., forcing a "Wait" or "So") at these precise structural breaks, we can actively steer the model out of suboptimal paths (such as the Excessive Reflection identified in smaller models) and toward the superior paths that optimize both accuracy and efficiency.

E.2 How do we choose the operator set $\mathcal{O}$

We determine the composition of the operator set $\mathcal{O}$ through a statistical analysis of the vocabulary distribution in the training corpus. Specifically, we calculate the frequency of tokens that immediately follow the step delimiter `"\n\n"`. By selecting the tokens that most frequently initiate a new reasoning step, we ensure that our constructed search space aligns with the model's natural generation patterns and covers the most probable reasoning transitions.

E.2.1 Operator Set for Neural CoT Search

In our proposed method, we employ a comprehensive operator set $\mathcal{O} = \{$"The", "Thus", "Therefore", "So", "Then", "Let", "Wait", "Alternatively", "Now", "I", "First", "Option", "**", "-", "\[", "\"$\}$. We selected these tokens because they are not only statistically frequent but also serve distinct and necessary functional roles in structuring the reasoning process. Beyond the standard thinking tokens that guide logical flow (e.g., "Thus", "Wait"), we explicitly include functional markers. For instance, "Option" is crucial for analyzing specific choices in multiple-choice questions; "**" and "-" are widely used for emphasis and enumeration to organize complex arguments; and "\[" and "\" are essential for initiating mathematical derivation blocks. Each of these tokens represents a specific mode of operation that the model frequently utilizes to construct valid reasoning steps.

E.2.2 Operator Set for Random Search

In contrast, for our random search experiments, we utilize a restricted subset: $\mathcal{O}_{\text{random}} = \{$"The", "Thus", "Therefore", "So", "Then", "Let", "Wait", "Alternatively"$\}$. The rationale for this difference lies in the inherent limitation of random sampling. Unlike our learned policy, a random search agent lacks the semantic understanding to apply context-dependent formatting tokens correctly. Randomly selecting structural tokens such as "First", "Option", "**", "-", or "\[" without appropriate context often leads to syntactically broken or incoherent generations (e.g., opening a math block with "\[" when no calculation is needed, or starting a list with "-" in the middle of a sentence). Therefore, we limit the random search space to purely connective thinking tokens to ensure that the sampled paths remain semantically plausible.

E.2.3 Impact of Operator Set

The selection of the operator set involves a fundamental trade-off between search potential and computational cost. Expanding the operator set to encompass a broader vocabulary increases the coverage of the search space, which theoretically allows for the discovery of even higher-quality reasoning paths. Specifically, while our current set is primarily optimized for English STEM reasoning, the framework allows for straightforward extension to multilingual domains by incorporating thinking tokens from other languages. Furthermore, the set can be recalibrated to support creative tasks by adding tokens that guide narrative planning or brainstorming. However, a larger set also increases the branching factor at each decision point, which raises both the training complexity of the policy network and the computational overhead during inference. Our empirical observations indicate that the current selection strikes an effective balance, enabling the discovery of superior reasoning paths while keeping the search process computationally efficient.

Question: "Plants have specialized tissues at the ends of their stems. What do these tissues help the plants do? Choices: A. digest food B. grow taller C. make food D. absorb water" (Answer: B)
Original (2032 tokens): " Okay, so I have this question about plants and their specialized tissues … First, the main types of tissues in plants are probably similar to animals Wait, no, the epidermis is the outer layer of the plant, … But at the ends of stems, maybe there are specific tissues … Let me think about what each tissue does. Root hair cells are … Oh, right! The ends of stems are called leaves. … Wait, the question is about specialized tissues at the ends of their stems. … Wait, maybe it's the xylem or phloem at the tip? … But wait, the question says ""at the ends of their stems."" Leaves are at the ends of branches, not just stems … But I'm a bit confused because leaves are the main site for photosynthesis, which involves making food. So, maybe the answer is C. Wait, the question says ""the ends of their stems."" … Alternatively, maybe the question is referring to the terminals of the stem, which are where leaves grow. So, the cells there might be specialized for photosynthesis. So, if the cells are specialized for making food, then the answer is C. But I'm not entirely sure. Let me think again. … So, the ends of stems … Then, the mesophyll is where photosynthesis occurs … So, the tissues themselves are the epidermis and mesophyll, … Wait, maybe I'm overcomplicating it. … So, the question is asking what these tissues help the plant do … Wait, perhaps the answer is both D and C. Looking at the options: A. digest food … So, if the specialized tissues at the ends of stems are leaves … Alternatively, maybe it's the xylem or phloem at the ends of stems. … But I'm still a bit confused because I know that the epidermis is specialized for absorbing water … I think I need to decide. If the question is about the specialized tissues at the ends of stems, … But if it's about the epidermis, then D. I think the correct answer is D, absorb water, because … I think I'll go with D, absorb water. **Final Answer** $\boxed{D}$ "
NCoTS (433 tokens): " Okay, so I have this question about plants and their specialized tissues. … First, I remember that plants have different kinds of tissues. … Wait, what do those tissues do? I recall that in the stems, the ends are responsible for … Let me go through the options again to make sure. Option A is digest food. That's more about root hairs … Option B is grow taller. As I thought earlier … Option C is make food. Making food is done by leaves … Option D is absorb water. I think water absorption is more related to the roots … So, putting it all together, the correct answer should be B … **Final Answer** $\boxed{B}$ "

Table 12: Comparison of reasoning paths between original DeepSeek-R1-Distill-Qwen-7B (top) and our NCoTS (bottom). Blue boxes: reflection/verification steps. Green boxes: divergence steps. Yellow boxes: statement/summarization steps. The original model drifts into irrelevant reasoning branches, resulting in an incorrect answer. NCoTS enables the model to reach the correct conclusion with higher efficiency.

F Case Study

We present two qualitative comparisons between the original generation and our proposed NCoTS. As detailed in Table 12 and Table 13, these cases illustrate how current reasoning models frequently lack foresight and consequently fail to navigate the solution space efficiently.

In the commonsense science query shown in Table 12, the original model experienced severe reasoning drift. Lacking a high level plan, it conflates concepts like ends of stems with leaves and wanders into irrelevant topics such as water absorption, which ultimately leads to an incorrect conclusion In contrast, NCoTS mitigates this issue by actively searching for optimal thinking mode at decision points. This mechanism allows the model to prune these inefficient branches early, effectively steering the path toward the correct solution with significantly reduced token usage. Similarly, the mathematical reasoning task in Table 13 highlights the inefficiency of myopic next token prediction. While the original model eventually finds the solution, it suffers from a lack of confidence, evidenced by getting stuck in redundant verification loops and exploring complex but unnecessary branches. NCoTS eliminates this overhead by prioritizing decisive operators that advance the solution state. Our approach performs one succinct verification step, reflecting a more confident and efficient reasoning process.

These case studies form a closed loop with our initial motivation. They confirm that the inefficiency of current reasoning models stems from the failure to foresee the appropriate reasoning direction at decision points. By treating reasoning as a dynamic search for the optimal thinking mode, NCoTS actively selects the suitable reasoning direction and effectively prunes redundant branches. This yields a superior reasoning path that is both accurate and concise, avoiding reasoning traps such as getting stuck in redundant verification loops or excessive exploration.

Question: There is a unique sequence of integers $a_1, a_2, \cdots a_{2023}$ such that $$\tan 2023x = \frac{a_1 \tan x + a_3 \tan^3 x + a_5 \tan^5 x + \cdots + a_{2023} \tan^{2023} x}{1 + a_2 \tan^2 x + a_4 \tan^4 x \cdots + a_{2022} \tan^{2022} x}$$ whenever $\tan 2023x$ is defined. What is a_{2023}? (Answer: -1)
Original (4416 tokens): "Okay, so I have this problem here where I need to find the value of a_{2023} in a unique sequence of integers. ... Hmm, okay. I remember that multiple-angle identities for tangent can be expressed using ... Given that, I think the coefficients $a_1, a_3, \ldots, a_{2023}$ are related to ... Alternatively, I remember that there is a formula involving Bernoulli numbers or some recursive relation for the coefficients in the multiple-angle expansion of tangent. ... Wait, another thought: the tangent function can be expressed using complex numbers. ... Alternatively, perhaps using the expansion in terms of Bernoulli polynomials or some generating function? ... Wait, maybe I can use the identity for $\tan(nx)$. ... Yes, for example, for $\tan(3x)$, as I mentioned earlier, it's $\frac{3\tan x - \tan^3 x}{1-3\tan^2 x}$. ... Wait, looking at $\tan(3x)$, the numerator is $3\tan x - \tan^3 x$, which is similar to the expansion of $\sin(3x)/\cos(3x)$, ... But perhaps for the general case, the expansion of $\tan(nx)$ as a rational function ... Wait, another approach: perhaps using the formula for $\tan(A + B)$. ... But 2023 is a large number, so recursively applying this would be tedious. ... After a quick recall, I remember ... Moreover, the leading term in the numerator is ... Wait, let's test this with small n. ... So, in general, it seems that for $\tan(nx)$ ... Wait, but in the case of n=3 ... Therefore, for n=2023, the leading term in the numerator would be ... Wait, but in the problem statement ... But hold on, let me check with n=3 again. ... Wait, that's a problem. ... Looking at n=3 ... So, for n=3, the leading coefficient is -1 ... Wait, for n=3: $(-1)^{(3-1)/2} = (-1)^1 = -1$ ... But in n=3, the leading coefficient is -1, which matches $(-1)^1 = -1$. ... So, for n=2023, the leading coefficient is -1, so $a_{2023} = -2023$. ... Wait, no, in n=3, the leading term is $-\tan^3 x$, so the coefficient is -1. ... Therefore, for n=2023 ... But in the numerator, for n=3 ... So, in general, for the numerator of ... Wait, hold on. In n=3, the leading term is ... Wait, but in the problem statement ... But hold on, in n=3 ... So, the coefficient of $\tan^n x$ is ... Therefore, for n=2023, the coefficient a_{2023} is $(-1)^{(2023-1)/2} = (-1)^{1011} = -1$. ... But hold on a second, in n=3, the leading term is $-\tan^3 x$ ... Yes, $\tan(3x) = \frac{3\tan x - \tan^3 x}{1-3\tan^2 x}$. ... Similarly, for n=5, the numerator is $5\tan x - 10\tan^3 x + \tan^5 x$ ... So, the leading term is $-\tan^3 x$. So, the coefficient is -1. ... So, in general, for each odd n ... Wait, but in n=3, the leading coefficient is -1 ... Therefore, $a_{2023} = (-1)^{(2023-1)/2} = (-1)^{1011} = -1$. ... But wait, another thought ... But let's think about how the expansion is constructed ... So, based on the pattern observed in smaller cases ... **Final Answer** $\boxed{-1}$"
NCoTS (1771 tokens): " Okay, so I have this problem here about a unique sequence of integers $a_1, a_2, \ldots, a_{2023}$ ... I need to find the value of a_{2023}. First, I recall that there's a formula for the tangent of multiple angles. ... So, for $n = 2023$, the numerator will be a polynomial of degree 2023 in $\tan x$, ... The problem is asking for the coefficient a_{2023} of the highest degree term ... Now, how do I find this coefficient? I remember that the tangent multiple-angle formula can be derived using the addition formula for tangent. The addition formula is: $\tan(A+B) = \frac{\tan A + \tan B}{1 - \tan A \tan B}$ So, if we apply this repeatedly, we can express $\tan(nx)$ in terms of $\tan x$. ... Therefore, in general, for $\tan(nx)$, the numerator is: $\sum_{k=1}^{n} (-1)^{k-1} \binom{n}{2k-1} \tan^{2k-1} x$ So, the coefficient a_{2023} is $(-1)^{1011} \binom{2023}{2023}$ ... Let's compute this: $a_{2023} = (-1)^{1011} \binom{2023}{2023}$ I know that $\binom{2023}{2023} = 1$, so this simplifies to: ... Now, 1011 is an odd number, so $(-1)^{1011} = -1$. Therefore, $a_{2023} = -1$. Wait, let me just verify if this makes sense. ... So, the numerator is $3\tan x - \tan^3 x$ ... So, similarly, for $n = 2023$, $a_{2023} = (-1)^{1011} \times 1 = -1$. So, that seems consistent. I think that's the answer. The coefficient of the highest degree term in the numerator of $\tan 2023x$ is $(-1)^{1011} \times 1 = -1$. **Final Answer** $\boxed{-1}$ "

Table 13: Comparison of reasoning paths between original DeepSeek-R1-Distill-Qwen-1.5B (top) and our NCoTS (bottom). Blue boxes: reflection/verification steps. Green boxes: divergence steps. Yellow boxes: statement/summarization steps. This case demonstrates that the original model struggles with a lack of foresight, leading to excessive exploration and redundant verification. In contrast, NCoTS delivers a coherent derivation that solves the problem using fewer than 50% of the tokens.