Article

Not peer-reviewed version

Toward Efficient and Faithful Reasoning in Large Language Models

Lukas Schneider , Anna Muller , Mareike Gerhardt *

Posted Date: 18 July 2025

doi: 10.20944/preprints202507.1531.v1

Keywords: large language models; efficient reasoning; Chain-of-Thought; tool-augmented inference; neuro-symbolic AI; prompt engineering; program induction; adaptive computation; reasoning benchmarks; interpretability

 doi:10.20944/preprints202507.1531.v1

Article

Toward Efficient and Faithful Reasoning in Large Language Models

Lukas Schneider, Anna Muller and Mareike Gerhardt *

Karlsruhe Institute of Technology, Germany
* Correspondence: mareike.gerhardt@kit.edu

Abstract

Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of tasks, from natural language understanding to arithmetic reasoning and code generation. However, enabling these models to reason efficiently—achieving high performance with minimal computational overhead and maximal interpretability—remains an open challenge. This survey presents a comprehensive overview of methodologies for building efficient reasoning models with LLMs. We categorize the landscape into prompt-based methods (e.g., chain-of-thought, self-consistency), architectural and tool-augmented enhancements (e.g., retrieval-augmented generation, program-aided reasoning, memory systems), and training-time techniques (e.g., distillation, curriculum learning). We also review evaluation protocols and benchmark datasets that capture diverse reasoning requirements, from symbolic logic and mathematical problem solving to multi-hop question answering. In addition to characterizing the trade-offs between accuracy and inference cost, we highlight emerging trends in neuro-symbolic integration, adaptive computation, lifelong learning, and interpretable reasoning. We conclude by identifying open challenges and future directions toward general-purpose reasoning agents. This survey aims to serve both as a structured map of recent developments and a call to advance reasoning efficiency as a first-class objective in the next generation of LLM research.

Keywords: large language models; efficient reasoning; Chain-of-Thought; tool-augmented inference; neuro-symbolic AI; prompt engineering; program induction; adaptive computation; reasoning benchmarks; interpretability

1. Introduction

Large Language Models (LLMs), such as GPT, BERT, and their variants, have revolutionized natural language processing (NLP) by demonstrating unprecedented capabilities in a wide array of tasks, including language understanding, generation, summarization, and reasoning. These models, typically based on the Transformer architecture, possess billions of parameters and are pre-trained on massive corpora, enabling them to capture intricate linguistic patterns and world knowledge [1]. However, despite their remarkable performance, LLMs encounter significant challenges when tasked with complex reasoning processes due to their inherent architectural and computational constraints. Reasoning, the ability to draw logical conclusions, infer implicit information, and synthesize knowledge, is central to many advanced NLP applications such as question answering, dialogue systems, code generation, and scientific discovery. While early NLP models primarily relied on symbolic and rule-based approaches for reasoning, LLMs have shifted the paradigm by implicitly encoding reasoning skills within distributed representations [2]. Nonetheless, achieving efficient and reliable reasoning in LLMs remains a formidable challenge, as reasoning often requires multi-step, compositional, and context-dependent inference that can strain the models' memory, computation, and interpretability. This survey focuses on the development and advancement of *efficient reasoning models* for LLMs, aiming to bridge the gap between the expressive power of large-scale pre-trained models and the practical demands of reasoning-intensive applications [3]. By "efficient reasoning models,"

 doi:10.20944/preprints202507.1531.v1

we refer to techniques, architectures, and frameworks that enhance the reasoning abilities of LLMs while optimizing resource consumption such as computational cost, memory footprint, latency, and scalability [4]. The pursuit of efficiency is motivated by the need to deploy reasoning-capable LLMs in real-world scenarios where computational resources are limited, response times are critical, and interpretability is desired [5]. Several key challenges underscore the complexity of designing efficient reasoning models for LLMs:

- **Computational Complexity:** Large-scale transformers involve quadratic complexity with respect to sequence length, making long-horizon reasoning prohibitively expensive. Efficient models seek to reduce this burden through architectural innovations or approximations.
- **Memory Constraints:** Reasoning tasks often require maintaining and manipulating large context windows or knowledge bases, which can exceed the memory capacity of standard LLMs. Methods to compress, retrieve, or summarize context play a critical role [6].
- **Interpretability and Transparency:** Unlike symbolic reasoning systems, neural reasoning models are often black boxes, limiting their explainability. Efficient reasoning models aim to improve interpretability while maintaining performance [7].
- **Generalization and Compositionality:** Reasoning frequently involves applying learned knowledge in novel combinations and contexts [8]. Efficient models must generalize beyond training distributions without excessive retraining or parameter increase.
- **Multi-step and Hierarchical Reasoning:** Complex reasoning may require sequential inference steps or hierarchical decomposition of problems. Models must balance reasoning depth with efficiency and error accumulation.

In response to these challenges, the research community has proposed a diverse array of approaches spanning algorithmic innovations, architectural modifications, training paradigms, and auxiliary components. These include sparse attention mechanisms, retrieval-augmented models, modular and compositional architectures, memory-augmented networks, and hybrid neuro-symbolic frameworks. Moreover, advances in efficient fine-tuning, knowledge distillation, and pruning techniques contribute to making reasoning-capable LLMs more practical and accessible. This survey aims to provide a comprehensive and systematic overview of the state-of-the-art in efficient reasoning models for LLMs. We categorize and analyze recent developments, highlighting their principles, advantages, limitations, and potential application domains [9]. We also discuss benchmarking strategies, evaluation metrics, and open research questions that are critical for advancing this vibrant area [10]. The remainder of this paper is structured as follows [11]. Section 2 reviews the background and foundational concepts in LLMs and reasoning [12]. Section 3 delves into architectural approaches for efficiency, including sparse attention and memory enhancements. Section 4 covers algorithmic and training strategies such as retrieval augmentation and multi-step reasoning frameworks. Section 5 explores hybrid models integrating symbolic and neural reasoning paradigms. Section 6 discusses evaluation methodologies and benchmarks. Finally, Section 7 outlines future directions and concluding remarks. By synthesizing the current landscape of efficient reasoning models for LLMs, this survey aims to facilitate informed research and development that push the boundaries of intelligent language understanding and reasoning while maintaining practical feasibility.

2. Background and Foundations

Large Language Models (LLMs) are predominantly built upon the Transformer architecture [**?**], which utilizes self-attention mechanisms to capture contextual dependencies across input se-

 doi:10.20944/preprints202507.1531.v1

quences [13]. Formally, given an input token sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the self-attention operation computes attention weights α_{ij} between tokens x_i and x_j as

$$\alpha_{ij} = \frac{\exp\left(\frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d_k}}\right)}{\sum_{m=1}^{n} \exp\left(\frac{\mathbf{q}_i^\top \mathbf{k}_m}{\sqrt{d_k}}\right)},$$

where $\mathbf{q}_i = W^Q x_i$ and $\mathbf{k}_j = W^K x_j$ are the query and key projections respectively, and d_k is the dimensionality of the key vectors [14]. The output representation at position i is then a weighted sum of the value vectors $\mathbf{v}_j = W^V x_j$,

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{v}_j [15].$$

This mechanism allows the model to dynamically attend to relevant parts of the input, enabling effective capture of long-range dependencies. However, the self-attention's quadratic complexity in sequence length n, i.e., $O(n^2)$, poses substantial computational challenges for reasoning tasks that require processing extensive contexts or multi-hop inference chains. Reasoning within LLMs can be characterized by various forms, including deductive, inductive, abductive, and analogical reasoning. These can be abstracted as transformations on knowledge representations, where the model learns to approximate inference functions $f : \mathcal{K} \times \mathcal{Q} \to \mathcal{A}$, mapping a knowledge base $\mathcal{K}$ and query $\mathcal{Q}$ to an answer $\mathcal{A}$ [16]. Efficient reasoning demands models that not only approximate f with high fidelity but also do so within tractable computational budgets and time constraints. Historically, classical symbolic reasoning systems leverage logic programming, theorem proving, or rule-based inference, offering interpretability and soundness guarantees but suffering from brittleness and poor scalability to natural language understanding. Neural approaches, in contrast, embed reasoning implicitly within distributed representations, enabling robust generalization and end-to-end training, but often lacking explicit transparency and efficiency [17]. To bridge these paradigms, several foundations are key [18]. Memory-augmented neural networks introduce external memory modules $\mathcal{M}$ that can be read and written over multiple reasoning steps, extending the model's capacity to store intermediate conclusions. Formally, the memory state at step t can be described as $\mathbf{M}_t \in \mathbb{R}^{m \times d}$, updated via

$$\mathbf{M}_{t+1} = \text{Update}(\mathbf{M}_t, \mathbf{h}_t),$$

where $\mathbf{h}_t$ denotes the current hidden state [19]. This facilitates iterative refinement and multi-hop reasoning [20]. Another foundational concept is retrieval-augmented generation, where an external knowledge retriever $\mathcal{R}$ dynamically fetches relevant documents or facts $\mathbf{D}$ based on the query $\mathcal{Q}$. The model then conditions its reasoning on the augmented input $(\mathbf{x}, \mathbf{D})$, effectively integrating symbolic knowledge bases with learned representations [21]. The retriever is often trained to optimize

$$\max_{\theta} \mathbb{E}_{(\mathcal{Q},\mathcal{A})} \log P_\theta(\mathcal{A} \mid \mathcal{Q}, \mathcal{R}(\mathcal{Q})),$$

where θ are the model parameters [22]. In addition, compositionality plays a critical role in reasoning efficiency. Compositional models factorize complex reasoning tasks into sequences of simpler subproblems, leveraging the principle that complex functions can be decomposed as

$$f(\mathcal{K}, \mathcal{Q}) = f_n \circ f_{n-1} \circ \cdots \circ f_1(\mathcal{K}, \mathcal{Q}),$$

where each f_i represents a distinct reasoning step or module. This modularity enables focused computation, reuse of sub-solutions, and improved interpretability. In terms of evaluation, reasoning capability is measured through diverse benchmarks encompassing multi-hop question answering (e.g., HotpotQA [**?**]), logical inference, commonsense reasoning (e.g., CommonsenseQA [**?**]), and

 doi:10.20944/preprints202507.1531.v1

code generation challenges. Metrics typically involve accuracy, reasoning step efficiency, and resource consumption [23]. Table 3 summarizes some prominent benchmarks, their reasoning type, and key evaluation metrics [24].

Table 1. Summary of prominent reasoning benchmarks used for evaluating LLM reasoning capabilities.

Benchmark	Reasoning Type	Domain	Input Format	Evaluation Metric
HotpotQA [**?**]	Multi-hop QA	Wikipedia Articles	Textual Questions	Exact Match, F1
CommonsenseQA [**?**]	Commonsense	General Knowledge	Multiple Choice	Accuracy
CLUTRR [**?**]	Relational Reasoning	Synthetic Stories	Textual Stories	Accuracy
ProofWriter [**?**]	Logical Deduction	Synthetic Rules	Rule Sets	Accuracy, Stepwise Correctness
GSM8K [**?**]	Mathematical Reasoning	Math Problems	Word Problems	Accuracy
CodeXGLUE [**?**]	Program Synthesis	Code Repositories	NL-to-Code	BLEU, Exact Match

The convergence of architectural innovations, training paradigms, and external knowledge integration constitutes the foundation upon which efficient reasoning models for LLMs are built. In subsequent sections, we explore these advances in detail, analyzing how they address the dual objectives of enhancing reasoning performance and optimizing computational efficiency [25].

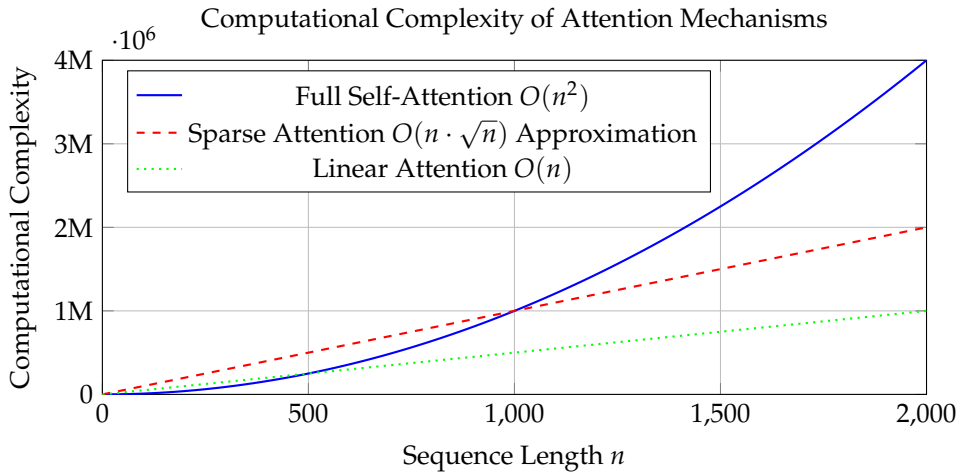

Figure 1. Comparison of computational complexity as a function of input sequence length n for various attention mechanisms [26]. Full self-attention exhibits quadratic growth, while sparse and linear attention mechanisms approximate this with reduced complexity, enabling more efficient reasoning over longer contexts.

3. Architectural Approaches for Efficient Reasoning

The core computational bottleneck in Large Language Models (LLMs) arises primarily from the self-attention mechanism, which scales quadratically with the input sequence length, thereby limiting the practical applicability of reasoning tasks that involve long contexts or multi-hop inference chains [27]. To mitigate these constraints, numerous architectural innovations have been proposed to enhance reasoning efficiency by reducing complexity, improving memory usage, and enabling hierarchical or modular computation. One prominent class of approaches focuses on sparse attention mechanisms that selectively attend to a subset of tokens rather than the full sequence[28,29], thus lowering the effective complexity from $O(n^2)$ to near-linear or sub-quadratic scales. Formally, a sparse attention mask $M \in \{0,1\}^{n \times n}$ restricts the attention computation such that

$$\alpha_{ij} = \frac{\exp\left(\frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d_k}}\right) M_{ij}}{\sum_{m=1}^{n} \exp\left(\frac{\mathbf{q}_i^\top \mathbf{k}_m}{\sqrt{d_k}}\right) M_{im}},$$

where $M_{ij} = 1$ indicates allowed attention, and zero otherwise. Various sparsity patterns have been explored, including local windows [**?**], fixed patterns (strided, block), learnable sparse topologies [**?**], and random attention [30]. These designs trade off expressivity and coverage to achieve substantial

 doi:10.20944/preprints202507.1531.v1

computational savings without significant performance degradation [31]. Complementing sparse attention, linear attention mechanisms reformulate the attention computation by leveraging kernel feature maps $\phi(\cdot)$ to express attention as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \phi(\mathbf{Q})\Big(\phi(\mathbf{K})^\top \mathbf{V}\Big),$$

which can be computed in $O(n)$ time and space [**?**]. Although these approximations enable scaling to very long sequences, they may introduce approximation errors affecting the fidelity of multi-step reasoning chains [32]. Another architectural advancement involves hierarchical and chunk-based processing [33]. Models such as Longformer [**?**] and BigBird [**?**] partition the input sequence into overlapping or non-overlapping chunks processed at multiple scales [34]. This hierarchical representation allows the model to first extract local features and then aggregate global context via sparse or global tokens, effectively balancing local detail with long-range dependencies crucial for reasoning tasks. Formally, the hierarchical attention can be expressed as

$$\mathbf{Z}^{(l)} = \text{Attention}\Big(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}\Big), \quad l = 1, 2, \ldots, L,$$

where each layer l operates at different resolution or chunk granularity. Memory-augmented architectures extend the Transformer paradigm by introducing explicit memory modules that decouple context size from model complexity. Models like Compressive Transformers [**?**] and Memory-Enhanced Transformers [**?**] maintain external memory buffers $\mathbf{M} \in \mathbb{R}^{m \times d}$ updated dynamically to store past computations or intermediate reasoning states [35]. The update mechanism often employs a combination of writing new information and compressing or forgetting obsolete content, defined as

$$\mathbf{M}_{t+1} = \text{Compress}(\mathbf{M}_t \cup \mathbf{h}_t),$$

where $\mathbf{h}_t$ is the current hidden representation. This approach enables effective long-horizon reasoning by preserving salient information over extended sequences while controlling memory size [36]. Modular and compositional architectures decompose reasoning into specialized sub-networks or modules tailored for different reasoning primitives [37]. Such models utilize gating or routing functions $g_i(\cdot)$ to dynamically select relevant modules,

$$\mathbf{z} = \sum_{i=1}^{M} g_i(\mathbf{x}) f_i(\mathbf{x}),$$

where each f_i is a module performing a distinct reasoning operation (e.g., arithmetic, logic, retrieval), and M is the number of modules [38]. This design enhances interpretability and allows reusing modules across different tasks, fostering sample efficiency and scalability. To further alleviate memory and computational demands, parameter-efficient fine-tuning techniques such as adapters [**?**], LoRA [**?**], and prompt tuning [**?**] have been integrated with efficient reasoning models [39]. These methods fine-tune a small subset of parameters or learn task-specific prompts to adapt LLMs for reasoning without retraining or storing full model weights, reducing overhead and enabling deployment on resource-constrained devices [40]. Table 2 summarizes representative architectural approaches for efficient reasoning, detailing their core mechanisms, complexity, and typical application scenarios.

Table 2. Architectural approaches for efficient reasoning in LLMs: mechanisms, complexity, and applications.

Approach	Core Mechanism	Computational Complexity	Typical Applications
Sparse Attention	Attention masking with local/global patterns	$O(n \cdot \sqrt{n})$ to $O(n \log n)$	Long context QA, document summarization
Linear Attention	Kernel feature map approximations	$O(n)$	Streaming data, real-time inference
Hierarchical Models	Multi-scale chunk processing	$O(n \log n)$	Multi-hop reasoning, long text modeling
Memory-Augmented Networks	External memory buffers with update/compress	$O(n)$ per step, constant memory size	Sequential reasoning, dialogue systems
Modular Architectures	Dynamic routing among specialized modules	Depends on number of active modules	Compositional tasks, multi-domain reasoning
Parameter-Efficient Fine-tuning	Adapters, LoRA, prompts	Minimal parameter updates	Task adaptation, resource-constrained deployment

 doi:10.20944/preprints202507.1531.v1

In summary, architectural innovations have significantly expanded the feasibility of applying LLMs to reasoning-intensive tasks by reducing computational bottlenecks and extending effective context length [41]. These techniques enable efficient multi-step inference, scalable memory management, and modular composition of reasoning capabilities. The integration of these architectural methods with advanced training strategies and retrieval augmentation further advances the frontier of efficient reasoning models, which we explore in the next section.

4. Training Paradigms and Optimization for Efficient Reasoning

Beyond architectural innovations, the training paradigms employed to teach Large Language Models (LLMs) to reason effectively and efficiently are central to their overall performance. Reasoning, particularly multi-step or compositional reasoning, poses challenges related to supervision, optimization stability, generalization, and computational efficiency. In this section, we explore curriculum learning, supervised vs [42]. self-supervised training, intermediate supervision, step-by-step reasoning supervision, and techniques such as distillation and reinforcement learning that enhance reasoning efficiency. A key strategy for improving reasoning efficiency is curriculum learning, wherein training data is presented in a structured progression from simple to complex examples. This pedagogical approach, inspired by human learning, facilitates more stable convergence and better generalization in LLMs. Formally, let $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k\}$ be a partitioned dataset where $\mathcal{D}_i$ contains tasks of difficulty i, with $i < j \Rightarrow \text{difficulty}(\mathcal{D}_i) < \text{difficulty}(\mathcal{D}_j)$ [43]. The model is trained iteratively on $\mathcal{D}_1$, then $\mathcal{D}_2$, etc., to minimize the empirical loss:

$$\mathcal{L}(\theta) = \sum_{i=1}^{k} \mathbb{E}_{(x,y)\in\mathcal{D}_i}[\ell(f_\theta(x), y)][44].$$

This results in improved reasoning performance on higher-difficulty tasks due to gradual complexity exposure. Supervised learning remains the dominant training method for fine-tuning LLMs for reasoning, particularly when using datasets like GSM8K, HotpotQA, and ProofWriter that contain structured input-output pairs [45]. However, labeled reasoning traces are expensive to obtain. Self-supervised methods, including masked language modeling (MLM), causal language modeling (CLM), and denoising objectives, offer scalability. Yet, they typically lack explicit reasoning supervision [**?**]. To compensate, auxiliary objectives have been introduced to enforce structure in the model's internal computation. For example, chain-of-thought (CoT) prompting and supervision require the model to generate intermediate reasoning steps:

$$\text{Input: } x \quad \Rightarrow \quad \text{Output: } (s_1, s_2, \ldots, s_k, y),$$

where $(s_1, \ldots, s_k)$ are intermediate reasoning steps and y is the final answer [46]. The model is trained with a multi-step cross-entropy loss:

$$\mathcal{L}_{\text{CoT}} = \sum_{t=1}^{k+1} \text{CE}(f_\theta(x)^{(t)}, s_t),$$

where $s_{k+1} = y$ [47]. This promotes decomposition of complex problems into smaller logical steps. Step-by-step supervision has been shown to improve not only the final accuracy but also the interpretability and controllability of model outputs. However, it increases training cost [48]. To mitigate this, a compromise is achieved using intermediate reward-based optimization, where correctness of intermediate steps is rewarded but not strictly enforced. Distillation techniques have also been used to compress complex reasoning into smaller, more efficient student models. Given a teacher model f_T and student model f_S, the distillation loss combines standard supervised loss $\mathcal{L}_{\text{hard}}$ and soft target loss $\mathcal{L}_{\text{soft}}$ as

$$\mathcal{L}_{\text{distill}} = \lambda\mathcal{L}_{\text{soft}} + (1-\lambda)\mathcal{L}_{\text{hard}},$$

 doi:10.20944/preprints202507.1531.v1

where $\mathcal{L}_{\text{soft}} = \text{KL}(f_S(x) \,\|\, f_T(x))$, and λ balances the trade-off. This strategy enables the student model to mimic reasoning behaviors of the teacher while maintaining reduced model size and faster inference [19]. Another paradigm is reinforcement learning (RL), especially Reinforcement Learning with Human Feedback (RLHF) [**?**], which aligns model outputs with human preferences. For reasoning, RL can be extended to optimize a sequence of decisions, with rewards defined over correctness, step efficiency, and answer compactness. Let s_t be the model's reasoning state at step t, and a_t the reasoning action [49]. The objective is to maximize the expected return:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} r(s_t, a_t)\right],$$

where $r(\cdot)$ measures reasoning utility [50]. RL fine-tuning, although unstable and sample inefficient, has been used successfully in models like InstructGPT to produce more coherent and correct reasoning outputs [51]. In addition, contrastive learning has emerged as a promising self-supervised paradigm to refine representations for reasoning [52]. Given a positive pair (x, x^+) and a set of negatives $\{x^-\}$, the model is trained to maximize the similarity of the positive pair while minimizing similarity to negatives, using a contrastive loss such as NT-Xent:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(x, x^+)/\tau)}{\sum_{x'} \exp(\text{sim}(x, x')/\tau)},$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and τ is a temperature hyperparameter. This objective improves clustering of semantically related reasoning paths in latent space [53]. Recent works have also explored meta-learning approaches to enable fast adaptation to new reasoning tasks with limited examples. Model-Agnostic Meta-Learning (MAML) trains a model on a distribution of reasoning tasks such that it can adapt quickly to a new task with minimal gradient steps [54]. The inner and outer loop optimization are defined respectively as:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta), \quad \theta \leftarrow \theta - \beta \sum_i \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta_i'),$$

where $\mathcal{T}_i$ is a task sampled from the task distribution. Figure 2 illustrates the conceptual differences between standard supervised learning, CoT supervision, and distillation-based reasoning optimization.

Figure 2. Comparison of reasoning training paradigms: direct answer supervision, step-by-step chain-of-thought training, and knowledge distillation from teacher to student model.

In conclusion, efficient reasoning in LLMs not only relies on architecture but also heavily depends on thoughtful training strategies [55]. Techniques such as stepwise supervision, distillation, RL fine-tuning, and curriculum learning contribute significantly to improving reasoning accuracy, sample efficiency, and inference speed. In the following section, we analyze the role of external retrieval and tool augmentation in enhancing reasoning efficiency.

5. Retrieval-Augmented Reasoning and Tool Use

As LLMs scale in size and capability, a persistent limitation remains: the inefficiency of internalizing and processing large quantities of world knowledge within finite context windows and parameters [56]. Retrieval-augmented reasoning (RAR) offers a compelling solution by allowing LLMs to access and incorporate external information dynamically during inference. In parallel, tool-augmented reasoning enables LLMs to delegate complex sub-tasks—such as calculation, symbolic logic, or code execution—to specialized tools [57]. These strategies shift the computational burden away from internal model capacity and toward interaction with structured external systems. The retrieval-augmented framework introduces a modular paradigm comprising three main components: (1) a retriever R,

 doi:10.20944/preprints202507.1531.v1

which selects relevant context $\mathcal{C} \subseteq \mathcal{D}$ from a corpus $\mathcal{D}$, (2) a generator G, typically an LLM, that produces output y conditioned on input x and retrieved content $\mathcal{C}$, and (3) a scorer or ranker for refining the context [58]. Formally, the reasoning objective is redefined as:

$$y = G(x, \mathcal{C}) \quad \text{where} \quad \mathcal{C} = R(x, \mathcal{D}),$$

and R is trained or heuristically designed to optimize retrieval quality. In this setting, LLMs perform reasoning not purely from internal memory but through compositional use of retrieved facts, thereby enabling few-shot generalization with minimal additional training [34]. A key challenge in RAR is the integration of retrieved passages into the reasoning process [49]. Traditional models concatenate retrieved passages with input text, leading to inefficient use of the attention mechanism and input budget. Recent approaches like Fusion-in-Decoder (FiD) [**?**] address this by encoding each retrieved document separately and fusing them at the decoding stage:

$$y = \text{Decoder}(\sum_{i=1}^{k} \text{Encoder}(d_i), x),$$

where $d_i \in \mathcal{C}$ [59]. This preserves document granularity while maintaining scalability. Tool-augmented models extend reasoning capabilities by interfacing with external APIs, calculators, code interpreters, databases, or symbolic engines [60]. These models decompose a reasoning task into a series of actions $a_t \in \mathcal{A}$, where each action invokes a tool or emits a reasoning step. Let $\pi_\theta(a_t|s_t)$ be the policy over actions given state s_t. The model's goal is to maximize task reward by choosing optimal sequences of actions:

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} r(s_t, a_t)\right],$$

where $r(\cdot)$ encodes correctness, efficiency, or tool usage costs. For example, a mathematical reasoning model might first generate an equation, call a calculator tool to compute a result, and then compare or refine the output. These tool-use pipelines are often modeled as programs, with LLMs trained to generate executable code or symbolic expressions [61]. This is formalized as a program induction task: for input x, the model emits a program $p = f_\theta(x)$ such that

$$y = \text{Exec}(p),$$

where Exec is a deterministic interpreter [62]. This method allows the model to offload high-precision computation or symbolic logic to external environments while focusing its capacity on planning and program generation. Hybrid retrieval+tool pipelines integrate these paradigms. For example, in WebGPT [**?**], the model first retrieves documents from the web and then generates responses conditioned on those documents, sometimes invoking citation generators or ranking modules as tools [63]. This interaction creates a dynamic reasoning graph rather than a linear text completion task [64]. Similarly, Toolformer [**?**] learns to invoke tools in-context during self-supervised training, resulting in implicit planning and execution capabilities embedded into the language model. Figure 3 illustrates a typical retrieval-augmented tool reasoning pipeline [65].

 doi:10.20944/preprints202507.1531.v1

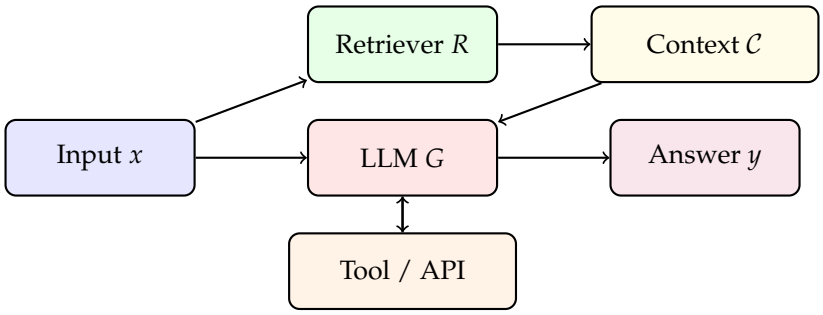

Figure 3. Retrieval- and tool-augmented reasoning pipeline: input is processed through a retriever to obtain relevant context, a language model conditions on both to optionally invoke tools, and finally outputs an answer.

RAR and tool use dramatically improve both reasoning accuracy and efficiency by extending the LLM's effective capacity beyond its parameters. However, these systems introduce new challenges in latency, tool reliability, external API access, and hallucination control [66]. Current research focuses on improving retrieval relevance through differentiable retrievers, learning tool invocation protocols through self-supervised training, and minimizing dependency chains for fast inference [67]. Overall, these methods represent a fundamental shift from monolithic reasoning toward distributed, modular computation paradigms [68]. In the next section, we discuss benchmark datasets and evaluation protocols that are essential for measuring the effectiveness and efficiency of reasoning in LLMs.

6. Benchmarks and Evaluation Protocols for Reasoning Efficiency

The proliferation of reasoning capabilities in large language models necessitates rigorous and standardized evaluation methodologies [69]. Efficient reasoning is not merely a question of accuracy but also involves considerations such as inference cost, latency, reasoning interpretability, step efficiency, and generalization across diverse domains [70]. This section surveys key benchmark datasets, evaluation metrics, and experimental protocols employed in assessing reasoning efficiency in LLMs. Benchmarks for reasoning are broadly categorized into arithmetic reasoning, symbolic logic, multi-hop question answering, mathematical problem solving, and program synthesis [71]. Table 3 summarizes several widely-used datasets, categorized by domain, complexity, and the nature of reasoning required.

 doi:10.20944/preprints202507.1531.v1

Table 3. Representative benchmark datasets for evaluating reasoning in LLMs.

Dataset	Reasoning Type	Domain	Key Characteristics
GSM8K [**?**]	Arithmetic	Elementary Math	Step-by-step numerical reasoning, free-form explanation required
SVAMP [**?**]	Algebraic Comparison	Math Word Problems	Requires semantic understanding and algebraic manipulation
DROP [**?**]	Discrete Reasoning	Reading Comprehension	Multi-hop reasoning with numerical and logical operations
HotpotQA [**?**]	Multi-hop QA	Wikipedia Text	Requires synthesizing multiple facts from distinct documents
ProofWriter [**?**]	Symbolic Logic	Synthetic	Theorem proving and proof step generation under natural language
ARC Challenge [**?**]	General Reasoning	Science Exams	Requires abstract commonsense and factual reasoning
MATH [**?**]	Advanced Math	Competition Math	Covers algebra, calculus, number theory, and geometry
BIG-Bench [**?**]	Diverse Tasks	Mixed Domains	Over 200 reasoning-related subtasks with varying difficulty

A central challenge in evaluation is defining metrics that go beyond accuracy [72]. For example, in multi-step reasoning, models may produce the correct final answer through incorrect or unfaithful intermediate steps—a phenomenon known as spurious reasoning. Hence, metrics such as *step accuracy*, *reasoning fidelity*, and *path consistency* are introduced. Let $s = (s_1, \ldots, s_T)$ denote the reasoning steps and y the final answer. Define the total reasoning loss as:

$$\mathcal{L}_{\text{reason}} = \sum_{t=1}^{T} \ell_{\text{step}}(s_t, s_t^*) + \ell_{\text{final}}(y, y^*),$$

where ℓ_{step} evaluates the semantic correctness of each intermediate step. Efficiency metrics quantify resource use during inference [73]. These include:

- **Inference latency** (in milliseconds or FLOPs): Time or computation required to reach the answer [74].
- **Token economy**: Average number of generated tokens per answer, including intermediate steps.
- **Step length**: Average number of reasoning hops or function calls required.
- **Invocation count**: Number of external tool or retrieval calls made [75].
- **Error locality**: Position in the reasoning chain where the first error occurs.

One advanced metric is **faithfulness**, which measures whether the generated reasoning path truly supports the answer. Faithfulness can be estimated through logical consistency checks or automated theorem provers in symbolic settings. Let $\mathcal{P}(s)$ be the logical implications of reasoning steps s, then the faithfulness condition is:

$$\mathcal{P}(s) \models y \quad \Rightarrow \quad \text{faithful reasoning} citewang2025don.$$

 doi:10.20944/preprints202507.1531.v1

Benchmark protocols increasingly employ *multi-phase evaluation*, where models are tested in both open-book (with access to retrieval or tools) and closed-book (pure internal reasoning) settings. This distinction reveals the model's reliance on memory vs [77]. external knowledge. Furthermore, *adversarial evaluation*—using perturbed, counterfactual, or minimally-different inputs—is gaining traction for measuring robustness [78]. In program synthesis, execution-based accuracy is favored over string match:

$$\text{ExecAcc} = \mathbb{E}_{(x,y)}\left[\mathbf{1}_{\text{Exec}(f_\theta(x))=y}\right].$$

As shown in Figure 4, tool-augmented and retrieval-augmented models outperform closed-book models in accuracy but incur increased inference cost [80]. Efficient reasoning therefore entails careful tradeoffs across multiple dimensions, best evaluated through comprehensive benchmark suites and diverse metrics. In the next section, we will explore future directions and open challenges in building reasoning-efficient LLMs, including neuro-symbolic hybrids, continual learning, and adaptive inference strategies [81].

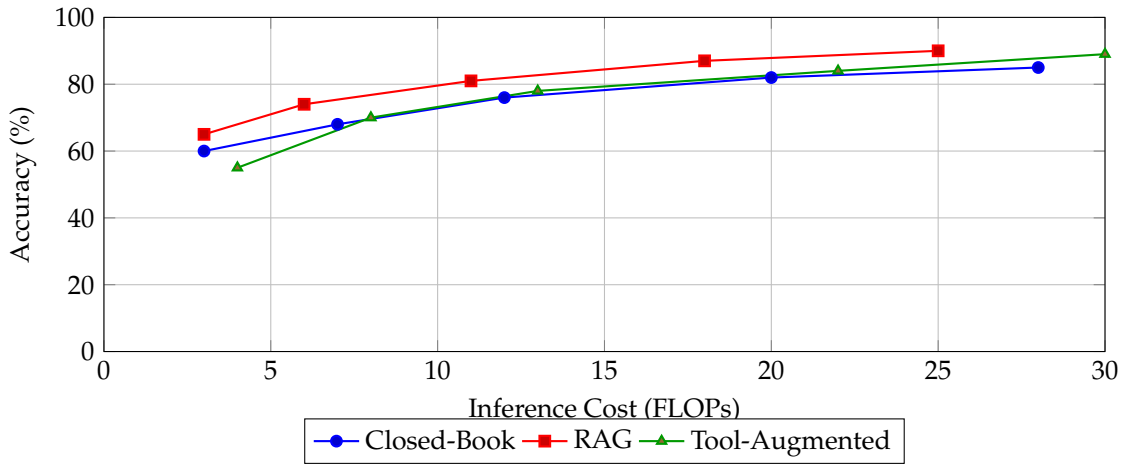

Figure 4. Accuracy vs [79]. Inference Cost tradeoff in different reasoning settings. Tool-augmented models reach higher accuracy but with higher cost.

7. Future Directions and Open Challenges

Despite significant progress in enabling efficient reasoning in large language models, numerous open challenges remain. These stem from both theoretical and engineering limitations, as well as the growing demand for reliability, interpretability, and generalizability in real-world applications. In this section, we outline key future directions and challenges that must be addressed to realize the full potential of reasoning-augmented LLMs [82].

Neuro-Symbolic Integration

A central avenue of research is the unification of neural and symbolic reasoning [83]. While neural models exhibit impressive generalization across fuzzy and high-dimensional inputs, they often struggle with precision, consistency, and compositionality. Symbolic systems, by contrast, excel at logic, abstraction, and rigorous semantics. The integration of these paradigms seeks to leverage their complementary strengths [84]. This could take the form of:

- Embedding symbolic execution environments inside LLM inference loops [85].
- Training models to emit formal programs or logical statements as intermediate representations.
- Using structured symbolic traces as supervision signals for pretraining and finetuning [86].

The challenge lies in making symbolic interfaces differentiable or aligning them effectively with gradient-based learning objectives. One promising direction is neuro-symbolic scaffolding, where

 doi:10.20944/preprints202507.1531.v1

an LLM constructs a structured representation (e.g., abstract syntax tree $\mathcal{T}$), and a symbolic engine verifies or executes it:

$$y = \text{Exec}(\mathcal{T}), \quad \text{where} \quad \mathcal{T} = f_\theta(x).$$

Continual and Lifelong Learning for Reasoning

Current reasoning models are mostly static; once trained, they retain fixed knowledge and reasoning strategies. Lifelong learning—the ability to incorporate new reasoning patterns or tools over time without catastrophic forgetting—is largely unsolved for LLMs. Efficient lifelong reasoning requires mechanisms for:

- Dynamically updating knowledge and procedural templates.
- Memorizing, abstracting, and generalizing from novel task distributions.
- Adapting inference strategies to user feedback or task drift.

A potential solution is modular meta-learning, where the model maintains a library of reasoning modules $\{M_i\}$ and learns to reuse or fine-tune them incrementally based on incoming tasks [87]. Formally, let $\mathcal{T}_t$ denote task t; the objective becomes:

$$\min_\theta \sum_{t=1}^{T} \mathcal{L}_{\mathcal{T}_t}(f_\theta; \{M_i\}),$$

subject to constraints on storage, latency, and update stability [88].

Interpretable and Faithful Reasoning Chains

Interpretability is critical for high-stakes applications such as scientific discovery, healthcare, and legal reasoning [89]. While chain-of-thought prompting has made progress in exposing intermediate reasoning steps, the generated chains are often not faithful—i.e., they do not causally support the final answer. Improving interpretability requires:

- Training models on datasets with verified logical traces and proofs.
- Developing metrics for semantic faithfulness and causal attribution [90].
- Encouraging consistency between different reasoning paths that yield the same answer [91].

A promising idea is to enforce traceability constraints during decoding. For example, models can be trained to maximize the mutual information between intermediate steps s and the output y, conditioned on input x:

$$\max I(s; y \mid x) \quad \text{s.t.} \quad \text{valid}(s, y).$$

Adaptive and Budget-Aware Inference

Most existing models perform reasoning using fixed-length generation and uniform computational budgets [45]. However, efficient reasoning must be adaptive—spending more resources only when necessary. This motivates dynamic computation strategies such as early exiting, adaptive sampling, or budget-constrained planning [92]. Let $\mathcal{B}$ be the total allowed budget (e.g., in FLOPs or time), the goal is:

$$\max_{s,y} \Pr(y \mid x, s) \quad \text{s.t.} \quad \text{Cost}(s, y) \leq \mathcal{B}.$$

Such models require meta-reasoning abilities—i.e., reasoning about the reasoning process—to decide whether to retrieve additional information, call tools, or terminate inference early. Incorporating utility-based decision policies, such as reinforcement learning with budget-sensitive rewards, remains an open research area.

Robustness and Adversarial Reasoning

Finally, the robustness of reasoning is a critical issue [93]. Current models are highly sensitive to input perturbations, adversarial prompts, and misleading intermediate steps [94]. Reasoning chains can be derailed by subtle errors early in the chain [95,96]. Future models must be robust to:

 doi:10.20944/preprints202507.1531.v1

- Semantic paraphrasing or rephrasing of inputs.
- Adversarial distractors introduced in context.
- Noisy or unreliable retrieved or tool-generated content.

Approaches such as ensemble reasoning, redundancy via self-consistency, and verification using auxiliary models may mitigate this brittleness [97]. For example, in self-consistency decoding, a model generates K reasoning paths $\{s^{(i)}\}_{i=1}^{K}$, and aggregates the final answers via majority vote:

$$y = \text{mode}\Big(\{f(s^{(i)})\}_{i=1}^{K}\Big),$$

where $f(s^{(i)})$ maps reasoning chains to final predictions [98].

Toward General-Purpose Reasoning Agents

In the long term, the goal is to build general-purpose reasoning agents—models capable of solving a wide spectrum of cognitive tasks with high efficiency, correctness, and autonomy [26]. Such agents will need to integrate language understanding, formal reasoning, retrieval, tool-use, and interactive planning in a unified architecture [99]. They should be capable of formulating subgoals, using tools strategically, learning new procedures on the fly, and engaging in reflective reasoning. Achieving this vision will require innovations in architecture (e.g., modular reasoning graphs), learning algorithms (e.g., meta-RL for reasoning), training data (e.g., curated symbolic traces), and evaluation frameworks (e.g., simulated agent environments). Despite the challenges, the trajectory of research in efficient reasoning is rapidly accelerating, bringing us closer to LLMs that can not only generate fluent text, but also think with clarity, rigor, and efficiency.

8. Conclusion

The emergence of large language models as general-purpose problem solvers has revitalized interest in machine reasoning—a core component of artificial intelligence that encompasses logical inference, mathematical deduction, strategic planning, and structured decision-making. Yet, scaling reasoning capabilities in LLMs remains a formidable challenge, particularly when considering the dual imperatives of accuracy and efficiency. This survey has aimed to provide a comprehensive overview of recent progress in building efficient reasoning models for LLMs, by examining methodologies, architectural innovations, benchmark ecosystems, and open research frontiers.

We have seen that reasoning in LLMs takes multiple forms—ranging from arithmetic and symbolic logic to tool use and multi-hop natural language inference. Each of these tasks demands different forms of inductive and deductive capabilities, as well as different kinds of internal and external knowledge representations. We categorized the primary approaches to efficient reasoning into three broad classes: prompt-based reasoning (e.g., chain-of-thought and self-consistency), architecture-level enhancements (e.g., modularity, memory, and external tool integration), and training-time strategies (e.g., distillation, imitation learning, and curriculum learning). These approaches are often complementary, and when combined, they can significantly improve the sample-efficiency, latency, and faithfulness of LLM reasoning.

Nonetheless, evaluating reasoning efficiency is nontrivial. We emphasized the importance of going beyond final-answer accuracy, by incorporating step-wise fidelity, interpretability, execution correctness, and cost-aware metrics such as inference time, computation, and invocation budgets. We surveyed a variety of benchmarks—ranging from GSM8K and MATH to DROP and ProofWriter—and illustrated the tradeoffs between reasoning power and computational cost across different configurations, including closed-book inference, retrieval augmentation, and tool use.

Looking ahead, we outlined several promising research directions: neuro-symbolic integration for compositional generalization, lifelong learning frameworks for procedural acquisition, interpretable and faithful reasoning chain supervision, budget-aware dynamic inference, and robustness under adversarial and noisy environments. Perhaps the most ambitious frontier is the construction of general-

 doi:10.20944/preprints202507.1531.v1

purpose reasoning agents that can autonomously plan, reason, and interact with complex external environments across extended timescales.

Ultimately, efficient reasoning in LLMs is not only a matter of engineering; it is a deeply cognitive and epistemological endeavor. It requires us to ask: What does it mean for a machine to reason? How can we evaluate and trust its conclusions? And how do we ensure that the process leading to those conclusions is reliable, transparent, and robust? As models grow in size, scope, and responsibility, answering these questions will be central to aligning LLMs with human values and aspirations in science, education, governance, and beyond.

References

1. Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* **2024**.
2. OpenAI. Learning to Reason with LLMs. urlhttps://openai.com/index/learning-to-reason-with-llms/. Accessed: 15 March 2025.
3. Sakr, C.; Khailany, B. Espace: Dimensionality reduction of activations for model compression. *arXiv preprint arXiv:2410.05437* **2024**.
4. Chen, X.; Zhou, S.; Liang, K.; Liu, X. Distilling Reasoning Ability from Large Language Models with Adaptive Thinking. *arXiv preprint arXiv:2404.09170* **2024**.
5. Pan, Z.; Luo, H.; Li, M.; Liu, H. Chain-of-action: Faithful and multimodal question answering through large language models. *arXiv preprint arXiv:2403.17359* **2024**.
6. Hu, M.; Chen, T.; Chen, Q.; Mu, Y.; Shao, W.; Luo, P. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. *arXiv preprint arXiv:2408.09559* **2024**.
7. Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J.T.; Li, Z.; Weller, A.; Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284* **2023**.
8. Liu, T.; Chen, Z.; Liu, Z.; Tian, M.; Luo, W. Expediting and Elevating Large Language Model Reasoning via Hidden Chain-of-Thought Decoding. *arXiv preprint arXiv:2409.08561* **2024**.
9. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
10. Ma, C.; Zhao, H.; Zhang, J.; He, J.; Kong, L. Non-myopic Generation of Language Models for Reasoning and Planning. *arXiv preprint arXiv:2410.17195* **2024**.
11. Goel, V. *Sketches of thought*; MIT press, 1995.
12. Zhang, Y.; Khalifa, M.; Logeswaran, L.; Kim, J.; Lee, M.; Lee, H.; Wang, L. Small language models need strong verifiers to self-correct reasoning. *arXiv preprint arXiv:2404.17140* **2024**.
13. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, *1*, 3.
14. Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476* **2025**.
15. Yang, W.; Ma, S.; Lin, Y.; Wei, F. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv preprint arXiv:2502.18080* **2025**.
16. Cuadron, A.; Li, D.; Ma, W.; Wang, X.; Wang, Y.; Zhuang, S.; Liu, S.; Schroeder, L.G.; Xia, T.; Mao, H.; et al. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. *arXiv preprint arXiv:2502.08235* **2025**.
17. Besta, M.; Barth, J.; Schreiber, E.; Kubicek, A.; Catarino, A.; Gerstenberger, R.; Nyczyk, P.; Iff, P.; Li, Y.; Houliston, S.; et al. Reasoning Language Models: A Blueprint. *arXiv preprint arXiv:2501.11223* **2025**.
18. Teng, F.; Yu, Z.; Shi, Q.; Zhang, J.; Wu, C.; Luo, Y. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018* **2025**.
19. Feng, T.; Li, Y.; Chenglin, L.; Chen, H.; Yu, F.; Zhang, Y. Teaching Small Language Models Reasoning through Counterfactual Distillation. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 5831–5842.
20. Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems* **2023**, *36*, 34661–34710.
21. Yu, P.; Xu, J.; Weston, J.; Kulikov, I. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023* **2024**.

 doi:10.20944/preprints202507.1531.v1

22. Shen, Y.; Zhang, J.; Huang, J.; Shi, S.; Zhang, W.; Yan, J.; Wang, N.; Wang, K.; Lian, S. DAST: Difficulty-Adaptive Slow-Thinking for Large Reasoning Models. *arXiv preprint arXiv:2503.04472* **2025**.
23. Muennighoff, N.; Yang, Z.; Shi, W.; Li, X.L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; Hashimoto, T. s1: Simple test-time scaling, 2025, [arXiv:cs.CL/2501.19393].
24. Wang, Y.; Liu, Q.; Xu, J.; Liang, T.; Chen, X.; He, Z.; Song, L.; Yu, D.; Li, J.; Zhang, Z.; et al. Thoughts Are All Over the Place: On the Underthinking of o1-Like LLMs. *arXiv preprint arXiv:2501.18585* **2025**.
25. Saparov, A.; He, H. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In Proceedings of the ICLR, 2023.
26. Liu, T.; Guo, Q.; Hu, X.; Jiayang, C.; Zhang, Y.; Qiu, X.; Zhang, Z. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855* **2024**.
27. Gray, R.M.; Neuhoff, D.L. Quantization. *IEEE transactions on information theory* **1998**.
28. Atil, B.; Chittams, A.; Fu, L.; Ture, F.; Xu, L.; Baldwin, B. LLM Stability: A detailed analysis with some surprises. *arXiv preprint arXiv:2408.04667* **2024**.
29. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
30. Zhu, J.; Shen, Y.; Zhao, J.; Zou, A. Path-Consistency: Prefix Enhancement for Efficient Inference in LLM. *arXiv preprint arXiv:2409.01281* **2024**.
31. Xu, J.; Zhou, M.; Liu, W.; Liu, H.; Han, S.; Zhang, D. TwT: Thinking without Tokens by Habitual Reasoning Distillation with Multi-Teachers' Guidance, 2025, [arXiv:cs.CL/2503.24198].
32. LeCun, Y.; Denker, J.; Solla, S. Optimal brain damage. *Advances in neural information processing systems* **1989**, *2*.
33. Fang, J.; Wang, Y.; Wang, R.; Yao, Z.; Wang, K.; Zhang, A.; Wang, X.; Chua, T.S. SafeMLRM: Demystifying Safety in Multi-modal Large Reasoning Models. *arXiv preprint arXiv:2504.08813* **2025**.
34. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* **2022**, *35*, 24824–24837.
35. Ma, W.; He, J.; Snell, C.; Griggs, T.; Min, S.; Zaharia, M. Reasoning Models Can Be Effective Without Thinking. *arXiv preprint arXiv:2504.09858* **2025**.
36. Lee, A.; Che, E.; Peng, T. How Well do LLMs Compress Their Own Chain-of-Thought? A Token Complexity Approach. *arXiv preprint arXiv:2503.01141* **2025**.
37. Valmeekam, K.; Marquez, M.; Sreedharan, S.; Kambhampati, S. On the planning abilities of large language models-a critical investigation. In Proceedings of the NeurIPS, 2023.
38. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* **2017**.
39. Liao, B.; Xu, Y.; Dong, H.; Li, J.; Monz, C.; Savarese, S.; Sahoo, D.; Xiong, C. Reward-Guided Speculative Decoding for Efficient LLM Reasoning. *arXiv preprint arXiv:2501.19324* **2025**.
40. Luo, Y.; Song, Y.; Zhang, X.; Liu, J.; Wang, W.; Chen, G.; Su, W.; Zheng, B. Deconstructing Long Chain-of-Thought: A Structured Reasoning Optimization Framework for Long CoT Distillation. *arXiv preprint arXiv:2503.16385* **2025**.
41. Yan, Y.; Shen, Y.; Liu, Y.; Jiang, J.; Zhang, M.; Shao, J.; Zhuang, Y. InftyThink: Breaking the Length Limits of Long-Context Reasoning in Large Language Models. *arXiv preprint arXiv:2503.06692* **2025**.
42. Cheng, J.; Van Durme, B. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171* **2024**.
43. Yu, Z.; Wu, Y.; Zhao, Y.; Cohan, A.; Zhang, X.P. Z1: Efficient Test-time Scaling with Code, 2025, [arXiv:cs.CL/2504.00810].
44. Yu, Z.; Xu, T.; Jin, D.; Sankararaman, K.A.; He, Y.; Zhou, W.; Zeng, Z.; Helenowski, E.; Zhu, C.; Wang, S.; et al. Think Smarter not Harder: Adaptive Reasoning with Inference Aware Optimization. *arXiv preprint arXiv:2501.17974* **2025**.
45. Srivastava, G.; Cao, S.; Wang, X. Towards Reasoning Ability of Small Language Models. *arXiv preprint arXiv:2502.11569* **2025**.
46. Shi, L.; Zhang, H.; Yao, Y.; Li, Z.; Zhao, H. Keep the cost down: A review on methods to optimize llm's kv-cache consumption. *arXiv preprint arXiv:2407.18003* **2024**.
47. Sui, Y.; He, Y.; Cao, T.; Han, S.; Hooi, B. Meta-Reasoner: Dynamic Guidance for Optimized Inference-time Reasoning in Large Language Models. *arXiv preprint arXiv:2502.19918* **2025**.
48. Arora, D.; Zanette, A. Training Language Models to Reason Efficiently. *arXiv preprint arXiv:2502.04463* **2025**.

 doi:10.20944/preprints202507.1531.v1

49. Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599* **2025**.
50. Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; Steinhardt, J. Measuring Mathematical Problem Solving With the MATH Dataset. In Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021.
51. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* **2024**.
52. Pan, R.; Dai, Y.; Zhang, Z.; Oliaro, G.; Jia, Z.; Netravali, R. SpecReason: Fast and Accurate Inference-Time Compute via Speculative Reasoning. *arXiv preprint arXiv:2504.07891* **2025**.
53. Huang, C.; Huang, L.; Leng, J.; Liu, J.; Huang, J. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031* **2025**.
54. Li, Y.; Yue, X.; Xu, Z.; Jiang, F.; Niu, L.; Lin, B.Y.; Ramasubramanian, B.; Poovendran, R. Small Models Struggle to Learn from Strong Reasoners. *arXiv preprint arXiv:2502.12143* **2025**.
55. Zhao, Y.; Zhou, S.; Zhu, H. Probe then retrieve and reason: Distilling probing and reasoning capabilities into smaller language models. In Proceedings of the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, pp. 13026–13032.
56. Wang, J.; Zhu, S.; Saad-Falcon, J.; Athiwaratkun, B.; Wu, Q.; Wang, J.; Song, S.L.; Zhang, C.; Dhingra, B.; Zou, J. Think Deep, Think Fast: Investigating Efficiency of Verifier-free Inference-time-scaling Methods, 2025, [arXiv:cs.AI/2504.14047].
57. Xiang, K.; Liu, Z.; Jiang, Z.; Nie, Y.; Cai, K.; Yin, Y.; Huang, R.; Fan, H.; Li, H.; Huang, W.; et al. Can Atomic Step Decomposition Enhance the Self-structured Reasoning of Multimodal Large Models? *arXiv preprint arXiv:2503.06252* **2025**.
58. She, J.; Li, Z.; Huang, Z.; Li, Q.; Xu, P.; Li, H.; Ho, Q. Hawkeye:Efficient Reasoning with Model Collaboration, 2025, [arXiv:cs.AI/2504.00424].
59. Song, M.; Zheng, M.; Li, Z.; Yang, W.; Luo, X.; Pan, Y.; Zhang, F. FastCuRL: Curriculum Reinforcement Learning with Progressive Context Extension for Efficient Training R1-like Reasoning Models. *arXiv preprint arXiv:2503.17287* **2025**.
60. Aggarwal, P.; Welleck, S. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. *arXiv preprint arXiv:2503.04697* **2025**.
61. Zhang, N.; Zhang, Y.; Mitra, P.; Zhang, R. When Reasoning Meets Compression: Benchmarking Compressed Large Reasoning Models on Complex Reasoning Tasks. *arXiv preprint arXiv:2504.02010* **2025**.
62. Liu, R.; Sun, Y.; Zhang, M.; Bai, H.; Yu, X.; Yu, T.; Yuan, C.; Hou, L. Quantization Hurts Reasoning? An Empirical Study on Quantized Reasoning Models. *arXiv preprint arXiv:2504.04823* **2025**.
63. Chenglin, L.; Chen, Q.; Li, L.; Wang, C.; Tao, F.; Li, Y.; Chen, Z.; Zhang, Y. Mixed Distillation Helps Smaller Language Models Reason Better. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 1673–1690.
64. Jin, M.; Yu, Q.; Shu, D.; Zhao, H.; Hua, W.; Meng, Y.; Zhang, Y.; Du, M. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925* **2024**.
65. Liu, Z.; Yuan, J.; Jin, H.; Zhong, S.; Xu, Z.; Braverman, V.; Chen, B.; Hu, X. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750* **2024**.
66. Deng, Y.; Choi, Y.; Shieber, S. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838* **2024**.
67. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.V.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
68. Chen, W.; Ma, X.; Wang, X.; Cohen, W.W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588* **2022**.
69. Light, J.; Cheng, W.; Yue, W.; Oyamada, M.; Wang, M.; Paternain, S.; Chen, H. DISC: Dynamic Decomposition Improves LLM Inference Scaling. *arXiv preprint arXiv:2502.16706* **2025**.
70. Wu, Y.; Wang, Y.; Du, T.; Jegelka, S.; Wang, Y. When More is Less: Understanding Chain-of-Thought Length in LLMs. *arXiv preprint arXiv:2502.07266* **2025**.
71. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* **2025**.

 doi:10.20944/preprints202507.1531.v1

72. Ma, X.; Wan, G.; Yu, R.; Fang, G.; Wang, X. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. *arXiv preprint arXiv:2502.09601* **2025**.
73. Meng, Y.; Xia, M.; Chen, D. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems* **2024**, *37*, 124198–124235.
74. Zhang, W.; Nie, S.; Zhang, X.; Zhang, Z.; Liu, T. S1-Bench: A Simple Benchmark for Evaluating System 1 Thinking Capability of Large Reasoning Models. *arXiv preprint arXiv:2504.10368* **2025**.
75. Li, Y.; Niu, L.; Zhang, X.; Liu, K.; Zhu, J.; Kang, Z. E-sparse: Boosting the large language model inference through entropy-based n: M sparsity. *arXiv preprint arXiv:2310.15929* **2023**.
76. Wang, A.; Song, L.; Tian, Y.; Yu, D.; Mi, H.; Duan, X.; Tu, Z.; Su, J.; Yu, D. Don't Get Lost in the Trees: Streamlining LLM Reasoning by Overcoming Tree Search Exploration Pitfalls. *arXiv preprint arXiv:2502.11183* **2025**.
77. Ding, M.; Liu, H.; Fu, Z.; Song, J.; Xie, W.; Zhang, Y. Break the chain: Large language models can be shortcut reasoners. *arXiv preprint arXiv:2406.06580* **2024**.
78. Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.L.; Wu, T.; Gonzalez, J.E.; Kadous, M.W.; Stoica, I. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665* **2024**.
79. Luo, M.; Tan, S.; Wong, J.; Shi, X.; Tang, W.Y.; Roongta, M.; Cai, C.; Luo, J.; Zhang, T.; Li, L.E.; et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog* **2025**.
80. Xu, Y.; Guo, X.; Zeng, Z.; Miao, C. SoftCoT: Soft Chain-of-Thought for Efficient Reasoning with LLMs. *arXiv preprint arXiv:2502.12134* **2025**.
81. Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. Graph of thoughts: Solving elaborate problems with large language models. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 17682–17690.
82. Aytes, S.A.; Baek, J.; Hwang, S.J. Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching. *arXiv preprint arXiv:2503.05179* **2025**.
83. Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; et al. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. *arXiv preprint arXiv:2501.09686* **2025**.
84. Qu, Y.; Yang, M.Y.; Setlur, A.; Tunstall, L.; Beeching, E.E.; Salakhutdinov, R.; Kumar, A. Optimizing Test-Time Compute via Meta Reinforcement Fine-Tuning. *arXiv preprint arXiv:2503.07572* **2025**.
85. Duan, J.; Yu, S.; Tan, H.L.; Zhu, H.; Tan, C. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2022**, *6*, 230–244.
86. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* **2023**, *36*, 11809–11822.
87. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* **2023**.
88. Shen, X.; Wang, Y.; Shi, X.; Wang, Y.; Zhao, P.; Gu, J. Efficient Reasoning with Hidden Thinking. *arXiv preprint arXiv:2501.19201* **2025**.
89. Li, C.; Zhang, C.; Lu, Y.; Zhang, J.; Sun, Q.; Wang, X.; Wei, J.; Wang, G.; Yang, Y.; Shen, H.T. Syzygy of Thoughts: Improving LLM CoT with the Minimal Free Resolution. *arXiv preprint arXiv:2504.09566* **2025**.
90. Magister, L.C.; Mallinson, J.; Adamek, J.; Malmi, E.; Severyn, A. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410* **2022**.
91. Shen, Z.; Yan, H.; Zhang, L.; Hu, Z.; Du, Y.; He, Y. CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation. *arXiv preprint arXiv:2502.21074* **2025**.
92. Xing, S.; Qian, C.; Wang, Y.; Hua, H.; Tian, K.; Zhou, Y.; Tu, Z. Openemma: Open-source multimodal model for end-to-end autonomous driving. In Proceedings of the Proceedings of the Winter Conference on Applications of Computer Vision, 2025, pp. 1001–1009.
93. Liu, R.; Gao, J.; Zhao, J.; Zhang, K.; Li, X.; Qi, B.; Ouyang, W.; Zhou, B. Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. *arXiv preprint arXiv:2502.06703* **2025**.
94. Geiping, J.; McLeish, S.; Jain, N.; Kirchenbauer, J.; Singh, S.; Bartoldson, B.R.; Kailkhura, B.; Bhatele, A.; Goldstein, T. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. *arXiv preprint arXiv:2502.05171* **2025**.
95. Pfau, J.; Merrill, W.; Bowman, S.R. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758* **2024**.

 doi:10.20944/preprints202507.1531.v1

96. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
97. Cuadron, A.; Li, D.; Ma, W.; Wang, X.; Wang, Y.; Zhuang, S.; Liu, S.; Schroeder, L.G.; Xia, T.; Mao, H.; et al. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks, 2025, [arXiv:cs.AI/2502.08235].
98. Li, C.; Chen, Q.; Li, L.; Wang, C.; Li, Y.; Chen, Z.; Zhang, Y. Mixed distillation helps smaller language model better reasoning. *arXiv preprint arXiv:2312.10730* **2023**.
99. Ning, X.; Lin, Z.; Zhou, Z.; Wang, Z.; Yang, H.; Wang, Y. Skeleton-of-thought: Prompting llms for efficient parallel generation. *arXiv preprint arXiv:2307.15337* **2023**.