Evaluating Atom of Thoughts Across Diverse Language Models: A Framework for Enhancing Non-reasoning LLMs performance

Prof. Vaishali Patil[1], Darshan Patil[2], Sanskar Patil[3], Tanuja Patange[4], Roshan Patel[5], Dnyanesh Patil[6], Srinivas Patil[7], Faaiz Khan[8]

[1,2,3,4,5,6,7,8]Department of Engineering, Sciences and Humanities (DESH) Vishwakarma Institute of Technology, Pune, Maharashtra, India

Abstract — Large Language Models (LLMs) have shown remarkable progress, yet enhancing their complex reasoning capabilities, especially during inference (test-time), remains a challenge. Traditional methods often struggle with computational overhead or fail to optimally guide the reasoning process. Atom of Thoughts (AoT) was recently proposed as a novel test-time scaling technique that models reasoning as a Markov process of atomic questions, aiming to improve efficiency and focus. This paper presents the implementation, extension, and critical evaluation of the AoT framework. We developed a comprehensive system featuring multi-model support (integrating OpenAI, Gemini, and OpenRouter models), a user-friendly web interface for experiment configuration and execution, and extended capabilities for mathematical reasoning tasks. Our evaluation across various benchmarks confirms that AoT can enhance the performance of smaller or non-reasoning-focused LLMs. However, our most significant finding reveals a counter-intuitive trend: AoT often *degrades* the performance of several state-of-the-art reasoning models (e.g., DeepSeek R1, Grok 3, GPT o3-mini). We hypothesize this is due to interference with their specialized internal reasoning mechanisms, potentially involving planning and differing internal vs. external thought processes, which conflicts with AoT's structured decomposition prompts.

Notably, highly instruction-following models like Gemini 2.5 Pro Thinking showed minimal performance change, suggesting instruction adherence mitigates this negative interaction. This work provides a practical AoT implementation and offers crucial insights into the interaction between structured reasoning frameworks and the non-reasoning models.

Keywords: Atom of Thoughts, LLM Reasoning, Test-Time Scaling, Large Language Models, Model Evaluation, Reasoning Models, Web Interface, Instruction Following, Gemini, OpenRouter, DeepSeek.

I. INTRODUCTION

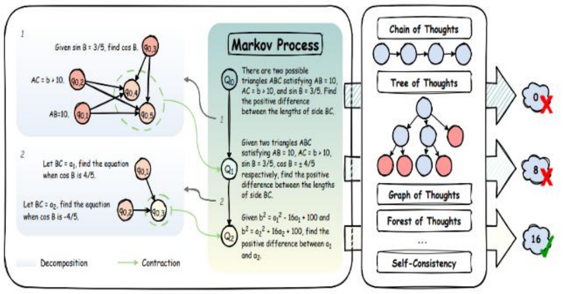
The rapid advancement of Large Language Models (LLMs) has significantly impacted various domains. However, eliciting complex, multi-step reasoning from these models during inference remains an active area of research. Techniques like Chain-of-Thought (CoT) [1] prompt models to generate intermediate steps, while more advanced methods explore complex reasoning structures like trees [2] or graphs [3]. These approaches, while effective, often suffer from accumulating historical context, leading to computational inefficiency and potential interference with the reasoning process itself [4].

Recently, Teng et al. proposed Atom of Thoughts (AoT) [4], a novel test-time scaling framework designed to address these limitations. AoT decomposes complex problems into a sequence of simpler, self-contained "atomic questions". The transition between these question states is modeled as a Markov process, meaning the next state depends only on the current one, thus discarding potentially redundant historical information. This is achieved through an iterative two-phase mechanism: (1) Decomposing the current question into a dependency graph of sub-questions, and (2) Contracting this graph by treating solved independent sub-questions as known conditions and reformulating dependent sub-questions into a new, single atomic question. AoT aims to focus computational resources effectively and enhance reasoning performance, particularly for smaller models.

This project undertook the implementation, extension, and critical evaluation of the AoT framework. Building upon an initial open-source implementation [5], our contributions include:

1. Multi-Model Integration: Extended the framework to support models beyond the original GPT-4o implementation, incorporating native Gemini models and a wide range of models accessible via OpenRouter.
2. Web-Based Experimentation Platform: Developed a user-friendly web interface using Flask and SocketIO, enabling researchers to easily configure API keys, select datasets and models, run experiments, and visualize results in real-time.
3. Mathematical Reasoning Enhancement: Implemented specific prompters and logic tailored for mathematical problem-solving within the AoT framework.
4. Extensive Evaluation & Analysis: Conducted experiments across various models, including state-of-the-art reasoning models, leading to novel insights about AoT's applicability and limitations.

This paper details our implementation, the architecture of the web platform, and presents our key findings, particularly the unexpected performance degradation observed when applying AoT to several specialized reasoning models. We propose a hypothesis grounded in recent research on LLM internals to explain this phenomenon.


1
Given sin B = 3/5, find cos B.
q0,3
AC = b > 10.
q0,2
q0,4
AB=10.
q0,1
q0,5
2
Let BC = a1, find the equation when cos B is 4/5.
q0,1
Let BC = a2, find the equation when cos B is -4/5.
q0,2
q0,3
Decomposition
Contraction
Markov Process
Q0
There are two possible triangles ABC satisfying AB = 10, AC = b > 10, and sin B = 3/5. Find the positive difference between the lengths of side BC.
1
Q1
Given two triangles ABC satisfying AB = 10, AC = b > 10, sin B = 3/5, cos B = ± 4/5 respectively, find the positive difference between the lengths of side BC.
2
Q2
Given b² = a1² - 16a1 + 100 and b² = a2² + 16a2 + 100, find the positive difference between a1 and a2.
Chain of Thoughts
Tree of Thoughts
Graph of Thoughts
Forest of Thoughts
...
Self-Consistency
0 ✗
8 ✗
16 ✓

II. LITERATURE REVIEW

The quest to improve LLM reasoning has spurred various techniques.

Chain-of-Thought (CoT) prompting [1] demonstrated that simply asking models to "think step-by-step" significantly boosts performance on reasoning tasks. Building on this, methods like Tree of Thoughts (ToT) [2] and Graph of Thoughts (GoT) [3] explore multiple reasoning paths simultaneously, allowing for exploration and backtracking, albeit often at increased computational cost.

Other approaches focus on refining generated thoughts. Self-Refine [6] uses the LLM to critique and improve its own outputs iteratively. Chain of Drafts [7] iteratively refines drafts of an answer, focusing on improving quality through revision. These methods highlight the potential for iterative improvement but often maintain significant context.

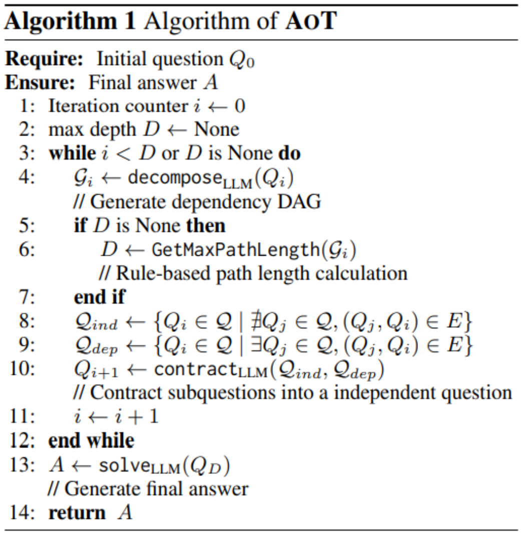
Recent work delves into the scaling laws and computational aspects of reasoning. Research like "Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling" [8] explores how allocating more computational resources during *inference* (test-time), rather than just increasing model size during training, can lead to significant performance gains. Atom of Thoughts [4] fits within this paradigm of test-time scaling, proposing a specific *structure* for utilizing extended computation efficiently by focusing on Markovian state transitions between atomic questions, thereby minimizing redundant context processing compared to traditional CoT or graph-based methods.

Furthermore, understanding the *internal* workings of LLMs during reasoning is becoming crucial. The concept that LLMs might possess internal "world models" [9] suggests their internal state representation might be richer than their textual output implies. A recent blog post by Anthropic on Attribution Graphs [10] provides compelling evidence, particularly for models like DeepSeek R1 and Claude 3 models, that the explicit reasoning steps shown in the output (e.g., CoT) might not accurately reflect the model's internal problem-solving process. They observed models planning steps ahead and potentially using different internal strategies than what they externalize.

Our work directly engages with these concepts. We implement AoT [4] as a method for structured test-time scaling and evaluate its interaction with models hypothesized to have complex internal reasoning processes [10]. Our findings contribute to the understanding of how structured prompting frameworks like AoT align (or conflict) with the internal mechanisms of different types of LLMs, especially specialized reasoning models

III. METHODOLOGY/EXPERIMENTAL

Our project involved implementing the core AoT algorithm and building a platform for its evaluation across diverse models.



Algorithm 1 Algorithm of **AoT**

```
Require: Initial question $Q_0$
Ensure: Final answer $A$
Iteration counter $i \leftarrow 0$
max depth $D \leftarrow$ None
while $i < D$ or $D$ is None do
    $\mathcal{G}_i \leftarrow \texttt{decompose}_{\text{LLM}}(Q_i)$
    // Generate dependency DAG
    if $D$ is None then
        $D \leftarrow \texttt{GetMaxPathLength}(\mathcal{G}_i)$
        // Rule-based path length calculation
    end if
    $\mathcal{Q}_{ind} \leftarrow \{Q_i \in \mathcal{Q} \mid \nexists Q_j \in \mathcal{Q}, (Q_j, Q_i) \in E\}$
    $\mathcal{Q}_{dep} \leftarrow \{Q_i \in \mathcal{Q} \mid \exists Q_j \in \mathcal{Q}, (Q_j, Q_i) \in E\}$
    $Q_{i+1} \leftarrow \texttt{contract}_{\text{LLM}}(\mathcal{Q}_{ind}, \mathcal{Q}_{dep})$
    // Contract subquestions into a independent question
    $i \leftarrow i + 1$
end while
$A \leftarrow \texttt{solve}_{\text{LLM}}(Q_D)$
// Generate final answer
return $A$
```

A. Core Framework and AoT Implementation

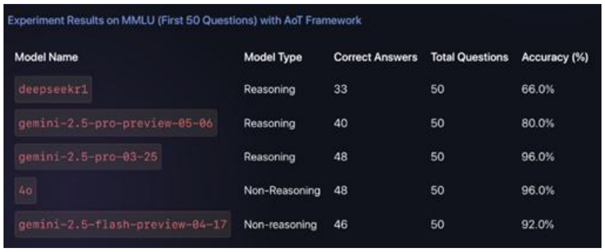
The core logic implements the iterative decompose-contract loop described in AoT Algorithm 1. Key Python components include: an LLM interaction layer (llm.py) handling multi-provider API calls (OpenAI, Gemini, OpenRouter) with rate limiting and retries; task-specific prompters (experiment/prompter/) defining instructions for decomposition (label), contraction (contract), and solving (direct); and the main orchestration logic (experiment/module.py) managing the iterative state transitions and final problem-solving. Mathematical reasoning was specifically addressed by developing tailored prompts within this framework

B. Web-Based Experimentation Platform

To facilitate broader testing, a web interface was developed using Flask for the backend and standard HTML/CSS/JavaScript for the frontend. It allows users to manage API keys for different providers, select datasets and models, configure experiment parameters (e.g., question range), run experiments via background processes, and monitor progress and results in real-time through SocketIO updates. This platform significantly streamlines the evaluation of AoT across various configurations.

IV. RESULTS AND DISCUSSIONS

We conducted experiments using our framework across multiple datasets (MATH, GSM8K, MMLU, HotpotQA) and a diverse set of LLMs, including standard models (GPT-4o-mini), instruction-following models (Gemini 2.5 Pro Thinking, O4 Mini), and specialized reasoning models (DeepSeek R1, Grok 3 Beta, Gemini 2.0 Flash Thinking Experimental).



Experiment Results on MMLU (First 50 Questions) with AoT Framework

Model Name	Model Type	Correct Answers	Total Questions	Accuracy (%)
deepseekr1	Reasoning	33	50	66.0%
gemini-2.5-pro-preview-05-06	Reasoning	40	50	80.0%
gemini-2.5-pro-03-25	Reasoning	48	50	96.0%
4o	Non-Reasoning	48	50	96.0%
gemini-2.5-flash-preview-04-17	Non-reasoning	46	50	92.0%

A. Confirmation of AoT Benefits for Smaller Models

Consistent with the original AoT paper [4], our experiments showed that applying the AoT algorithm generally improved the performance of smaller or non-reasoning-focused models compared to their baseline performance (direct CoT). For instance, gpt-4o-mini demonstrated notable gains on datasets like HotpotQA when using AoT, aligning with the claim that AoT helps focus computation effectively for models that might otherwise struggle with long-context reasoning or complex dependencies. The added math prompter also showed similar benefits for smaller models on MATH/GSM8K benchmarks.

B. Performance Degradation on Reasoning Models

The most striking and unexpected finding emerged when applying AoT to models explicitly designed or fine-tuned for reasoning tasks.

- Negative Impact: Models such as DeepSeek R1 and Grok 3 Beta consistently performed *worse* on reasoning benchmarks when guided by the AoT framework compared to their standard execution. The structured prompts for decomposition and contraction appeared to hinder rather than help their performance.
- Qualitative Observation: Examining the reasoning traces of some models, particularly the Gemini 2.0 Flash Thinking experimental model, revealed potential interference. The model sometimes appeared to generate answers or conclusions first and then attempted to retroactively fit the decomposition/sub-question structure requested by the AoT prompts, rather than following the intended step-by-step simplification process. This suggests a conflict between AoT's explicit structural guidance and the model's internal, possibly pre-trained, reasoning pathways.

C. Hypothesis: Conflict with Internal Reasoning Mechanisms

We hypothesize that the observed performance degradation stems from a fundamental mismatch between AoT's explicit, structured prompting and the internal problem-solving mechanisms of some reasoning-focused LLMs. These models might be fine-tuned extensively on CoT or similar sequential generation patterns, or employ reinforcement learning strategies that optimize for specific output formats. Alternatively, as suggested by recent research [9, 10], these models might possess internal planning capabilities or world models, allowing them to "think ahead" or solve problems using internal representations that differ significantly from the explicit step-by-step output they generate.

The AoT prompts, demanding a specific breakdown (label) and reformulation (contract), might force these models out of their optimized internal strategies. Instead of genuinely decomposing the problem as instructed, they might solve it internally first and then struggle to reverse-engineer the sub-questions and dependencies to match the prompt's requirements, leading to errors or inefficient processing. This aligns with findings from Anthropic [10] suggesting a divergence between internal processes and externalized reasoning steps in models like DeepSeek.

D. Outliers and Instruction Following

Interestingly, the current state-of-the-art models known for strong instruction-following capabilities, such as Gemini 2.5 Pro 03-25 Thinking did *not* exhibit significant performance degradation with AoT. Our results evidently show this model scoring 96%, the same level as the AoT+Non-reasoning models. Their performance was largely similar with or without the AoT framework.

This suggests that superior instruction-following ability might allow these models to adhere more closely to the AoT prompts, even if their internal process differs, thereby avoiding the negative interference seen in other reasoning models. However, the fact that AoT did not provide a significant *boost* to Gemini 2.5 Pro Thinking might imply that its own internal reasoning is already highly optimized, or that AoT's current decomposition/contraction steps aren't sophisticated enough to simplify problems further for such capable models.

Our Hypothesis is strongly supported by Gemini 2.5 Pro Preview 05-06 performing worse than Gemini 2.5 Pro 03-25, as evidently Gemini 2.5 Pro Preview 05-06 is worse at instructions following than 03-25, Google launched it with more coding capabilities, and it has been shown that more coding capabilities align with more reasoning capabilities, but not necessarily better instructions following [11]. The 05-06 preview version still performs better than deepseek r1, but not better than the 03-25 version, because our observations show that the 05-06 preview version is bad at following instructions.

V. FUTURE SCOPE

The rapidly changing LLM field and unpredictable nature of LLM models makes it very hard to be certain about the authenticity of algorithms like AoT on the next generation LLMs. Gemini 2.5 Pro 03-25 and Gemini 2.5 Pro Preview 05-06 are the best examples of this. It's quite possible LLMs are fine-tuned fundamentally with these kinds of techniques or a variant of this to top the benchmarks. Techniques like these work best as a tool for LLM to avoid mistakes, think sequentially and avoid hallucinations. So this framework has its best place as a MCP tool. And there is already such an MCP tool available that uses these techniques whenever necessary. For our future scope we think of a better decomposition strategy, re-evaluating the model responses and curiously – providing LLMs with tools like python interpreter, search, calculator and agentic workflows with these prompts and techniques to see how better or worse they perform.

VI. ACKNOWLEDGMENT

I would like to express my sincere gratitude to Vaishali Patil mam, our guide for the Applied Sciences and Engineering Project, for her continuous support and mentorship throughout this project. Her invaluable guidance has been instrumental in helping me navigate the challenges and intricacies of this work, providing me with the clarity and confidence needed to achieve my objectives.

Additionally, I am deeply grateful to Vishwakarma Institute of Technology, Pune, for providing a conducive environment for learning and research. The resources and support offered by the institute have significantly contributed to the successful completion of this project. I am truly appreciative of the opportunity to work and grow within such a prestigious

institution.

REFERENCE

[1] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2201.11903

[2] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2305.10601

[3] Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., & Hoefler, T. (2023). Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *arXiv preprint arXiv:2308.09687*. https://arxiv.org/abs/2308.09687

[4] Teng, F., Yu, Z., Shi, Q., Zhang, J., Wu, C., & Luo, Y. (2025). Atom of Thoughts for Markov LLM Test-Time Scaling. *arXiv preprint arXiv:2502.12018*. https://arxiv.org/abs/2502.12018

[5] qixucen. (2025). *Atom of Thoughts GitHub Repository*. https://github.com/qixucen/atom

[6] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2303.17651

[7] Xie, S. M., Kim, K., Singh, A., Raghunathan, A., & Narayan, A. (2024). Chain of Drafts: Fusing Chain-of-Thought and Iterative Refinement for Logic Problems. *arXiv preprint arXiv:2405.02216*. https://arxiv.org/abs/2405.02216

[8] Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., & Hashimoto, T. (2025). Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. *Preprint, arXiv:2501.19393*. https://arxiv.org/abs/2501.19393

[9] Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., & Wattenberg, M. (2024). Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2210.13382

[10] Anthropic Interpretability Team. (2025). Attribution Graphs: Methods. *Transformer Circuits*. https://transformer-circuits.pub/2025/attribution-graphs/methods.html

[11] Kuan, Y., Zhang, W., Wang, X., Liu, Z., & Sun, M. (2024). Code to Think, Think to Code: A Survey on Code-Enhanced Reasoning and Reasoning-Driven Code Intelligence in LLMs.https://arxiv.org/abs/2502.19411