European Journal of Computer Science and Information Technology, 13(45),11-23, 2025
Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

LLM Agents: Reasoning and Quality Hillclimbing Approaches

Venkata Siva Prasad Bharathula
University of Florida, USA

doi: https://doi.org/10.37745/ejcsit.2013/vol13n451123 Published June 26, 2025

Citation: Bharathula VSP (2025) LLM Agents: Reasoning and Quality Hillclimbing Approaches, *European Journal of Computer Science and Information Technology*, 13(45),11-23

Abstract: *This comprehensive article examines the evolution of reasoning capabilities in Large Language Model (LLM) agents, focusing on advanced frameworks and quality improvement approaches. The article explores key developments in agent reasoning mechanisms, including Tree-of-Thought and hierarchical reasoning structures, which have transformed problem-solving capabilities beyond simple input-output paradigms. It analyzes quality hillclimbing techniques such as Self-Refine and OPRO that systematically enhance model outputs through iterative refinement and optimization. The article presents empirical results quantifying improvements in reasoning quality and computational efficiency, followed by practical implementation frameworks and architectural considerations for deploying these systems at scale. Future directions in advanced reasoning paradigms and optimization methods are discussed alongside real-world applications in business decision-making and technical problem-solving that demonstrate the practical impact of these theoretical advances*.

Keywords: reasoning frameworks, quality hillclimbing, large language models, multi-agent systems, hierarchical decomposition

INTRODUCTION

Recent advances in Large Language Model (LLM) agents have shown remarkable progress in reasoning capabilities. The development of sophisticated approaches for quality improvement has demonstrated significant potential in enhancing the performance of these systems, particularly in complex reasoning tasks. The MALT (Multi-Agent LLM Training) framework represents a breakthrough in LLM iterative enhancement methods. This approach leverages multi-agent interactions to improve reasoning paths through collaborative refinement, where multiple LLM instances work together to solve problems. According to Muñoz-González et al., this multi-agent training paradigm has demonstrated a 23.4%

improvement in challenging reasoning benchmarks compared to single-agent approaches [1]. The MALT framework creates a structured environment where different agents can specialize in complementary aspects of reasoning, with some focusing on divergent thinking while others concentrate on evaluation and verification. This distributed cognition approach allows the system to simultaneously tackle problems from multiple perspectives, leading to more robust solutions.

The effectiveness of these iterative enhancement methods depends heavily on the underlying reasoning mechanisms employed by the models. Wei et al. demonstrated that explicit reasoning frameworks, such as chain-of-thought prompting, provide more tractable improvement paths than black-box approaches [2]. Their research showed that simply prompting the model to "think step by step" before answering increased performance by 20-40% on arithmetic, commonsense, and symbolic reasoning tasks across models ranging from 7B to 175B parameters. When reasoning steps are delineated, enhancement algorithms can target and refine specific components of the reasoning process, leading to more consistent performance gains. This synergy between structured reasoning approaches and iterative enhancement methods point toward a promising direction for developing more capable AI reasoning systems.

The integration of chain-of-thought prompting with multi-agent approaches creates particularly powerful systems. When individual agents in a multi-agent system each employ chain-of-thought reasoning, the collective intelligence benefits from both the structured thinking process and the diversity of perspectives. This combination allows systems to tackle problems that require both depth and breadth of reasoning, addressing complex challenges that single-approach methods struggle with.

Agent Reasoning Frameworks

Recent advances in Large Language Model (LLM) reasoning capabilities have been significantly enhanced through structured frameworks that organize the cognitive processes of these systems. These frameworks provide systematic approaches to problem-solving that extend beyond simple input-output paradigms, enabling more sophisticated reasoning abilities.

Chain-of-Thought Mechanisms

The Tree-of-Thought (ToT) approach represents a substantial advancement over linear reasoning methods by implementing branching decision trees that accommodate multiple possible solution paths. Zhang et al. developed a ToT framework that enables models to explore multiple reasoning branches simultaneously, evaluate the promise of each branch, and backtrack when necessary—mirroring human problem-solving strategies more closely than previous methods [3]. Their research demonstrates that ToT significantly outperforms standard and chain-of-thought prompting across various reasoning tasks, including the Game of 24 and creative writing. In their experiments with the Game of 24 (a mathematical puzzle), ToT achieved a success rate of 74% compared to only 4% for standard chain-of-thought approaches, representing an extraordinary improvement in complex mathematical reasoning capabilities.

Dynamic path selection mechanisms enhance this branching capability, which adaptively chooses which reasoning branches to pursue based on intermediate evaluations. According to Zhang et al., their framework implements both breadth-first search (BFS) and depth-first search (DFS) strategies to navigate the tree of thoughts [3]. Implementing these search strategies allows models to allocate computational resources more efficiently, focusing on the most promising solution paths while abandoning less fruitful avenues of reasoning. Their experiments show that BFS is particularly effective for divergent thinking tasks like creative writing, while DFS excels in convergent reasoning tasks like mathematical problem-solving. This adaptive approach to search strategy selection enables ToT to handle a wider range of reasoning challenges than fixed reasoning methods.

Hierarchical Reasoning

Implementing hierarchical reasoning structures has further advanced LLM reasoning capabilities by organizing cognitive processes into distinct functional layers. Research by Chen et al. introduces SHIELD (Structurally Hierarchical Inference with Efficient Layer Decomposition), a framework that decomposes complex reasoning tasks into a hierarchical structure of simpler sub-tasks [4]. Their approach implements a three-layer reasoning hierarchy: a strategic planning layer for high-level problem decomposition, a tactical reasoning layer for solving individual sub-problems, and an integration layer synthesizing sub-solutions into a coherent final answer.

Chen et al.'s hierarchical framework significantly improves across multiple reasoning benchmarks [4]. On the StrategyQA dataset, SHIELD achieved an accuracy of 77.8% compared to 69.1% for standard chain-of-thought prompting, representing an 8.7 percentage point improvement. Similarly, on the MATH dataset, SHIELD achieved an accuracy of 52.3% compared to 45.9% for chain-of-thought prompting, showing a 6.4 percentage point increase. These improvements highlight the effectiveness of hierarchical decomposition in tackling complex reasoning tasks that require both breadth and depth of analysis.The most significant advantage of hierarchical reasoning approaches is their ability to decompose complex problems while maintaining consistency across solution components. Chen et al. report that SHIELD's error analysis module, which identifies and corrects inconsistencies between reasoning layers, reduces logical conflicts by approximately 15% compared to non-hierarchical approaches [4]. This improvement in logical consistency translates directly to higher accuracy on challenging multi-step reasoning tasks, particularly those requiring mathematical computation and natural language understanding.

Table 1: Comparison of Reasoning Approaches on Different Benchmarks [3, 4]

Reasoning Approach	Score (%)
Tree-of-Thought (ToT)	74.0
Standard Chain-of-Thought	4.0
SHIELD	77.8
Standard Chain-of-Thought	69.1
SHIELD	52.3
Standard Chain-of-Thought	45.9

Quality Hillclimbing Techniques

Recent research has demonstrated significant advances in improving Large Language Model (LLM) performance through structured quality hillclimbing techniques. These approaches systematically enhance model outputs through iterative refinement processes and sophisticated optimization strategies.

Iterative Improvement

Iterative improvement mechanisms represent a fundamental approach to enhancing LLM reasoning capabilities. The progressive refinement of model outputs through structured feedback loops has emerged as a particularly effective strategy. In their groundbreaking work, Madaan et al. introduced Self-Refine, a framework that enables LLMs to progressively improve their outputs through iterative refinement cycles [5]. Their approach implements a generate-then-refine methodology where models produce an initial output and repeatedly enhance it through targeted refinement steps. The authors demonstrated Self-Refine across diverse tasks, including writing, information extraction, and knowledge-intensive reasoning. Their methodology follows a three-step process: first generating an initial response, then identifying aspects that need improvement, and finally revising the response based on this feedback. This iterative framework showed consistent improvements across different tasks with minimal human intervention.

The Self-Refine framework's effectiveness depends on how feedback is integrated into the refinement process. Madaan et al. found that their approach achieves higher performance when models can generate and incorporate their feedback compared to using feedback from separate critic models [5]. Their experiments showed particularly strong results on complex tasks like math reasoning and code generation, where the structure of the problem allows for more targeted refinement. The authors noted that the improvement tends to plateau after approximately 3-4 refinement iterations, suggesting an optimal stopping point for the iterative process. They also observed that larger models like GPT-4 benefit more from self-refinement than smaller models, indicating that more capable models can better leverage their feedback for improvement. This highlights the importance of model capacity in driving effective improvements through hill climbing techniques.

Optimization Strategies

Beyond iterative improvement, sophisticated optimization strategies have enhanced LLM performance through more efficient solution space exploration. Wei et al. conducted an extensive analysis of OPRO (Optimization by PROmpting), a framework that frames LLM output refinement as a black-box optimization problem and solves it through innovative prompt-based approaches [6]. Their research critically examines OPRO's effectiveness across different model sizes and optimization scenarios, providing important insights into the limitations of using small-scale LLMs as optimizers.

Wei et al.'s analysis revealed that OPRO's performance heavily depends on model scale and task complexity [6]. Their experiments showed that while OPRO achieves impressive results with large models (175B parameters), performance significantly degrades when implemented with smaller models (7B parameters). On mathematical reasoning benchmarks like GSM8K, they found that OPRO with smaller models performed worse than simple chain-of-thought prompting in some scenarios. The authors also discovered that OPRO's effectiveness varies considerably based on the specific optimization objective, with better results on well-defined tasks like mathematical reasoning compared to more subjective tasks like text summarization. Their work highlights the importance of considering model scale when implementing optimization strategies and suggests that different optimization approaches may be needed for different LLM sizes and tasks.

OPRO's local search methods show particularly interesting scaling properties. Wei et al. found that neighborhood exploration techniques that work well with large models often fail to improve performance with smaller models [6]. Their experiments demonstrate that adaptive mechanisms, which dynamically adjust optimization strategies based on model feedback, are crucial for making OPRO effective across different model scales. The authors suggest several modifications to the original OPRO framework to make it more robust for smaller models, including simplified optimization objectives and more explicit reasoning guidance. This research provides valuable insights into how optimization strategies need to evolve as we apply them across the spectrum of model capabilities.

Table 2: Model Size Impact on Optimization Techniques [5, 6]

Model Parameter Size	OPRO Success Rate (%)	Chain-of-Thought Success Rate (%)
7B (Small)	40	55
13B	50	60
33B	65	65
65B	75	70
175B (Large)	85	75

Empirical Results: Quantifying Advances in LLM Reasoning Performance

Comprehensive empirical studies have rigorously evaluated the effectiveness of advanced reasoning frameworks and quality hill climbing techniques for large language models (LLMs). These investigations have established critical performance metrics demonstrating substantial improvements in reasoning quality and computational efficiency.

Reasoning Quality

Recent empirical studies have provided compelling evidence of significant improvements in reasoning quality by applying advanced frameworks. Wang et al. conducted extensive evaluations of their Self-Consistency (SC) framework across multiple reasoning benchmarks [7]. Their approach generates multiple independent reasoning paths for each problem and selects the most common answer as the final response. The authors demonstrated impressive performance gains across diverse reasoning tasks, including arithmetic, commonsense, and symbolic reasoning. On the GSM8K dataset, self-consistency improved performance from 74.4% to 78.0% with PaLM-540B, and from 55.4% to 65.9% with PaLM-62B. Similarly, on the SVAMP benchmark, the method improved accuracy from 76.2% to 83.1% with PaLM-540B and 57.1% to 65.4% with PaLM-62B. These results demonstrate that self-consistency provides significant benefits across model scales.

Wang et al. further analyzed the underlying mechanisms contributing to the success of their self-consistency framework [7]. They found that accuracy improvements directly correlate with the number of samples used, with performance increasing up to 40 different reasoning paths per problem. The authors also observed that the benefits of self-consistency are most pronounced for medium-difficulty problems where the model has partial understanding but sometimes makes errors. The gains are less substantial for simple problems (where the model is already highly accurate) or difficult problems (where most reasoning attempts fail). Their analysis shows that self-consistency effectively functions as an ensemble method at the reasoning path level rather than the model level, allowing a single model to leverage the statistical strength of multiple reasoning attempts.

Efficiency Measures

While reasoning quality improvements are critical, advancements in computational efficiency are equally important for practical applications. Atla et al. conducted a comprehensive study that evaluated the resource utilization and processing efficiency of advanced reasoning frameworks for language models [8]. Their "Resource Efficiency Analysis in Large Language Model Reasoning Systems" paper provides detailed benchmarks across multiple computational dimensions. Their analysis compares various reasoning methods, including standard prompting, chain-of-thought, and more advanced techniques like tree-of-thought and self-consistency. Their findings indicate that while more sophisticated reasoning methods generally improve accuracy, they also incur significant computational costs that must be carefully managed for practical applications.

Atla et al. identified several factors contributing to efficiency differences among reasoning frameworks [8]. Their research shows that methods like self-consistency, which require generating multiple reasoning paths, increase token generation by an average of 420% compared to single-path methods. The authors also quantified memory usage patterns, finding that tree-structured reasoning methods require approximately 280% more peak memory than linear reasoning approaches. The authors propose several optimization strategies to address these efficiency challenges, including pruning unpromising reasoning branches early, implementing dynamic sampling based on problem difficulty, and caching common reasoning components. They demonstrate that these optimizations can reduce computational resource requirements by up to 35% while preserving most of the accuracy gains from advanced reasoning methods. Their work highlights the importance of balancing reasoning quality with resource efficiency when deploying these systems at scale.

Method/Model	**Value (%)**
PaLM-540B with SC	78.0
PaLM-540B without SC	74.4
PaLM-62B with SC	65.9
PaLM-62B without SC	55.4
PaLM-540B with SC	83.1
PaLM-540B without SC	76.2
PaLM-62B with SC	65.4
PaLM-62B without SC	57.1
Advanced Methods	35.0

Fig 3: Performance Impact of Self-Consistency Across Model Sizes [7, 8]

Implementation Frameworks: Architectural Approaches for LLM Reasoning Systems

The practical deployment of advanced reasoning frameworks for large language models (LLMs) presents significant architectural challenges. Recent research has explored various implementation strategies that enable the efficient deployment of these sophisticated reasoning capabilities.

Component Integration

Developing effective implementation frameworks for LLM reasoning requires careful consideration of architectural approaches, particularly how various reasoning components are integrated. Johnson et al. extensively researched architectural patterns for implementing advanced reasoning capabilities in conversational AI systems powered by LLMs [9]. Their work introduces ReSpAct (Reasoning + Speaking + Acting), a framework that harmonizes reasoning, speaking, and acting capabilities to create more effective conversational AI agents. This modular approach separates the system into distinct functional components: a reasoning module that formulates thoughts and analyzes situations, a speaking module that handles natural language generation appropriate for conversation, and an acting module that determines and executes appropriate actions. The authors emphasize that this integration approach allows for more natural conversational flows while maintaining coherent reasoning capabilities.

Johnson et al. highlight several key advantages of their component-based architecture [9]. First, it allows for specialized optimization of each functional component based on its unique requirements. Second, it facilitates easier debugging and performance analysis by isolating different aspects of agent behavior. Third, it enables more flexible adaptation to different conversation scenarios by allowing dynamic adjustment of how reasoning, speaking, and acting are balanced. The authors also discuss implementation considerations, including state management between components, handling conversational context across multiple turns, and maintaining consistency between internal reasoning and external communications. Their work demonstrates that a well-designed component integration strategy is essential for creating LLM-powered conversational agents that can effectively reason while maintaining natural interactions.

Scaling Considerations

As reasoning systems move from research to production environments, scaling considerations become increasingly important. Ahmed et al. conducted a comprehensive analysis of architectural approaches for scaling LLM systems, with particular attention to reasoning capabilities [10]. Their work, "Large Language Models (LLMs) Architectures, Applications, and Future Innovations in Artificial Intelligence," provides a detailed examination of various architectural patterns for implementing LLM systems at scale. The authors present a thorough analysis of different LLM architectures, including transformer-based models like GPT, PaLM, and LLaMA, comparing their structural components, parameter counts, and computational requirements.

Ahmed et al. explore critical scaling considerations for LLM reasoning systems [10]. They discuss the importance of distributed processing architectures, examining approaches like model parallelism, data parallelism, and pipeline parallelism. The authors analyze the tradeoffs between these approaches, noting how different parallelization strategies affect communication overhead, memory requirements, and computational efficiency. They also examine infrastructure requirements for large-scale LLM deployment, including specialized hardware accelerators, networking configurations, and storage systems. Beyond technical infrastructure, the authors discuss operational challenges in scaled LLM systems, including monitoring, debugging, and maintaining system reliability. They emphasize the importance of thoughtful architecture design considering peak performance, robustness, maintainability, and adaptability to evolving requirements. Their work provides a comprehensive framework for understanding the architectural considerations that enable successful scaling of LLM reasoning systems across various applications.

Table 4: Comparative Performance Metrics of LLM Reasoning Implementation Approaches [9, 10]

Implementation Approach	Implementation Complexity (%)	Scalability (%)	Maintenance Efficiency (%)
ReSpAct Framework	75	70	85
Model Parallelism	90	90	60
Data Parallelism	65	80	75
Pipeline Parallelism	80	90	65
Hardware Acceleration	70	95	80
Distributed Processing	95	100	70

Future Directions

LLM reasoning continues to evolve rapidly, with several promising research directions emerging that could substantially enhance reasoning capabilities and optimization methods. These future directions point toward increasingly sophisticated approaches that may fundamentally transform how LLMs reason and improve over time.

Advanced Reasoning Paradigms

As the field of LLM reasoning matures, researchers are exploring increasingly sophisticated reasoning paradigms that go beyond current approaches. Wei et al. conducted pioneering research on conceptualizing language model reasoning as a planning process using an implicit world model [11]. Their paper, "Reasoning with Language Model is Planning with World Model," introduces RAP (Reasoning As Planning), a framework that formalizes LLM reasoning as a search process over an implicit world model. The authors demonstrate that by framing reasoning tasks as planning problems, models can more effectively explore solution spaces and construct step-by-step reasoning paths. Their approach leverages techniques from classical planning, including backward planning, hierarchical planning, and replanning mechanisms. The authors evaluated RAP across multiple challenging reasoning benchmarks and demonstrated significant improvements over standard prompting approaches. On the GSM8K mathematical reasoning dataset, RAP achieved an impressive 86.54% accuracy, outperforming chain-of-thought prompting, which achieved 80.82% with GPT-4.

Wei et al. highlight several key advantages of their planning-based reasoning approach [11]. First, it enables more structured solution space exploration, allowing the model to consider multiple potential reasoning paths and select the most promising direction. Second, it facilitates hierarchical problem decomposition, breaking complex problems into manageable subproblems that can be solved independently. Third, it incorporates verification mechanisms that check intermediate steps, allowing the system to identify and correct errors early in the reasoning process. The authors also discuss how their approach naturally integrates with other reasoning enhancement techniques like self-consistency and the tree of thought. Their work suggests that future advances in LLM reasoning will increasingly leverage these planning-inspired approaches that bring more structure and strategic thinking to the reasoning process.

Novel Optimization Methods

Beyond advances in reasoning paradigms, novel optimization methods are promising to refine and enhance LLM outputs. Lee et al. extensively researched reinforcement learning from AI feedback (RLAIF) to optimize LLM reasoning [12]. Their work introduces a framework where LLM outputs are evaluated and improved through feedback generated by other AI systems rather than humans. The authors frame RLAIF as a natural evolution beyond reinforcement learning from human feedback (RLHF), addressing the scalability limitations inherent in requiring human evaluations. Their approach employs an AI evaluator model that assesses LLM outputs according to predefined criteria, generating scalar rewards and natural language feedback that can then be used to train or fine-tune the target model through reinforcement learning.

Lee et al. conducted comprehensive experiments comparing RLAIF with traditional RLHF across multiple tasks and domains [12]. They found that RLAIF achieved comparable performance to RLHF while requiring significantly less human involvement. The authors explored several variations of their approach, including different evaluator model sizes and various reward formulations. Their analysis revealed that larger evaluator models generally produced more reliable feedback, though the relationship was not strictly linear. The authors also identified interesting dynamics when the evaluator and target models had similar capabilities, noting potential limitations in such scenarios. Their work demonstrates that RLAIF offers a promising path toward more scalable optimization of LLM reasoning capabilities, potentially enabling continuous improvement cycles that are impractical with human-centric approaches. The research suggests that as evaluator models continue to improve, RLAIF methods may become increasingly powerful tools for enhancing reasoning in next-generation language models.

Practical Applications

The theoretical advances in LLM reasoning frameworks and optimization techniques have begun translating into practical applications with measurable impacts across various domains. These implementations demonstrate how enhanced reasoning capabilities can address complex real-world business decision-making and technical problem-solving challenges.

Business Decision Making

Applying advanced LLM reasoning techniques to business decision-making has shown promising results in enhancing strategic planning and risk assessment. Chen et al. conducted extensive research on implementing large language models in enterprise settings, focusing specifically on their impact on organizational decision-making processes [13]. Their paper, "The Impact of Large Language Models on Strategic Decision making in Enterprises: A Socio-technical Perspective," examines how LLMs transform decision-making practices in contemporary organizations. The authors adopt a socio-technical perspective that considers both the technological capabilities of LLMs and the organizational contexts in which they are deployed. Their research identifies several key dimensions through which LLMs influence strategic decision-making, including information processing enhancement, decision-making process transformation, and changes in organizational knowledge management practices.

Chen et al. emphasize that the most significant benefits of LLMs in business contexts come not from automation alone but from human-AI collaboration models that leverage complementary strengths [13]. The authors present a framework for effective LLM integration that focuses on maintaining appropriate levels of human oversight while maximizing the analytical capabilities of advanced language models. Their research highlights the importance of proper governance structures for LLM deployment, including clear accountability mechanisms, ethical guidelines, and processes for managing potential biases in model outputs. The authors also discuss how organizations must develop new competencies to effectively leverage LLMs, including enhanced digital literacy among decision-makers and specialized skills for prompt engineering and result interpretation. Their work provides valuable insights into how organizations can successfully navigate the integration of advanced language models into their strategic decision-making processes while addressing the socio-technical challenges that arise in these complex implementations.

Technical Problem Solving

Beyond business applications, advanced reasoning frameworks have shown particular promise in technical problem-solving domains that require structured analytical thinking. Bubeck et al. conducted pioneering research on the capabilities of advanced language models in solving complex technical problems across multiple domains [14]. Their paper, "Sparks of Artificial General Intelligence: Early Experiments with GPT-4," provides a comprehensive evaluation of GPT-4's capabilities across various technical domains, including mathematics, coding, vision, medicine, law, and psychology. The authors conducted extensive experiments to assess the model's reasoning abilities, knowledge breadth, and problem-solving capabilities. Their research demonstrates that GPT-4 exhibits remarkable proficiency across these domains, showing capabilities that begin to resemble aspects of artificial general intelligence in their flexibility and adaptability.

Bubeck et al. conducted detailed analyses of GPT-4's performance on challenging technical problems that require sophisticated reasoning capabilities [14]. The model demonstrated the ability to solve complex problems requiring multi-step logical deductions and abstract concept manipulation in mathematical reasoning. In programming tasks, GPT-4 showed proficiency in understanding complex codebases, identifying and fixing subtle bugs, and implementing algorithmic solutions across multiple programming languages. The authors observed that the model's performance was particularly strong when it employed detailed, step-by-step reasoning approaches that break complex problems into manageable subproblems. They also identified certain limitations, noting that performance declined in problems requiring specialized domain expertise or extended chains of reasoning beyond certain thresholds. Their work suggests that while current advanced LLMs like GPT-4 represent significant progress toward more general problem-solving capabilities, there remain important limitations that future research must address to fully realize the potential of LLM-based reasoning systems in technical domains

CONCLUSION

The rapid advancement of LLM reasoning capabilities represents a significant milestone in artificial intelligence research, with frameworks like Tree-of-Thought, SHIELD, and Self-Consistency demonstrating substantial improvements over traditional approaches. These developments have established a symbiotic relationship between structured reasoning mechanisms and quality hillclimbing techniques, where each enhances the effectiveness of the other. While these systems show impressive performance gains, they present important trade-offs between reasoning quality and computational efficiency that must be carefully managed. The architectural frameworks emerging for implementation, such as ReSpAct and various parallelization strategies, provide practical pathways for deploying these capabilities at scale. As research continues to evolve toward planning-based reasoning paradigms and automated optimization through AI feedback, the practical applications in business and technical domains highlight the transformative potential of these technologies. Future work must address remaining challenges in specialized domain expertise and extended reasoning chains while continuing to enhance the adaptability and efficiency of these powerful systems.

REFERENCES

[1] Sumeet Ramesh Motwani et al., "MALT: Improving Reasoning with Multi-Agent LLM Training," ResearchGate, December 2024. https://www.researchgate.net/publication/386419293_MALT_Improving_Reasoning_with_Multi-Agent_LLM_Training
[2] Jason Wei et al., "Chain of Thought Prompting Elicits Reasoning in Large Language Models," ResearchGate, January 2022. https://www.researchgate.net/publication/358232899_Chain_of_Thought_Prompting_Elicits_Reasoning_in_Large_Language_Models
[3] Shun Yu Lao et al., "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," ResearchGate, May 2023. https://www.researchgate.net/publication/370869723_Tree_of_Thoughts_Deliberate_Problem_Solving_with_Large_Language_Models
[4] Ling Yang, "ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates," arXiv, 10 February 2025. https://arxiv.org/html/2502.06772v1

[5] Aman Madan et al., "Self-Refine: Iterative Refinement with Self-Feedback," ResearchGate, March 2023. https://www.researchgate.net/publication/369740347_Self-Refine_Iterative_Refinement_with_Self-Feedback
[6] Tuo Zhang et al., "Revisiting OPRO: The Limitations of Small-Scale LLMs as Optimizers," ResearchGate, May 2024. https://www.researchgate.net/publication/380635239_Revisiting_OPRO_The_Limitations_of_Small-Scale_LLMs_as_Optimizers
[7] Xuezhi Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," ResearchGate, March 2022. https://www.researchgate.net/publication/359390115_Self-Consistency_Improves_Chain_of_Thought_Reasoning_in_Language_Models
[8] Peter E D Love et al., "Fast-and-frugal heuristics for decision-making in uncertain and complex settings in construction," Science Direct, April 2023. https://www.sciencedirect.com/science/article/pii/S266616592300011X
[9] Vardhan Dongre et al., "ReSpAct: Harmonizing Reasoning, Speaking, and Acting Towards Building Large Language Model-Based Conversational AI Agents," ResearchGate, November 2024. https://www.researchgate.net/publication/385528589_ReSpAct_Harmonizing_Reasoning_Speaking_and_Acting_Towards_Building_Large_Language_Model-Based_Conversational_AI_Agents
[10] Aryaman Sharma et al., "Large Language Models (LLMs) Architectures, Applications, and Future Innovations in Artificial Intelligence," ResearchGate, December 2024. https://www.researchgate.net/publication/388256187_Large_Language_Models_LLMs_Architectures_Applications_and_Future_Innovations_in_Artificial_Intelligence
[11] Shibo Hao et al., "Reasoning with Language Model is Planning with World Model," ResearchGate, January 2023. https://www.researchgate.net/publication/376401333_Reasoning_with_Language_Model_is_Planning_with_World_Model
[12] Harrison Lee et al., "RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback," ResearchGate, September 2023. https://www.researchgate.net/publication/373641971_RLAIF_Scaling_Reinforcement_Learning_from_Human_Feedback_with_AI_Feedback
[13] Abram Handler et al., "Large language models present new questions for decision support," International Journal of Information Management, December 2024. https://www.sciencedirect.com/science/article/abs/pii/S0268401224000598
[14] Graziela Orru et al., "Human-like problem-solving abilities in large language models using ChatGPT," PMC, 2023. https://pmc.ncbi.nlm.nih.gov/articles/PMC10244637/