Vol. 16 (2026): Proceedings of the 2026 International Conference on AI and Technological Innovation Applications (AITIA 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/aitia.2026286292

Reasoning in Large Language Models: Methods, Challenges, and Future Directions

Zhuangchen Xu*

Tongji University, Shanghai, China

**Corresponding author: Zhuangchen Xu.*

Abstract

In recent years, large language models have developed rapidly and are now widely used in many everyday tasks, such as conversation, information retrieval, and code generation. Although these models can produce fluent and coherent text, their ability to perform reliable reasoning is still limited, especially in tasks that require multiple steps or logical consistency. This paper reviews several methods aimed at improving reasoning in LLMs. It first introduces prompt-based approaches, including Chain-of-Thought, self-consistency, and Auto-CoT, which try to guide models to generate intermediate reasoning steps. It then discusses search-based methods, such as Tree-of-Thought, in which reasoning is treated as a process of exploring multiple paths. In addition, the paper examines the hallucination problem and the use of retrieval-based techniques to improve factual accuracy. Benchmark datasets for evaluating reasoning ability are also briefly discussed. Furthermore, the paper examines recent developments in agent-based reasoning and tool use, in which models can interact with external systems and perform more complex tasks. While these methods show some improvements, several open challenges remain, including issues of reasoning reliability, error accumulation, and evaluation.

Keywords

large language models, reasoning, chain-of-thought, tool-augmented models, AI agents

1. Introduction

In recent years, large language models (LLMs) have developed very rapidly in the field of artificial intelligence. Models such as GPT, DeepSeek, Qwen, and Gemini have been introduced one after another, and their applications have gradually entered everyday life. People now use these models in many different situations, for example, having daily conversations, searching for information, analyzing documents, and even writing or debugging code.

From a technical perspective, LLMs generate text primarily through token prediction. That is, given the previous context, the model predicts the most likely next token step by step. Because of this mechanism, the generated text is often very fluent and coherent. However, fluency does not necessarily mean correctness. Especially when the task involves reasoning, such as solving multi-step mathematical problems or making logical deductions, the model may still produce incorrect answers. In many cases, the model's explanation

looks reasonable, but the conclusion is wrong. This mismatch between fluent generation and actual reasoning ability has become a major issue in current research.

2. Overview of Reasoning Methods

2.1 Prompt-Based Reasoning (Cot, Zero-Shot Cot, Auto-CoT)

To address the limitations of direct input-output generation, researchers proposed the Chain-of-Thought (CoT) approach. The key idea of CoT is to encourage the model to generate intermediate reasoning steps before producing the final answer. Compared with directly outputting an answer, this approach helps the model break complex problems into smaller steps, often leading to better results. One important feature of CoT is that it does not require additional training; instead, it only modifies the prompting strategy. Even so, it can significantly improve performance on many reasoning tasks, especially those requiring multiple steps [1].

Further analysis shows an interesting phenomenon. When the model produces a correct answer, the reasoning process is usually correct. When the answer is incorrect, the reasoning steps are often partially correct rather than completely meaningless. This suggests that the model is not simply guessing, but is actually performing some form of reasoning process[1]. Similar observations have also been reported in later studies on strong LLMs, where reasoning-like behavior appears even without explicit supervision [2].

However, traditional CoT relies on greedy decoding, which selects only one reasoning path based on probability. In practice, many problems can be solved in different ways, and different reasoning paths may lead to different conclusions [3]. To address this, the self-consistency strategy was proposed. Instead of generating a single reasoning path, the model samples multiple paths and aggregates their results. The final answer is usually determined by majority voting. This method improves robustness and reduces the risk of errors caused by a single incorrect reasoning path [4].

In terms of implementation, CoT can be divided into two main forms. The first is Zero-Shot-CoT, which uses simple prompts such as “let’s think step by step” to guide the model to generate reasoning without providing examples [5]. The second is Few-Shot-CoT, which uses manually designed examples that include both reasoning steps and answers. Few-shot methods can achieve better performance when examples are well-constructed, but they require more effort and are often task-specific.

To reduce the dependence on manually designed examples, Auto-CoT was proposed. In this approach, the model automatically generates reasoning examples rather than relying on human-written ones [2]. The method typically involves clustering the input questions, selecting representative samples from each cluster, generating reasoning chains, and then filtering them using simple heuristics. Experimental results show that Auto-CoT can achieve performance comparable to, and in some cases better than, manually constructed CoT.

At the same time, it should be noted that for more advanced models like ChatGPT, adding explicit “step-by-step” instructions does not always improve performance and, in some cases, may even have negative effects. One possible explanation is that these models have already learned similar reasoning patterns during instruction tuning, so additional prompts may become redundant or interfere with the generation process [6].

2.2 Search-Based Reasoning (Tree-of-Thought)

Building on CoT, researchers further proposed Tree-of-Thought (ToT), which extends the idea of reasoning from a single path to multiple paths. Instead of following a single reasoning trajectory, ToT allows the model to explore several possible reasoning paths simultaneously.

During this process, the model can evaluate different intermediate states and decide which path to continue. In addition, ToT introduces mechanisms such as lookahead and backtracking. This means the model can reconsider earlier steps and revise its reasoning if necessary. Compared with CoT, this makes the reasoning process more flexible and closer to how humans might approach complex problems.

Experimental results show that ToT performs better than standard CoT in tasks that require planning, exploration, or non-trivial decision making [3]. However, it also introduces additional computational cost and complexity, which may limit its practical application in some cases.

3. Hallucination

As LLMs become more powerful, they can handle a wider range of tasks, including writing, translation, and even multimodal processing. However, hallucination remains a significant challenge.

A hallucination is a situation in which the model generates information that appears plausible but is actually incorrect. This issue arises because the model predicts the next token based on statistical patterns in the training data, rather than retrieving verified facts. As a result, the model may produce detailed but fabricated information.

For example, when asked about the author of a specific paper, the model might generate a realistic name, publication year, and journal, even if the information is not true. Since the response is fluent and well-structured, users may not notice the error.

Several methods have been proposed to reduce hallucination. One approach is to improve prompting, for example, by asking the model to base its answer on the given information or to explicitly state uncertainty. Another more effective approach is Retrieval-Augmented Generation (RAG), which allows the model to access external knowledge sources such as databases or the internet, thereby improving the accuracy of the generated content [7].

In addition, some recent work attempts to reduce hallucination by introducing verification mechanisms. For example, in mathematical reasoning tasks, models can generate candidate answers and then use separate verification steps to check their correctness [8].

It is also important to note that simply increasing model size does not solve hallucination. A more effective approach is to use human feedback for fine-tuning. In this process, human annotators provide high-quality responses, and model outputs are ranked according to human preference. A reward model is trained, and reinforcement learning methods such as PPO are used to optimize it. This approach, known as RLHF, has been shown to improve alignment with human expectations significantly [9].

Overall, hallucination is not just a minor technical issue. It reflects a deeper limitation of current models, which generate text based on probability rather than factual correctness.

4. Benchmark

Because LLM performance varies across tasks, researchers have proposed numerous benchmarks to evaluate reasoning ability. These benchmarks are generally divided into three categories: symbolic reasoning, knowledge reasoning, and interactive or agent reasoning. This classification has been widely adopted in recent survey work [10].

For example, GSM8K and MATH are commonly used to evaluate mathematical reasoning ability, especially in multi-step problem-solving [11]. MMLU is widely used as a general benchmark for measuring knowledge and reasoning ability across multiple domains [12], while datasets such as StrategyQA focus on commonsense reasoning. In addition, benchmarks such as HotpotQA and ALFWorld are used to evaluate multi-step decision-making and interaction with environments.

Although these benchmarks have played an important role in advancing research, they also have some limitations. Most benchmarks focus only on whether the final answer is correct, without evaluating the reasoning process itself. This makes it difficult to assess whether the model truly understands the problem or arrives at the correct answer by chance.

Moreover, many benchmark tasks are static and simplified, which does not fully reflect real-world applications. In real-world scenarios, tasks often involve dynamic environments, long-term planning, and multiple stages of decision-making.

In practice, benchmarks are used not only to compare models but also as a shared reference for the research community. Without such standards, it would be difficult to evaluate whether a new method truly improves performance. At the same time, benchmarks also influence research directions, since many methods are optimized specifically for commonly used evaluation tasks.

5. Current Challenges

Although LLMs have demonstrated strong capabilities, several challenges remain.

First, the reliability of reasoning is still an open issue. Even with methods such as CoT and self-consistency, models may make errors at intermediate reasoning steps, and these errors are often difficult to detect automatically [1,13].

Second, hallucination remains unresolved. Models may produce confident but incorrect answers, which is closely related to their probabilistic generation mechanism [14].

Third, the integration of external knowledge is not yet stable. While RAG can improve accuracy, issues such as retrieval quality, relevance filtering, and information integration still need further improvement [15].

Finally, long-term reasoning and planning abilities remain limited. Even methods like ToT do not always perform consistently on complex tasks [16].

6. Agent-Based Reasoning

Recent research has shown increasing interest in extending LLMs from passive text generators to more active problem-solving systems. One important direction is agent-based reasoning, where models interact with environments and use tools. This direction has been widely discussed in recent work on LLM-based agents [17].

Compared with standard prompting methods such as CoT, agent-based approaches treat reasoning as a more dynamic process. Instead of generating a fixed sequence of reasoning steps, the model can decide what to do next based on the current state. For example, it may choose to search for information, perform a calculation, or call an external API before continuing the reasoning process. This shift from static generation to interactive reasoning has also been explored in recent agent-oriented frameworks [15].

A typical framework in this direction involves three main components: planning, acting, and observing. First, the model generates a plan or decides the next action. Then, it executes the action, such as querying a database or using a tool. Finally, it observes the result and updates its internal state. This cycle may repeat multiple times until a final answer is produced. In this sense, reasoning becomes a loop rather than a single forward generation process. This type of interaction loop is also reflected in frameworks such as ReAct, which combines reasoning and action in a unified process [15].

This paradigm is closely related to earlier methods such as Tree-of-Thought. In ToT, the model explores multiple reasoning paths and evaluates them. Similarly, in agent-based systems, the model may explore different actions and adjust its behavior based on feedback from the environment [3]. However, agent-based reasoning is often more flexible, since it allows interaction with external systems rather than relying only on internal text generation.

Another important aspect is tool usage. Traditional LLMs are limited by the knowledge encoded in their parameters. Even with large-scale training, they may still lack up-to-date information or precise computational ability. By integrating external tools, these limitations can be partially addressed.

For example, a model may use a calculator for arithmetic operations, a search engine to retrieve recent information, or a database to access structured data. This idea is explored in work such as Toolformer, where the model learns when and how to use tools through its own training process [14]. Instead of relying on manually designed rules, the model can decide whether calling a tool is helpful in a given context. Similar ideas have also been extended to more general tool-learning frameworks [16].

Tool usage is also closely connected to Retrieval-Augmented Generation. While RAG focuses on retrieving relevant documents, tool-augmented systems generalize this idea to a broader set of operations, including computation and interaction with external services [7]. In this sense, RAG can be seen as a specific case within a larger framework of tool use.

However, agent-based reasoning also introduces new challenges. One issue is error accumulation. Since the reasoning process involves multiple steps and interactions, errors in earlier steps may propagate and affect later decisions. Unlike single-step generation, it is more difficult to ensure consistency across the entire process.

Another challenge is decision-making. The model needs to decide not only what to say, but also what action to take. This requires a different type of capability, closer to planning and control rather than pure language generation. In some cases, the model may choose suboptimal actions or fail to use available tools effectively.

In addition, evaluating agent-based systems is more complicated. Traditional benchmarks focus on final answers, but in agent settings, the quality of intermediate actions and the efficiency of the overall process also matter. This makes evaluation more difficult and less standardized.

Despite these challenges, agent-based reasoning is considered a promising direction, and recent survey work also highlights its potential for building more autonomous AI systems [17]. It provides a way to extend LLM capabilities beyond static text generation, making them more suitable for real-world tasks that require interaction, adaptation, and decision-making over time.

7. Future Directions

Looking forward, several directions appear worth further exploration.

First, improving the interpretability of reasoning is important. Instead of focusing only on final answers, future work may pay more attention to intermediate reasoning steps, for example, through process supervision [4].

Second, evaluation methods may need improvement. Future benchmarks should better reflect real-world scenarios, including dynamic environments and multi-stage tasks.

Third, combining LLMs with external tools and agent frameworks is becoming an important trend. By using tools such as search engines, calculators, or databases, models can extend their capabilities beyond text generation [3]. Recent work, such as Toolformer, also shows that models can learn to use external tools in a more autonomous way [14].

In addition, multimodal reasoning may further expand the range of applications, enabling models to process and combine diverse types of information, such as text and images.

Finally, alignment remains a key issue. Although methods like RLHF have improved model behavior, ensuring safety, controllability, and consistency remains an ongoing challenge [8].

8. Conclusion

Overall, LLMs have made significant progress in their reasoning abilities. Compared with earlier models, they are now capable of handling multi-step reasoning and even some level of planning. Methods such as CoT, self-consistency, and ToT have all contributed to these improvements.

However, these methods mainly enhance existing capabilities, rather than solving fundamental problems. Issues such as hallucination and unreliable reasoning still exist. There is also still a gap between benchmark performance and real-world applications.

At the current stage, LLM reasoning is still developing. Some problems have been partially addressed, but not fully solved. Future progress will depend not only on model improvements but also on better evaluation methods, integration with external tools, and more effective human-AI interaction.

References

[1] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35, 24824-24837.

[2] Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. arXiv preprint arXiv:2210.03493.

[3] 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, 36, 11809-11822.

[4] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

[5] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, 22199-22213.

[6] Chen, J., Chen, L., Huang, H., & Zhou, T. (2023). When do you need Chain-of-Thought Prompting for ChatGPT?. arXiv preprint arXiv:2304.03262.

[7] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33, 9459-9474.

[8] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.

[9] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[10] Ferrag, M. A., Tihanyi, N., & Debbah, M. (2025). From llm reasoning to autonomous ai agents: A comprehensive review. arXiv preprint arXiv:2504.19678.

[11] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

[12] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... & Schulman, J. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

[13] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Sharick, E. (2023). Paper Review:'Sparks of Artificial General Intelligence: Early experiments with GPT-4'.

[14] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., ... & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, *36*, 68539-68551.

[15] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., & Cao, Y. (2022, October). React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

[16] Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., ... & Sun, M. (2024). Tool learning with foundation models. *ACM Computing Surveys*, *57*(4), 1-40.

[17] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., ... & Gui, T. (2025). The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, *68*(2), 121101.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access