# A Survey on Enhancing Large Language Models with Symbolic Reasoning

Anonymous ACL submission

## Abstract

Reasoning is one of the fundamental human abilities, central to problem-solving, decision-making, and planning. With the development of large language models (LLMs), significant attention has been given to enhancing and understanding their reasoning capabilities. Most existing works attempt to directly apply LLMs to natural-language-based reasoning. However, due to the inherent semantic ambiguity and grammatical complexity of natural language, LLMs often struggle with complex problems, leading to challenges such as hallucinations and inconsistent reasoning. Therefore, methods for constructing formal language representations, most of which are symbolic languages, have emerged to obtain more reliable solutions recently. In this work, we focus on symbolic reasoning in LLMs and provide a comprehensive review of the related research. This includes examining the applications of symbolic reasoning, types of symbolic languages used, techniques for enhancing LLMs' symbolic reasoning abilities, and benchmarks employed to evaluate their performance. Our goal is to offer a thorough review of symbolic reasoning in LLMs, highlighting key findings and challenges while providing a reference for future research in this area.

## 1 Introduction

Reasoning involves deriving conclusions or solutions from limited information by applying logical analysis, pattern recognition, and knowledge integration. Researchers have found that once large language models (LLMs) reach a certain threshold in terms of parameters and training data, they can exhibit reasoning capabilities (Wei et al., 2022), leading researchers to explore a variety of methods to enhance the reasoning capabilities of LLMs (Zhang et al., 2022; Yao et al., 2024; Ning et al., 2023). However, LLMs encounter substantial challenges in their reasoning processes, most notably the issue of hallucination. This issue becomes especially prominent in complex reasoning tasks, where LLMs may fail to account for all relevant factors, omit key information, or fall into logical traps.

To address these challenges and further enhance the reasoning capabilities of LLMs, researchers have begun to explore an innovative framework that enables LLMs to focus on question comprehension and symbolic representation generation, while delegating the execution of reasoning steps to external solvers (Olausson et al., 2023; Pan et al., 2023; Gao et al., 2023).This approach helps alleviate the limitations of LLMs in reasoning tasks by combining the strengths of both symbolic representation and specialized solvers.

As long as LLMs generate accurate symbolic representations, external solvers can take over and efficiently solve complex reasoning tasks based on these representations. This strategy not only reduces the burden on the language model itself but also fully leverages the professional advantages of the external solver in dealing with specific problems, achieving complementary advantages and collaborative work between the two. Symbolic language, with its precision, interpretability, and logicality, brings significant advantages to the reasoning process. (Olausson et al., 2023; Lyu et al., 2023; Chen et al., 2022) have demonstrated significant improvements in accuracy through the implementation of symbolic reasoning approaches.

Research on LLMs reasoning has progressed significantly, and numerous review articles have discussed it from different perspectives. Some papers provide an overall review of reasoning tasks (Plaat et al., 2024; Xu et al., 2025; Huang and Chang, 2022). Others focus on specific reasoning tasks and deeply analyze the technological advancements within particular fields (Lu et al., 2022). (Yu et al., 2024) conducts a comprehensive review of natural language reasoning. However, despite its potential to greatly enhance LLMs reasoning, symbolic reasoning remains underexplored in terms of systematic review and analysis, leaving a gap in understanding its full impact and utility in complex

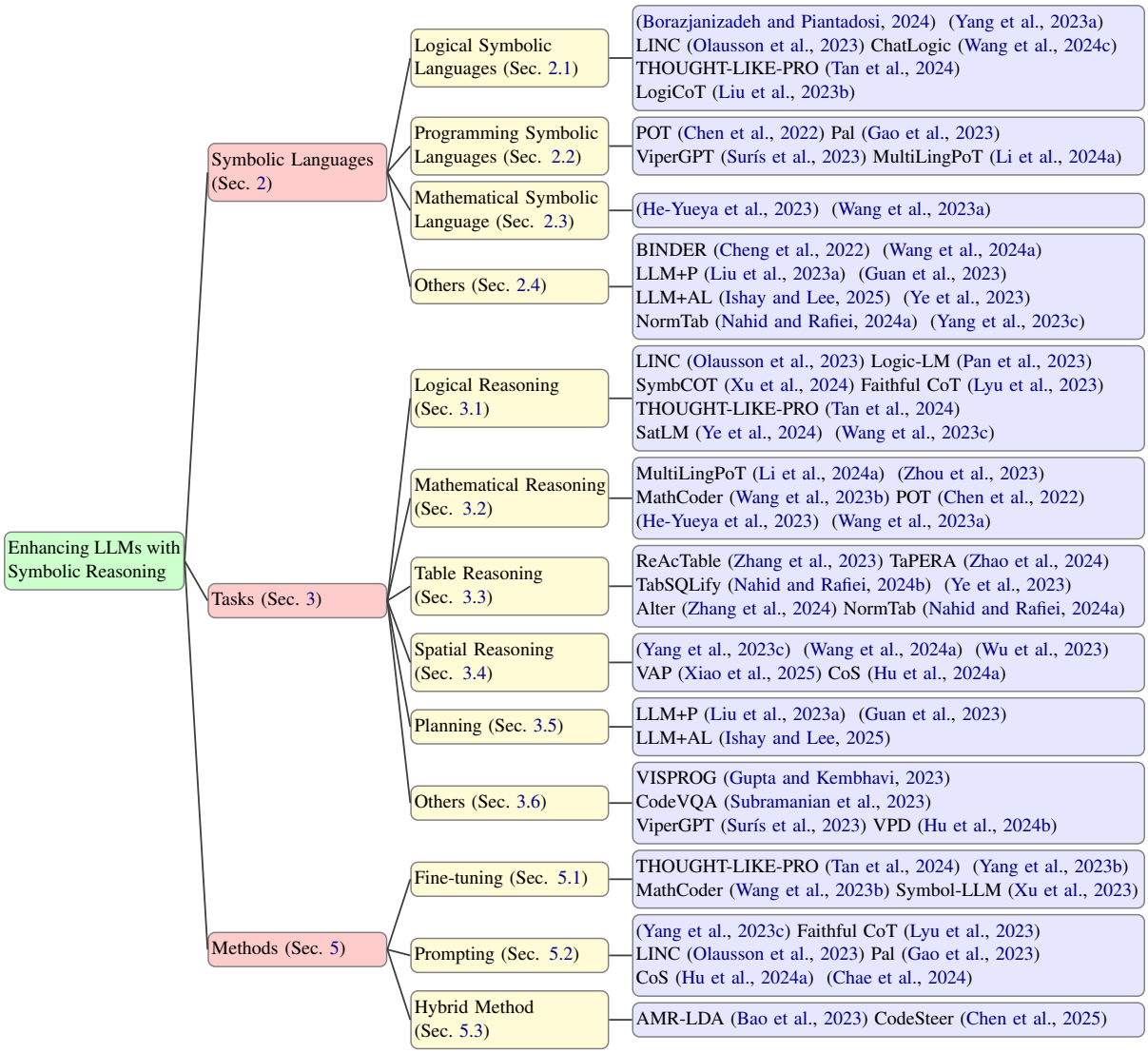


Figure 1: The structure of this survey.

reasoning tasks.

To address this research gap, we present this comprehensive survey that systematically examines the roles, mechanisms, and evolutionary trajectories of symbolic language in LLMs reasoning. As illustrated in Figure 1, we systematically organize these studies along three dimensions: tasks, symbolic languages, and methods. Through an extensive synthesis and forward-looking analysis of existing research, this paper aims to establish a clear conceptual framework and provide valuable reference points for future researchers, thereby facilitating further advancements and breakthroughs in the application of symbolic language for LLMs reasoning.

## 2 Symbolic Languages

Symbolic languages serve as vital tools within the realms of artificial intelligence and logic. When faced with a variety of reasoning tasks, selecting the appropriate symbolic language becomes particularly crucial. Below, we have summarized several commonly used symbolic languages.

### 2.1 Logical Symbolic Languages

In (Newton, 1934), the language of logical symbols is rigorously defined, encompassing both the symbols themselves and the rules for their use. Common logical symbols include conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and quantifiers ($\forall$, $\exists$). These symbols form the foundation of logical expressions and, through precise rules, ensure the rigor and consistency of logical reasoning. Compared to natural language, the language of logical symbols adheres to strict rules of reasoning. As long as the premises are correct, it ensures that the derivation of conclusions is error-free.

It is pointed out in (Olausson et al., 2023; Liu

et al., 2023b) that utilizing LLMs to transform natural language into first-order logic and then processing them with Prover9 can achieve significant results in reasoning tasks. The programming language Prolog, based on first-order logic and known for its powerful logical reasoning capabilities, is also widely applied in reasoning tasks (Borazjanizadeh and Piantadosi, 2024; Yang et al., 2023a; Tan et al., 2024; Wang et al., 2024c).

### 2.2 Programming Symbolic Languages

A programming language is a formal language specifically designed for writing computer programs, aimed at enabling computers to execute tasks with precision and efficiency. Today, a wide variety of symbolic languages exist worldwide, and they exhibit several shared traits. First, programming languages establish rigorous syntactic rules. Moreover, they commonly incorporate fundamental logical constructs, including selection and loop structures. The advantage of programming languages lies in their comprehensive support for data structures and efficient algorithms, enabling them to excel at handling a wide range of complex reasoning tasks.

Python, renowned for its concise and clear syntax and robust library support, is widely used to address reasoning problems (Chen et al., 2022; Gao et al., 2023; Surís et al., 2023). In the view of (Li et al., 2024a), the capacity of LLMs to select the best-suited programming language (such as Java, Matlab, and others) for diverse tasks is essential, especially since different programming languages have their own areas of expertise.

### 2.3 Mathematical Symbolic Language

Mathematical Symbolic Language is the cornerstone of mathematical expression and communication. It utilizes a standardized set of symbols (such as $+$, $-$, $\times$, $\div$) and rules to precisely describe mathematical concepts, relationships, and operations (Newton, 1934). Mathematical Symbolic Language is characterized by its exceptional conciseness and precision, making it highly effective in enhancing the accuracy of LLMs when tackling reasoning tasks, particularly those involving mathematical reasoning.

When faced with mathematical problems, (He-Yueya et al., 2023; Wang et al., 2023a) advocate LLMs should transform these problems into mathematical equations and subsequently solve them using external solvers.

### 2.4 Others

There are other symbolic languages specifically tailored for specialized domains. Answer Set Programming (ASP) is a declarative programming paradigm. (Yang et al., 2023c) uses prompts to guide LLMs to convert natural language into ASP code and then invokes a solver to obtain faithful results. (Wang et al., 2024a) introduces DSPy to conduct self-refinement on the prompts to better obtain the ASP code.

Planning Domain Definition Language (PDDL) is a formal language for describing planning problems. Proposed by (Liu et al., 2023a), the approach uses PDDL to formally describe planning tasks and relies on prompts to generate the problem code. (Guan et al., 2023) takes the LLMs as an intermediate layer. It also brings in human experts to modify the generated PDDL code. The limitations of PDDL are described by (Ishay and Lee, 2025), who propose Action Language (AL) as a replacement with clear syntactic and semantic rules for describing and changing the state of the world.

SQL is a declarative language specifically designed for managing and manipulating relational databases, wildly used in table reasoning (Nahid and Rafiei, 2024a; Ye et al., 2023; Cheng et al., 2022).

## 3 Tasks

In this section, we will introduce LLMs reasoning combined with symbolic languages in different tasks, each task with unique challenges and specific requirements.

### 3.1 Logical Reasoning

Logical reasoning tasks require LLMs to derive conclusions or verify their accuracy based on given information, such as rules and facts (Olausson et al., 2023). This critically depends on the LLMs' capability to understand logical rules and conditions while consistently upholding rigor throughout the reasoning process. However, due to the ambiguity and complexity of natural language, LLMs are prone to generating hallucinations and errors. Symbolic reasoning effectively mitigates this ambiguity by employing precise symbolic representations and formal rules to express concepts and relationships. There are three main strategies used to integrate symbolic languages into the reasoning process.

The first strategy is decomposition, which involves breaking a problem into smaller subproblems or dividing the reasoning process into multi-

ple steps, followed by the deployment of symbolic languages.

(Pan et al., 2023) decomposes logical reasoning problems into three stages: problem formulation, symbolic reasoning, and result interpretation. The framework, which integrates four modules—Translator, Planner, Solver, and Verifier—is introduced by (Xu et al., 2024) to ensure faithful logical reasoning.

Secondly, seeking the assistance of external tools to solve formal symbolic expressions and having LLMs only undertake translation tasks is also a strategy to enhance the reliability of reasoning. LLMs are utilized as semantic parsers by (Olausson et al., 2023), converting natural language into first-order logic and employing an external theorem prover to determine the truth value of conclusions. (Lyu et al., 2023) enhances the faithfulness of Chain of Thought (CoT) by incorporating external solvers.

Thirdly, logical expansion based on symbolic representation provides a more complete semantic expression, reducing the risk of semantic conversion errors in LLMs (Liu et al., 2023b). (Liu et al., 2024) utilizes LLMs to extract sentences containing conditional reasoning relationships from the input and implement logical expansion using Python. According to (Bao et al., 2023), the Abstract Meaning Representation (AMR) graph is used to carry out logical expansion by applying the principle of logical equivalence. Imitation learning has been leveraged by (Tan et al., 2024) to enable LLMs to mimic the reasoning trajectories generated by the Prolog logic engine.

Several methods mentioned above are typically applied comprehensively in practical work to achieve a higher level of reasoning ability.

### 3.2 Mathematical Reasoning

Mathematical tasks encompass a wide range of problems, including algebra, geometry, calculus, and more. During the reasoning process, precise analysis of mathematical conditions is essential, often accompanied by extensive computations. This poses significant challenges to LLMs, demanding both strong analytical understanding and accurate computational capabilities (Chen et al., 2022). Therefore, a viable strategy involves directing LLMs to focus on generating symbolic representations, such as mathematical formulations or Python code, and utilizing these representations to delegate specific computational tasks to external solvers, which significantly improves the accuracy of mathematical reasoning (Wang et al., 2023b). Python code is generated by prompting LLMs, and the execution of this code is used to solve problems, a method proposed by (Chen et al., 2022) that effectively improves result accuracy.

The introduction of external solvers has effectively addressed the limitations of large language models in terms of precise computation. Currently, the core challenge in the field of mathematical reasoning lies in ensuring that large language models can accurately generate standardized symbolic expressions. Just as humans can verify the correctness of their answers to mathematical problems by substituting them back into the original question, (Zhou et al., 2023) proposes a method called "code-based explicit self-verification", which introduces an additional verification stage. When the verification result is "False", the model can automatically correct the solution. Emphasizing the importance of programming language selection, (Li et al., 2024a) notes that different mathematical problems are best addressed by different programming languages, thereby reducing the burden on LLMs to generate accurate code. Mathematical equations, as highlighted by (He-Yueya et al., 2023; Wang et al., 2023a), provide a more accurate representation of problems and are more reliably generated by LLMs compared to programming languages.

### 3.3 Table Reasoning

The task of table reasoning aims to enable models to generate corresponding results as answers based on specific task requirements when receiving one or more tables as input (Wang et al., 2024d). The complexity and huge volume of information in table reasoning may obscure key details, potentially weakening the decision-making ability of LLMs (Liu et al., 2023c).

To address the challenges posed by large table data, researchers have proposed that LLMs focus exclusively on sub-tables relevant to the problem at hand. SQL queries are generated using LLMs, and sub-tables containing only the essential information required to answer questions or verify statements are extracted (Nahid and Rafiei, 2024b). Similarly, (Zhang et al., 2023) decomposes large tables into smaller sub-tables and employs Python to perform fine-grained splits of semantically complex rows and columns.

When dealing with complex table-related problems that involve multiple tables, decomposing the problem can significantly reduce the burden

on LLMs (Zhao et al., 2024). Some researchers have adopted a hybrid approach. (Ye et al., 2023) uses LLMs to predict the relevant row and column indices involved in the problem and decomposed complex problems into numerical, logically related sub-problems. According to (Zhang et al., 2024), complex problems are decomposed into multiple sub-problems, symbolic language sub-tables are generated, and the answers are integrated to derive the results.

The structural complexity of tables has also been addressed. (Nahid and Rafiei, 2024a) enhances the symbolic reasoning capabilities of LLMs by normalizing tables into relational structures that adhere to standard paradigms.

### 3.4 Spatial Reasoning

Spatial reasoning is a fundamental aspect of human cognition, enabling us to interact effectively with our environment. It plays a crucial role in tasks that involve understanding and reasoning about the spatial relationships between objects and their movements. However, spatial reasoning tasks present a significant challenge for LLMs. It is usually hard for LLMs to understand the complex relationship descriptions formed by natural language in spatial Reasoning. A common approach is to modify the CoT with symbols. *Chain-of-Symbol Prompting* has been proposed by (Hu et al., 2024a), leveraging specific symbols to simplify the spatial-relationship information expressed in natural language in CoT.

Modeling spatial relationships with specially developed formal expressions helps to obtain faithful results using solvers. (Yang et al., 2023c) combines LLMs with ASP programming. Iteratively refining the generated ASP programs and employing the DSPy(Declarative Self-improving Language Programs in Python) framework, (Wang et al., 2024a) manages complex workflows and optimizes prompts effectively.

Multi-modal methods can also enhance performance. (Xiao et al., 2025) uses LLMs as a planner to generate plans for problems and Multi-Modal Large Language Model (MLLM) to invoke external image synthesis toolkits. A visual extractor based on a System-1-like VLM is employed, and LLMs are used to approximate a symbolic system with broad-coverage symbols and rational rules, as proposed by (Wu et al., 2023).

### 3.5 Planning

At the heart of planning tasks is crafting a viable path from the initial state to the goal. This involves starting from the task's origin and guiding the system or environment toward the desired outcome through a sequence of well-chosen actions. Symbolic languages enable the flexible construction of planning algorithms tailored to specific needs, allowing for precise definition and optimization of state transitions. This significantly boosts the overall effectiveness of planning tasks, particularly in ensuring plans are both executable and reliable.

The approach converts natural language into PDDL using LLMs, and then employs a planner to generate a plan, as proposed by (Liu et al., 2023a). (Guan et al., 2023) generates a PDDL model through prompting and combines the PDDL verification tool with the feedback from human domain experts to correct model errors. (Ishay and Lee, 2025) points out that PDDL lacks flexibility, and introduces Action Language (AL) BC+ for expressing more general forms of knowledge about actions like indirect effects.

### 3.6 Others

Given that multi-modal reasoning tasks typically span multiple categories discussed previously, they cannot be strictly classified into a single specific category. However, considering the unique advantages of multi-modal reasoning in integrating heterogeneous information, enhancing model performance, and simulating human multisensory cognition, we provide a dedicated discussion on this topic here. The greatest challenge in multi-modal reasoning lies in how to efficiently integrate information from different modalities while effectively managing the complexity of the multi-modal reasoning process. The use of symbolic languages offers a promising solution, as it facilitates the development of flexible multi-modal data processing frameworks. These frameworks enable seamless interaction and collaborative cooperation among different modalities, thereby enhancing the overall reasoning capabilities.

(Gupta and Kembhavi, 2023) addresses complex tasks by decomposing them into multiple modules, each processed using visual models, Python, and other tools. Prompts are utilized to guide the LLMs in generating Python code, and existing visual models are directly invoked to facilitate the reasoning process, as described in (Subramanian et al., 2023). (Surís et al., 2023) opts for customizing APIs to more precisely assist Python code in calling visual models, thereby optimizing for specific requirements. The field is advanced by the development of an efficient model specifically designed for vi-

sual reasoning using symbolic language, and by employing knowledge distillation techniques, as presented in (Hu et al., 2024b).

## 4 Datasets and Benchmarks

In this section, we summarize the datasets and benchmarks (see details in Table 1) used to evaluate the symbolic reasoning capabilities of LLMs.

**Arithmetic Reasoning.** The data of arithmetic reasoning consists of problem descriptions, rationales, and answers. GSM8K (Cobbe et al., 2021) contains approximately 8500 primary-school-level math word problems and is used to evaluate a model's multi-step arithmetic reasoning ability. Math (Hendrycks et al., 2021) includes various mathematical problems, and it is used to evaluate a model's arithmetic reasoning and problem-solving skills. AQuA (Ling et al., 2017) contains multiple-choice math reasoning questions and detailed explanations. SVAMP (Patel et al., 2021) is a collection of basic math problems with diverse expressions. ASDiv (Miao et al., 2020) contains a large number of primary-school-level math division word problems with different numbers of operation steps and varying levels of difficulty. MAWPS (Koncel-Kedziorski et al., 2016) is a benchmark that contains various math word problems and their solutions. ALGEBRA (He-Yueya et al., 2023) mainly focuses on algebraic problems.

**Logical Reasoning.** The data of logical reasoning includes problem scenarios or premise information, conclusions or questions, as well as annotation information used to evaluate the model's output. ProofWriter (Tafjord et al., 2020) is a synthetic dataset dedicated to multi-step logical reasoning. PrOntoQA (Saparov and He, 2022) is a logical reasoning dataset constructed based on formal ontology structures. FOLIO (Han et al., 2022) is a natural language inference benchmark constructed based on first-order logic. AR-LSAT (Zhong et al., 2022) serves as a benchmark formulated around the analytical reasoning questions of the Law School Admission Test. LogiQA (Liu et al., 2020) is a logical reasoning question-answering dataset constructed based on legal and daily scenarios. CLUTRR (Sinha et al., 2019) is a dataset built around the synthetic kinship relation reasoning tasks.

**Table Reasoning.** The data of the table reasoning consists of tables, questions, and answers. WikiTQ (Pasupat and Liang, 2015) is constructed from Wikipedia tables. FetaQA (Nan et al., 2022) is built based on multi-source factual tables. TabFact (Chen et al., 2020) is a fact-checking dataset constructed from Wikipedia tables. WikiSQL (Zhong et al., 2017) is a large-scale text-to-SQL dataset built upon Wikipedia tables.

**Spatial Reasoning.** The data format of the spatial reasoning dataset is composed of scene descriptions, questions, and answers. StepGame (Shi et al., 2022) is designed to test the ability to multi-hop spatial reasoning, SparTUN (Mirzaee and Kordjamshidi, 2022) is built upon the NLVR (Natural Language for Visual Reasoning) images.

**Others.** Some other benchmarks contain tasks from multiple domains and can test the comprehensive performance of LLMs. BIG-bench (Srivastava et al., 2022) encompasses over 200 tasks designed to test various reasoning capabilities of LLMs. Some benchmarks are designed for testing the performance of the framework they proposed to solve some real-world problems (Hu et al., 2023). There are corresponding datasets available for testing some works that apply VLMs (Goyal et al., 2017).

## 5 Methods

Leveraging LLMs to conduct reasoning tasks directly is subpar. Past work has employed several methods to enhance the reasoning capabilities of LLMs. In general, they mainly adopt one of three methods: fine-tuning, prompting, and hybrid methods. The main processes of enhancing LLMs with symbolic reasoning through prompting and fine-tuning are shown in Figure 2. The hybrid method incorporates fine-tuning and prompting as sub-structures to form the overall framework.

### 5.1 Fine-tuning

Fine-tuning refers to further training the model on the basis of the pre-trained model by using the data of specific reasoning tasks to better adapt it to specific scenarios. Developing a symbolic reasoning-enhanced dataset to fine-tune LLMs allows the model to focus on the characteristics of reasoning tasks and learn corresponding behavioral patterns. THOUGHT-LIKE-PRO is proposed, where LLMs are fine-tuned with datasets verified by the Prolog engine (Tan et al., 2024). A high-quality dataset has been constructed by (Wang et al., 2023b) to improve the mathematical reasoning ability of LLMs through customizing supervised fine-tuning methods. In (Yang et al., 2023b), a dataset constructed from diverse sentence-level NL-FOL pairs is designed to fine-tune LLMs for the translation task from natural language (NL) to first-order

| Domains | Benchmarks | Size | Citations |
|---|---|---|---|
| Arithmetic Reasoning | GSM8K (Cobbe et al., 2021) | 8.5K | (Borazjanizadeh and Piantadosi, 2024; Lyu et al., 2023; Chen et al., 2022; Gao et al., 2023) |
| | Math (Hendrycks et al., 2021) | 12.5K | (Zhou et al., 2023; Wang et al., 2023b; Li et al., 2024a; Tan et al., 2024) |
| | AQuA (Ling et al., 2017) | 100K | (Lyu et al., 2023; Chen et al., 2022; Leang et al., 2024) |
| | SVAMP (Patel et al., 2021) | 1K | (Lyu et al., 2023; Chen et al., 2022; Gao et al., 2023; Leang et al., 2024) |
| | ASDiv (Miao et al., 2020) | 2305 | (Lyu et al., 2023; Gao et al., 2023) |
| | MAWPS (Koncel-Kedziorski et al., 2016) | 3320 | (Gao et al., 2023) |
| | ALGEBRA (He-Yueya et al., 2023) | 222 | (He-Yueya et al., 2023; Wang et al., 2023a) |
| Logical Reasoning | ProofWriter (Tafjord et al., 2020) | 500K | (Yang et al., 2023a; Lee and Hwang, 2024; Pan et al., 2023; Xu et al., 2024) |
| | PrOntoQA (Saparov and He, 2022) | 10K | (Pan et al., 2023; Xu et al., 2024; Tan et al., 2024) |
| | FOLIO (Han et al., 2022) | 1435 | (Li et al., 2024b; Kalyanpur et al., 2024; Liu et al., 2024) |
| | AR-LSAT (Zhong et al., 2022) | 2046 | (Pan et al., 2023; Xu et al., 2024; Wang et al., 2024b) |
| | LogiQA (Liu et al., 2020) | 8678 | (Liu et al., 2024; Li et al., 2024b; Bao et al., 2023) |
| | CLUTRR (Sinha et al., 2019) | 10K | (Ye et al., 2024; Yang et al., 2023c) |
| Table Reasoning | WikiTQ (Pasupat and Liang, 2015) | 22033 | (Zhang et al., 2023; Nahid and Rafiei, 2024b; Mouravieff et al., 2024; Cheng et al., 2022; Zhang et al., 2024) |
| | FetaQA (Nan et al., 2022) | 10K | (Ye et al., 2023; Zhang et al., 2023; Nahid and Rafiei, 2024b) |
| | TabFact (Chen et al., 2020) | 117854 | (Ye et al., 2023; Nahid and Rafiei, 2024b; Wang et al., 2024d; Cheng et al., 2022; Zhang et al., 2024) |
| | WikiSQL (Zhong et al., 2017) | 80654 | (Nahid and Rafiei, 2024b) |
| Spatial Reasoning | StepGame (Shi et al., 2022) | 6.1K | (Wang et al., 2024a; Yang et al., 2023c) |
| | SparTUN (Mirzaee and Kordjamshidi, 2022) | 50K | (Wang et al., 2024a) |
| Others | BIG-bench (Srivastava et al., 2022) | - | (Borazjanizadeh and Piantadosi, 2024; Gao et al., 2023; Li et al., 2023) |
| | Fruit Shop (Hu et al., 2023) | 70 | (Hu et al., 2023) |
| | VQAv2 (Goyal et al., 2017) | 1105904 | (Hu et al., 2024b) |

Table 1: Common datasets and benchmarks used to evaluate the symbolic reasoning capabilities of LLMs. We classify these datasets into different domains, marked their sizes, and noted some representative works that used them for evaluation.

logic (FOL). (Xu et al., 2023) designs a 2-stage fine-tuning framework aiming at improving the balanced capabilities of LLMs in symbolic tasks and natural language tasks.

### 5.2 Prompting

Prompting methods in Large Language Models (LLMs) encompass a range of techniques for crafting specialized input prompts that direct the model to produce outputs aligned with users' specifications. LLMs can utilize prompting strategies to generate diverse symbolic languages, which are then processed by external solvers to execute reasoning tasks, thereby enhancing overall accuracy. A natural idea is to incorporate symbolic expressions into the Chain of Thought (CoT) prompts, guiding the LLMs to generate symbolic language step by step, which does not require extensive model training. A method of optimizing the prompts of the original CoT has been raised by (Lyu et al., 2023). They decompose the original problems into subproblems and generate more refined prompts consisting of natural language and symbolic language. For (Olausson et al., 2023), the prompt requires the LLMs to convert the premises and conclusions into FOL expressions to obtain a formal representation. The prompt of (Gao et al., 2023) consists of a natural-language question and intermediate reasoning steps combined with programming symbolic language. (Yang et al., 2023c) designs the prompt used to guide the LLMs to convert natural language into atomic facts as symbols that are suitable for ASP reasoning. (Hu et al., 2024a) constructs prompt through self-refinement with objects and symbols.

### 5.3 Hybrid Methods

The fine-tuning method requires a carefully designed dataset which consumes substantial time and expenses. In addition, the fine-tuning approach usually focuses on specific tasks, which limits the generalization ability. Leveraging prompts can guide LLMs in generating optimized rationales. However, this approach does not improve the actual capability of LLMs.

The hybrid method integrates their advantages, enhancing the performance of LLMs while enabling them to generate better rationales and answers. (Bao et al., 2023) first adopts Abstract Meaning Representation-Based Logic-Driven Data Augmentation (AMR-LDA) to generate diverse symbolic logical expressions. Then they use Logical-Equivalence-Identification Contrastive Learning and fine-tuning methods for discriminative LLMs and prompts for generative LLMs based on the augmented expressions. A framework for guiding the generation of LLMs code/text is proposed by (Chen et al., 2025). They fine-tune a guiding model called CodeSteerLLM and introduce a symbolic checker and a self-answer checker (prompting method) to conduct multi-round interactions with a TaskLLM.

## 6 Future Directions

**Customized Symbolic Languages and Solvers.** Recent research relies mainly on existing formal symbolic languages to assist in reasoning, without innovating the grammar for specific tasks (Olausson et al., 2023; Yang et al., 2023c; Liu et al.,

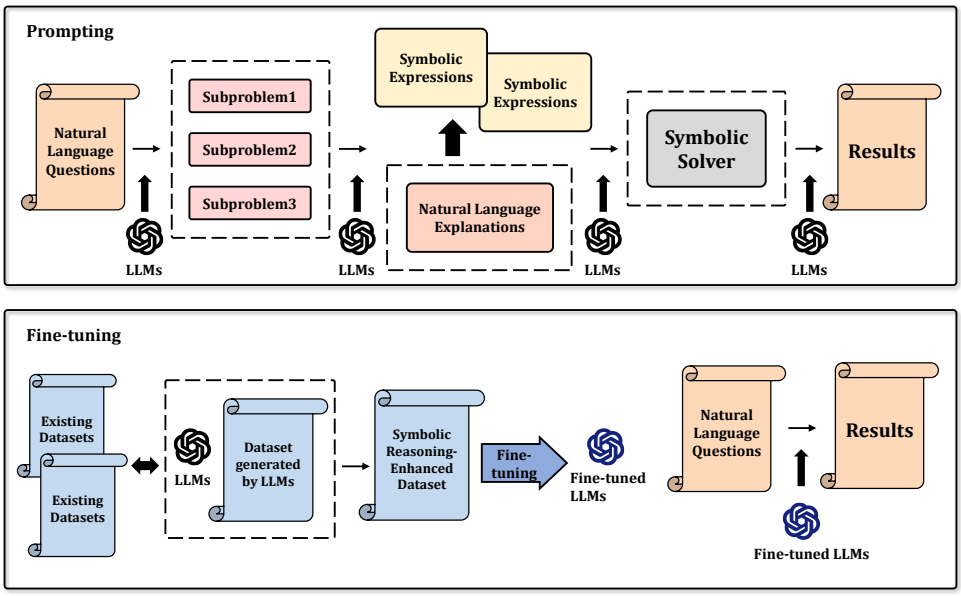


Figure 2: The main processes of enhancing the symbolic reasoning performance of LLMs through prompting and fine-tuning. The parts enclosed by dashed lines represent the special designs of some works.

2023a). However, these existing formal symbolic languages cannot fully cover all application scenarios in the real world (Ishay and Lee, 2025) and are unable to entirely meet the requirements. Therefore, customizing language parsers and related symbolic grammar according to application scenarios may be the focus of future work.

**Multilingual Symbolic Languages Integration.** Reasoning tasks are inherently complex and typically cannot be effectively addressed using a single language alone. (Li et al., 2024a; Pan et al., 2023) advocate for the use of different symbolic languages tailored to various tasks. However, in addressing specific problems, they predominantly rely on a single symbolic language, which indicates that the integration between different symbolic languages remains relatively superficial. In contrast, (Zhang et al., 2023) demonstrates a more integrated approach by utilizing both Python and SQL for solving tabular problems. We encourage future research to explore more effective multi-language integration strategies across a broader range of reasoning tasks, leveraging the strengths of multiple symbolic languages to enhance the reasoning capabilities of LLMs.

**Inherent Structured Languages for LLMs.** When leveraging symbolic languages for reasoning in LLMs, many errors stem from how these models generate task-specific symbolic expressions. Various solutions have been proposed—such as using execution feedback, human-like debugging (Zhong et al., 2024), and strategies that avoid generating entire code segments in one go (Zhou et al., 2023)—yet none fully resolve the problem. The appeal of symbolic languages lies in their structured and rigorous nature. We encourage researchers to explore alternatives that either replace symbolic languages altogether or imbue LLMs with inherent structured and rigorous reasoning capabilities.

## 7 Conclusion

In this paper, we conduct a systematic and detailed review of the symbolic reasoning tasks in large language models. We illustrate the applications of LLMs symbolic reasoning in different reasoning tasks, summarize the types of symbolic languages used in reasoning, discuss the techniques for enhancing and eliciting the symbolic reasoning capabilities of LLMs, as well as the benchmarks employed to evaluate and analyze the symbolic reasoning abilities of LLMs. We also discuss the promising future directions of symbolic reasoning. We hope that this paper can offer a comprehensive and valuable overview of the present status of the field and facilitate further advancements in the application of symbolic language for LLM reasoning.

## 8 Limitations

While this survey aims to provide a comprehensive overview of the integration of symbolic reasoning with large language models (LLMs), it has its limitations, which we acknowledge to provide a balanced perspective on our work.

First, the field of enhancing LLMs with symbolic reasoning is evolving at an unprecedented pace. New methodologies, frameworks, and applications are being published frequently, making it challenging to capture the most recent advancements. Despite our rigorous efforts to include the latest research up to the submission deadline, some cutting-edge developments may have emerged during the final stages of this survey's preparation.

Second, in an effort to provide a broad overview of the field, this survey categorizes the research from three perspectives: tasks, symbolic methods, and languages. While this approach offers a structured framework for understanding the landscape, the breadth of coverage inevitably comes at the expense of depth in certain areas. Some technical nuances, domain-specific challenges, and emerging sub-fields may have been underexplored or oversimplified.

Finally, the majority of reasoning benchmarks are collected and categorized from the experimental sections of mainstream industry works, potentially leading to insufficient coverage of niche or domain-specific reasoning tasks.

Despite these limitations, we believe this survey provides a valuable foundation for understanding the current state of research and identifying future directions in symbolic reasoning with LLMs. We encourage researchers to build upon this work and address the gaps identified here to further advance the field.

## References


Qiming Bao, Alex Yuxuan Peng, Zhenyun Deng, Wanjun Zhong, Gael Gendron, Timothy Pistotti, Neset Tan, Nathan Young, Yang Chen, Yonghua Zhu, et al. 2023. Abstract meaning representation-based logic-driven data augmentation for logical reasoning. *arXiv preprint arXiv:2305.12599*.

Nasim Borazjanizadeh and Steven T Piantadosi. 2024. Reliable reasoning beyond natural language. *arXiv preprint arXiv:2407.11373*.

Hyungjoo Chae, Yeonghyeon Kim, Seungone Kim, Kai Tzu-iunn Ong, Beong-woo Kwak, Moohyeon Kim, Seonghwan Kim, Taeyoon Kwon, Jiwan Chung, Youngjae Yu, et al. 2024. Language models as compilers: Simulating pseudocode execution improves algorithmic reasoning in language models. *arXiv preprint arXiv:2404.02575*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale dataset for table-based fact verification. *Preprint*, arXiv:1909.02164.

Yongchao Chen, Yilun Hao, Yueying Liu, Yang Zhang, and Chuchu Fan. 2025. CodeSteer: Symbolic-augmented language models via code/text guidance. *arXiv preprint arXiv:2502.04350*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2022. Binding language models in symbolic languages. *arXiv preprint arXiv:2210.02875*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Leveraging Pretrained Large Language Models to Construct and Utilize World Models for Model-based Task Planning. *Advances in Neural Information Processing Systems*, 36:79081–79094.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual Programming: Compositional Visual Reasoning without Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. 2022. FOLIO: Natural Language Reasoning with First-Order Logic. *arXiv preprint arXiv:2209.00840*.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the Math Dataset. *arXiv preprint arXiv:2103.03874*.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. ChatDB: Augmenting LLMs with Databases as Their Symbolic Memory. *arXiv preprint arXiv:2306.03901*.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. 2024a. Chain-of-Symbol Prompting For Spatial Reasoning in Large Language Models. In *First Conference on Language Modeling*.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. 2024b. Visual Program Distillation: Distilling Tools and Programmatic Reasoning into Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9590–9601.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards Reasoning in Large Language Models: A survey. *arXiv preprint arXiv:2212.10403*.

Adam Ishay and Joohyung Lee. 2025. LLM+AL: Bridging Large Language Models and Action Languages for Complex Reasoning about Actions. *arXiv preprint arXiv:2501.00830*.

Aditya Kalyanpur, Kailash Karthik Saravanakumar, Victor Barres, Jennifer Chu-Carroll, David Melville, and David Ferrucci. 2024. LLM-ARC: Enhancing LLMs with an Automated Reasoning Critic. *arXiv preprint arXiv:2406.17663*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A Math Word Problem Repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Joshua Ong Jun Leang, Aryo Pradipta Gema, and Shay B Cohen. 2024. CoMAT: Chain of Mathematically Annotated Thought Improves Mathematical Reasoning. *arXiv preprint arXiv:2410.10336*.

Jinu Lee and Wonseok Hwang. 2024. SymBa: Symbolic Backward Chaining for Multi-step Natural Language Reasoning. *arXiv preprint arXiv:2402.12806*.

Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. 2023. Chain of Code: Reasoning with a Language Model-Augmented Code Emulator. *arXiv preprint arXiv:2312.04474*.

Nianqi Li, Zujie Liang, Siyu Yuan, Jiaqing Liang, Feng Wei, and Yanghua Xiao. 2024a. MultiLingPoT: Enhancing Mathematical Reasoning with Multilingual Program Fine-tuning. *arXiv preprint arXiv:2412.12609*.

Qingchuan Li, Jiatong Li, Tongxuan Liu, Yuting Zeng, Mingyue Cheng, Weizhe Huang, and Qi Liu. 2024b. Leveraging LLMs for Hypothetical Deduction in Logical Inference: A Neuro-Symbolic Approach. *arXiv preprint arXiv:2410.21779*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023a. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. *arXiv preprint arXiv:2304.11477*.

Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. 2023b. LogiCoT: Logical Chain-of-Thought Instruction Tuning. In *Proc. of EMNLP Findings*, pages 2908–2921.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. *arXiv preprint arXiv:2007.08124*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023c. Rethinking Tabular Data Understanding with Large Language Models. *arXiv preprint arXiv:2312.16702*.

Tongxuan Liu, Wenjiang Xu, Weizhe Huang, Xingyu Wang, Jiaxing Wang, Hailong Yang, and Jing Li. 2024. Logic-of-Thought: Injecting Logic into Contexts for Full Reasoning in Large Language Models. *arXiv preprint arXiv:2409.17539*.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2022. A Survey of Deep Learning for Mathematical Reasoning. *arXiv preprint arXiv:2212.10535*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful Chain-of-Thought Reasoning. *arXiv preprint arXiv:2301.13379*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Roshanak Mirzaee and Parisa Kordjamshidi. 2022. Transfer Learning with Synthetic Corpora for Spatial Role Labeling and Reasoning. *arXiv preprint arXiv:2210.16952*.

Raphaël Mouravieff, Benjamin Piwowarski, and Sylvain Lamprier. 2024. Training Table Question Answering via SQL Query Decomposition. *arXiv preprint arXiv:2402.13288*.

Md Mahadi Hasan Nahid and Davood Rafiei. 2024a. NormTab: Improving Symbolic Reasoning in LLMs through Tabular Data Normalization. *arXiv preprint arXiv:2406.17961*.

Md Mahadi Hasan Nahid and Davood Rafiei. 2024b. TabSQLify: Enhancing Reasoning Capabilities of LLMs Through Table Decomposition. *arXiv preprint arXiv:2404.10150*.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form Table Question Answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Isaac Newton. 1934. Principia mathematica. *Book III, Lemma V, Case*, 1:1687.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-Thought: Large Language Models Can Do Parallel Decoding. *Proceedings ENLSP-III*.

Theo X Olausson, Alex Gu, Benjamin Lipkin, Cedegao E Zhang, Armando Solar-Lezama, Joshua B Tenenbaum, and Roger Levy. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. *arXiv preprint arXiv:2310.15164*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. *arXiv preprint arXiv:2305.12295*.

Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. StepGame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11321–11329.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Sanjay Subramanian, Medhini Narasimhan, Kushal Khangaonkar, Kevin Yang, Arsha Nagrani, Cordelia Schmid, Andy Zeng, Trevor Darrell, and Dan Klein. 2023. Modular visual question answering via code generation. *arXiv preprint arXiv:2306.05392*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. ProofWriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.

Xiaoyu Tan, Yongxin Deng, Xihe Qiu, Weidi Xu, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2024. THOUGHT-LIKE-PRO: Enhancing Reasoning of Large Language Models through Self-Driven Prolog-based Chain-of-Thought. *arXiv preprint arXiv:2407.14562*.

Dingzirui Wang, Longxu Dou, Wenbin Zhang, Junyu Zeng, and Wanxiang Che. 2023a. Exploring equation as a better intermediate meaning representation for numerical reasoning. *arXiv preprint arXiv:2308.10585*.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023b. MathCoder: Seamless code integration in llms for enhanced mathematical reasoning. *arXiv preprint arXiv:2310.03731*.

Rong Wang, Kun Sun, and Jonas Kuhn. 2024a. Dspy-based neural-symbolic pipeline to enhance spatial reasoning in llms. *arXiv preprint arXiv:2411.18564*.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D Goodman. 2023c. Hypothesis Search: Inductive reasoning with language models. *arXiv preprint arXiv:2309.05660*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024b. Symbolic working memory enhances language models for complex rule application. *arXiv preprint arXiv:2408.13654*.

Zhongsheng Wang, Jiamou Liu, Qiming Bao, Hongfei Rong, and Jingfeng Zhang. 2024c. ChatLogic: Integrating logic programming with large language models for multi-step reasoning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2024d. Chain-of-Table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Xiaoqian Wu, Yong-Lu Li, Jianhua Sun, and Cewu Lu. 2023. Symbol-LLM: Leverage language models for symbolic system in visual human activity reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pages 29680–29691. Curran Associates, Inc.

Ziyang Xiao, Dongxiang Zhang, Xiongwei Han, Xiaojin Fu, Wing Yin Yu, Tao Zhong, Sai Wu, Yuan Wang, Jianwei Yin, and Gang Chen. 2025. Enhancing llm reasoning via vision-augmented prompting. *Advances in Neural Information Processing Systems*, 37:28772–28797.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. 2023. Symbol-LLM: Towards foundational symbol-centric interface for large language models. *arXiv preprint arXiv:2311.09278*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Sen Yang, Xin Li, Leyang Cui, Lidong Bing, and Wai Lam. 2023a. Neuro-symbolic integration brings causal and reliable reasoning proofs. *arXiv preprint arXiv:2311.09802*.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2023b. Harnessing the power of large language models for natural language to first-order logic translation. *arXiv preprint arXiv:2305.15541*.

Zhun Yang, Adam Ishay, and Joohyung Lee. 2023c. Coupling large language models with logic programming for robust and general reasoning from text. *arXiv preprint arXiv:2307.07696*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of Thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2024. SatLM: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large Language Models are Versatile Decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–184.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2024. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39.

Han Zhang, Yuheng Ma, and Hanfang Yang. 2024. Alter: Augmentation for large-table-based reasoning. *arXiv preprint arXiv:2407.03061*.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M Patel. 2023. ReAcTable: Enhancing react for table question answering. *arXiv preprint arXiv:2310.00815*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024. TaPERA: enhancing faithfulness and interpretability in long-form table qa by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840.

Li Zhong, Zilong Wang, and Jingbo Shang. 2024. Ldb: A large language model debugger via verifying runtime execution step-by-step. *arXiv preprint arXiv:2402.16906*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Yining Chen, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2022. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2306–2319, Seattle, United States. Association for Computational Linguistics.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*.