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 paid to enhancing and understanding their reasoning capabilities. Most existing works attempt to directly use LLMs for natural-language-based reasoning. However, due to the inherent semantic ambiguity and complexity of natural language, LLMs often struggle with complex problems, leading to challenges such as hallucinations and inconsistent reasoning. Therefore, techniques for constructing formal language representations, most of which are symbolic languages, have emerged. In this work, we focus on symbolic reasoning in LLMs and provide a comprehensive review of the related research. This includes the types of symbolic languages used, different symbolic reasoning tasks and related benchmarks, and typical techniques for enhancing LLMs' symbolic reasoning abilities. 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 techniques to enhance the reasoning capabilities of LLMs (Zhang et al., 2022; Yao et al., 2024; Ning et al., 2023). However, LLMs face significant challenges in their reasoning processes, with hallucination being one of the most notable. 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 framework stems from classic neuro-symbolic approaches (Andreas et al., 2016; Liang et al., 2017; Ebrahimi et al., 2021) and effectively alleviates the limitations of LLMs in reasoning tasks by integrating the strengths of 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 technique not only reduces the burden on the language model but also fully leverages the professional advantages of the external solver in dealing with specific problems, achieving complementary advantages and collaborative work between the LLMs and the external solver. Symbolic languages, with their precision, interpretability, and logicality, bring significant advantages to the reasoning process (Olausson et al., 2023; Lyu et al., 2023a; Chen et al., 2022).

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. conduct a comprehensive review of natural language reasoning (Yu et al., 2024). 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 reasoning tasks.

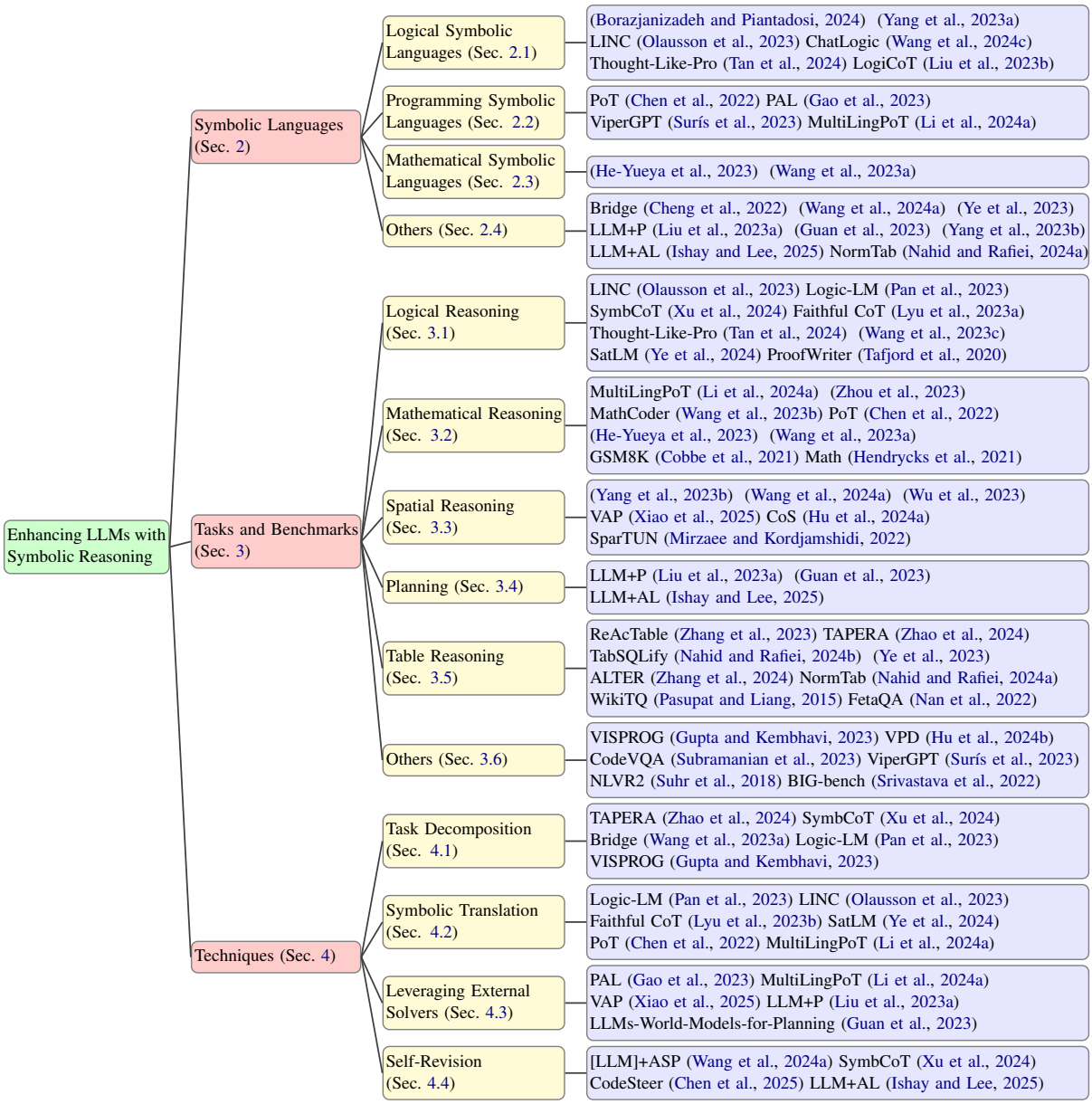

Figure 1: The structure of this survey.

To address this research gap, we present this comprehensive survey that systematically examines how to use symbolic language to enhance the reasoning capabilities of LLMs. As illustrated in Figure 1, we systematically organize these studies along three dimensions: symbolic languages, tasks and benchmarks, and techniques. Through an extensive synthesis and forward-looking analysis of existing research, this paper aims to establish a clear conceptual framework for future researchers, thereby facilitating further advancements and breakthroughs in the application of symbolic languages 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. In the following text, we will introduce various symbolic languages.

2.1 Logical Symbolic Languages

In Principia Mathematica (Newton, 1934), the language of logical symbols is rigorously defined, encompassing both the symbols and the rules. 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 LINC (Olausson et al., 2023) that utilizing LLMs to transform natural language into first-order logic and then processing them with Prover9 can help improve the accuracy of logical 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. Nowadays, there is a wide variety of programming languages, yet they share some common characteristics. 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 capability to handle various complex reasoning tasks with the assistance of data structures and algorithms.

Different programming languages possess unique advantages. For example, Matlab is better at matrix operations than Java, while Python provides a rich library for number theory (Li et al., 2024a). These advantages make Matlab and Python excel in mathematical reasoning tasks. The experimental results conducted by Luo et al. (Luo et al., 2024) indicate that Java performs the best in table reasoning tasks, and C++ demonstrates higher efficiency in mathematical applications and spatial reasoning tasks.

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 et al. (He-Yueya et al., 2023) and Wang et al. (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. In [LLM]+ASP (Yang et al., 2023b), prompts are employed to steer LLMs in transforming natural language to ASP code and then invoking a solver to obtain faithful results. Wang et al. (Wang et al., 2024a) introduces DSPy (Declarative Self-improving Language Programs in Python) 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. The approach proposed by LLM+P (Liu et al., 2023a) uses PDDL to formally describe planning tasks and relies on prompts to generate the problem code. LLMs-World-Models-for-Planning (Guan et al., 2023) takes the LLMs as an intermediate layer. It also brings in human experts to modify the generated PDDL code. LLM+AL (Ishay and Lee, 2025) expounds on the limitations of the PDDL and proposes Action Language (AL) as an alternative. This language is designed to be more flexible than PDDL, capable of expressing complex causal relationships, temporal constraints, and uncertain events, while maintaining the rigor of formalization.

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

3 Tasks and Benchmarks

This section focuses on the symbolic reasoning tasks of LLMs and their associated datasets and benchmarks. Each task faces unique challenges and has specific requirements. Detailed information about these datasets is presented in Table 1.

3.1 Logical Reasoning

Logical reasoning tasks (Pan et al., 2023; Xu et al., 2024; Lyu et al., 2023a) require LLMs to infer the truth value of a conclusion from a set of rules and conditions. This critically depends on the LLMs' capability to understand logical rules and conditions while consistently upholding rigor throughout

Domains	Benchmarks	Size	Representative Works
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., 2025)
	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., 2025; Li et al., 2024b; Bao et al., 2024)
	CLUTRR (Sinha et al., 2019)	10K	(Ye et al., 2024; Yang et al., 2023b)
Mathematical Reasoning	GSM8K (Cobbe et al., 2021)	8.5K	(Borazjanizadeh and Piantadosi, 2024; Lyu et al., 2023a; 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., 2023a; Chen et al., 2022; Leang et al., 2024)
	SVAMP (Patel et al., 2021)	1K	(Lyu et al., 2023a; Chen et al., 2022; Gao et al., 2023; Leang et al., 2024)
	ASDiv (Miao et al., 2020)	2305	(Lyu et al., 2023a; 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)
Spatial Reasoning	StepGame (Shi et al., 2022)	6.1K	(Wang et al., 2024a; Yang et al., 2023b)
	SparTUN (Mirzaee and Kordjamshidi, 2022)	50K	(Wang et al., 2024a)
Planning	International Planning Competition (IPC) domains (Seipp et al., 2022)	-	(Liu et al., 2023a; Guan et al., 2023)
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)
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, mark their sizes, and note some representative works that used them for evaluation.

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 (Olausson et al., 2023; Tan et al., 2024; Ye et al., 2024).

The content of the datasets used for logical reasoning tasks mainly includes rules, conditions, conclusions, and conclusion validity labels. 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.

3.2 Mathematical Reasoning

Mathematical tasks (Zhou et al., 2023; He-Yueya et al., 2023; Wang et al., 2023a) encompass a wide range of problems, including algebra, geometry, and calculus. 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, by guiding LLMs to focus on generating symbolic representations (such as mathematical formulas or Python code) and leveraging these representations to delegate specific computational tasks to external solvers, not only is the burden on LLMs reduced, but the accuracy of mathematical reasoning is also significantly enhanced (Wang et al., 2023b; Chen et al., 2022).

The content of the datasets used for mathematical reasoning tasks consists of problem descriptions, rationales, and answers. The ASDiv (Miao et al., 2020), SVAMP (Patel et al., 2021), and GSM8K (Cobbe et al., 2021) are primarily composed of primary-school-level math word problems. Math (Hendrycks et al., 2021) mainly consists of challenging math competition problems. AQuA (Ling et al., 2017) contains multiple-choice math reasoning questions and detailed explanations. 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.

3.3 Spatial Reasoning

Spatial reasoning (Hu et al., 2024a; Xiao et al., 2025) is a fundamental aspect of human cognition, enabling human to interact effectively with the 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 pose a significant challenge for LLMs. It is usually hard for LLMs to understand the complex relationship descriptions formed by natural language in spatial reasoning. Utilizing specialized symbolic expressions to model spatial relationships helps to achieve reliable results (Yang et al., 2023b; Wang et al., 2024a).

The data format of the spatial reasoning datasets are 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.

3.4 Planning

At the heart of planning tasks (Liu et al., 2023a; Guan et al., 2023; Ishay and Lee, 2025) is crafting a viable path from the initial state to the goal. This requires guiding the system or environment towards the desired outcome based on the initial state, through a series of carefully selected 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 enhances the executability and reliability of plans.

Planning datasets contain a scenario in which some decisions need to be made by robots. Some works conduct experiments in the domains (e.g., GRIPPERS, TYREWORLD) of the International Planning Competition (IPC) (Seipp et al., 2022).

3.5 Table Reasoning

The task of table reasoning (Zhang et al., 2023; Zhao et al., 2024; Nahid and Rafiei, 2024b; Ye et al., 2023; Zhang et al., 2024) aims to enable models to generate corresponding results as answers based on task requirements when receiving one or more tables as input (Wang et al., 2024d).

The complexity and huge volume of information in table may obscure key details, potentially weakening the decision-making ability of LLMs (Liu et al., 2023c). Compared to relying on natural language processing, employing domain-specific symbolic languages designed for structured data, such as SQL, can effectively reduce the burden on LLMs.

The content of the datasets used for table reasoning tasks 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.

3.6 Others

Given that multi-modal reasoning tasks (Surís et al., 2023; Hu et al., 2024b) typically span multiple categories discussed previously, they cannot be strictly classified into a single specific category. The main challenge in multi-modal reasoning lies in how to efficiently integrate information from different modalities. Leveraging symbolic languages enable seamless interaction and collaborative cooperation among different modalities, thereby enhancing the overall reasoning capabilities(Gupta and Kembhavi, 2023).

The content of the datasets used for multi-modal reasoning tasks consists of visual information and textual information. The NLVR2 dataset (Suhr et al., 2018) consists of images and their corresponding descriptive sentences. The CoVR dataset (Ventura et al., 2024) is a dataset for composite video retrieval tasks.

Some datasets are challenging to precisely match with specific tasks. We provide a brief introduction here. 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).

4 Techniques

In general, four typical techniques are used to enhance the reasoning capabilities of LLMs, including task decomposition, symbolic translation, leveraging external solvers, and self-revision. The typical processes of enhancing LLMs with symbolic reasoning are shown in Figure 2. Many works adopt task decomposition as a fundamental technique. Symbolic translation and leveraging external solvers are usually used together. The mapping relationship between techniques and tasks is shown in Table 2.

4.1 Task Decomposition

By decomposing problems into smaller and manageable sub-problems, LLMs can handle these problems effectively, which makes the problem-solving process clear and trackable. Task decomposition can serve as a fundamental technique, followed by the deployment of symbolic languages. Bridge (Wang et al., 2023a) improves the accuracy of generating equations by decomposing complex problems into independent sub-problems. VISPROG (Gupta and Kembhavi, 2023) addresses complex tasks by decomposing them into multiple modules, each processing using visual models, Python, and other tools.

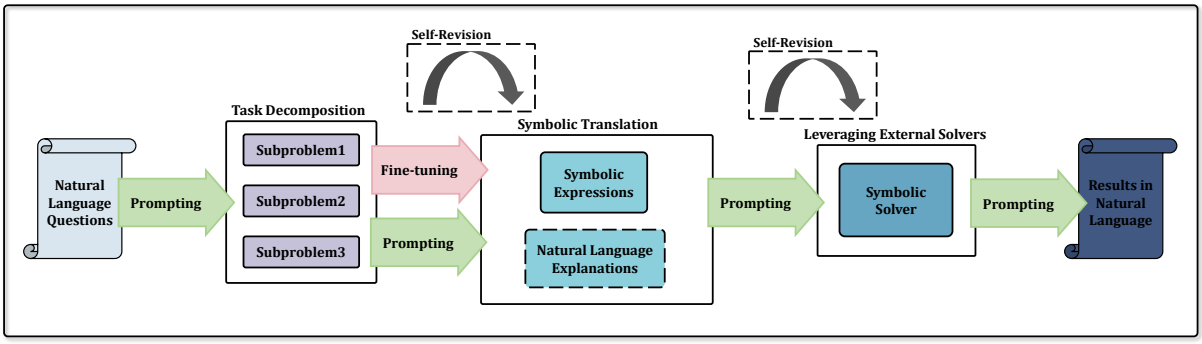

Figure 2: Typical processes of enhancing LLMs with symbolic reasoning.

Task decomposition is widely used in table reasoning. The implementation forms of task decomposition in table reasoning include problem decomposition and table decomposition. Chain-of-Table (Wang et al., 2024d) framework decomposes tables by dynamically generating a chain of table operations, and presents intermediate reasoning results in the form of structured tables. Aiming at long-form table question answering, TAPERA (Zhao et al., 2024) decomposes complex questions into sub-problems through three modules: QA content planner, executable table reasoner, and answer generator. In TabSQLify (Nahid and Rafiei, 2024b) and ReAcTable (Zhang et al., 2023), symbolic languages have been used to decompose the large tables into small sub-tables containing only the essential information required to answer questions. DATER (Ye et al., 2023) and ALTER (Zhang et al., 2024) adopt a dual decomposition mechanism, extracting sub-tables relevant to the question from large tables and breaking down complex problems into logical sub-problems. The issue of table structure has been addressed by NormTab (Nahid and Rafiei, 2024a), which optimizes tables through value normalization and structural normalization, and decomposes tables into sub-tables.

4.2 Symbolic Translation

In most of the works, symbolic translation appears together with leveraging external solvers. The process of these two techniques is first to use LLMs to translate the natural language into symbolic expressions, then select an appropriate solver to address the problem. There are some differences in the implementation forms of symbolic translation. Some works use prompts to guide LLMs to conduct symbolic translation, while some works adopt fine-tuning to enhance the translation capabilities of LLMs.

Translating natural language into logical expressions is a prevalent practice in logical reasoning. Logic-LM (Pan et al., 2023) and LINC (Olausson et al., 2023) generate a symbolic representation for the input problem with LLMs via in-context learning. Faithful CoT (Lyu et al., 2023b) prompts LLMs to translate the problems into a reasoning chain, which interleaves natural language comments and symbolic language programs. SatLM (Ye et al., 2024) uses LLMs to parse natural language problems into declarative task specifications (e.g., logical formulas), which have relative solvers.

Programming symbolic languages and mathematical symbolic languages can represent quantitative relationships and are suitable for solving mathematical problems. PoT (Chen et al., 2022) prompts LLMs to generate Python code. Considering the differences between programming symbolic languages, MultiLingPoT (Li et al., 2024a) fine-tunes LLMs to generate code in four distinct programming symbolic languages. Declarative-math-word-problem (He-Yueya et al., 2023) translates natural language problems into systems of equations according to set principles. Bridge (Wang et al., 2023a) erases information irrelevant to equation generation from the sub-problems and transforms them into equations.

Spatial reasoning and planning have domain-specific symbolic languages. CoS has been proposed by (Hu et al., 2024a), leveraging specific symbols to simplify the spatial-relationship information expressed in natural language in CoT. [LLM]+ASP and DSPy-based LLM+ASP (Yang et al., 2023b; Wang et al., 2024a) introduces ASP (Answer Set Programming), translates the natural language into ASP code. VAP (Xiao et al., 2025) guides the LLMs to call upon Multi-Modal Large Language Model (MLLM) to understand image information and assist in plan generation.

Compared to other tasks, planning is more dependent on symbolic languages because it is necessary to ensure that the plans are executable and reliable.

Tasks	Papers	Techniques
Logical Reasoning	Logic-LM (Pan et al., 2023)	Symbolic Translation, Leveraging External Solvers
	SymbCoT (Xu et al., 2024)	Self-Revision
	LINC (Olausson et al., 2023)	Symbolic Translation, Leveraging External Solvers
	AMR-LDA (Bao et al., 2024)	Symbolic Translation
	Faithful CoT (Lyu et al., 2023b)	Symbolic Translation, Leveraging External Solvers
	SatLM (Ye et al., 2024)	Symbolic Translation, Leveraging External Solvers
	LoT (Liu et al., 2025)	Symbolic Translation
	AMR-LDA (Bao et al., 2024)	Symbolic Translation
	Thought-Like-Pro (Tan et al., 2024)	Symbolic Translation
Mathematical Reasoning	PoT (Chen et al., 2022)	Symbolic Translation, Leveraging External Solvers
	CSV (Zhou et al., 2023)	Leveraging External Solvers, Self-Revision
	MultiLingPoT (Li et al., 2024a)	Symbolic Translation, Leveraging External Solvers
	Declarative-math-word-problem (He-Yueya et al., 2023)	Symbolic Translation
	Bridge (Wang et al., 2023a)	Task Decomposition, Symbolic Translation
Spatial Reasoning	CoS (Hu et al., 2024a)	Symbolic Translation
	[LLM]+ASP (Yang et al., 2023b)	Symbolic Translation, Leveraging External Solvers
	DSPy-based LLM+ASP (Wang et al., 2024a)	Symbolic Translation, Leveraging External Solvers, Self-Revision
	VAP (Xiao et al., 2025)	Symbolic Translation, Leveraging External Solvers
Planning	LLM+P (Liu et al., 2023a)	Symbolic Translation, Leveraging External Solvers
	LLMs-World-Models-for-Planning (Guan et al., 2023)	Symbolic Translation, Leveraging External Solvers, Self-Revision
	LLM+AL (Ishay and Lee, 2025)	Symbolic Translation, Leveraging External Solvers, Self-Revision
Table Reasoning	Chain-of-Table (Wang et al., 2024d)	Task Decomposition
	TAPERA (Zhao et al., 2024)	Task Decomposition, Leveraging External Solvers
	TabSQLify (Nahid and Rafiei, 2024b)	Task Decomposition
	ReAcTable (Zhang et al., 2023)	Task Decomposition
	DATER (Ye et al., 2023)	Task Decomposition
	ALTER (Zhang et al., 2024)	Task Decomposition
	NormTab (Nahid and Rafiei, 2024a)	Task Decomposition
Others	VISPROG (Gupta and Kembhavi, 2023)	Task Decomposition
	CodeVQA (Subramanian et al., 2023)	Leveraging External Solvers
	CodeSteer (Chen et al., 2025)	Self-Revision
	ViperGPT (Surís et al., 2023)	Leveraging External Solvers
	PAL (Gao et al., 2023)	Leveraging External Solvers

Table 2: Techniques used in different reasoning tasks.

Symbolic languages enable flexible construction of planning algorithms, allowing for the precise definition and optimization of state transitions. The approach proposed by LLM+P (Liu et al., 2023a) converts natural language into PDDL using LLMs. LLMs-World-Models-for-Planning (Guan et al., 2023) incorporates corrections for errors in the content translated by LLMs. LLM+AL (Ishay and Lee, 2025) introduces an AL called BC+ (Babb and Lee, 2020) as an alternative to PDDL and prompts LLMs to translate natural language into AL.

Logical expansion based on symbolic representation provides a more complete semantic expression, reducing the risk of semantic conversion errors in LLMs (Liu et al., 2025). LoT (Liu et al., 2025) utilizes LLMs to extract sentences containing conditional reasoning relationships from the input and implement logical expansion using Python. According to AMR-LDA (Bao et al., 2024), 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 Thought-Like-Pro (Tan et al., 2024) to enable LLMs to mimic the reasoning trajectories generated by the Prolog logic engine.

4.3 Leveraging External Solvers

Leveraging external solvers is a technique that uses solvers to address formal symbolic expressions. This technique has effectively addressed the limitations of LLMs in terms of precise computation. By guiding the LLMs to generate content in the input format required by these solvers, reliable results can be obtained.

For logical reasoning problems, after obtaining the symbolic expressions, using a solver to derive the answers is prevalent. LINC (Olausson et al., 2023) focuses on first-order logic and Logic-LM (Pan et al., 2023) adopts various solvers to address different kinds of problems. Faithful CoT (Lyu et al., 2023b) incorporates external solvers and ensures that the answer is the deterministic result of executing the reasoning chain. SatLM (Ye et al., 2024) adopts relative solvers to derive answers based on logical formulas.

Programming symbolic languages are suitable for solving mathematical problems. PoT (Chen et al., 2022) executes the generated Python code to solve mathematical problems. PAL (Gao et al., 2023) and CSV (Zhou et al., 2023) use generated Python code as intermediate reasoning steps. Mul-

tiLingPoT (Li et al., 2024a) considers multiple programming symbolic languages, selects an appropriate one from four different options.

Spatial reasoning and planning require considering complex spatial relationships, and have exclusive symbolic languages and solvers. [LLM]+ASP (Yang et al., 2023b) and DSPy-based LLM+ASP (Wang et al., 2024a) obtain results using an ASP solver. Visual Question Answering (VQA) is transformed into a modular code generation task, where CodeVQA (Subramanian et al., 2023) and ViperGPT (Surís et al., 2023) prompt the LLMs to generate Python code that invokes the APIs of visual language models (VLMs) to process images. LLM+P (Liu et al., 2023a) and LLMs-World-Models-for-Planning (Guan et al., 2023) employ a planner to generate a plan after the symbolic translation. LLM+AL (Ishay and Lee, 2025) invokes a BC+ solver to obtain plans.

4.4 Self-Revision

For some complex problems, due to issues such as hallucinations, using LLMs to conduct symbolic translation or leveraging external solvers may result in errors. Self-Revision can be used multiple times within the framework to correct these errors. Logic-LM (Pan et al., 2023) designs a module to modify inaccurate logical formulations using the error messages from the symbolic solver as feedback. SymbCoT (Xu et al., 2024) designs a Verifier in the framework. For the found invalid logic, the Verifier refines the reasoning steps. CSV (Zhou et al., 2023) proposes "code-based explicit self-verification", which introduces an additional verification stage. Iteratively refining the generated ASP programs and employing the DSPy framework, DSPy-based LLM+ASP (Wang et al., 2024a) manages complex workflows and optimizes prompts effectively. By fine-tuning a small model CodeSteerLLM as an assistant, combined with a symbolic checker and a self-answer verifier, CodeSteer (Chen et al., 2025) guides LLMs to switch between text reasoning and code generation and continuously refines the results.

For the planning task, it is prone to result in factual errors and syntax errors when generating plans. Introducing a self-revision module is a prevalent practice in planning. LLMs-World-Models-for-Planning (Guan et al., 2023) combines the PDDL verification tool with the feedback from human domain experts to correct model errors. LLM+AL (Ishay and Lee, 2025) introduces multiple verification processes to ensure the executability and correctness of the plan.

5 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., 2023b; Liu et al., 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.

Multi-Symbolic Language Integration. Reasoning tasks are inherently complex and typically cannot be effectively addressed using a single language alone. Multi-Symbolic language integration can fully leverage the strengths of each symbolic language, but current methods are superficial in integration and lack flexibility (Li et al., 2024a; Pan et al., 2023; Zhang et al., 2023). We encourage researchers to explore better ways of multilingual integration to resolve the grammatical and semantic conflicts between symbolic languages.

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. The currently proposed methods of generating code in multiple steps (Zhou et al., 2023) and human-like debugging (Zhong et al., 2024) cannot fully resolve the problem. We encourage researchers to explore alternatives that either replace symbolic languages altogether or imbue LLMs with inherent structured and rigorous reasoning capabilities. A possible direction is to design a neuro-symbolic framework that enables LLMs to implicitly align free-form reasoning processes with symbolic representations.

6 Conclusion

In this paper, we conduct a systematic review of the symbolic reasoning in LLMs. We summarize the types of symbolic languages used in reasoning, illustrate different reasoning tasks and relative benchmarks, and discuss the typical techniques for enhancing the symbolic reasoning capabilities of LLMs. We also discuss the promising future directions of symbolic reasoning. We hope 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.

7 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: symbolic languages, tasks and benchmarks, and techniques. 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

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

Joseph Babb and Joohyung Lee. 2020. Action language +. *Journal of Logic and Computation*, 30(4):899–922.

Qiming Bao, Alex Yuxuan Peng, Zhenyun Deng, Wanjun Zhong, Gaël Gendron, Timothy Pistotti, Neşet Tan, Nathan Young, Yang Chen, Yonghua Zhu, Paul Denny, Michael Witbrock, and Jiamou Liu. 2024. Abstract Meaning Representation-based logic-driven data augmentation for logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5914–5934, Bangkok, Thailand. Association for Computational Linguistics.

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

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*.

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, and Pascal Hitzler. 2021. Towards bridging the neuro-symbolic gap: Deep deductive reasoners. *Applied Intelligence*, 51:6326–6348.

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*.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33.

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, Yuting Zeng, Jiaxing Wang, Xingyu Wang, Hailong Yang, and Jing Li. 2025. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10168–10185, Albuquerque, New Mexico. Association for Computational Linguistics.

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*.

Xianzhen Luo, Qingfu Zhu, Zhiming Zhang, Libo Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Python is not always the best choice: Embracing multilingual program of thoughts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7185–7212.

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

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023b. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

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*.

Jendrik Seipp, Álvaro Torralba, and Jörg Hoffmann. 2022. Pddl generators.

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*.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2018. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*.

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*.

Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. 2024. Covr: Learning composed video retrieval from web video captions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5270–5279.

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.

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*.

Zhun Yang, Adam Ishay, and Joohyung Lee. 2023b. 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*.