Empowering LLMs with Logical Reasoning: A Comprehensive Survey

Fengxiang Cheng[1] , **Haoxuan Li**[2,3,4*] , **Fenrong Liu**[3,1] , **Robert van Rooij**[1] ,
Kun Zhang[4,5] and **Zhouchen Lin**[6,7,8*]
[1]Institute for Logic, Language and Computation, University of Amsterdam
[2]Center for Data Science, Peking University
[3]Tsinghua-UvA JRC for Logic, Department of Philosophy, Tsinghua University
[4]Machine Learning Department, MBZUAI
[5]Department of Philosophy, CMU
[6]State Key Lab of General AI, School of Intelligence Science and Technology, Peking University
[7]Institute for Artificial Intelligence, Peking University
[8]Pazhou Laboratory (Huangpu), Guangzhou
{f.cheng, r.a.m.vanrooij}@uva.nl, hxli@stu.pku.edu.cn, fenrong@tsinghua.edu.cn,
kunz1@cmu.edu, zlin@pku.edu.cn

Abstract

Large language models (LLMs) have achieved remarkable successes on various tasks. However, recent studies have found that there are still significant challenges to the logical reasoning abilities of LLMs, which can be categorized into the following two aspects: (1) **Logical question answering**: LLMs often fail to generate the correct answer *within* a complex logical problem which requires sophisticated deductive, inductive or abductive reasoning given a collection of premises. (2) **Logical consistency**: LLMs are prone to producing responses contradicting themselves *across* different questions. For example, a state-of-the-art question-answering LLM Macaw, answers *Yes* to both questions *Is a magpie a bird?* and *Does a bird have wings?* but answers *No* to *Does a magpie have wings?*. To facilitate this research direction, we comprehensively investigate the most cutting-edge methods and propose a detailed taxonomy. Specifically, to accurately answer complex logic questions, previous methods can be categorized based on reliance on external solvers, prompts, and fine-tuning. To avoid logical contradictions, we discuss concepts and solutions of various logical consistencies, including implication, negation, transitivity, factuality consistencies, and their composites. In addition, we review commonly used benchmark datasets and evaluation metrics, and discuss promising research directions, such as extending to modal logic to account for uncertainty and developing efficient algorithms that simultaneously satisfy multiple logical consistencies.

*Haoxuan Li and Zhouchen Lin are the corresponding authors.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance in a broad range of natural language tasks including language generation, classification and translation. However, recent studies have found that there are still significant challenges to the logical reasoning abilities of LLMs. On the one hand, learning syntax, semantics, and world knowledge through tasks such as next-word prediction or masked language modeling does not ensure the logical reasoning ability of LLMs [Luo *et al.*, 2023]. On the other hand, the pre-training corpus of LLMs primarily consists of human-written texts, which lack high-quality logical reasoning samples such as logical deduction and proofs [Morishita *et al.*, 2024]. These challenges significantly limit the applicability of LLMs due to the following two summarized aspects.

LLMs often fail to generate the correct answer in **logical question answering**, which requires sophisticated deductive, inductive or abductive reasoning given a collection of premises and constraints. Specifically, these logical questions can be broadly divided into two categories: (1) Determine whether a statement can be deduced from the given information, namely, output the truth value of the statement: true, false or unknown. For example, the premise and constraints could be *Metals conduct electricity. If something is made of iron, then it is metal. Nails are made of iron.*, followed by the logical question *Is the following statement true, false, or unknown? Nails cannot conduct electricity*. To answer this question correctly, LLMs need to conduct logical reasoning *nails→made of iron→metal→conduct electricity* before concluding that the statement is actually *false*. (2) Find the correct option that can satisfy all the given premises from the multiple choices. Surprisingly, LLaMA-13B achieves 33.63% accuracy under 8-shot prompting on the logical questions dataset FOLIO, which is only slightly better than a random guess from true, false and unknown with an accuracy of 33.33% [Han *et al.*, 2024]. This significantly restricts the application of LLMs in complicated real-world situations, such

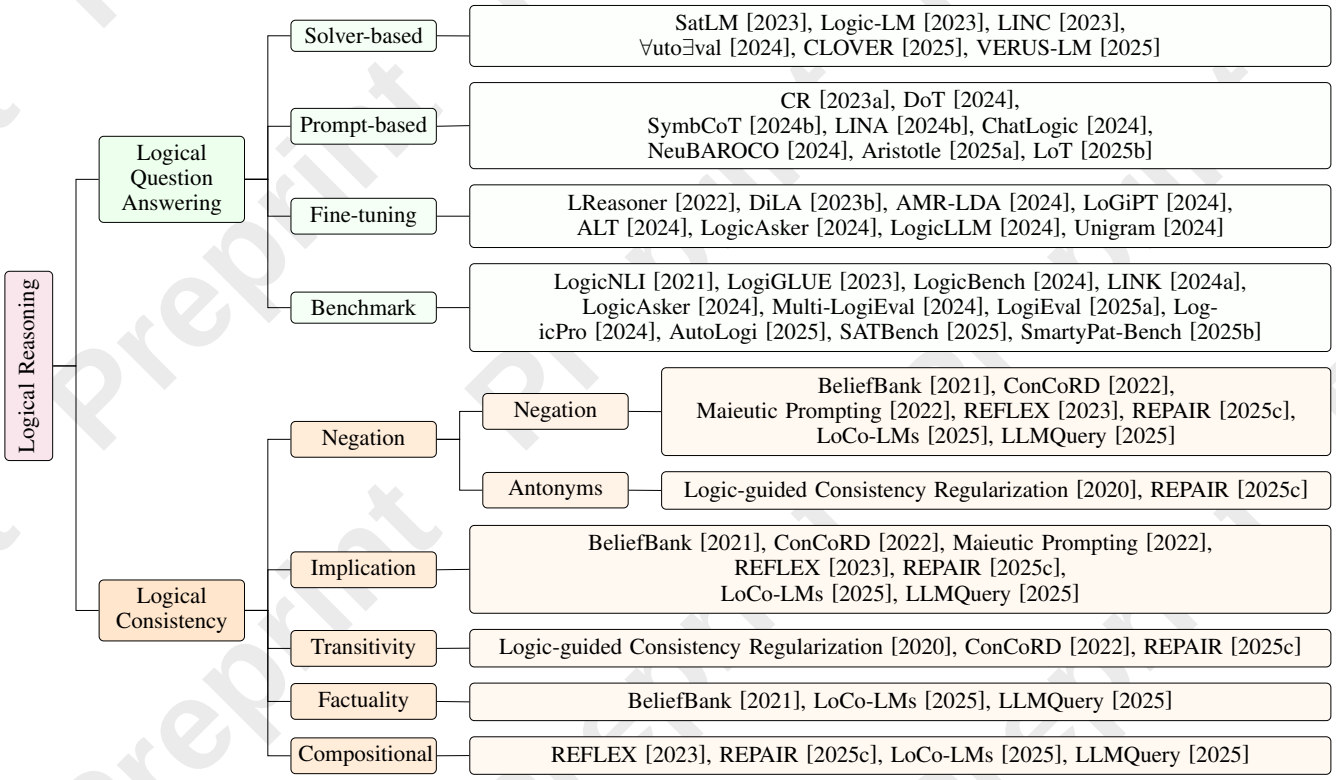

Figure 1: Our taxonomy tree of Logical Reasoning, which is primarily divided into Logical Question Answering and Logical Consistency.

as problem-solving and decision-making.

LLMs are also prone to producing responses contradicting themselves *across* different questions, which is regarded as a violation of **logical consistency**. Note that the form of logical consistency can be diverse. For example, LLaMA-2-70B answers *true* to both questions *Is an albatross an organism?* and *Is an albatross not an organism?* [Calanzone *et al.*, 2025], which violates the negation consistency (see Section 3 for formal definition). In addition, a state-of-the-art Macaw question-answering LLM answers *Yes* to both questions *Is a magpie a bird?* and *Does a bird have wings?* but answers *No* to *Does a magpie have wings?* [Mitchell *et al.*, 2022], which violates the transitivity consistency. Unfortunately, many studies have shown that training on large question-answering datasets alone cannot ensure the logical consistency of LLMs [Kassner *et al.*, 2021; Jung *et al.*, 2022]. As a result, these contradictory outputs concern the reliability and trustworthiness of LLMs, which limits the practical deployment in especially high-stakes scenarios.

In this paper, we consider accurately answering isolated complex logical questions and ensuring logical consistency across outputs to different questions as two sides of the coin in improving the logical reasoning capabilities of LLMs. We comprehensively investigate the most cutting-edge methods for both and propose a detailed taxonomy, as shown in Figure 1. Specifically, for logical question answering, these methods are classified into solver-based, prompt-based, and fine-tuning methods. In general, solver-based methods translate natural language problems to symbolic language expressions, and then solve them via external logical solvers [Olausson *et al.*, 2023]. Prompt-based methods either explicitly model the logical chain when answering the questions [Zhang *et al.*, 2024], or translate natural language into symbolic language by well-defined prompts, and then utilize LLMs for better reasoning [Xu *et al.*, 2024a]. Fine-tuning methods train LLMs with augmented deductive proofs and natural language examples explicitly including the logical reasoning process. For logical consistency, we formulate the most common types of violations, including negation, implication, transitivity, factuality consistencies, and their composites, discuss the state-of-the-art solutions, and review commonly used benchmark datasets and evaluation metrics. Lastly, we discuss promising research directions, such as extending to modal logic to account for uncertainty and developing efficient algorithms satisfying multiple logical consistencies.

To the best of our knowledge, this is the first work to comprehensively investigate the most cutting-edge research on improving LLM logical reasoning capabilities, covering complex logical question answering and logical consistency. A relevant survey work [Zong and Lin, 2024] focuses on categorical syllogisms, as a specific logical reasoning paradigm, but ignores other complex logic inference rules. In addition, another survey [Lam *et al.*, 2024] only discuss the methods based on external tools such as logical solvers.

2 Logical Question Answering

Leveraging LLMs for complex reasoning problems has been a key focus of recent research [Luo *et al.*, 2023; Sun *et*

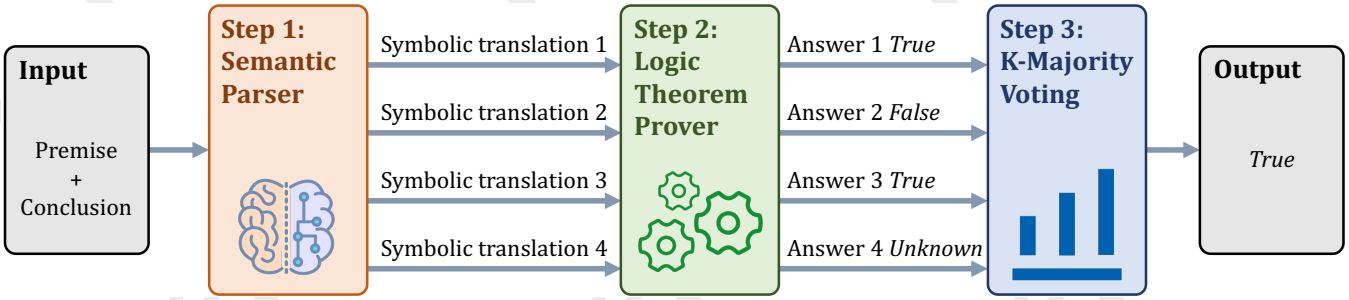

Figure 2: The overview of workflow in solver-based methods for logical question answering.

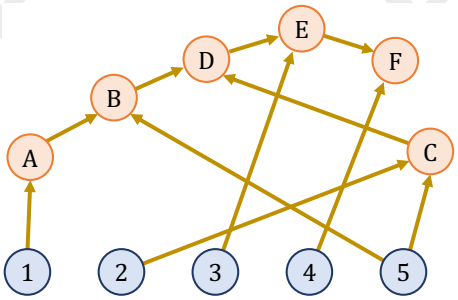

Figure 3: The logical derivation process could be represented as a DAG, where 1-5 represent premises, A-F represent the derivation proceed and every edge represents an individual inference.

al., 2024; Zhang *et al.*, 2024]. Numerous studies have explored methods to enhance the logical reasoning capabilities of LLMs, which can be broadly categorized into solver-based (Sec. 2.1), prompt-based (Sec. 2.2), and fine-tuning methods (Sec. 2.3). In addition, many benchmarks are developed to evaluate the LLM logical reasoning capabilities (Sec. 2.4).

2.1 Solver-Based Methods

The solver-based methods translate **natural language (NL)** questions to **symbolic language (SL)** expressions, and then solve them via external solvers. The workflow is shown in Figure 2, including the following three steps. First, these methods use LLMs to translate NL into SL (such as logic programming (LP), first-order logic (FOL), constraint satisfaction (CSP), or boolean satisfiability (SAT) formulation) that can be recognized by corresponding solvers. Next, they adopt an external solver for logical reasoning to output the desired answer. Finally, LLMs are used to translate the SL answer provided by the solver into NL to generate the final answer based on ensemble algorithms such as majority vote.

Specifically, Faithful Chain-of-Thought [Lyu *et al.*, 2023] translates an NL query into an SL reasoning chain with a deterministic solver (e.g., Python/Datalog interpreter) to improve the faithfulness of LLM-generated answers. Focusing on logical reasoning motivated by this approach, SatLM [Ye *et al.*, 2023] and LINC [Olausson *et al.*, 2023] propose to integrate logical solvers into the reasoning processes of LLMs. In particular, LINC leverages prompting to enable LLMs to translate NL inputs comprising multiple premises and a conclusion into SL expressions. To mitigate translation errors that arise from LLMs, LINC generates multiple SL samples for a single NL problem. The solver then produces the corresponding results as candidate outputs, after which k-majority voting is applied to select the optimal output as the final answer to the logic reasoning question. SatLM exploits the LLM to translate the NL question into a set of logical constraints and then employs an SAT solver to execute the proof plan and derive answers. Unlike LINC and SatLM adopt a single type of SL and use its corresponding solver, LogicLM [Pan *et al.*, 2023] utilizes LLMs to translate NL problems into a task-specific formulation, which could be the LP language, FOL, CSP, and SAT formulation, tailored to different datasets. LogicLM then utilizes the corresponding solvers to obtain the answers and finally translates them back to NL.

While the above methods demonstrate remarkable performance, they heavily rely on the correctness of the translation between NL and SL. To enhance translation accuracy, CLOVER [Ryu *et al.*, 2025] first translates the raw NL paragraph to atomic NL subsentences with their logical dependency structure, then translates to the target SL. VERUS-LM [Callewaert *et al.*, 2025] introduces a self-refinement step that uses feedback from the reasoning engine to correct erroneous logical statements, performing both syntactic refinement and semantic refinement. ∀uto∃val [Karia *et al.*, 2024] adds a bidirectional translation loop of SL→NL (interpretation) and NL→SL (compilation), and then uses logical solvers to automatically check the logical equivalence of SL expressions before and after the translation loop, to ensure the correctness of LLM answers without human annotation.

The limitations of the solver-based methods can be summarized as follows. First, transforming logical questions into formal expressions will result in information loss [Li *et al.*, 2024b; Liu *et al.*, 2025b], leading to unsolvable problems. For example, in the symbolic translation of "When a person reads a book, that person gains knowledge. *Harry* reads *Walden*. Whether this inference is correct: Harry gains knowledge", the solver will output uncertain, since the symbolic translation process loses the vital hidden information "Harry is a person" and "Walden is a book", which can be easily inferred by humans [Liu *et al.*, 2025b]. Second, a small mistake in translation from the NL questions to the SL formulation will severely affect the results derived from SL solvers. For example, GPT 3.5 is not able to add a complete bracket in a logical formula, leading to wrong SL with different meanings [Feng *et al.*, 2024; Lam *et al.*, 2024]. Lastly, as the complexity of the question

increases, the search space expands exponentially, decreasing both the effectiveness and efficiency [Zhang *et al.*, 2023b].

2.2 Prompt-Based Methods

Prompting is a direct and effective technique for eliciting the logical reasoning capabilities of LLMs, which can be broadly divided into two categories. The first is to explicitly model the logical chain when answering the questions. For example, the Chain-of-Thought (CoT) [Wei *et al.*, 2022] prompting strategy enables LLMs to output the reasoning process step-by-step. Based on this, Tree-of-Thought (ToT) [Yao *et al.*, 2024] is proposed to let LLMs self-evaluate and select among multiple inference paths. Graph-of-Thought (GoT) [Besta *et al.*, 2024] enhances LLMs' capabilities through networked reasoning, by modeling LLM thought as a vertex and representing the dependencies between such thoughts via edges. To better mirror human reasoning process, as shown in Figure 3, cumulative reasoning (CR) [Zhang *et al.*, 2023a] decomposes the complex problem into small and manageable components, and modeling the logical derivation process as a direct acyclic graph (DAG) among these components, in which previous propositions are leveraged for effective composition. Diagram-of-Thought (DoT) [Zhang *et al.*, 2024] prompts a single LLM to construct and navigate the DAG by role-specific tokens (e.g., proposer, critic, summarizer).

The second category is to obtain the symbolic expressions and to leverage its advantages by well-defined LLMs' prompts. For example, SymbCoT [Xu *et al.*, 2024a] prompts LLMs to first translate NL problems to SL formulation, then generate a step-by-step solution to address the task with logic inference rules such as modus ponens (MP) rule, and finally to verify the correctness of the translation and answers. Instead of using linguistic token relations for decomposing logical problems, which may lead to disconnected sub-problems and faulty reasoning, the Aristotle framework [Xu *et al.*, 2025a] proposes to exploit the underlying logical structure for decomposition to improve both efficacy and efficiency. In addition, Logic-of-Thought (LoT) [Liu *et al.*, 2025b] instructs LLMs to translate and expands the implicated logical expressions based on the logic rules, then adding these expanded logical information into the input prompt to derive the final answers. LINA [Li *et al.*, 2024b] design LLMs' prompts via hypothetical-deductive reasoning paradigm to reduce the expansive search space compared to the traditional forward reasoning methods. ChatLogic [Wang *et al.*, 2024] integrates symbolic logic programming (e.g., pyDatalog) through interactive prompts, which corrects semantic and syntactic errors of the programming to improve the deductive accuracy.

Despite the transparency and interpretability of prompt-based methods, they still have the following limitations. First, despite the prompt-based methods can correctly translate from NL to SL, reasoning errors remain, suggesting that the source of these errors is not the misinterpretation of the premises but rather the reasoning process itself [Ozeki *et al.*, 2024]. Second, compared with solver-based methods, the prompt-based methods do rely more on the LLM's own reasoning capabilities, leading to sub-optimal performance when facing the complex reasoning tasks. Lastly, solving a problem typically requires numerous inference steps, in which the LLMs are iteratively prompted with various search strategies [Yao *et al.*, 2024], resulting in a significant computational cost [Yang *et al.*, 2025].

2.3 Fine-Tuning Methods

The limited reasoning abilities of LLMs can be attributed to the lack of high-quality reasoning samples (especially logical multi-step deduction or proofs) in the pre-training corpus, which is composed mainly of human-written texts [Morishita *et al.*, 2024]. Human-written texts usually show reflexive thinking rather than rigid reasoning, let alone logical deduction or inference. Intuitively, training LLMs with deductive proofs and NL examples including the explicit logical reasoning process provides an effective way to enhance the intrinsic logical reasoning capabilities of LLMs.

In general, the fine-tuning data for enhancing the logical reasoning capabilities of LLMs are generated via logic rules. Specifically, LogicAsker [Wan *et al.*, 2024] formally defines a comprehensive set of 34 atomic logic rules and 208 extended rules necessary for LLMs to execute formal reasoning based on propositional and predicate logic. By evaluating reasoning ability on each rule, LogicAsker figures out the reasoning rules that LLM performs poorly, based on which creating corresponding in-context-learning examples and fine-tuning data to improve reasoning abilities. Similar to LogicAsker, ALT [Morishita *et al.*, 2024] builds a synthetic logic corpus based on diverse logical reasoning rules such as syllogism, contraposition, and De Morgan's laws, comprising numerous samples of multi-step deduction with unknown facts, diverse linguistic expressions, and challenging distractors. Instead of directly augmenting NL via logical rules, AMR-LDA [Bao *et al.*, 2024] first converts the original NL into an Abstract Meaning Representation (AMR) graph to capture the logical structure of the sentence, then augments NL via logically augmented AMR graphs. In addition, incorporating deeper reasoning steps for fine-tuning the LLMs can further enhance the logical reasoning capabilities of LLMs [Bao *et al.*, 2022].

The effectiveness of fine-tuning methods can be further improved by incorporating more symbolic reasoning process and challenging instances, as well as solution refinement. Specifically, LoGiPT [Feng *et al.*, 2024] empowers LLMs by directly learning the reasoning process of logical solvers, avoiding the risk of no answer to the logical questions when facing parsing errors for the solver-based methods. LReasoner [Wang *et al.*, 2022] proposes to augment challenging instances with literally similar but logically different instances and incorporates contrastive learning to better capture logical information such as logical negative and conditional relationships. DiLA [Zhang *et al.*, 2023b] utilizes a LLM to generate an original solution, and then iteratively refines it by forward and backward passes through a network logic layer involving first-order logic constraints. Moreover, Unigram [Sileo, 2024] shows that simple declarative grammars paired with solvers can outperform complex proof tree generators for reasoning dataset generations. To reduce manual annotation costs, LogicLLM [Jiao *et al.*, 2024] introduces a fully self-supervised framework for integrating logical reasoning capabilities into LLMs.

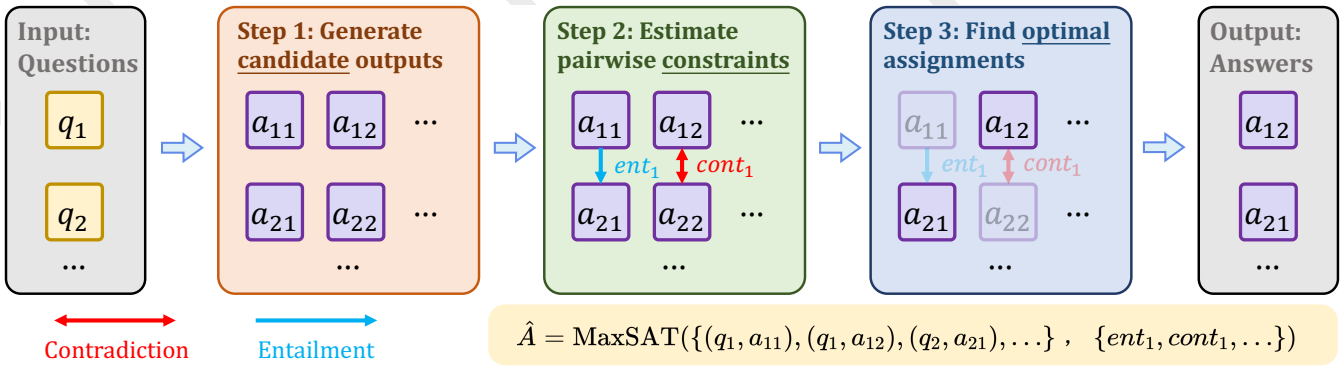

Figure 4: A general framework to enhance the logical consistency of LLM outputs across different questions.

2.4 Evaluation: Tasks and Benchmark Datasets

Typically, a logical reasoning question contains an NL description that outlines a set of propositions or constraints related to certain objects, along with a corresponding question about these objects. The objective is to infer the correct answer to the question based on the given information [Ye *et al.*, 2023]. These question-answering pairs can be typically categorized into two primary types [Luo *et al.*, 2023]:

(1) Logical Statement Checking: Assess whether a given question or statement can be logically inferred from the provided information, producing an output choosing from true, false, or unknown. The formulation includes logic programming, propositional logic, and first-order logic language. Typical datasets include Proofwriter [Tafjord *et al.*, 2021], FOLIO [Han *et al.*, 2024], Ruletaker [Clark *et al.*, 2021], etc.

(2) Multiple Choice Question-Answering: Identify the correct option from multiple choices satisfying all given premises and constraints. The formulation includes constraint satisfaction problems and Boolean SAT formulation. Typical datasets include ReClor [Yu *et al.*, 2020], LogiQA [Liu *et al.*, 2021], AR-LSAT [Zhong *et al.*, 2022], etc.

In addition, several studies have established benchmarks for evaluating LLMs' various types logical reasoning. Specifically, LogicNLI [Tian *et al.*, 2021] assesses FOL reasoning via natural language inference. LogiGLUE [Luo *et al.*, 2023] and LogiEval [Liu *et al.*, 2025a] cover deductive, abductive, inductive, and analogical reasoning. LogicPro [Jiang *et al.*, 2024] synthesizes complex logical data using algorithm problems with Python code, while LINK [Li *et al.*, 2024a] systematically generates long-tail reasoning data via logical rules. LogicBench [Parmar *et al.*, 2024] and Multi-LogiEval [Patel *et al.*, 2024] consist of propositional, first-order, and non-monotonic logic, in which Multi-LogiEval contains samples with reasoning steps up to five. AutoLogi [Zhu *et al.*, 2025] and SATBench [Wei *et al.*, 2025] use logical puzzles to create datasets with varying difficulties, while SmartyPat-Bench [Xu *et al.*, 2025b] generates annotated logical fallacies using Prolog rules.

3 Logical Consistency

Logical consistency requires that LLM responses to different questions do not conflict with each other or with established knowledge bases, in accordance with logical principles [Mitchell *et al.*, 2022; Tafjord *et al.*, 2022]. This prevents LLMs from contradicting themselves in multiple outputs and is consistent with external knowledge [Paleka *et al.*, 2025; Ghosh *et al.*, 2025], improving the reliability and trustworthiness especially in high-stakes real-world deployment so as to strengthening end-users' experience and confidence.

Unfortunately, merely training on QA datasets is insufficient to meet the consistency requirements. To address this issue, many methods have been proposed to improve the logical consistency of LLMs. Hereafter, we will take a closer look at various logical consistencies, including logical relationships among one, two, three statements and their composites. Meanwhile, we will summarize corresponding methods for enhancing LLM logical consistencies as well as the evaluation metrics.

3.1 Negation Consistency

Negation consistency, as the basis of logical consistency, requires that p and $\neg p$ cannot hold simultaneously while one of them is true $p \oplus \neg p$, *i.e.*, $(p \vee \neg p) \wedge \neg(\neg p \wedge p)$ [Kassner *et al.*, 2023]. In NL, there are two forms of negation consistency: the first is to directly add *not* to the statement, and the second is to replace the original adjective in the statement with its antonym [Asai and Hajishirzi, 2020], e.g., replacing *heavier* with *lighter*. Empirically, recent studies found that LLMs are prone to producing responses that contradict themselves across different questions [Kassner *et al.*, 2023; Ghosh *et al.*, 2025]. For example, LLaMA-2-70B answers true to both questions *Is an albatross an organism?* and *Is an albatross not an organism?*.

To address this challenge, Beliefbank [Kassner *et al.*, 2021] stores LLM's raw answers (or beliefs) and flips the beliefs that clash significantly with others using a constraint solver (e.g., MaxSAT solver). The modified beliefs are then used as query context via a feedback mechanism for improving both consistency and accuracy of the LLM when facing relevant questions. Similarly, as shown in Figure 4, ConCoRD [Mitchell *et al.*, 2022] first generates several candidate outputs from the initial model, then estimates soft pairwise inconsistencies between the outputs using natural language inference, and finally finds the most satisfactory output for each question via a MaxSAT solver.

3.2 Implication Consistency

The implication consistency is based on the logical rule $p \rightarrow q, p \vDash q$. It means that given a constraint "$p \rightarrow q$" and the premise p, it can be inferred that "q is True". If the model outputs that "q is False", then we say the output violates implication consistency. For example, given the physical fact that *All Iron (p) is metal (q)*, LLMs are not to be expected to answer *True* to *This material is iron (p)* and *False* to *This material is metal (q)* at the same time.

Intuitively, implication consistency is closely related to the CoT method because each chain can be regarded as a logical implication. The incorrectness of LLM-generated explanations may lead to logically inconsistent and unreliable outputs. To infer a correct output to a question even from the unreliable generations of LLMs, Maieutic Prompting [Jung *et al.*, 2022] offers a novel few-shot inference method that infers a correct answer by inducing the LLMs to generate abductive explanations (e.g., X is true, because ...) for diverse hypotheses and enumerating a structure of explanations — possibly noisy and contradictory — and resolving them with a symbolic inference algorithm.

3.3 Transitivity Consistency

Transitivity represents the logical relationships among three logical statements. Formally, given two premises $p \rightarrow q$ and $q \rightarrow r$, it can be inferred that $p \rightarrow r$, which is regarded as the transitivity consistency. It has been shown that LLMs lack transitivity consistency. For example, the state-of-the-art QA model Macaw answers both *Yes* to *Is a sparrow a bird?* and *Does a bird have feet?*, but answers *No* to *Does a sparrow have feet?* [Mitchell *et al.*, 2022]. From the two affirmative answers, it can be implied that *A sparrow has the feet* according to the transitivity rule, which is incompatible with the negative answer to *Does a sparrow have feet?*.

To mitigate the transitivity inconsistency of LLMs, a logic-guided data augmentation and consistency regularization approach [Asai and Hajishirzi, 2020] leverages logical and linguistic knowledge to augment labeled training data and then uses a consistency-based regularizer to train the LLMs. For example, based on the questions *If a tsunami happens, will wood be more moist?* and *If wood is more moist, is more weathering occurring?* along with answers *More* and *More*, the implicated information *If a tsunami happens, is more weathering occurring?* together with the answer *More* will then be augmented by the transitivity. Similarly, for improving the negation consistency, the antonyms as well as the negation of the statements will also be added to the dataset. In this way, this approach achieves large improvements over previous methods in a variety of logical QA tasks.

3.4 Factuality Consistency

Factuality consistency refers to the degree of alignment between the generated outputs from LLMs and the real-world knowledge base (KB). It is closely related to fact-checking tasks, which involve evaluating the factual accuracy of model outputs by comparing them with reliable knowledge sources, while also detecting potential logical errors or factual inaccuracies. Factuality consistency differs from other logical consistencies in that it requires LLMs not to violate an external knowledge base [Calanzone *et al.*, 2025]. In contrast, the latter requires that logical rules not be violated.

To address the issue of insufficient factuality consistency in existing LLMs under complex query scenarios, an evaluation and improvement framework is proposed through a combination of knowledge graph (KG) integration and supervised fine-tuning [Ghosh *et al.*, 2025]. Using a retrieval-augmented generation (RAG) framework, they provide LLMs with KG contexts (subgraphs extracted via breadth first search) to ground fact-checking queries. The key approach involves supervised fine-tuning on three datasets constructed in the RAG manner to enforce consistency, where the model learns to align responses with logical rules (e.g., commutative laws, negation consistency) and notably KG facts. By leveraging structured KG facts and explicit logical constraints, the approach enhances LLMs' factuality consistency in fact-checking tasks involving propositional logic.

3.5 Compositional Consistency

Compositional consistency refers to the ability to maintain various types of logical consistency when combining multiple facts or logical constraints. Specifically, when a model needs to combine independent facts into a complex chain of reasoning by logical operators (e.g., implication, conjunction, etc.), it should ensure that each step of the derivation conforms to the logical rules, and makes final conclusions self-consistent and logically correct to avoid contradictions. This requires LLMs not only understand the meaning of individual fact, but also to correctly capture the logical relationships when they are combined, avoiding LLMs' inference errors.

For LLMs' outputs to satisfy compositional consistency, LoCo-LMs [Calanzone *et al.*, 2025] introduces a loss based on neuro-symbolic reasoning that teaches an LLM to be logically consistent. The loss encourages the LLM to perform principled probabilistic reasoning at training time by maximizing the probability of LLM's beliefs that comply with the provided set of logical constraints. In addition, REPAIR [Liu *et al.*, 2025c] proposes a universal framework to quantify the compositional logical consistency including three fundamental properties: *transitivity*, *commutativity*, and *negation invariance*, then refines noisy pairwise comparisons using rank aggregation and augments logically consistent comparisons for instruction-tuning. This leads to improved logical consistency while maintaining alignment with human preferences.

3.6 Evaluation

BeliefBank [Kassner *et al.*, 2021] introduces a gold-standard for evaluating the accuracy and consistency of LLM-generated answers, which has been widely adopted by subsequent work [Mitchell *et al.*, 2022; Kassner *et al.*, 2023]. The evaluation framework typically includes:

- A set of entities $S = \{s_m\}_{m=1}^{M}$;
- A set of unary predicates $P = \{P_n\}_{n=1}^{N}$;
- A collection of facts $\{(P_n(s_m))_i\}_{i=1}^{I}$, whose binary truth value $\{0, 1\}$ is known;
- A directed graph of constraints $G(P, E)$, whose edges $(P_n, P_{n'}) \in E$ represent $\forall s_m\,(P_n(s_m) \rightarrow P_{n'}(s_m))$.

From these, the simple yes/no questions are constructed using NL templates. For example, for fact $P_n\left(s_m\right)$, if entity s_m represents a student and predicate P_n represents an ability to play the piano, then the corresponding template-generate QA pair (q_i, a_i) means the question Q: *Is it true that a student is able to play piano?* together with the answer A: *Yes* or *No*.

To assess the logical consistency within LLM-generated answers across different questions, Beliefbank [Kassner *et al.*, 2021], ConCoRD [Mitchell *et al.*, 2022] and many subsequent work apply the complement of *conditional constraint violation metric* τ [Li *et al.*, 2019], that is, $1-\tau$, in which τ is defined as the proportion of *relevant* constraints in G which are *violated*. Specifically, we say a constraint $\forall x\left(P_n(x) \rightarrow P_{n'}(x)\right)$ is relevant if for some entity s_m, there exists some LLM outputs considering the premise $\left(P_n(s_m)\right)_i$ is true, and there also exists some LLM outputs corresponding to the conclusion $\left(P_{n'}(s_m)\right)_j$; the constraint is violated when the LLM outputs considers the conclusion $\left(P_{n'}(s_m)\right)_j$ is false. As for true answers are highly biased towards *No* answers, F1 metric between LLM outputs and the truth value of facts $\left(P_n(s_m)\right)_i$ is employed in practice.

4 Future Research Directions

4.1 Extension to Complex Conditional and Modal Logic Reasoning

Despite state-of-the-art approaches having greatly enhanced several logical reasoning capabilities of LLMs, there still remains a lack of exploration into the more complicated reasoning abilities such as complex conditional and modal logic reasoning. Specifically, sentences involving conditional reasoning can be written in the form of ***If*** *...*, ***then*** *...*. Conditional reasoning is also related to the non-monotonic reasoning, and a recent corresponding work reveals significant limitations in current LLMs in this capability [Leidinger *et al.*, 2024]. In addition, modal logic extends propositional logic by further incorporating ***must*** and ***might*** to account for the certainty and possibility, respectively. For example, *Mary* ***might*** *($\Diamond$) not ($\neg$) have been at the wedding (p)* implies *It's not ($\neg$) the case that Mary* ***must*** *($\Box$) have been at the wedding (p)*, which can be formally formulated as $\Diamond \neg p \vDash \neg \Box p$. Despite expanding to conditional and modal logic reasoning can significantly increase the applicability of LLMs, recent work shows almost all LLMs make some basic mistakes with conditionals or modals, display logically inconsistent judgments across their inference patterns [Holliday *et al.*, 2024]. Therefore, it is still worthwhile to explore new methods to enable LLMs to reason about conditional and uncertain events.

4.2 Higher-Order Logical Reasoning of LLMs

Compared to first-order logic, higher-order logic (HOL) enables reasoning about properties and functions, thus facilitating more complex statements and proofs. First-order logic only quantifies over individual variables, which can only perform reasoning about properties of objects, *not* properties of properties. In contrast, HOL allows quantification over functions and predicates, enabling reasoning about properties of properties and function relationships. For example, a first-order logic statement *All cats are mammals* can be formulated as $\forall x(\mathrm{Cat}(x) \rightarrow \mathrm{Mammal}(x))$. A related HOL example would be *There is a property that all cats have, such that any animal with that property is a mammal*, which can be formulated as $\exists P \forall x(\mathrm{Cat}(x) \rightarrow P(x)) \wedge \forall y(P(y) \rightarrow \mathrm{Mammal}(y))$. This demonstrates the ability to quantify over properties (not just individual objects), allowing for more complex expressions and reasoning on LLMs.

4.3 Efficient Algorithms Satisfying Multiple Logical Consistencies

While numerous methods have been proposed to enhance various types of logical consistency in LLMs, two critical challenges still remain. On the one hand, most methods are limited to enhancing a specific type of logical consistency, failing to satisfy multiple logical consistencies simultaneously. For example, the logic-guided data augmentation [Asai and Hajishirzi, 2020] only simulates reverse samples and transitive logical rules to enhance negation consistency and transitivity consistency of LLMs, respectively. Nonetheless, improving a specific type of logical consistency doesn't necessarily enhance others. On the other hand, enumerating all possible combinations of answers to all questions to verify logical consistencies of LLMs would cost exponential space storage and unexpectedly large computational overhead. Taking the transitivity consistency checking as an example, one need to determine the answer to each of the three associated questions, which requires time complexity $\mathcal{O}(n^3)$. ZebraLogic [Lin *et al.*, 2025] reveals LLMs' logical reasoning performance declines significantly with increasing search space and logical conflicts, named as "curse of complexity". Efficient LLM algorithms are needed in large-scale real-world applications such as accounting [Li and Vasarhelyi, 2024]. Therefore, it is essential to develop more efficient methods that can simultaneously satisfy the various logical consistencies of LLMs.

5 Conclusion

In summary, this survey provides a comprehensive overview of the most cutting-edge methods for enhancing LLM logical reasoning capabilities. Despite impressive advances in most NLP tasks, LLMs still face significant challenges in their logical reasoning abilities, especially in logical question answering and logical consistency. Through a thorough taxonomy, we classify state-of-the-art methods for tackling these challenges, including logical question answering via external solvers, prompting, and fine-tuning, as well as a variety of logical consistency concepts and corresponding solutions, including negation, implication, transitivity, factuality consistencies, and their composites. In addition, we review benchmark datasets and evaluation metrics used in this research direction, which are critical for assessing LLM performance in logical reasoning tasks. Looking ahead, promising research directions include extending LLMs' logical reasoning abilities to modal logic for addressing questions with uncertainty and developing efficient algorithms to satisfy multiple logical consistencies simultaneously. These advancements are pivotal for enhancing the reliability and robustness of LLMs in applications requiring precise logical reasoning, and will ultimately bridge the gap between current abilities of LLMs and the demands of complex reasoning in real-world scenarios.

Acknowledgments

ZL was supported by National Key R&D Program of China (2022ZD0160300), the Beijing Natural Science Foundation (L257007), and the NSF China (62276004). HL was supported by the NSF China (623B2002). FL was supported by the Tsinghua University Initiative Scientific Research Program. RvR was supported by the Dutch Research Council (NWO) (406.18.TW.007).

References

[Asai and Hajishirzi, 2020] Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. In *ACL*, 2020.

[Bao *et al.*, 2022] Qiming Bao, Alex Yuxuan Peng, et al. Multi-step deductive reasoning over natural language: An empirical study on out-of-distribution generalisation. In *NeSy@IJCLR*, 2022.

[Bao *et al.*, 2024] Qiming Bao, Alex Peng, et al. Abstract Meaning Representation-based logic-driven data augmentation for logical reasoning. In *Findings of ACL*, 2024.

[Besta *et al.*, 2024] Maciej Besta, Nils Blach, et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, 2024.

[Calanzone *et al.*, 2025] Diego Calanzone, Stefano Teso, et al. Logically consistent language models via neuro-symbolic integration. In *ICLR*, 2025.

[Callewaert *et al.*, 2025] Benjamin Callewaert, Simon Vandevelde, et al. Verus-LM: a versatile framework for combining LLMs with symbolic reasoning. *arXiv preprint arXiv:2501.14540*, 2025.

[Clark *et al.*, 2021] Peter Clark, Oyvind Tafjord, et al. Transformers as soft reasoners over language. In *IJCAI*, 2021.

[Feng *et al.*, 2024] Jiazhan Feng, Ruochen Xu, et al. Language models can be deductive solvers. In *Findings of NAACL*, 2024.

[Ghosh *et al.*, 2025] Bishwamittra Ghosh, Sarah Hasan, et al. Logical consistency of large language models in fact-checking. In *ICLR*, 2025.

[Han *et al.*, 2024] Simeng Han, Hailey Schoelkopf, et al. FOLIO: Natural language reasoning with first-order logic. In *EMNLP*, 2024.

[Holliday *et al.*, 2024] Wesley H. Holliday, Matthew Mandelkern, et al. Conditional and modal reasoning in large language models. In *ACL*, 2024.

[Jiang *et al.*, 2024] Jin Jiang, Yuchen Yan, et al. LogicPro: Improving complex logical reasoning via program-guided learning. *arXiv preprint arXiv:2409.12929*, 2024.

[Jiao *et al.*, 2024] Fangkai Jiao, Zhiyang Teng, et al. Exploring self-supervised logic-enhanced training for large language models. In *NAACL*, 2024.

[Jung *et al.*, 2022] Jaehun Jung, Lianhui Qin, et al. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *EMNLP*, 2022.

[Karia *et al.*, 2024] Rushang Karia, Daniel Bramblett, et al. ∀uto∃val: Autonomous assessment of llms in formal synthesis and interpretation tasks. *arXiv preprint arXiv:2403.18327*, 2024.

[Kassner *et al.*, 2021] Nora Kassner, Oyvind Tafjord, et al. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *EMNLP*, 2021.

[Kassner *et al.*, 2023] Nora Kassner, Oyvind Tafjord, et al. Language models with rationality. In *EMNLP*, 2023.

[Lam *et al.*, 2024] Long Hei Matthew Lam, Ramya Keerthy Thatikonda, et al. A closer look at tool-based logical reasoning with LLMs: The choice of tool matters. In *ALTC*, 2024.

[Leidinger *et al.*, 2024] Alina Leidinger, Robert Van Rooij, et al. Are llms classical or nonmonotonic reasoners? lessons from generics. In *ACL*, 2024.

[Li and Vasarhelyi, 2024] Huaxia Li and Miklos A Vasarhelyi. Applying large language models in accounting: A comparative analysis of different methodologies and off-the-shelf examples. *JETA*, 2024.

[Li *et al.*, 2019] Tao Li, Vivek Gupta, et al. A logic-driven framework for consistency of neural models. In *EMNLP*, 2019.

[Li *et al.*, 2024a] Huihan Li, Yuting Ning, et al. In search of the long-tail: Systematic generation of long-tail inferential knowledge via logical rule guided search. In *EMNLP*, 2024.

[Li *et al.*, 2024b] Qingchuan Li, Jiatong Li, et al. Leveraging LLMs for hypothetical deduction in logical inference: A neuro-symbolic approach. *arXiv preprint arXiv:2410.21779*, 2024.

[Lin *et al.*, 2025] Bill Yuchen Lin, Ronan Le Bras, et al. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *ICML*, 2025.

[Liu *et al.*, 2021] Jian Liu, Leyang Cui, et al. LogiQA: a challenge dataset for machine reading comprehension with logical reasoning. In *IJCAI*, 2021.

[Liu *et al.*, 2025a] Hanmeng Liu, Yiran Ding, et al. Evaluating the logical reasoning abilities of large reasoning models. *arXiv preprint arXiv:2505.11854*, 2025.

[Liu *et al.*, 2025b] Tongxuan Liu, Wenjiang Xu, et al. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models. In *NAACL*, 2025.

[Liu *et al.*, 2025c] Yinhong Liu, Zhijiang Guo, et al. Aligning with logic: Measuring, evaluating and improving logical consistency in large language models. In *ICML*, 2025.

[Luo *et al.*, 2023] Man Luo, Shrinidhi Kumbhar, et al. Towards LogiGLUE: A brief survey and a benchmark for analyzing logical reasoning capabilities of language models. *arXiv preprint arXiv:2310.00836*, 2023.

[Lyu *et al.*, 2023] Qing Lyu, Shreya Havaldar, et al. Faithful chain-of-thought reasoning. In *IJCNLP-AACL*, 2023.

[Mitchell *et al.*, 2022] Eric Mitchell, Joseph Noh, et al. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *EMNLP*, 2022.

[Morishita *et al.*, 2024] Terufumi Morishita, Gaku Morio, et al. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. In *NeurIPS*, 2024.

[Olausson *et al.*, 2023] Theo Olausson, Alex Gu, et al. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *EMNLP*, 2023.

[Ozeki *et al.*, 2024] Kentaro Ozeki, Risako Ando, et al. Exploring reasoning biases in large language models through syllogism: Insights from the NeuBAROCO dataset. In *Findings of ACL*, 2024.

[Paleka *et al.*, 2025] Daniel Paleka, Abhimanyu Pallavi Sudhir, et al. Consistency checks for language model forecasters. In *ICLR*, 2025.

[Pan *et al.*, 2023] Liangming Pan, Alon Albalak, et al. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of EMNLP*, 2023.

[Parmar *et al.*, 2024] Mihir Parmar, Nisarg Patel, et al. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *ACL*, 2024.

[Patel *et al.*, 2024] Nisarg Patel, Mohith Kulkarni, et al. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. In *EMNLP*, 2024.

[Ryu *et al.*, 2025] Hyun Ryu, Gyeongman Kim, et al. Divide and translate: Compositional first-order logic translation and verification for complex logical reasoning. In *ICLR*, 2025.

[Sileo, 2024] Damien Sileo. Scaling synthetic logical reasoning datasets with context-sensitive declarative grammars. In *EMNLP*, 2024.

[Sun *et al.*, 2024] Hongda Sun, Weikai Xu, et al. DetermLR: Augmenting LLM-based logical reasoning from indeterminacy to determinacy. In *ACL*, 2024.

[Tafjord *et al.*, 2021] Oyvind Tafjord, Bhavana Dalvi, et al. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of ACL*, 2021.

[Tafjord *et al.*, 2022] Oyvind Tafjord, Bhavana Dalvi, et al. Entailer: Answering questions with faithful and truthful chains of reasoning. In *EMNLP*, 2022.

[Tian *et al.*, 2021] Jidong Tian, Yitian Li, et al. Diagnosing the first-order logical reasoning ability through LogicNLI. In *EMNLP*, 2021.

[Wan *et al.*, 2024] Yuxuan Wan, Wenxuan Wang, et al. LogicAsker: Evaluating and improving the logical reasoning ability of large language models. In *EMNLP*, 2024.

[Wang *et al.*, 2022] Siyuan Wang, Wanjun Zhong, et al. Logic-driven context extension and data augmentation for logical reasoning of text. In *Findings of ACL*, 2022.

[Wang *et al.*, 2024] Zhongsheng Wang, Jiamou Liu, et al. Chatlogic: Integrating logic programming with large language models for multi-step reasoning. In *IJCNN*, 2024.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[Wei *et al.*, 2025] Anjiang Wei, Yuheng Wu, et al. SAT-Bench: Benchmarking LLMs' logical reasoning via automated puzzle generation from SAT formulas. *arXiv preprint arXiv:2505.14615*, 2025.

[Xu *et al.*, 2024a] Fangzhi Xu, Zhiyong Wu, et al. Symbol-LLM: Towards foundational symbol-centric interface for large language models. In *ACL*, 2024.

[Xu *et al.*, 2024b] Jundong Xu, Hao Fei, et al. Faithful logical reasoning via symbolic chain-of-thought. In *ACL*, 2024.

[Xu *et al.*, 2025a] Jundong Xu, Hao Fei, et al. Aristotle: Mastering logical reasoning with a logic-complete decompose-search-resolve framework. In *ACL*, 2025.

[Xu *et al.*, 2025b] Zihao Xu, Junchen Ding, et al. Socrates or Smartypants: Testing logic reasoning capabilities of large language models with logic programming-based test oracles. *arXiv preprint arXiv:2504.12312*, 2025.

[Yang *et al.*, 2025] Sen Yang, Xin Li, et al. Neuro-symbolic integration brings causal and reliable reasoning proofs. In *Findings of NAACL*, 2025.

[Yao *et al.*, 2024] Shunyu Yao, Dian Yu, et al. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2024.

[Ye *et al.*, 2023] Xi Ye, Qiaochu Chen, et al. Satlm: Satisfiability-aided language models using declarative prompting. In *NeurIPS*, 2023.

[Yu *et al.*, 2020] Weihao Yu, Zihang Jiang, et al. ReClor: A reading comprehension dataset requiring logical reasoning. In *ICLR*, 2020.

[Zhang *et al.*, 2023a] Yifan Zhang, Jingqin Yang, et al. Cumulative reasoning with large language models. *arXiv preprint arXiv:2308.04371*, 2023.

[Zhang *et al.*, 2023b] Yu Zhang, Hui-Ling Zhen, et al. DiLA: Enhancing LLM tool learning with differential logic layer. *arXiv preprint arXiv:2402.11903*, 2023.

[Zhang *et al.*, 2024] Yifan Zhang, Yang Yuan, et al. On the diagram of thought. *arXiv preprint arXiv:2409.10038*, 2024.

[Zhong *et al.*, 2022] Wanjun Zhong, Siyuan Wang, et al. Analytical reasoning of text. In *Findings of NAACL*, 2022.

[Zhu *et al.*, 2025] Qin Zhu, Fei Huang, et al. AutoLogi: Automated generation of logic puzzles for evaluating reasoning abilities of large language models. *arXiv preprint arXiv:2502.16906*, 2025.

[Zong and Lin, 2024] Shi Zong and Jimmy Lin. Categorical syllogisms revisited: A review of the logical reasoning abilities of LLMs for analyzing categorical syllogisms. In *NLP4Science@EMNLP*, 2024.