Adaptive Selection of Symbolic Languages for Improving LLM Logical Reasoning

Xiangyu Wang[1], **Haocheng Yang**[2], **Fengxiang Cheng**[3*], **Fenrong Liu**[1,3*]

[1]Tsinghua University [2]National University of Singapore [3]University of Amsterdam
f.cheng@uva.nl, fenrong@tsinghua.edu.cn

Abstract

Large Language Models (LLMs) still struggle with complex logical reasoning. To improve LLMs' logical reasoning abilities, a large group of approaches employ external logical solvers, which first translate logical reasoning problems in natural language (NL) into a pre-defined type of symbolic language (SL) expressions and then leverage existing solvers for inference. However, the solution of the same logical reasoning problem can vary significantly when expressed and solved in different SLs. For example, first-order logic language specializes in logical reasoning with categorical syllogisms and complex quantifiers, whereas Boolean satisfiability formalism excels at constraint satisfaction problems. While previous works overlook the varying selection of the target SL for each logical reasoning problem, this is the first paper to propose a method to improve the logical reasoning performance of LLMs by adaptively selecting the most suitable SL for each problem prior to translation. Specifically, for each logical reasoning problem, we prompt LLMs to adaptively select the most suitable SL among first-order logic, logic programming, and Boolean satisfiability formalisms based on the characteristics of the problem. Then we leverage LLMs to translate the problem in NL to the target SL expressions as well as employ the corresponding logical solver to derive the final answer. Experimental results on benchmarks show that our adaptive SL selection method significantly outperforms translating all into a single SL and randomly selecting the SL with an average improvement of 25%.

Introduction

Large language models (LLMs) have exhibited outstanding performance on diverse natural language processing tasks, but they still face significant challenges in complex logical reasoning, which limits their practical applicability in real-world scenarios (Cheng et al. 2025a,b; Lv et al. 2025; Yu et al. 2025). Efforts to enhance the logical question answering (QA) abilities of LLMs can be categorized into three approaches: prompt-based methods (Xu et al. 2024b, 2025), fine-tuning methods (Morishita et al. 2024a; Wan et al. 2024b), and external solver-based methods (Ye et al. 2023; Ryu et al. 2025a). Prompt-based methods leverage LLMs to translate and reason logical question-answering problems directly while fine-tuning approaches enhance logical reasoning performance by constructing synthetic datasets that expose detailed logical deduction steps. This work focuses on the solver-based methods, which translate natural language (NL) questions into symbolic language (SL) expressions and then employ existing solvers for reasoning.

Employing external solvers offers reliable reasoning, as these solvers provide deterministic and verifiable execution (Olausson et al. 2023; Ye et al. 2023). Since the reasoning performance of logical solvers is acutely sensitive to the NL-to-SL translation, numerous approaches focus on the translation stage, including translating the raw NL paragraph to simpler and atomic NL subsentences (Ryu et al. 2025b), self-refinement loops where feedback from a logical solver is used to correct erroneous logical statements (Callewaert, Vandevelde, and Vennekens 2025), and bidirectional translation checks (SL→NL→SL) to automatically verify logical equivalence without human annotation (Karia et al. 2024).

However, previous works only focus on how to translate logical reasoning problems better into a fixed type of SL, overlooking selecting the type of SL that is the most suitable for each logical problem before the translation stage. In fact, the optimal choice of an SL varies significantly depending on the nature of the logical reasoning problem. For instance, First-Order Logic (FOL) language excels at addressing logical reasoning problems requiring categorical syllogisms with complex quantifiers, whereas the Boolean Satisfiability (SAT) formalism is highly proficient in representing constraint satisfaction tasks such as ordering problems. Thus, adopting an inappropriate SL type, which is unable to properly formalize and capture all content of the NL problem, will still lead to lower overall accuracy. We empirically found that for the LogicalDeduction dataset, translating all questions into SAT formulization reaches 90% reasoning accuracy with the corresponding solver, while FOL translation only achieves 42%; conversely, for ProofWriter, FOL achieves 95.50% accuracy but SAT drops to 68.33%. These results intuitively demonstrates that different formalisms are suited to different logical QA problems due to the different expression capabilities and complexity of these languages.

Therefore, to fill this research gap, we are the first to reveal that the selection of the symbolic language significantly impacts translation accuracy and consequently the reasoning performance including the execution rate of solvers. We

[*] Fengxiang Cheng and Fenrong Liu are corresponding authors.

also propose an approach that adaptively selects the most appropriate SL for a given reasoning problem. Our method prompts an LLM to choose the most appropriate SL for each problem by comparing their expressive features, advantages, and disadvantages of several candidate SL. Subsequently, the model then translates the NL query into the chosen SL, and a corresponding logic solver is employed to derive the final answer. Through experiments, we demonstrate the remarkable effectiveness and feasibility of our proposed method.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to reveal and empirically verify that each NL logical reasoning problem corresponds to an optimal SL.
- We propose an adaptive SL selection method to improve LLMs logical reasoning abilities, prompting LLMs to choose the most suitable SL for each logical QA problem and employing the corresponding solver.
- We design experiments showing that (1) comparisons across three types of SL translations of the same dataset verify that each logical problem has an optimal SL, and (2) our methods of adaptive selection of SL significantly outperforms random choice.

Related Work

Logical Reasoning in LLMs

Efforts to enhance the logical reasoning capabilities of LLMs can be categorized and summarized into three approaches: solver-based, prompt-based, and fine-tuning methods (Cheng et al. 2025a; Cheng 2025). Solver-based methods initially translate NL queries into SL formulations and subsequently leverage dedicated solvers to execute the inference task (Lyu et al. 2023; Ye et al. 2023; Olausson et al. 2023; Ryu et al. 2025a). Prompt-based techniques follow two main pipelines: one involves the explicit generation of NL reasoning steps to derive the conclusion (Wei et al. 2022; Yao et al. 2024; Zhang, Yuan, and Yao 2024), while the other prompts LLMs to perform NL-to-SL translation, sequential reasoning, and answer validation (Xu et al. 2024a; Liu et al. 2025; Li et al. 2024; Fu et al. 2025). Finally, fine-tuning strategies enhance LLMs' logical reasoning capability either by synthetically creating datasets that feature logical derivation proofs (Bao et al. 2024; Morishita et al. 2024b) or by augmenting existing training corpora with logical reasoning examples (Feng et al. 2024; Wan et al. 2024a; Jiao et al. 2024). In contrast to these methods generally considered only one SL formalization for all logical QAs, in this paper we introduce a crucial preliminary step before the translation process: adaptively selecting the most suitable SL for each distinct problem.

Methods to Improve the Accuracy of Translation

A primary challenge for solver-based reasoning systems is the accuracy of the translation from NL to SL (Lyu et al. 2023; Ye et al. 2023; Olausson et al. 2023; Ryu et al. 2025a). The inference performance of solvers is critically dependent on the correctness of this initial translation step. Several strategies have been proposed to mitigate translation errors. CLOVER (Ryu et al. 2025b) decomposes complex NL statements into simpler units before converting them into the SL. VERUS-LM (Callewaert, Vandevelde, and Vennekens 2025) employs an iterative refinement loop where the reasoning engine provides feedback to correct both syntactic and semantic errors in the generated logical forms. ∀uto∃val (Karia et al. 2024), proposes a self-verification technique based on a round-trip translation (SL→NL→SL) to check for logical consistency without manual supervision. While these techniques focus on refining the translation process, our approach addresses a more fundamental aspect: selecting the optimal target SL before translation begins, which directly influences both translation feasibility and accuracy.

Logical Question Answering Problem Setup

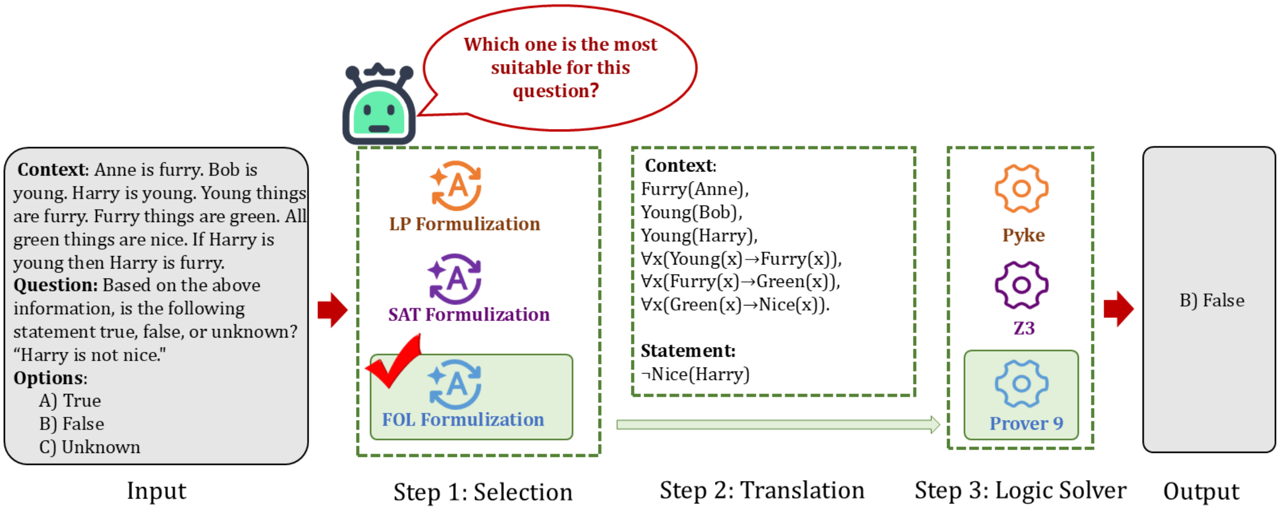
Logical QA tasks are centered around assessing that: according to logical inference rules, whether a given statement can be validly inferred from provided contextual information through strict logical deduction. For LLMs tackling such tasks, the core requirement is to accurately answer the truth status of this statement among three options: *true* (when the statement can be logically deduced by the premises), *false* (when it contradicts the given information), or *unknown* (when the given evidence is insufficient to confirm or refute it). An illustrative example is provided below.

An example for Logical QA

Contexts (Premises):
• The tiger is big.
• If something is big then it visits the rabbit.
• The rabbit visits the tiger.
• If something visits the rabbit then the rabbit needs the lion.
• If something sees the tiger then it is rough.
• ......
Question: Based on the above information, is the following statement true, false, or unknown? The rabbit does not need the lion.
Options: A) True B) False C) Unknown
Answer: B) False

Proposed Method

Our method for solving logical QA tasks consists of three main stages: an adaptive symbolic language selection step, a translation step, and a reasoning step. As illustrated in Figure 1, given a natural language logical reasoning problem, we prompt an LLM first determines the most suitable SL among three candidates: FOL, LP, and SAT. The problem is then translated into the chosen SL by the LLM. Finally, an external logical solver is used to perform the reasoning and generate the final answer. This pipeline ensures that each problem is processed using the most appropriate formalization, leading to higher accuracy of final answer.



Context: Anne is furry. Bob is young. Harry is young. Young things are furry. Furry things are green. All green things are nice. If Harry is young then Harry is furry.
Question: Based on the above information, is the following statement true, false, or unknown? "Harry is not nice."
Options:
A) True
B) False
C) Unknown

Context:
Furry(Anne),
Young(Bob),
Young(Harry),
∀x(Young(x)→Furry(x)),
∀x(Furry(x)→Green(x)),
∀x(Green(x)→Nice(x)).

Statement:
¬Nice(Harry)

Figure 1: The framework of our methods to adaptively select symbolic languages to translate logical reasoning problems.

Three Formalization of Symbolic Languages

We use three distinct symbolic languages tailored for different types of logical reasoning problems. **FOL** is suitable for problems involving complex quantification and relationships between entities. Its expressive power allows for a direct translation of premises and rules that contain universal or existential quantifiers. **LP** is well-suited for deductive reasoning based on a set of facts and rules. They are particularly effective for problems where a clear chain of forward or backward inference can be established. **SAT** formalizes problems as constraints to check for satisfiability. It is highly efficient for problems that can be reduced to boolean constraints, such as those with a large number of relationships.

Adaptively Selection to the Symbolic Languages

Our framework employs an adaptive selection mechanism that prompts a LLM to choose the most suitable SL for each problem. The prompt provides the LLM with a comparative analysis of the candidate SLs, detailing their distinct expressive features, advantages, and disadvantages. Based on this information and the specific problem at hand, the LLM determines which language—FOL, LP or SAT—is best suited for the translation and reasoning task.

Prompts for Adaptively Selection of SL

You are an expert in symbolic logic and reasoning systems. Your task is to analyze a logic problem and select the most appropriate symbolic language for solving it. You have three symbolic languages to choose from:

1. FOL (First-Order Logic):
   -Best for: Complex quantifiers, mathematical relationships, formal proofs. **-Features**: Universal ($\forall$) and existential ($\exists$) quantifiers, logical operators ($\neg, \vee, \wedge, \rightarrow$), predicates, functions, variables. **-Typical problems**: Mathematical theorems, complex logical relationships, nested quantifications, categorical syllogisms. **-Example patterns**: "For all X, there exists Y such that...", "If and only if...", "All X are Y".
2. **LP (Logic Programming)**:
   -Best for: Deductive reasoning, propositions, relationship between sentences. **-Features**: Fact as a simple statement with predicates and arguments. Rules written in the form of clauses. Query as another fact required to be proved based on known facts and rules. **-Typical problems**: Deductive reasoning, propositional logical reasoning. **-Example patterns**: "If something is X then it is Y".
3. **SAT (Boolean Satisfiability Problem)**: **-Best for**: Constraint satisfaction, spatial/ordering problems, discrete choices. **-Features**: Boolean variables, constraints, position/ordering relationships. **-Typical problems**: Arrangement puzzles, scheduling, spatial reasoning. **-Example patterns**: "X is to the left of Y", "X is between Y and Z".

Given the following logic problem:
Context: `${context}`
Question: `${question}`
Options: `${options}`

Analyze the problem structure carefully and select the symbolic language that best matches the problem.

Translation via LLMs and Reasoning via Solvers

The translation from natural language to the chosen symbolic language is performed by a LLM. This LLM is prompted to convert the given premises and question intoically correct and semantically faithful expressions in the selected SL. The generated symbolic expressions are then passed to their corresponding external solvers. Specifically,

Table 1: Performance comparison of different symbolic language selection strategies on three benchmarks. The best results are highlighted in bold.

		ProntoQA			ProofWriter			LogicalDeduction		
Model	**Strategy**	Overall-Acc	Exec-Rate	Exec-Acc	Overall-Acc	Exec-Rate	Exec-Acc	Overall-Acc	Exec-Rate	Exec-Acc
GPT-4	Chance	50.00%	/	/	33.33%	/	/	20.00%	/	/
	LP	93.60%	87.80%	99.66%	78.83%	98.50%	79.53%	36.33%	74.33%	41.97%
	FOL	98.80%	98.60%	99.59%	95.50%	97.83%	96.88%	42.00%	32.00%	88.75%
	SAT	80.60%	65.20%	96.93%	63.83%	68.33%	77.97%	90.00%	93.67%	94.73%
	Random selection	88.80%	82.20%	97.20%	77.50%	86.83%	84.20%	53.33%	63.67%	72.36%
	Adaptive selection	**99.80%**	**98.80%**	**100.00%**	**96.00%**	**98.17%**	**97.17%**	**91.33%**	**94.33%**	**95.62%**
Claude 3.7	Chance	50.00%	/	/	33.33%	/	/	20.00%	/	/
	LP	94.60%	90.40%	99.31%	79.67%	98.33%	80.22%	37.33%	75.00%	42.87%
	FOL	99.40%	**99.40%**	99.73%	96.33%	98.67%	97.30%	43.67%	33.67%	88.62%
	SAT	82.80%	69.00%	97.80%	65.33%	70.00%	79.17%	93.00%	95.00%	96.20%
	Random selection	92.20%	86.20%	98.93%	79.83%	89.17%	85.33%	58.33%	68.00%	76.68%
	Adaptive selection	**99.80%**	99.20%	**99.85%**	**96.83%**	**98.83%**	**97.52%**	**93.67%**	**95.33%**	**97.13%**

Table 2: Performance comparison on the mixed dataset.

Model		Mixed		
		Overall-Acc	Exec-Rate	Exec-Acc
GPT-4	LP	69.33%	83.33%	76.32%
	FOL	79.67%	76.00%	93.96%
	SAT	78.67%	77.00%	91.89%
	Random selection	70.67%	74.00%	83.41%
	Adaptive selection	**96.00%**	**96.33%**	**98.34%**
Claude 3.7	LP	70.67%	85.67%	76.11%
	FOL	80.00%	77.33%	96.26%
	SAT	80.33%	78.00%	91.77%
	Random selection	76.67%	81.33%	87.65%
	Adaptive selection	**96.67%**	**97.00%**	**98.61%**

we use **Prover9** (McCune 2010) for FOL, **Pyke** (Frederiksen 2008) for LP, and **Z3** (De Moura and Bjørner 2008) for SAT. These solvers execute the logical reasoning, such as theorem proving or satisfiability checking, and return a definitive logical result. This result is then transformed back to the final answer (e.g., True, False, or Unknown).

Experiments

Experimental Setup

The experiments are conducted on GPT-4 (OpenAI 2023) and Claude 3.7 sonnet (Anthropic 2025). We evaluate our method on three distinct logical reasoning benchmarks: **ProntoQA** (Saparov and He 2023), **ProofWriter** (Tafjord, Dalvi, and Clark 2021), and **LogicalDeduction** (Srivastava et al. 2023). The performance is measured by three metrics: Overall-Acc (Overall Accuracy), Exec-Rate (Execution Rate, representing the proportion of total samples for which the solver can read and execute reasoning on the translated results), and Exec-Acc (Execution Accuracy, estimating the proportion of samples with correct reasoning results relative to the samples can be reasoned by the solver). We compare our adaptive selection method against four baselines, including translating all samples into a single SL (including LP, FOL and SAT) and a random selection baseline. A further test is performed on a **mixed** dataset constructed by selecting 100 samples in each dataset (ProntoQA, ProofWriter, and LogicalDeduction) to evaluate the LLM's performance in a setting closed to real-world serving.

Results Analysis

The results from Table 1 demonstrate that our adaptive selection method performs best on all benchmarks among translating all samples into one single SL and randomly selecting the SL. The results also verify our claim that different NL logical reasoning problems correspond to different optimal SL formalizations for translation. Apart from our adaptive selection methods, we find that on the three datasets of ProntoQA, ProofWriter, and LogicalDeduction, FOL (corresponding to the first two) and SAT respectively achieve the highest accuracy. This highlights that a "one-SL-fits-all" approach to translate is sub-optimal, and selecting the appropriate language is crucial for maximizing performance on different logical QA problems.

The comparison between the random selection and adaptive selection methods confirms the effectiveness of our approach. As shown in both Table 1 and Table 2, the adaptive selection method consistently outperforms the random selection method across all benchmarks and the mixed dataset. On the mixed dataset, adaptive selection on GPT-4 achieves a remarkable 96.00% Overall-Acc, a significant improvement over the 70.67% for Random selection. This difference demonstrates that intelligently selecting the most suitable symbolic language for each problem instance is far superior to approaches which fail to consider the SL selection and validates the effectiveness of our adaptive selection method.

Case Study

To illustrate the critical impact of SL selection, we conduct a case study on a specific example from the LogicalDeduction dataset (ID: `logical_deduction_3`). This problem requires choosing the right price order of five fruits based on a set of relational constraints (e.g., "apples are less expensive than peaches", "plums are the cheapest", "watermelons are more expensive than peaches"). The gold-standard answer is "A) The watermelons are the most expensive."

We analyze the translation of this problem into SAT, FOL, and LP respectively. This comparison reveals how an inappropriate SL choice can lead to translation failures and incorrect answers, whereas the correctly selected SL (SAT) successfully solves the problem.

This problem is inherently a constraint satisfaction problem, for which SAT is the optimal formalism. Our adaptive selection method correctly identifies SAT. The successful translation and final answers from the solver are as follows:

```
# Declarations
fruits = EnumSort([Watermelon, Plum,
    ↪ Apple, Peach, Kiwi])
prices = IntSort([1, 2, 3, 4, 5])
price = Function([fruits] -> [prices])

# Constraints
Distinct([f:fruits], price(f))
price(Apple) < price(Peach)
price(Plum) == 1
price(Kiwi) == 2
price(Watermelon) > price(Peach)

# Options
is_valid(price(Watermelon) == 5) #
    ↪ Option A
```

This SAT formulation accurately models the problem as a function mapping each fruit to a unique integer price from 1 to 5. The solver (Z3) processes these constraints and correctly validates that `price(Watermelon) == 5` is the only satisfiable conclusion, yielding the correct answer ‘A’.

When forced to use FOL, the translation becomes overly complex, semantically flawed, and ultimately fails.

```
### Predicates:
Rank(fruit, pos) # pos in {one, ...,
    ↪ five}
Cheaper(x, y)

### Premises:
Rank(plums, five)
Rank(kiwis, four)
Cheaper(apples, peaches)
Cheaper(peaches, watermelons)
# (One rank per fruit / One fruit per
    ↪ rank constraints)
...
#  Higher rank -> more expensive  (10
    ↪ ordered pairs)
Forall X Forall Y (Rank(X, one) &
    ↪ Rank(Y, two) -> Cheaper(Y, X))
Forall X Forall Y (Rank(X, one) &
    ↪ Rank(Y, three) -> Cheaper(Y, X))
...
Forall X Forall Y (Rank(X, four) &
    ↪ Rank(Y, five) -> Cheaper(Y, X))
```

The FOL translation attempts to define the ordering explicitly using `Rank` predicates and some premises to link `Rank` to the `Cheaper` predicate. This formulation contains a critical semantic error: the premise “watermelons are more expensive than peaches” was incorrectly translated to `Cheaper(peaches, watermelons)`. Compounded by this error, the solver encountered an “execution error”, indicating a failure to reason over this complex formalization, resulting in an incorrect final answer (‘E’).

Similarly, the LP translation struggles to capture the global nature of the ranking constraint, leading to a logical flaw in its translation.

```
Predicates:
CheaperThan($x, $y, bool)
MoreExpensiveThan($x, $y, bool)
...
MostExpensive($x, bool)

Facts:
CheaperThan(apple, peach, True)
Cheapest(plum, True)
SecondCheapest(kiwi, True)
MoreExpensiveThan(watermelon, peach,
    ↪ True)

Rules:
...
# Rule 5:
MoreExpensiveThan($a, $b, True) &&
MoreExpensiveThan($a, $c, True) &&
MoreExpensiveThan($a, $d, True) &&
MoreExpensiveThan($a, $e, True) >>>
    ↪ MostExpensive($a, True)
```

The LP formulation defines `MostExpensive` using `Rule 5`, which requires a fruit `$a` to be more expensive than four other fruits (`$b, $c, $d, $e`). However, the solver’s reasoning trace reveals that it incorrectly applies this rule by binding `$b, $c, $d,` and `$e` to the *same* fruit (e.g., binding all to ‘peach’ when `$a` is ‘watermelon’). This faulty reasoning leads the solver to erroneously deduce both `MostExpensive(‘peach’, True)` and `MostExpensive(‘watermelon’, True)`, ultimately returning the incorrect answer ‘D’.

To sum up, for this ordering problem, SAT provides a direct and solvable representation, while FOL and LP resulted in semantically incorrect and wrong answers. This case study clearly demonstrates that the choice of SL should not be arbitrary, but adaptive to each problem.

Conclusion

In this paper, we are the first to reveal that the optimal choice of SL for solver-based logical reasoning dependent on each problem is crucial to the performance of both translation and final reasoning. We introduce an approach to improve LLMs’ logical reasoning capabilities by prompting LLM to adaptively select the most suitable formalism—from FOL, LP, SAT—for each individual question before the translation process. Our experiments confirm the effectiveness of our approach, showing that it significantly improves overall accuracy compared to fixed-SL and random-selection baselines. This work determines that considering SL selection and regarding it as a dynamic decision problem is a vital step to improve LLMs logical reasoning abilities. For future work, we plan to expand our framework by exploring a broader range of SL and aim to establish a more formal, theoretical foundation to guide the SL selection process.

Acknowledgments

FL was supported by the Beijing Natural Science Foundation (No. L257007) and Tsinghua University's Initiative for Advancing First-Class and World-Leading Disciplines in the Humanities and Social Sciences.

References

Anthropic. 2025. Claude 3.7 Sonnet System Card. Technical report, Anthropic PBC. Model/system card.

Bao, Q.; Peng, A.; Deng, Z.; Zhong, W.; Gendron, G.; Pistotti, T.; Tan, N.; Young, N.; Chen, Y.; Zhu, Y.; Denny, P.; Witbrock, M.; and Liu, J. 2024. Abstract Meaning Representation-Based Logic-Driven Data Augmentation for Logical Reasoning. In *Findings of the Association for Computational Linguistics ACL*.

Callewaert, B.; Vandevelde, S.; and Vennekens, J. 2025. VERUS-LM: a Versatile Framework for Combining LLMs with Symbolic Reasoning. *arXiv preprint arXiv:2501.14540*.

Cheng, F. 2025. Enhancing the Logical Reasoning Abilities of Large Language Models. *International Joint Conference on Artificial Intelligence, Doctoral Consortuim*.

Cheng, F.; Li, H.; Liu, F.; van Rooij, R.; Zhang, K.; and Lin, Z. 2025a. Empowering llms with logical reasoning: A comprehensive survey. *International Joint Conference on Artificial Intelligence, Survey Track*.

Cheng, F.; Zhou, C.; Li, X.; Leidinger, A.; Li, H.; Gong, M.; Liu, F.; and Rooij, R. V. 2025b. Mitigating Spurious Correlations via Counterfactual Contrastive Learning. In *Findings of the Association for Computational Linguistics: EMNLP*.

De Moura, L.; and Bjørner, N. 2008. Z3: an efficient SMT solver. In *Proceedings of the Theory and practice of software, international conference on Tools and algorithms for the construction and analysis of systems*.

Feng, J.; Xu, R.; Hao, J.; Sharma, H.; Shen, Y.; Zhao, D.; and Chen, W. 2024. Language Models can be Deductive Solvers. In *Findings of the Association for Computational Linguistics: NAACL*.

Frederiksen, B. 2008. Applying expert system technology to code reuse with pyke. *PyCon: Chicago*.

Fu, W.; Yang, H.; Cheng, F.; and Liu, F. 2025. Efficient First-Order Logic-Based Method for Enhancing Logical Reasoning Capabilities of LLMs. In *NeurIPS Workshop on Foundations of Reasoning in Language Models*.

Jiao, F.; Teng, Z.; Ding, B.; Liu, Z.; Chen, N.; and Joty, S. 2024. Exploring Self-supervised Logic-enhanced Training for Large Language Models. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Karia, R.; Bramblett, D.; Dobhal, D.; Verma, P.; and Srivastava, S. 2024. ∀uto∃val: Autonomous Assessment of LLMs in Formal Synthesis and Interpretation Tasks. *arXiv preprint arXiv:2403.18327*.

Li, Q.; Li, J.; Liu, T.; Zeng, Y.; Cheng, M.; Huang, W.; and Liu, Q. 2024. Leveraging LLMs for Hypothetical Deduction in Logical Inference: A Neuro-Symbolic Approach. *arXiv preprint arXiv:2410.21779*.

Liu, T.; Xu, W.; Huang, W.; Zeng, Y.; Wang, J.; Wang, X.; Yang, H.; and Li, J. 2025. Logic-of-Thought: Injecting Logic into Contexts for Full Reasoning in Large Language Models. In *Proceedings of Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Lv, X.; Wang, H.; Mao, Y.; Liang, K.; Li, H.; Huang, W.; Lan, L.; Chi, H.; Chen, H.; Yang, J.; Cyuanlong; and Yang, W. 2025. Breaking the Gradient Barrier: Unveiling Large Language Models for Strategic Classification. *Advances in Neural Information Processing Systems*.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

McCune, W. 2010. Prover9 and Mace4. *URL: http://www.cs.unm.edu/ mccune/Prover9*.

Morishita, T.; Morio, G.; Yamaguchi, A.; and Sogawa, Y. 2024a. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *Advances in Neural Information Processing Systems*.

Morishita, T.; Morio, G.; Yamaguchi, A.; and Sogawa, Y. 2024b. Enhancing Reasoning Capabilities of LLMs via Principled Synthetic Logic Corpus. *Advances in Neural Information Processing Systems*.

Olausson, T.; Gu, A.; Lipkin, B.; Zhang, C.; Solar-Lezama, A.; Tenenbaum, J.; and Levy, R. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

OpenAI. 2023. GPT-4 Technical Report.

Ryu, H.; Kim, G.; Lee, H. S.; and Yang, E. 2025a. Divide and Translate: Compositional First-Order Logic Translation and Verification for Complex Logical Reasoning. In *The International Conference on Learning Representations*.

Ryu, H.; Kim, G.; Lee, H. S.; and Yang, E. 2025b. Divide and Translate: Compositional First-Order Logic Translation and Verification for Complex Logical Reasoning. In *The International Conference on Learning Representations*.

Saparov, A.; and He, H. 2023. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. In *The International Conference on Learning Representations*.

Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A. A.; Abid, A.; Fisch, A.; Brown, A. R.; Santoro, A.; Gupta, A.; Garriga-Alonso, A.; et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Tafjord, O.; Dalvi, B.; and Clark, P. 2021. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*.

Wan, Y.; Wang, W.; Yang, Y.; Yuan, Y.; Huang, J.-t.; He, P.; Jiao, W.; and Lyu, M. 2024a. LogicAsker: Evaluating and Improving the Logical Reasoning Ability of Large Language Models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Wan, Y.; Wang, W.; Yang, Y.; Yuan, Y.; Huang, J.-t.; He, P.; Jiao, W.; and Lyu, M. R. 2024b. LogicAsker: Evaluating and Improving the Logical Reasoning Ability of Large Language Models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*.

Xu, F.; Wu, Z.; Sun, Q.; Ren, S.; Yuan, F.; Yuan, S.; Lin, Q.; Qiao, Y.; and Liu, J. 2024a. Symbol-LLM: Towards Foundational Symbol-centric Interface For Large Language Models. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Xu, J.; Fei, H.; Luo, M.; Liu, Q.; Pan, L.; Wang, W. Y.; Nakov, P.; Lee, M.-L.; and Hsu, W. 2025. Aristotle: Mastering logical reasoning with a logic-complete decompose-search-resolve framework. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.-L.; and Hsu, W. 2024b. Faithful Logical Reasoning via Symbolic Chain-of-Thought. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*.

Ye, X.; Chen, Q.; Dillig, I.; and Durrett, G. 2023. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*.

Yu, X.; Wang, Z.; Yang, L.; Li, H.; Liu, A.; Xue, X.; Wang, J.; and Yang, M. 2025. Causal Sufficiency and Necessity Improves Chain-of-Thought Reasoning. *Advances in Neural Information Processing Systems*.

Zhang, Y.; Yuan, Y.; and Yao, A. C.-C. 2024. On the diagram of thought. *arXiv preprint arXiv:2409.10038*.