Neuro-Symbolic Integration Brings Causal and Reliable Reasoning Proofs [*] [†]

Sen Yang [1 2] **Xin Li** [2 ‡] **Leyang Cui** **Lidong Bing** [3] **Wai Lam** [1]

[1] The Chinese University of Hong Kong

[2] DAMO Academy, Alibaba Group

[3] Shanda AI Research Institute

senyang.stu@gmail.com xinting.lx@alibaba-inc.com

lidong.bing@shanda.com neal19951101@gmail.com

wlam@se.cuhk.edu.hk

Abstract

Two lines of approaches are adopted for complex reasoning with LLMs. One line of work prompts LLMs with various reasoning structures, while the structural outputs can be naturally regarded as intermediate reasoning steps. Another line of work adopt LLM-free declarative solvers to do the reasoning task, rendering higher reasoning accuracy but lacking interpretability due to the black-box nature of the solvers. Aiming to resolve the trade-off between answer accuracy and interpretability, we present a simple extension to the latter line of work. Specifically, we showcase that the intermediate search logs generated by Prolog interpreters can be accessed and interpreted into human-readable reasoning proofs. As long as LLMs correctly translate problem descriptions into Prolog representations, the corresponding reasoning proofs are ensured to be causal and reliable. On two logical reasoning and one arithmetic reasoning datasets, our framework obtains significant improvements in terms of both answer accuracy and reasoning proof accuracy. Our code is released at https://github.com/DAMO-NLP-SG/CaRing.

1 Introduction

Large language models (LLMs), like LLaMA-3 (Dubey et al., 2024) and GPT-4 (OpenAI, 2023), are shown to be powerful, but they still struggle with structurally complex reasoning problems, such as logical reasoning (Tafjord et al., 2021) and complex arithmetic reasoning (Cobbe et al., 2021; Ribeiro et al., 2023), as shown in Figure 1. Related work argued that such defects result from the fact that existing LLMs fall short on planning (Valmeekam et al., 2023; Huang et al., 2024; Kambhampati et al., 2024). This is predictable, since one should not expect a next-token-prediction LLM to solve an n-step reasoning problem whose reasoning space grows exponentially regarding n.

Existing LLM-based approaches targeting at this issue can be divided into two-fold. One line of work, such as Self-Consistency (Wang et al., 2023), Tree-of-Thoughts (Yao et al., 2023) and RAP (Hao et al., 2023), adopt various search strategies (e.g., Monte Carlo tree search) to iteratively prompt LLMs, expecting that the correct reasoning path could be achieved using proper process supervision that guides the searching process. Their intermediate search steps are explicit, making the reasoning process interpretable. However, they still rely heavily on LLMs themselves to do planning, thus not fully addressing the aforementioned issue. Besides, LLM inference are performed in each searching step, while solving one problem usually requires many search steps, leading to excessive computational cost. Another line of work, such as Logic-LM (Pan et al., 2023) and SatLM (Ye et al., 2023), translate problem descriptions into declarative symbolic representations (i.e., code) and then adopt off-the-shelf solvers to execute the symbolic representations. Such methods enjoy two key advantages: (1) The declarative symbolic representation is closer to the problem description than the reasoning steps are, so it would be easier for LLMs to do problem translation rather than producing the reasoning steps themselves. (2) The reasoning task is undertaken by an external solver, guaranteeing the correctness of the reasoning process as long as the symbolic representations are correct. However, their search steps are implicitly performed within the solver, making it difficult for humans to inspect whether the reasoning process is correct.

Tracking the above, this work presents a framework that gets rid of LLMs when performing the

[*] The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200620).

[†] This work was supported by Alibaba Group through the Alibaba Innovative Research (AIR) Program (TA2217728).

[‡] XL is the corresponding author.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 5747–5759

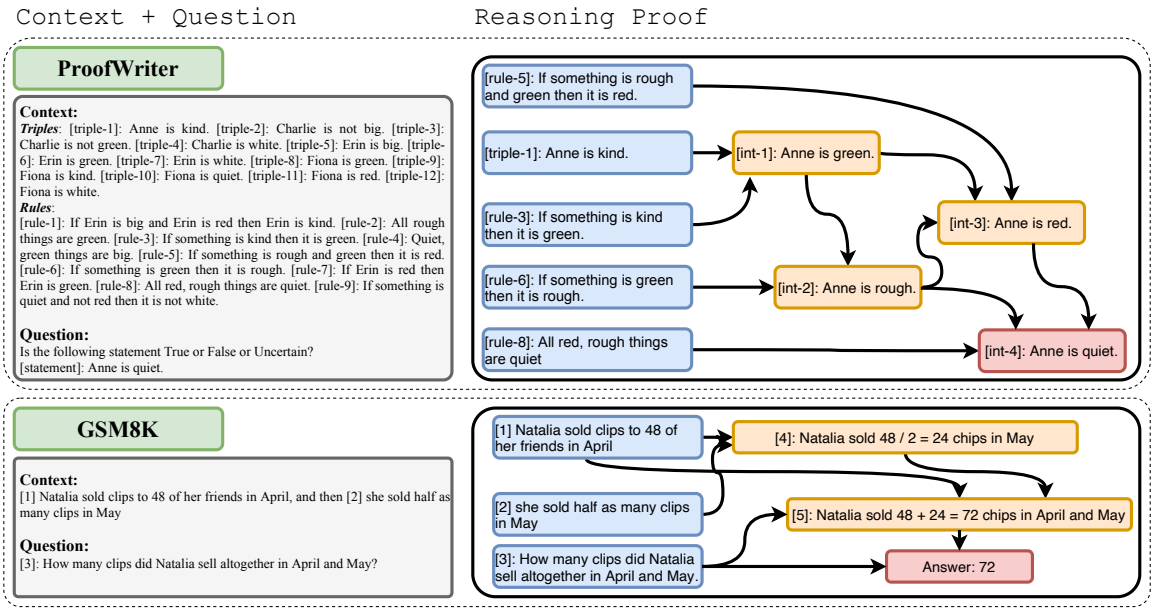


Figure 1: Two examples of complex/structured reasoning problems from ProofWriter and GSM8K, respectively. The reasoning proofs in such problems formulate directed acyclic graphs (DAGs) in a multi-step and multi-premise manner.

reasoning task while still yielding interpretable reasoning steps. Our idea is simple: Searching with symbolic solvers record automatically-generated logs (i.e., trace), which can be translated into human-readable intermediate steps upon further processing. Following this idea, we implement an approach using Prolog, a declarative programming language based on formal logic. We evaluate our approach, CARING (Causal and Reliable Reasoning), on three reasoning datasets that contain reasoning proof annotations, including two logical reasoning datasets, ProofWriter (Tafjord et al., 2021) and PrOntoQA (Saparov and He, 2023), and one arithmetic reasoning dataset, GSM8K (Ribeiro et al., 2023). CARING consistently outperforms the CoT baseline and existing methods in terms of answer accuracy, reasoning proof similarity, and reasoning proof accuracy. On the challenging ProofWriter dataset, CARING using Code-LLaMA-34B yields an answer accuracy of 96.5% and a reasoning proof similarity of 81.0%, while previous SoTA achieved an answer accuracy of 79.7%. Further analysis indicates CARING remains robust when the reasoning problem becomes more complex.

2 Related Work

2.1 Explainable Complex Reasoning

The Chain-of-Thought prompting method, which found out reasoning with LLMs benefits from generating intermediate steps, has sparked a recent trend in how to better do reasoning while remaining explainable. Several works investigated using other reasoning structures, such as trees (Yao et al., 2023; Long, 2023) and graphs (Besta et al., 2023; Zhang et al., 2023). These approaches have shown improved performance, particularly in complex reasoning tasks where the processes involved are often more intricate than simple linear chains. However, despite the alignment of their reasoning proof structures with the gold-standard proofs, these methods still face challenges in ensuring causality and reliability. This limitation stems from their reliance on LLMs for deliberate reasoning, which are prone to hallucinations and may compromise causality.

Some other recent works adopted a less structured manner (Sanyal et al., 2022; Tafjord et al., 2022; Creswell et al., 2023; Kazemi et al., 2023). For example, Selection-Inference (Creswell et al., 2023) divides the reasoning process into two phases: (1) the Selection phase for selecting the premises that might be relevant for the next round of inference, and (2) the Inference phase for conducting a single reasoning step with the selected knowledge fragments.

2.2 Neuro-symbolic Reasoning

Neuro-symbolic systems attempt to leverage the strengths of both neural networks and symbolic reasoning (Andreas et al., 2016; Neelakantan et al.,

2017; Hudson and Manning, 2019; Gupta et al., 2020; Nye et al., 2021). This includes the use of neural networks for pattern recognition and learning from unstructured data, integrated with symbolic systems for rule-based reasoning and knowledge representation. Despite significant progress, neuro-symbolic reasoning faces challenges, notably in scalability and the efficient integration of learning and reasoning components.

Recent advancements in neuro-symbolic research, particularly in reasoning over text, have utilized LLMs to encapsulate knowledge from unstructured human languages, as noted in Lyu et al. (2023); Pan et al. (2023). These methods typically translate natural language into symbolic representations for subsequent execution-based reasoning. However, they have not fully explored the capabilities of symbolic solvers in generating detailed reasoning proofs. In contrast, our approach leverages customized meta-interpreters in conjunction with symbolic solvers to uncover and articulate the underlying reasoning proofs. This not only enhances the transparency of automatic reasoning systems but also simplifies the process for humans to verify their correctness and safety.

3 Method

The problem we are interested in is featured with structured or complex reasoning. As depicted in Figure 1, these problems typically necessitate multi-step and multi-premise reasoning over a directed acyclic graph (DAG), where individual nodes signify distinct knowledge fragments and directed edges denote reasoning steps. Each reasoning step uses existing knowledge to infer new relevant knowledge. Numerous knowledge fragments are often aggregated to infer a new one, which we denote as "multi-premise". The solver usually performs multiple such steps to reach an ultimate goal, which we denote as "multi-step". This entire reasoning process naturally composes a DAG.

We are interested in providing accurate answers along with causal and reliable explanations for such reasoning problems. Kowalski (1979) proposed that *Algorithm* = *Logic* + *Control*, where *Logic* refers to the knowledge which can be used to solve the problem and *Control* refers to the problem-solving strategy in which the knowledge can be used. They further proved that an algorithm benefits from separating the *Logic* component and the *Control* component. Existing works that adopt a symbolic solver to search for valid reasoning paths have followed this design principle (Pan et al., 2023; Ye et al., 2023). Our work also follows this principle.

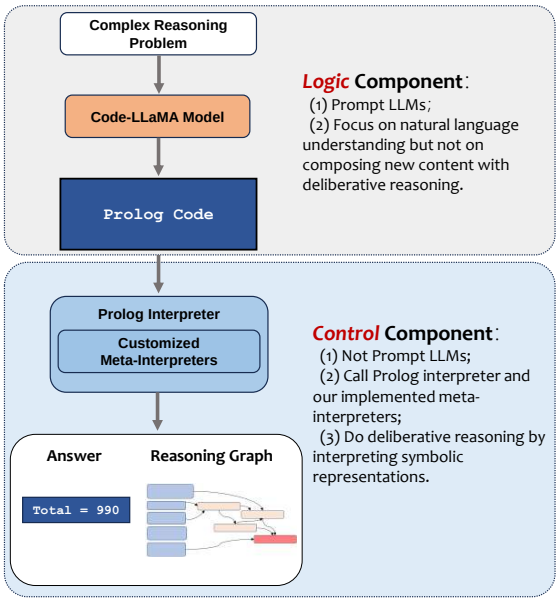


Figure 2: Illustration of our CARING framework, consisting of a *Logic* component and a *Control* component.

We present CARING (Causal and Reliable Reasoning), a modular approach consisting of two components:

- SYMGEN: LLM-based symbolic representation generator (§3.1), which translates problem descriptions into formal symbolic knowledge representations that can be used for symbolic inference.
- SYMINFER: LLM-free symbolic inference engine (§3.2), which performs deliberate reasoning by executing the symbolic representations provided by SYMGEN. With customized Prolog meta-interpreters, SYMINFER supports (i) causal and reliable tracing of the reasoning process (§3.2.1); (ii) various search strategies, such as Depth-First Search (DFS) and Iterative Deepening Search (IDS) (§3.2.2).

The machine-execution nature of SYMINFER guarantees both causality and reliability, as shown in Figure 3. Under the principle of **Causality**, the inference of a new knowledge piece is strictly linked to those existing fragments that are relevant, ensuring precise and limited attribution. This implies that a causal relationship is established only when the preceding event (at the base of the edge) directly influences the subsequent event (at the

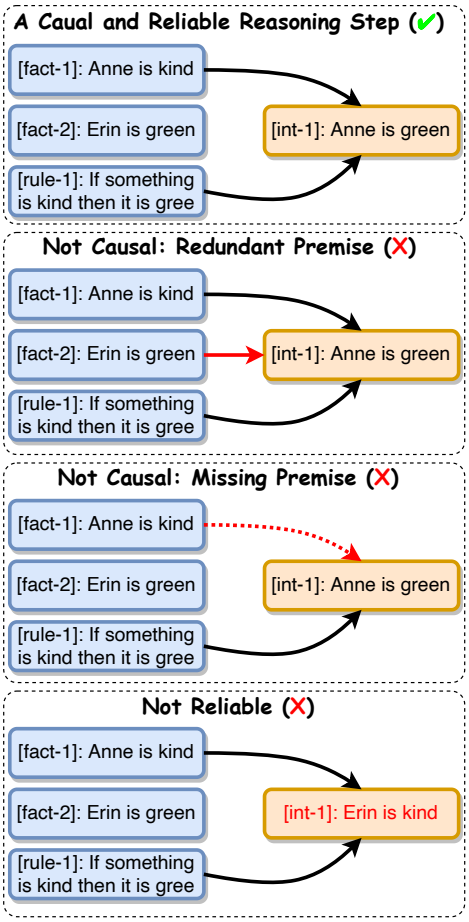


Figure 3: Illustrations of how causality and reliability play important roles in reasoning. LLMs may be (i) non-causal by selecting redundant premises or ignoring relevant ones and (ii) non-reliable by hallucinating erroneous contents.

apex). Regarding **Reliability**, the content within each newly inferred node is the result of a deterministic process, safeguarding it from the kinds of erroneous hallucinations often encountered in outputs from LLMs.

3.1 SymGen: Symbolic Representation Generator

To represent *Logic* (i.e., the knowledge which can be used to solve the problem), we adopt a popular logic programming language, Prolog (Colmerauer and Roussel, 1996). Prolog is a declarative programming language, in which *Logic* is expressed as relations (called Facts and Rules), with several examples shown in Table 1. A computation is initiated by running a query over these relations, which will be further explained in §3.2.

Though LLMs are prone to hallucinate erroneous facts when composing new knowledge (i.e., generating reasoning steps), they are shown to be powerful at understanding natural languages and directly translating them into other formats (Ye and Durrett, 2022; Saparov and He, 2023). Prompting LLMs to translate problem descriptions into symbolic representations just enjoys this strong point of LLMs, as previously shown by Ye et al. (2023). As for implementation, we few-shot prompt LLMs with several human-written in-context demonstrations, each containing a problem and corresponding Prolog representations.

Natural Language	Prolog Code
Fiona is green.	`green(fiona).`
All red, rough things are quiet.	`quiet(X) :- red(X), rough(X).`
Tina makes $18.00 an hour.	`wage(18.00).`
(she is eligible for overtime,) which is paid by your hourly wage + 1/2 your hourly wage.	`overtime_wage(W) :- wage(W1), W is 1.5 * W1.`

Table 1: Examples of problem description snippets and their Prolog representations. It can be seen that the Prolog code is highly declarative, rendering less challenges for LLMs. In other words, the LLM does not need to infer new knowledge, thus avoiding hallucination or planning reasoning paths.

3.2 SymInfer: Symbolic Inference Engine

We use SymInfer to produce answers and reasoning traces by executing the aforementioned symbolic representations. Since we adopt Prolog to represent knowledge, our symbolic inference engine is naturally instantiated with Prolog interpreters. By default, the SWI-Prolog (Wielemaker et al., 2012) interpreter adopts the Depth-First Search (DFS) backtracking strategy and does not yield reasoning proofs. We adopt customized Prolog-based meta-interpreters[1] to achieve two goals: (i) To produce reasoning proofs; (ii) To adopt better search algorithms other than DFS.

3.2.1 Reasoning Tracer

Below is a Prolog meta-interpreter to show the reasoning proofs (Triska, 2024):

```
% Define the operator for proofs
:- op(750, xfy, =>).

% Proof tree generation
mi_tree(true, true).
mi_tree((A,B), (TA,TB)) :-
```

[1]The meta-interpreters used in this work are built upon the implementations of Triska (2024).

```
        mi_tree(A, TA),
        mi_tree(B, TB).
mi_tree(G, builtin(G)) :-
        predicate_property(G, built-in
    ),
        !,
        call(G).
mi_tree(g(G), TBody => G) :-
        mi_clause(G, Body),
        mi_tree(Body, TBody).
```

We showcase how a reasoning trace is induced using the example below. Given a knowledge base like:

```
parent_of(X, Y) :- mother_of(X, Y).
parent_of(X, Y) :- father_of(X, Y).
grandparent_of(X, Y) :-
    parent_of(X, Z), parent_of(Z, Y).
mother_of(morty, beth).
father_of(beth, rick).
```

and a query:

```
?- mi_tree(g(
    grandparent_of(morty, Who)),
    Proof
    ).
```

the output would be

```
Who=rick,
Proof=
(
  (
    (
      true
      =>mother_of(morty, beth)
    )
    =>parent_of(morty, beth),
    (
      true
      =>father_of(beth, rick)
    )
    =>parent_of(beth, rick)
  )
  =>grandparent_of(morty, rick)
).
```

The output proof is ensured to be causal and reliable since it is deterministically generated. A notable advantage is that the reasoning process remains uninfluenced no matter how many distractors are added to the problem description, in contrast to LLMs that can be easily distracted by irrelevant context (Shi et al., 2023).

3.2.2 Search Strategy

The default search strategy of Prolog is DFS, which may lead to infinite loops. For example, given the knowledge base:

```
parent_of(X, Y) :- offspring_of(Y, X).
offspring_of(X, Y) :- parent_of(Y, X).
parent_of(X, Y) :- mother_of(X, Y).
parent_of(X, Y) :- father_of(X, Y).
mother_of(jack, anna).
```

and a query `?- parent_of(jack, Who).` , the backtracking process would repeat over the first two lines without resorting to other lines due to DFS. To address this issue, we adopt Iterative Deepening Search (IDS), in which the backtracking process performs a series of depth-limited searches, each with an increasing depth limit. This leverages the strengths of both Breadth-First Search (BFS) and DFS.

4 Experiments

We briefly introduce our experimental settings in §4.1 and show the experiment results in §4.2.

4.1 Experimental Settings

We present our experimental settings in this section, including our implementation details of the two components (§4.1.1), a brief introduction of the adopted datasets (§4.1.2) and the baselines (§4.1.4).

4.1.1 Implementation

SymGen We adopt the Code-LLaMA (Rozière et al., 2023) family as the base LLMs to translate natural languages into Prolog representations. Our prompting paradigm is in a pure few-shot in-context-learning (ICL) prompting style, without detailed human-written instructions. Each ICL demonstration comprises a question and a piece of Prolog code.

SymInfer We adopt SWI-Prolog (Wielemaker et al., 2012) and PySwip[2] packages to implement the symbolic inference engine. We set the maximum depth to be 20 for Iterative Deepening Search and the number of generated reasoning paths to 20.

4.1.2 Datasets

We evaluate CARING on three popular complex reasoning datasets, including two logical reasoning datasets (ProofWriter (Tafjord et al., 2021) and PrOntoQA (Saparov and He, 2023)) and one arithmetic dataset (GSM8K (Cobbe et al., 2021; Ribeiro et al., 2023)).

ProofWriter ProofWriter (Tafjord et al., 2021) is a commonly-used logical reasoning dataset. It contains many small rulebases of facts and rules, expressed in English. Each rulebase has a set of questions (English statements) that can either be proven true or false using proofs of various depths, or the answer is "Unknown" (in open-world setting,

[2] https://github.com/yuce/pyswip

OWA). The proofs can naturally be represented as directed acyclic graphs (DAGs). The dataset is divided into several sub-sets according to maximum proof depth, namely $\{0, \leq 1, \leq 2, \leq 3, \leq 5\}$. We follow previous work (Pan et al., 2023) to use a 600-instance subset sampled from the most difficult depth-5 test set. We also report additional results on the full depth-5 test set in Appendix §A.1.1.

PrOntoQA PrOntoQA (Saparov and He, 2023) is a synthetic question answering dataset designed for diagnosing the logical reasoning ability of LLMs. Each example aims to validate the feasibility of a statement given a context. We report results on two subsets so we can compare CARING with previous methods. As for the results reported in Table 4, we follow Pan et al. (2023) to adopt the most difficult depth-5 *fictional characters* sub-set, which contains 500 statement-context pairs. As for the results in Table 3, we use the subset adopted by Hao et al. (2023). Similar to ProofWriter, the proofs provided by the dataset can be naturally represented as DAGs.

GSM8K GSM8K (Cobbe et al., 2021) is a multi-step arithmetic reasoning dataset composed of high-quality grade school math word problems. The original GSM8K dataset contains reasoning explanations written in natural language, which raises difficulties in evaluating intermediate steps automatically. Recently, Ribeiro et al. (2023) released a subset that contains 270 questions annotated with structured reasoning proofs in the format of DAGs. We adopt this subset to enable the evaluation of reasoning proofs.

4.1.3 Evaluation Metrics

Ribeiro et al. (2023) proposed two novel metrics to evaluate the quality of the generated reasoning proofs in addition to the prevalent answer accuracy metric. Similar to them, we adopt the following metrics to evaluate both the answers and the generated reasoning proofs.

Answer Accuracy Answer accuracy measures a model's ability to predict the correct answer. A prediction is deemed correct if it is (i) the same as the gold option for multi-choice problems and (ii) the same integer as the gold answer for arithmetic reasoning problems. This metric is the upper bound for other metrics since a reasoning graph would be marked as incorrect without evaluation if the answer is marked as incorrect. We report this metric for all datasets.

	Method	Acc (%)	Proof Sim (%)	
			All	Correct
GPT-4*	CoT	67.41	–	–
	ToT	70.33	–	–
	CR	71.67	–	–
	DetermLR	79.17	–	–
	Logic-LM	79.66	–	–
Code-LLaMA 7B	CoT	46.33	9.69	14.95
	Ours	91.00	72.91	84.39
Code-LLaMA 13B	CoT	46.50	15.69	25.86
	Ours	95.67	80.65	86.00
Code-LLaMA 34B	CoT	52.00	15.76	27.74
	Ours	**96.50**	**81.02**	**86.12**

Table 2: Results on the subset of ProofWriter adopted by Pan et al. (2023). The default setting is 2-shot. "All": on all instances. "Correct": on correctly-predicted instances. *All GPT-4 numbers are from Sun et al. (2023).

Reasoning Proof Similarity As shown in Figure 1, the problems that we are interested in naturally compose reasoning proofs in the format of directed acyclic graphs (DAGs). Reasoning proof similarity $\text{sim}(\mathcal{G}_g, \mathcal{G}_p)$ measures the graph similarity between the gold and the predicted reasoning graphs. We follow Ribeiro et al. (2023) to adopt the graph edit distance function $\delta(\mathcal{G}_g, \mathcal{G}_p)$. This function quantifies the graph edit distance by determining the minimum number of operations required over nodes and edges to transform one graph into the other, thereby enabling a comparison of $\mathcal{G}_g$ and $\mathcal{G}_p$ based on their structural similarities. The reasoning graph similarity is normalized to $[0, 1]$ as:

$$\text{sim}(\mathcal{G}_p, \mathcal{G}_g) = 1 - \frac{\delta(\mathcal{G}_p, \mathcal{G}_g)}{\max\{|N_p| + |E_p|, |N_g| + |E_g|\}} \quad (1)$$

where $|N_p|$ and $|E_p|$ denote the count of nodes and edges, respectively, within the predicted reasoning graph. A similar notation applies to $|N_g|$ and $|E_g|$, which represent the number of nodes and edges in the gold graph. Note that the reasoning graph similarity is set to zero if the predicted answer is incorrect. We report this metric for ProofWriter and GSM8K.

Reasoning Proof Accuracy This metric evaluates the exact match between the gold and the predicted reasoning proofs in terms of both reason-

Base LLM	#Param	#Shot	Method	Acc (%)	Proof Acc (%)	
					All	Correct
LLaMA-1*	33B	8-shot	CoT	87.8	64.8	–
			RAP	94.2	78.8	–
Code-LLaMA	13B	2-shot	CoT	80.2	52.4	53.4
			Ours	**99.0**	**98.2**	**99.2**

Table 3: Results on the PrOntoQA subset that was adopted by RAP (Hao et al., 2023) for comparison with their method. The results marked with * are from their paper.

		Method	Acc (%)	Proof Acc (%)	
			All	Correct	
GPT4*		CoT	98.8	–	–
		Logic-LM	83.2	–	–
Code-LLaMA	7B	CoT	52.0	24.8	28.5
		Ours	98.8	98.4	99.6
	13B	CoT	61.0	32.2	35.9
		Ours	99.4	98.8	99.4
	34B	CoT	82.8	41.0	41.0
		Ours	**100.0**	**100.0**	**100.0**

Table 4: Results on the depth-5 subset of PrOntoQA. The default setting is 2-shot. Results marked with * are GPT-4 results reported by Logic-LM (Pan et al., 2023).

	Method	Acc	Proof Sim (%)	
			All	Correct
7B	CoT	13.70	4.99	36.39
	Ours	12.22	6.57	53.72
13B	CoT	15.56	5.76	37.03
	Ours	21.48	11.66	**54.26**
34B	CoT	35.19	13.04	37.07
	Ours	**42.22**	**22.91**	54.25

Table 5: Results on GSM8K. The default setting is 5-shot.

ing graph structures and textual contents[3]. The reasoning proof accuracy is either 1 or 0 for a single instance, making it a discrete version of reasoning proof similarity. Since this metric requires the dataset to have structured content to enable automatic evaluation, we can only apply it to PrOntoQA, which is specifically designed for easy parsing of the proofs.

4.1.4 Baselines

All baselines prompt the LLMs with few-shot in-context-learning (ICL) demonstrations.

Chain-of-Thought (CoT) CoT prompting (Wei et al., 2022) prompts LLMs with ICL demonstrations that contain both intermediate reasoning steps and answers. It serves as a popular and strong baseline for prompting LLMs to solve problems.

Logic-LM Logic-LM (Pan et al., 2023) is a neuro-symbolic method that adopts symbolic solvers for logical reasoning problems. The main difference between Logic-LM and CARING is: Logic-LM adopts various solvers for multiple datasets and only focuses on answer accuracy, while CARING universally uses one solver (i.e., SWI-Prolog); and more importantly, CARING showcases how SWI-Prolog interpreters can be customized to generate intermediate reasoning proofs and to adopt various search (i.e., problem-solving) strategies.

Search-based Methods We include several search-based methods as baselines. We directly adopt the released results in their papers, since it is too time-consuming to implement these methods on our own. For ProofWriter, we compare our method with GPT-4 based Tree-of-Thoughts (ToT; (Yao et al., 2023)), Cumulative Reasoning (CR; (Zhang et al., 2023)), and DetermLR (Sun et al., 2023). We cannot make comparisons on reasoning proofs because these methods only reported reasoning accuracy. For PrOntoQA, we compare our method with RAP (Hao et al., 2023), in terms of both reasoning accuracy and reasoning proof accuracy.

4.2 Main Results

Tables 2, 3, 4 and 5 show the experimental results on our adopted datasets.

ProofWriter The results on ProofWriter are presented in Table 2. CARING demonstrates notable improvements over existing baselines, particularly in terms of reasoning proof similarity. Utilizing Code-LLaMA-34B, CARING achieves a remark-

[3]Note that our implementation here is simpler than that of Ribeiro et al. (2023) because we only apply this metric to PrOntoQA, which is easy to get evaluated.

able answer accuracy of 96.50% and a reasoning proof similarity of 81.02%, significantly surpassing the most powerful method using GPT-4 that obtains an accuracy of 79.66%.

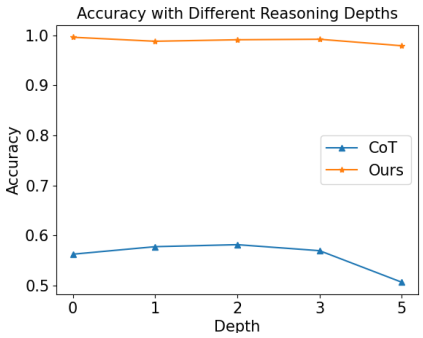
PrOntoQA The results on PrOntoQA are presented in Tables 3 and 4. CARING achieves almost full accuracy with the 13B model. Comparing with RAP, CARING obtains better results in terms of both answer accuracy and proof accuracy even using a smaller base LLM and fewer ICL demonstrations. CARING also outperforms CoT and Logic-LM that use GPT-4.

GSM8K The results on GSM8K are presented in Table 5. This dataset is more challenging than the previous two logical reasoning datasets for CARING, since it is generally believed that symbolic languages are restricted by their limited expressiveness and cannot properly handle the ambiguity in real-world human languages. Surprisingly, with the 34B model, CARING outperforms the strong CoT baseline by a large margin and almost doubles the reasoning proof similarity (22.91% vs. 13.04%). We attribute such improvements to increasingly powerful LLMs, which can correctly translate ambiguous human languages into formal symbolic representations.

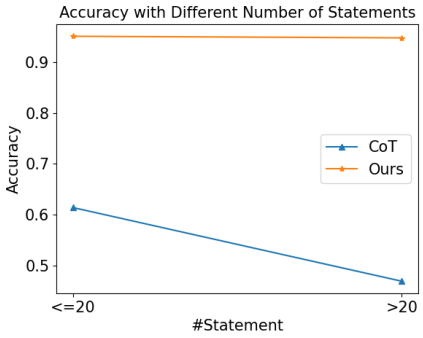
4.3 When Reasoning Becomes More Complex

A key difficulty confronted by reasoning systems is the rapid expansion of possible states as the reasoning process becomes more complex, such as when additional statements are considered or the depth of inference is greater. To investigate how our method handles more complex reasoning problems, we conduct experiments under two controlled settings: (1) **#Depth** ↑: How does the answer accuracy change with #Depth being $<= 0$, $<= 1$, $<= 2$, $<= 3$ and $<= 5$, respectively; (2) **#Statements** ↑: How does the answer accuracy change with #Statements being $<= 20$ and > 20, respectively.

As shown in Figures 4a and 4b, with increasing levels of reasoning intricacy, the answer accuracy of CARING remains steady. In contrast, CoT sees significant decreases in answer accuracy under both settings. This verifies the robustness of CARING against complex reasoning.



(a) Answer accuracy with different reasoning depths.



(b) Answer accuracy with different number of statements.

Figure 4: Answer accuracy when reasoning problems become more complex.

5 Conclusion

This paper presents a framework to address the erroneous reasoning proof problem of LLM-based reasoning systems. Specifically, we develop a neuro-symbolic method called CARING, which produces high-quality reasoning proofs for complex reasoning problems. By implementing customized meta-interpreters for executing Prolog representations and putting LLMs under quarantine during the reasoning phase, CARING ensures the reasoning proofs to be causal and reliable. We conduct experiments on two logical reasoning datasets and one arithmetic reasoning dataset. Experimental results demonstrate our method achieves significant improvements with both final answers and intermediate reasoning proofs. Further analysis indicates CARING remains robust when the reasoning problems become more complex.

Limitations

We observe two limitations regarding our framework:

- The generalization ability of CARING is restricted by the expressiveness of the concerning symbolic representations. In this paper, we showcase a Prolog-based implementation. Other symbolic representations could be explored to generalize CARING to more reason-

ing tasks.

- CARING requires powerful LLMs as symbolic representation generators, which is suggested by the results on GSM8K. This dependence might prevent it from being applied to productions.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 39–48. IEEE Computer Society.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of thoughts: Solving elaborate problems with large language models.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Alain Colmerauer and Philippe Roussel. 1996. *The Birth of Prolog*, page 331–367. Association for Computing Machinery, New York, NY, USA.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu

Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Drew A. Hudson and Christopher D. Manning. 2019. Learning by abstraction: The neural state machine. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5901–5914.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. 2024. Llms can't plan, but can help planning in llm-modulo frameworks.

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2023. LAMBADA: Backward chaining for automated reasoning in natural language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6547–6568, Toronto, Canada. Association for Computational Linguistics.

Robert Kowalski. 1979. Algorithm = logic + control. *Commun. ACM*, 22(7):424–436.

Jieyi Long. 2023. Large language model guided tree-of-thought.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *ArXiv preprint*, abs/2301.13379.

Arvind Neelakantan, Quoc V. Le, Martín Abadi, Andrew McCallum, and Dario Amodei. 2017. Learning a natural language interface with neural programmer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Maxwell Nye, Michael Henry Tessler, Joshua B. Tenenbaum, and Brenden M. Lake. 2021. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning.

OpenAI. 2023. Gpt-4 technical report.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-LM: empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*, Singapore.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Henghui Zhu, Rui Dong, Deguang Kong, Juliette Burger, Anjelica Ramos, zhiheng huang, William Yang Wang, George Karypis, Bing Xiang, and Dan Roth. 2023. STREET: A MULTI-TASK STRUCTURED REASONING AND EXPLANATION BENCHMARK. In *The Eleventh International Conference on Learning Representations*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. FaiRR: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093, Dublin, Ireland. Association for Computational Linguistics.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2023. From indeterminacy to determinacy: Augmenting logical reasoning capabilities with large language models.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2078–2093, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Markus Triska. 2024. A couple of meta-interpreters in prolog. `https://www.metalevel.at/acomip/`, Last accessed on 2024-08-16.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. 2012. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate problem solving with large language models.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. SatLM: Satisfiability-aided language models using declarative prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. In *Advances in Neural Information Processing Systems*.

Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew Chi-Chih Yao. 2023. Cumulative reasoning with large language models. *ArXiv preprint*, abs/2308.04371.

		Acc	Proof Sim	
			All	Correct
7B	Direct	41.78	–	–
	Direct (3-Shot)	43.32	–	–
	CoT	40.95	11.27	17.20
	CoT (3-shot)	42.58	11.52	21.58
	Ours	92.43	75.85	**86.68**
13B	Direct	43.44	–	–
	Direct (3-shot)	44.31	–	–
	CoT	45.88	16.16	27.32
	CoT (3-shot)	54.70	23.18	32.48
	Ours	96.16	80.74	86.34
34B	Direct	44.00	–	–
	Direct (3-shot)	45.93	–	–
	CoT	52.32	15.08	26.30
	CoT (3-shot)	56.50	24.12	34.61
	Ours	**98.11**	**83.17**	85.65

Table 6: Results on ProofWriter. "All" and "Correct" refer to "on all instances" and "on correctly-predicted instances", respectively. "Proof Sim" refers to "Proof Graph Similarity" while "Proof EM" means "Proof Graph Exact Match". The default setting is 2-shot. We additionally conduct 3-shot experiments for baselines to include all types of labels in the in-context demonstrations because this dataset contains three labels: {`true`, `false`, `uncertain`}. We do not conduct 3-shot experiments for our method because it is not sensitive to the number of labels due to its reasoning-by-execution nature.

A Appendix

A.1 Additional Results

A.1.1 Results on ProofWriter

Table 6 shows the results from our implementation on the depth-5 test set of ProofWriter.