Leibniz: Theory-of-Mind Driven Neuro-Symbolic Logical Reasoning via Multi-Agent Collaboration

Yue Fan[1], Hu Zhang[1,2*], Yunxiao Zhao[1], Guangjun Zhang[1], Hao Zhan[3], Ru Li[1,2*], Hongye Tan[1,2], Yuanlong Wang[1,2]

[1]School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, China
[3]School of Philosophy, Shanxi University, Taiyuan, China
yuefan24@163.com, zhanghu@sxu.edu.cn, yunxiaomr@163.com, zgj2866@gmail.com

Abstract

Logical reasoning with large language models (LLMs) has made significant progress in recent years. However, existing methods still suffer from insufficient rule semantic grounding and weak rule application mechanisms, making it difficult to achieve precise understanding and effective utilization of rules in complex multi-step reasoning. To address this, we propose **Leibniz**, a theory-of-mind driven neuro-symbolic reasoning framework. Specifically, we construct a bidirectional reasoning model based on multi-agent collaboration, which characterizes the reasoning process from two complementary perspectives, namely the Evolution Agent and the Reduction Agent. The former transforms belief-unstable propositions into stable ones that are beneficial for proving the target conclusion. The latter performs reverse reduction from the target to resolve belief conflicts and distill new inferential insights. Both share a belief state space and achieve efficient collaborative reasoning through continual belief updating. We integrate natural language and symbolic representations throughout the reasoning process. The experimental results demonstrate that Leibniz surpasses existing state-of-the-art models in reasoning accuracy, and further analyses reveal its substantial advantages in reliability and flexibility.

1 Introduction

Logical reasoning, as a core component of human intelligence, plays a crucial role in tasks such as semantic understanding, causal inference, and commonsense reasoning (Cheng et al., 2025; Liu et al., 2025a; Yang et al., 2025b). With the rapid advancement of large language models (LLMs), researchers propose various LLM-based reasoning methods and strategies (Ding et al., 2024; Besta et al., 2024; Bi et al., 2025), such as Chain-of-Thought (CoT) (Wei et al., 2022) and Tree-of-Thought (ToT) (Yao et al., 2023), which simulate human thought processes by explicitly generating intermediate reasoning steps, thereby substantially enhancing the ability of LLMs to solve complex problems. Nevertheless, these methods still perform poorly in rigorous logical reasoning scenarios. Unlike other forms of general reasoning, logical reasoning requires strict evidence evaluation, argument construction, and formal logical deduction to derive reliable conclusions (Cummins et al., 1991).

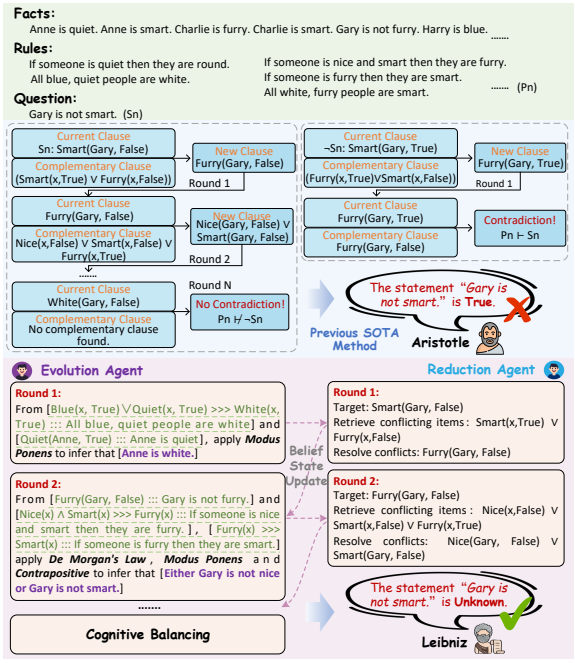

Figure 1: An illustrative example of LLM-based logical reasoning. The green box contains the input for logical reasoning. The blue box and purple box respectively show the reasoning processes of the previous SOTA method and our method.

In recent years, numerous studies have focused on exploring how to integrate LLMs with logical reasoning (Nathani et al., 2023; Gao et al., 2023; Dalal et al., 2024; Wan et al., 2024; Liu et al., 2025b; Qi et al., 2025; Chen et al., 2025). For example, Logic-LM (Pan et al., 2023) and LINC

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20186–20210

(Olausson et al., 2023) adopt a tool-augmented neuro-symbolic paradigm, first converting natural language into symbolic representations (e.g., first-order logic, FOL) and then delegating strict reasoning to external solvers. However, these methods exhibit limitations in flexibility and scalability due to their high sensitivity to symbolic representations, where minor errors can lead to reasoning failures. Subsequent researchers propose integrating logical symbols into CoT, significantly enhancing the flexibility of reasoning and the robustness of symbolic representations. For example, SymbCoT (Xu et al., 2024) combines symbolic expressions and logical rules with CoT prompts, leveraging LLMs' planning and solving capabilities to improve reasoning accuracy. Aristotle (Xu et al., 2025b) starts from the conclusion and simplifies the reasoning process by minimizing search errors, achieving the latest optimal performance across multiple tasks.

However, Aristotle relies on explicit premises to retrieve contradictory terms, making it difficult to exploit the implicit information in the premises and thereby diminishing the flexible reasoning capabilities of LLMs. As shown in Figure 1, Aristotle incorrectly outputs the answer "True" after retrieving a contradiction related to the conclusion from the given fact base. The implicit proposition "*Either Gary is not nice or Gary is not smart*," which can be derived from the facts and rules, is insufficient to determine that the conclusion is true. Therefore, the correct answer should be "unknown." Despite existing methods have made remarkable progress in logical reasoning, they still face two key challenges: **(i) Insufficient rule semantic grounding**: they struggle to accurately anchor appropriate logical rules in multi-step reasoning processes, leading to inconsistent reasoning paths or deviations from target logical structures. **(ii) Weak rule application mechanisms**: the accuracy of deriving new conclusions from given premises remains limited, especially when the reasoning chain is long or involves conflicting evidence. Existing methods fail to effectively perform self-monitoring and strategic adjustment for intermediate steps.

To address these challenges, we propose `Leibniz`, a Theory-of-Mind driven neuro-symbolic logical reasoning framework. Research in cognitive science indicates that humans rely on the "Theory of Mind" (ToM) ability during reasoning, which involves representing and inferring others' mental states (such as beliefs and intentions) to understand, predict, and regulate their behavior (Premack and Woodruff, 1978; Lake et al., 2017; Rabinowitz et al., 2018). Inspired by the collaborative mechanisms in ToM, we construct a bidirectional neuro-symbolic reasoning model composed of an evolution agent and a reduction agent. This enables agents to explicitly represent each other's beliefs and intentions, achieving dynamic understanding and flexible application of logical rules through belief update mechanisms.

Specifically, we first translate natural language statements into logical symbolic representations and employ both forms throughout the subsequent reasoning process. We then construct the bidirectional neuro-symbolic reasoning model: **(i) The evolution agent, acting as an exploratory reasoner**, first categorizes all facts and rules into stable and unstable beliefs based on their association with the target conclusion. Subsequently, we progressively evolve unstable beliefs into stable ones through symbolic rule grounding and rule implementation modules. During this process, the evolution agent will integrate historical information from the reduction agent to generate relevant propositions in a targeted manner. **(ii) The reduction agent, acting as a critical reasoner**, retrieves conflicting beliefs starting from the target conclusion and derives new insights through conflict resolution. This agent needs to understand the current knowledge boundaries of the evolution agent to avoid invalid reductions. Both agents share the belief state space and independently memorize historical beliefs during reasoning. They achieve effective collaborative reasoning through continuous belief transfer and updating. We evaluate `Leibniz` using multiple LLMs across several logical reasoning datasets, and the results show that it outperforms previous SOTA methods in reasoning accuracy. In summary, our technical contributions are:

- We propose a Theory-of-Mind driven neuro-symbolic framework that integrates ToM's collaborative mechanisms into the logical reasoning process of LLMs, enabling more effective multi-agent collaborative reasoning.
- We construct a bidirectional reasoning model comprising an evolution agent and a reduction agent, which can fully leverage latent information and enhance the ability of LLMs to accurately understand and apply complex logical rules.
- Experimental results across multiple LLMs and datasets demonstrate that `Leibniz` exhibits more adaptive and reliable logical reasoning capabilities.

2 Related Work

2.1 LLM-based Logical Reasoning

In recent years, LLM-based logical reasoning methods have achieved significant progress (Saparov et al., 2023; Dalal et al., 2024; Fan et al., 2024, 2026; Qi et al., 2025; Xu et al., 2025a; Yan et al., 2026). Previous researchers propose methods based on external symbol solvers (Olausson et al., 2023; Gao et al., 2023; Pan et al., 2023; Feng et al., 2024), where they first convert natural language statements into symbolic representations and then perform formal reasoning using symbol solvers such as `Z3` (de Moura and Bjørner, 2008), `Prover9`, and others. Although these methods offer strong logical rigor, they also exhibit notable limitations. The translation from natural language to symbolic formulas requires extremely high precision, and even minor symbolization errors may lead to reasoning failure. Moreover, the search space for reasoning grows exponentially with increasing problem complexity, making it difficult for existing symbolic solvers to perform reasoning efficiently.

To alleviate the above problems, subsequent researchers propose integrating symbolic representations into the reasoning process of LLMs themselves, thereby constructing LLM-based neural symbolic logic reasoning methods (Xu et al., 2024; Wang et al., 2024; Liu et al., 2025b; Xu et al., 2025b; He et al., 2025; He and Roy, 2025; Wang et al., 2025). These methods effectively combine the precision of symbolic representations with the generalization capabilities of language models, achieving substantial progress in logical reasoning tasks. However, accurate grounding and flexible application of rules remain key bottlenecks that constrain the logical reasoning capabilities of LLMs. To this end, we propose a theory-of-mind driven multi-agent collaboration framework, which enables agents to model each other's beliefs and intentions from both exploratory and critical reasoning perspectives, thereby enhancing the ability of LLMs to understand and apply logical rules.

2.2 Machine Theory of Mind

Theory of Mind (ToM) is a core cognitive ability that humans rely on heavily in cooperation and social interaction (Zhang et al., 2012; Li et al., 2023; Kosinski, 2023). Specifically, humans can infer others' mental states (including intentions, beliefs, expectations, etc.) by observing their behavior, thereby adjusting their own actions to achieve more efficient collaboration and reasoning. Recently, researchers have proposed various methods to explore and enhance the ToM reasoning capabilities of LLMs. For example, Sclar et al. (2023) and Wilf et al. (2024) employ prompt strategies to improve LLMs' ToM reasoning abilities in textual question-answering. The latest studies develop multimodal benchmarks and methods (Jin et al., 2024; Shi et al., 2025) to evaluate and strengthen LLMs' ToM reasoning performance in multimodal scenarios.

Prior research related to ToM has primarily focused on evaluating and enhancing the ToM reasoning capabilities of LLMs. These methods are typically tested on specially constructed ToM benchmark, such as ToM-bAbi (Nematzadeh et al., 2018) and ToMi (Le et al., 2019). Unlike the aforementioned studies, we propose integrating ToM into an LLM-based logical reasoning framework to fully leverage its critical role in intention understanding and collaborative reasoning, enabling models to more deeply understand the reasoning targets and achieve dynamic strategy adjustments.

3 Method

3.1 Problem Formulation

Logical reasoning tasks aim to progressively derive intermediate propositions from the given premises, ultimately arriving at the target conclusion. Formally, given a set of facts $F = \{f_1, f_2, \ldots, f_n\}$, a set of rules $R = \{r_1, r_2, \ldots, r_m\}$, and a query Q, our objective is to derive stepwise logical inferences, thereby determining whether Q is true, false, or unknown. Here f_i and r_j denote logical statements, and together they form the set of premises $P = \{p_1, p_2, \ldots, p_k\} \subseteq \{F \cup R\}$. As shown in Figure 2, `Leibniz` consists of three key modules: **Translator** (§3.2), **Evolution Agent** (§3.3.1), and **Reduction Agent** (§3.3.2).

3.2 Translator

We first convert the given premise P and question statement Q into a formal symbolic representation, thereby eliminating potential ambiguities in natural language and ensuring the clarity of logical structure and precision of reasoning statements. We use Prolog syntax (Clocksin and Mellish, 2003) to translate natural language into first-order logic (FOL). The translated facts and rules are represented as $F_t = \{f_{t1}, f_{t2}, \ldots, f_{tn}\}$ and $R_t = \{r_{t1}, r_{t2}, \ldots, r_{tm}\}$, while the problem statement is represented as Q_t. The details of the syntax can be found in ***Appendix A***.

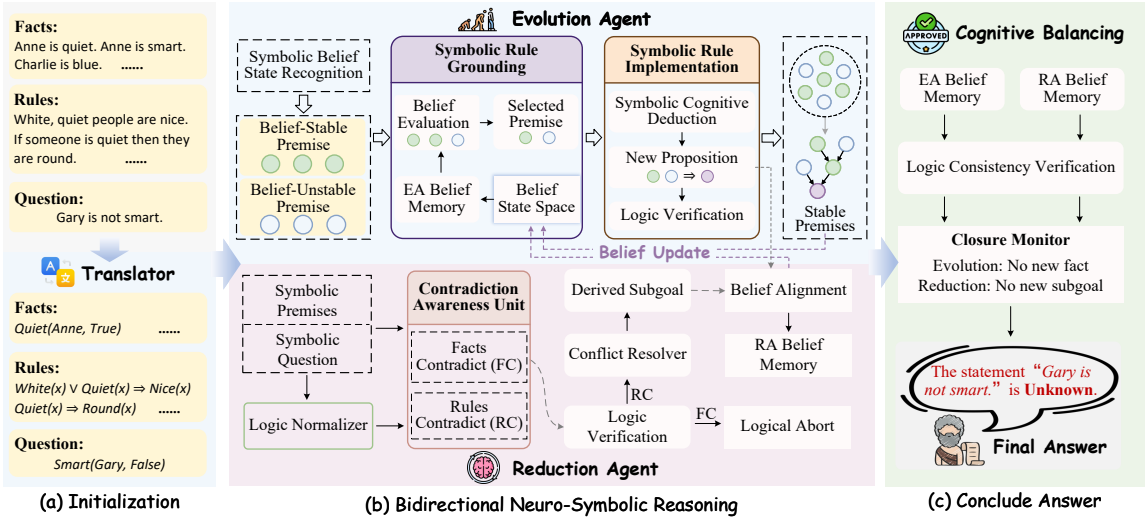

Figure 2: Overview of our proposed neuro-symbolic `Leibniz` framework.

► **Input:**
<Premises> (P)
The bear likes the cat.
If someone is kind and they visit cat then they are big.
<Question> (Q)
The rabbit does not like the cat.
► **Output:**
<Premises> (P_t)
`Likes(Bear, Cat, True)`
$\forall x\,(\texttt{Kind}(x) \wedge \texttt{Visits}(x, \texttt{Cat}) \rightarrow \texttt{Big}(x))$
<Question> (Q_t)
`Likes(Rabbit, Cat, False)`

3.3 Bidirectional Neuro-Symbolic Reasoning

We construct a bidirectional neuro-symbolic reasoning model to execute stepwise logical reasoning, comprising two core components: the **evolution agent (EA)** and the **reduction agent (RA)**. We explicitly model the reasoning intentions of two agents based on theory-of-mind, and simulate knowledge transfer and belief update mechanisms between them throughout the reasoning process. Each agent incorporates three mental variables in progressive reasoning: shared belief, own goal, and belief of others' goals.

3.3.1 Evolution Agent

When humans solve complex problems, they typically categorize known conditions according to their degree of relevance to the target. Stable premises can be directly used to determine the truth value of the target or provide explicit support, whereas unstable premises must be combined with other conditions and further inferred to generate new intermediate propositions. We model this process as a belief-evolution mechanism, which forms the foundation for constructing the evolution agent responsible for exploratory reasoning.

Specifically, we first categorize the premises into stable and unstable types according to their logical relevance to the target conclusion. We define stable premises as simple statements directly relevant to the target conclusion and presented as basic facts, which typically provide clear and reliable support for the reasoning process. In contrast, we define unstable premises as statements not directly relevant to the target conclusion and often containing disjunctive or hypothetical elements, which can be combined with other conditions to derive new insights. The above premise categorization process can be formalized as follows:

$$\mathcal{S}, \mathcal{U} = \text{Stability}(p_i, Q) \tag{1}$$

where premise $p_i \in \{F \cup R\}$, $\mathcal{S}$ and $\mathcal{U}$ denote the set of stable premises and unstable premises, respectively. We implement the function $\text{Stability}(\cdot)$ using LLM-based instructions. For example, regarding the target proposition "*Harry is not young*," stable propositions include "*Harry is furry. All rough, quiet people are young*", because they can directly or indirectly participate in reasoning related to the target. Unstable propositions include "*Anne is blue. Anne is furry. ...*" as they are not directly relevant to the target. We then design Symbolic Rule Grounding and Symbolic Rule Implementation to execute logical evolution.

(i) Symbolic Rule Grounding. The core of the evolution agent is to accurately anchor applicable rules during the stepwise reasoning. We prioritize stable and unstable premises separately and select the premise combinations that are most likely to generate new propositions. We design LLM-based

instructions to compute a relevance score for each premise p_i with respect to the target conclusion Q:

$$c_f, c_r = \text{Prioritizer}(p_i, Q, \mathcal{M}_E, \mathcal{M}_R) \quad (2)$$

where $\mathcal{M}_E$ and $\mathcal{M}_R$ denote the belief memories of EA and RA, respectively, which stores historical reasoning details. $p_i \in \mathcal{A}$, where $\mathcal{A}$ represents the dynamically updated premise belief state space. This space is continuously updated during the interaction between the two agents, and its initial content is $\mathcal{A} = \{F, F_t; R, R_t\}$. Therefore, EA's mental variables include shared belief $\mathcal{A}$, own goal Q, and RA's goal belief $\mathcal{M}_R$. The evolution agent's objective is to progressively evolve unstable propositions into stable ones, so its personal goal remains fixed on problem Q. c_f and c_r denote the stability scores for facts and rules, respectively. We then select from $\mathcal{S}$ the optimal premise p^*, and further quantify the likelihood that other premises in $\mathcal{S}$ and those in $\mathcal{U}$ can be combined with p^* based on their logical stability scores. This process yields a set of candidate premise combinations $\mathcal{H}$.

(ii) Symbolic Rule Implementation. After completing rule grounding and obtaining the most probable premise combinations, we further derive new intermediate propositions using symbolic rule implementation. Specifically, the derivation process can be formalized as follows:

$$p_{new} = \text{Deduce}(Q, Q_t, \mathcal{H}) \quad (3)$$

where $\mathcal{H}$ comprises both natural language and symbolic representations related to the current premises. This process achieves a more precise derivation of new propositions by fully integrating natural language semantics with logical symbolic structures.

► **Input:**
<Conclusion> (Q_t ::: Q)
`Eel(Sea_eel)` ::: Sea eel is an eel.
<Premises Combinations> ($\mathcal{H}$)
$\forall x\,(\texttt{Eel}(x) \rightarrow \texttt{Fish}(x))$ ::: All eels are fish.
$\forall x\,(\texttt{Plant}(x) \oplus \texttt{Animal}(x))$ ::: A thing is either a plant or animal.
(... More premises ...)
► **Output:**
Step 1: From "$\forall x\,(\texttt{Eel}(x) \rightarrow \texttt{Fish}(x))$" and "$\forall x\,(\texttt{Fish}(x) \rightarrow \texttt{Plant}(x, \texttt{False}))$", applying the **Transitivity Law** infers that "$\forall x\,(\texttt{Eel}(x, \texttt{True}) \rightarrow \texttt{Plant}(x, \texttt{False}))$ ::: No eels are plants".
Step 2: From "$\forall x\,(\texttt{Eel}(x) \rightarrow \texttt{Plant}(x, \texttt{False}))$" and "$\forall x\,(\texttt{Plant}(x) \oplus \texttt{Animal}(x))$", applying the **Disjunctive Syllogism** infers that "$\forall x\,(\texttt{Eel}(x) \rightarrow \texttt{Animal}(x))$ ::: All eels are animals".
<Proposition> (p_{new})
$\forall x\,\texttt{Eel}(x) \rightarrow \texttt{Animal}(x)$ ::: All eels are animals.

After generating a new proposition p_{new}, we design a two-stage verification mechanism to assess its validity. (i) Logical validity verification: This step examines whether the derivation of the new proposition conforms to the corresponding logical rules, ensuring that the reasoning chain is formally correct and self-consistent. (ii) Target contribution verification: This step evaluates the relevance between the proposition and the target conclusion, determining whether it provides substantive support for the final proof process. We incorporate the new proposition that passes the two-stage verification into the stable proposition set, that is, $\mathcal{S} \leftarrow \{\mathcal{S} \cup p_{new}\}$, $\mathcal{A} \leftarrow \{\mathcal{A} \cup p_{new}\}$. In addition, we update $\mathcal{M}_E$ with this step as part of the historical reasoning details.

3.3.2 Reduction Agent

In contrast to the Evolution Agent, we also design the Reduction Agent from a critical reasoning perspective, which retrieves potential conflicts from the target conclusion and derives new reasoning insights through conflict resolution. Both agents achieve knowledge transfer through a shared belief space, thereby explicitly modeling each other's reasoning intentions during interaction.

Specifically, we first transform the facts F_t and rules R_t in the belief state space $\mathcal{A}$ into standard logical form through Normalization (Davis and Putnam, 1960) and Skolemization (Nonnengart, 1996), which involves eliminating quantifiers and further converting them into conjunctive normal form. For example, the rule $\forall x\,(P(x) \rightarrow Q(x))$ can be transformed into $\neg P(x) \vee Q(x)$. We then design a contradiction awareness unit to retrieve the conflicting items from the target. We regard propositions that share the same predicate but have opposite polarity as conflicting items. For example, given the target `Smart(Gary, False)`, we search the premises for conditions or facts with opposite polarity, such as `Smart(x, True)` $\vee$ `Furry(x, False)`. These propositions contain the same predicate `Smart` but differ in polarity.

When the retrieved conflicting item constitutes a fact conflict (FC), the reasoning iteration can be immediately terminated, as direct evidence already exists to determine the truth value of the target proposition. This also enhances the overall reasoning efficiency of the framework. In contrast, when the retrieved conflicting item constitutes a rule conflict (RC), we further perform belief conflict resolution to derive new reasoning insights. We apply the resolution principle to eliminate conflicts, which can be formalized as follows:

$$\{X \vee Y, \neg X \vee Z\} \vdash Y \vee Z \quad (4)$$

For example, consider the target conclusion `Smart(Gary, False)` and its conflicting item `Smart(x, True)` $\vee$ `Furry(x, False)`, the resolution principle can be applied to eliminate the conflict and derive a new subgoal `Furry(Gary, False)`, denoted as p_{res}. We align this step with intermediate propositions from EA by performing consistency verification, which checks for conflicts among them. If no factual conflicts are found, we update them into the reduction agent's historical memory $\mathcal{M}_R$. We then continue to perform the above conflict detection and reduction process on p_{res} until a factual conflict emerges or the maximum iteration count of 10 is reached. Therefore, RA's mental variables during iteration include shared belief $\mathcal{A}$ and its own goal p_{res}.

During the reasoning process, the Evolution Agent and Reduction Agent do not operate independently but engage in collaborative planning based on explicit modeling of each other's objectives. By sharing the belief state space $\mathcal{A}$, EA provides RA with a broader set of premises derived from belief evolution, while RA in turn supplies EA with directions of propositional evolution derived from the reduction process. Finally, we design a cognitive balancing layer that performs consistency verification over the reasoning memories $\mathcal{M}_E$ and $\mathcal{M}_R$ of EA and RA, respectively. When the EA no longer produces new facts and the RA no longer generates new subgoals, we obtain a globally consistent final answer. All modules in our method are implemented as instruction prompts, with details provided in ***Appendix F***.

4 Experiments

We present the experimental datasets, baselines, and results in this section.

4.1 Datasets

We evaluate our method on three carefully selected logical reasoning datasets, including ProofWriter (Tafjord et al., 2021), FOLIO (Han et al., 2024), and ProntoQA (Saparov and He, 2023). All datasets adopt a multiple-choice format with accuracy as the primary evaluation metric. ProntoQA focuses on basic deductive logic relations, while ProofWriter introduces more complex structures such as compound operators like "and/or". FOLIO is among the most challenging benchmarks, containing complex first-order logic rules and richly expressed natural language statements. More details can be found in ***Appendix B***.

4.2 LLMs and Baselines

We evaluate the baselines and our method using three open-source large language models Qwen2.5-32B (Yang et al., 2025a), Qwen2.5-72B, and Llama3.3-70B (Grattafiori et al., 2024) and two closed-source models GPT-3.5-turbo (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023).

We compare with a broad range of established baselines, which can be classified into three main categories. **(i) Linear reasoning**, in which the model produces an answer directly based on an initial prompt, including *Naive Prompting* and *CoT* (Wei et al., 2022). **(ii) Aggregated reasoning**, in which the model performs multiple reasoning attempts or aggregates multiple reasoning paths to obtain the final answer, including *CoT-SC* (Wang et al., 2023), *Cumulative Reasoning* (CR; Zhang et al. (2023)), *ToT* (Yao et al., 2023), and *DetermLR* (Sun et al., 2024). **(iii) Symbolic reasoning**, which explicitly incorporates symbolic representations and logical rules into the reasoning framework, including *Logic-LM* (Pan et al., 2023), *SymbCoT* (Xu et al., 2024), and *Aristotle* (Xu et al., 2025b). More details on the baselines and implementation can be found in ***Appendix C*** and ***D***.

4.3 Main Results

The main results are shown in Tables 1 and 2, from which we can observe the following points:

Our method consistently outperforms the baseline across all three datasets. Specifically, for open-source LLMs, we achieve average improvements of 7.71%, 11.18%, and 2.79% over CoT-SC, ToT, and SymbCoT on Qwen2.5-32B, and average improvements of 7.63%, 7.02%, and 8.29% on Qwen2.5-72B. For closed-source LLMs, we achieve average improvements of 5.08% and 2.99% compared to SOTA methods on GPT-3.5-turbo and GPT-4, respectively. Overall results demonstrate that our method consistently exhibits advantages across different model scales and datasets.

Our method proves to be more effective in complex logical reasoning. As shown in Tables 1 and 2, our method does not exhibit significant improvement on the ProntoQA dataset. This can be attributed to the relatively simple nature of the dataset, and most baselines already achieve high accuracy, leaving limited room for further enhancement. In contrast, our method achieves more substantial improvements on logically more complex reasoning datasets. For example, we use Qwen2.5-32B, Qwen2.5-72B, and LLaMA3.3-70B

Method	Qwen2.5-32B				Qwen2.5-72B				LLaMA3.3-70B			
	ProofWriter	FOLIO	ProntoQA	**Avg.**	ProofWriter	FOLIO	ProntoQA	**Avg.**	ProofWriter	FOLIO	ProntoQA	**Avg.**
Linear Reasoning												
Naive	59.67	65.69	83.40	69.59	53.00	66.67	66.20	61.96	69.83	65.69	97.00	77.51
CoT	64.83	72.06	96.80	77.90	72.17	72.55	98.20	80.97	76.67	66.67	98.80	80.71
Aggregative Reasoning												
CoT-SC	61.50	73.04	98.00	77.51	67.00	74.51	98.00	79.84	53.17	67.65	99.40	73.41
CR	60.83	74.51	83.60	72.98	79.17	73.53	**99.60**	84.10	66.83	74.51	90.60	77.31
ToT	62.00	73.53	86.60	74.04	75.83	69.12	96.40	80.45	71.33	72.55	90.20	78.03
DetermLR	65.50	74.02	77.40	72.31	76.33	70.59	85.20	77.37	74.00	75.98	84.80	78.26
Symbolic Reasoning												
Logic-LM	60.17	57.35	89.20	68.91	66.50	66.18	72.80	68.49	73.17	61.77	80.80	71.91
SymbCoT	76.33	72.55	**98.40**	82.43	80.83	74.51	82.20	79.18	76.33	69.91	99.40	81.88
Aristotle	74.33	-	95.60	-	80.03	-	99.20	-	77.10	-	94.20	-
Leibniz(ours)	**79.33**	**77.94**	**98.40**	**85.22**	**86.33**	**76.47**	**99.60**	**87.47**	**82.17**	**76.96**	**99.60**	**86.24**
	(+3.00)	(+3.43)	(+0.00)	(+2.79)	(+5.50)	(+1.96)	(+0.00)	(+3.37)	(+5.07)	(+0.98)	(+0.20)	(+4.36)

Table 1: Experimental results for open-source LLMs. The second-best score is underlined and the best score is **bolded**. The values in parentheses indicate the corresponding improvements.

Method	GPT-3.5-turbo			GPT-4		
	ProofWriter	FOLIO	**Avg.**	ProofWriter	FOLIO	**Avg.**
Naive	35.50	45.09	40.30	52.67	69.11	60.89
CoT	49.17	57.35	53.26	68.11	70.58	69.35
CoT-SC	48.67	57.34	53.01	69.33	68.14	68.74
ToT	54.16	59.80	56.98	70.33	69.12	69.73
Logic-LM	58.33	62.74	60.54	79.66	75.45	77.56
SymbCoT	59.03	57.84	58.44	82.50	83.33	82.92
DetermLR	68.83	63.72	66.28	79.17	75.49	77.33
Aristotle	-	-	-	86.80	76.50	81.65
Leibniz(ours)	**70.17**	**72.55**	**71.36**	**87.50**	**84.31**	**85.91**
	(+1.34)	(+8.83)	(+5.08)	(+0.70)	(+0.98)	(+2.99)

Table 2: Performance on GPT-3.5-turbo and GPT-4.

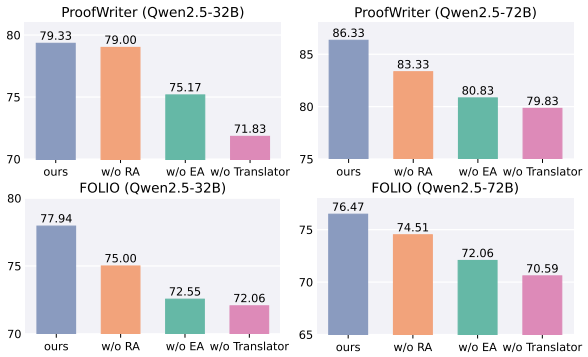

Figure 3: Ablation study results.

to achieve improvements of 3.00%, 5.50%, and 5.07% respectively over previous state-of-the-art methods on ProofWriter, and improvements of 3.43%, 1.96%, and 0.98% respectively on FOLIO. These results demonstrate that our method exhibits a more notable advantage when addressing reasoning tasks with complex logical structures.

5 Analysis and Discussion

We now conduct a more in-depth analysis of our system to explore the reasons behind its progress.

5.1 Ablation Study

To evaluate the effectiveness of each module in our framework, we conduct ablation experiments by sequentially removing three key components of our method: the Reduction Agent (RA), the Evolutionary Agent (EA), and the Translator. The results are shown in Figure 3. From the experimental results, we observe that removing any module leads to a significant performance drop, indicating that each component plays an indispensable role in the overall framework. Specifically, we observe significant performance degradation when removing either the reduction agent or the evolution agent. On ProofWriter and FOLIO, the average accuracy drops by 1.67% (w/o RA) and 4.83% (w/o EA), and by 2.45% (w/o RA) and 4.90% (w/o EA), respectively, highlighting the importance of the dual-agent collaborative reasoning mechanism. Notably, when we remove RA or EA, the model enters a state without ToM. Moreover, when we remove the translator, performance further declines, highlighting the necessity and effectiveness of integrating the neuro-symbolic reasoning mechanism.

5.2 Performance on Complex Reasoning

To further analyze the performance of our method under different reasoning difficulties, we evaluate various methods on the ProofWriter subsets corresponding to different reasoning depths. The questions in these subsets require 0, 1, 2, 3, or 5 reasoning hops, respectively, and the results are shown in Figure 4. We observe that `Leibniz` outperforms the other methods across all reasoning depths and exhibits stronger robustness as the rea-

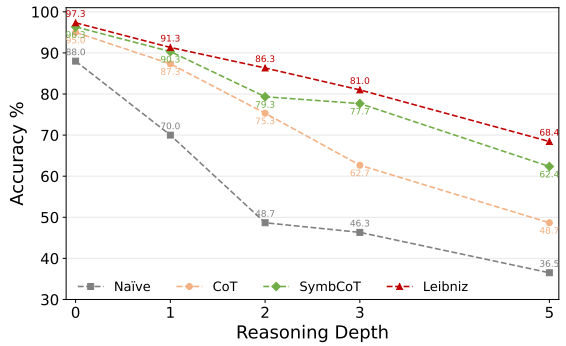

Figure 4: The effect of reasoning depth with Qwen2.5-72B on ProofWriter.

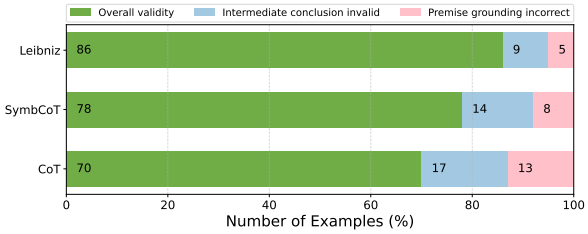

Figure 5: The intermediate reasoning step analysis results of Qwen2.5-72B on ProofWriter.

soning depth increases. **This demonstrates that our method offers better scalability in complex reasoning scenarios and can effectively handle more challenging problems.** These results are primarily attributed to our multi-agent collaborative framework, which enables LLMs to flexibly apply logical rules throughout the step-by-step reasoning process. By self-monitoring and strategic adjustment of intermediate steps, our framework can produce more reliable reasoning decisions.

5.3 Effect of Logical Rule Application

To further analyze the performance of our method in applying logical rules, we conduct a manual evaluation of the intermediate reasoning steps and final conclusions generated by different methods. We randomly sample 100 instances from the ProofWriter and compare the reasoning processes and answer quality along three dimensions: **(i) Premise grounding rationality:** whether the premises selected in the intermediate reasoning steps are appropriate and closely related to the current reasoning objective. **(ii) Intermediate conclusion validity:** whether the intermediate conclusion can be derived from the given premises through correct logical rules. **(iii) Overall validity:** whether the model's stepwise reasoning process can form a coherent and credible reasoning chain that effectively supports the final decision.

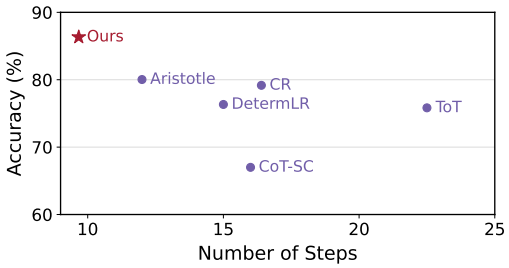

Figure 6: Comparison of reasoning steps and accuracy distribution for Qwen2.5-72B on ProofWriter.

	Qwen2.5-32B	Qwen2.5-72B	LLaMA3.3-70B
ProofWriter	93	96	94
FOLIO	92	95	94
ProntoQA	97	99	97

Table 3: Comparison of conflict resolution success rates(%).

The results are shown in Figure 5. We can observe that our method achieves substantially higher overall validity compared with SymbCoT and CoT, and also outperforms them in premise grounding and intermediate conclusion validity. This indicates that the collaborative reasoning between the Evolution Agent and the Reduction Agent can more effectively anchor and apply logical rules, thereby enhancing the logical reasoning capabilities of LLMs.

5.4 Comparison of Reasoning Step Counts

We further analyze the number of reasoning steps or visited nodes required by different methods to solve the problems in the ProofWriter dataset, and the results are shown in Figure 6. We can see that our method not only achieves the best accuracy compared to the baseline but also requires the fewest number of visited nodes. From the perspectives of exploratory reasoning and critical reasoning, we enhance the ability of LLMs to monitor and refine intermediate steps through stable belief evolution and conflict resolution mechanisms, thereby achieving more efficient and precise logical reasoning.

5.5 Analysis of Conflict Resolution Success Rates

We further analyze the conflict resolution success rate of RA after a conflict is detected. We randomly select 100 individual reasoning steps and calculate the success rate of conflict resolution, and the results are shown in Table 3. The success rate of conflict resolution based on Equation (4) by the LLM consistently remains above 92%, which provides crucial support for the subsequent overall

logical reasoning process. The primary source of errors arises from the failure to correctly introduce entity variables during the resolution procedure, for example:

► **Target:**
`Round(Dave, False).`
► **Contradictory items:**
`Green(x, False)` ∨ `Round(x, True).`
► **Conflict resolution (error):**
`Green(x, False).`
► **Conflict resolution (correct):**
`Green(Dave, False).`

Overall, when a conflict is detected, the model can effectively perform conflict resolution, thereby laying a solid foundation for subsequent accurate reasoning.

5.6 Error Analysis

We conduct an error analysis and identify the following limitations of LLM-based reasoning: **(i) Limited ability to model and capture implicit conditions.** For example, given the premise "`John is Mary's brother`," humans naturally infer that John is male and utilize this attribute in subsequent reasoning. However, LLMs often fail to explicitly recover such implicit information, resulting in reasoning chains that lack necessary semantic support and therefore lead to inconsistent or incomplete conclusions.

(ii) Incorrect selection and application of logical rules. When applying rules, the model may occasionally choose an invalid reasoning pattern or confuse similar logical rules, thereby deriving incorrect conclusions. For example, given the premises $p \rightarrow q$ and $\neg p$, the model may commit the fallacy of *denying the antecedent*. A corresponding natural language example is "`If it rains, the street gets wet. It is not raining. Therefore, the street is not wet.`"

(iii) Deviations arise in the formal translation process. The translation of logical symbols may also result in errors, as the process relies on the LLM's own ability to follow instructions and its formalization capabilities. For example, for the rule "`Nice(x, True)` ∨ `Smart(x, True)` → `Furry(x, True)`", the LLM incorrectly normalizes it using the instruction prompt to "`Nice(x, False)` ∨ `Smart(x, False)` ∨ `Furry(x, True)`", whereas it should be "`(Furry(x, True)` ∨ `Nice(x, False))` ∧ `(Furry(x, True)` ∨ `Smart(x, False))`". This formal error impacts subsequent RA and EA reasoning. More detailed analysis of our proposed method can be found in ***Appendix E***, including aspects such as the number of stable premises, the robustness of symbolic representations, computational efficiency, and case studies.

6 Conclusion

In this paper, we propose `Leibniz`, a theory-of-mind driven multi-agent neuro-symbolic logical reasoning framework. We integrate symbolic representations with natural language throughout the entire reasoning process and construct a bidirectional reasoning model composed of an Evolution Agent and a Reduction Agent. Our framework can mine additional potential propositions that support target conclusions through the belief stability evolution. Moreover, different agents can explicitly model each other's beliefs and intentions, achieving effective collaborative reasoning through continuous belief state updates. Experimental results demonstrate that our method enhances the ability of LLMs to dynamically understand and flexibly apply complex logical rules, thereby achieving more adaptive and reliable logical reasoning.

Limitations

Although our approach has achieved promising results in enhancing the reliability and flexibility of LLM logical reasoning, the following limitations remain. (i) First, our proposed neuro-symbolic method primarily focuses on first-order logic, evaluating its role in enhancing LLM logical reasoning. How more complex symbolic representations (such as higher-order logic) can be effectively integrated with LLM has yet to be systematically explored. (ii) Second, the current reasoning scenarios remain confined to pure text modalities. However, many critical real-world applications (such as high-stakes domains like healthcare, law, and finance) often involve both visual and textual information simultaneously (Xu et al., 2025c). This requires models to possess cross-modal alignment and rigorous reasoning capabilities. How to effectively model and enhance the cross-modal logical reasoning capabilities of LLMs remains an open direction for further investigation.

Ethics Statement

Logical reasoning plays a critical role in human cognition. It progressively transforms ambiguous and uncertain inputs into clear and credible conclusions through rigorous and coherent reasoning

processes, thereby supporting reliable decision-making. All datasets we utilize are sourced from publicly available benchmark or published corpora. Our proposed multi-agent neuro-symbolic framework can enhance the logical reasoning accuracy and reliability of large language models.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the Major Programs of the National Social Science Fund of China (24&ZD227).

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2025. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, Hao Zhou, and Mingxuan Wang. 2025. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. *CoRR*, abs/2505.19914.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 10400–10408. ijcai.org.

William F Clocksin and Christopher S Mellish. 2003. *Programming in PROLOG*. Springer Science & Business Media.

Denise D Cummins, Todd Lubart, Olaf Alksnis, and Robert Rist. 1991. Conditional reasoning and causation. *Memory & cognition*, 19(3):274–282.

Dhairya Dalal, Marco Valentino, André Freitas, and Paul Buitelaar. 2024. Inference to the best explanation in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 217–235. Association for Computational Linguistics.

Martin Davis and Hilary Putnam. 1960. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215.

Leonardo Mendonça de Moura and Nikolaj S. Bjørner. 2008. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2024. Everything of thoughts: Defying the law of penrose triangle for thought generation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 1638–1662. Association for Computational Linguistics.

Yue Fan, Hu Zhang, Ru Li, Yujie Wang, Hongye Tan, and Jiye Liang. 2024. FRVA: fact-retrieval and verification augmented entailment tree generation for explainable question answering. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9111–9128. Association for Computational Linguistics.

Yue Fan, Hu Zhang, Ru Li, Guangjun Zhang, Yujie Wang, Hongye Tan, Yuanlong Wang, Xiaoli Li, and Jiye Liang. 2026. SRCR: faithful structured reasoning with curriculum reinforcement learning for explainable question answering. *Information Processing & Management*, 63(5):104653.

Jiazhan Feng, Ruochen Xu, Junheng Hao, Hiteshi Sharma, Yelong Shen, Dongyan Zhao, and Weizhu Chen. 2024. Language models can be deductive solvers. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4026–4042. Association for Computational Linguistics.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun,

Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22017–22031. Association for Computational Linguistics.

Kang He and Kaushik Roy. 2025. LogicTree: Structured proof exploration for coherent and rigorous logical reasoning with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20863–20892, Suzhou, China. Association for Computational Linguistics.

Qianxi He, Qianyu He, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. 2025. Order doesn't matter, but reasoning does: Training LLMs with order-centric augmentation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27166–27180, Suzhou, China. Association for Computational Linguistics.

Chuanyang Jin, Yutong Wu, Jing Cao, Jiannan Xiang, Yen-Ling Kuo, Zhiting Hu, Tomer D. Ullman, Antonio Torralba, Joshua B. Tenenbaum, and Tianmin Shu. 2024. Mmtom-qa: Multimodal theory of mind question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 16077–16102. Association for Computational Linguistics.

Michal Kosinski. 2023. Theory of mind may have spontaneously emerged in large language models. *CoRR*, abs/2302.02083.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. 2019. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5871–5876. Association for Computational Linguistics.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia P. Sycara. 2023. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 180–192. Association for Computational Linguistics.

Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. 2025a. Logical reasoning in large language models: A survey. *CoRR*, abs/2502.09100.

Tongxuan Liu, Wenjiang Xu, Weizhe Huang, Yuting Zeng, Jiaxing Wang, Xingyu Wang, Hailong Yang, and Jing Li. 2025b. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 10168–10185. Association for Computational Linguistics.

Deepak Nathani, David Wang, Liangming Pan, and William Yang Wang. 2023. MAF: multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6591–6616. Association for Computational Linguistics.

Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Thomas L. Griffiths. 2018. Evaluating theory of mind in question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2392–2400. Association for Computational Linguistics.

Andreas Nonnengart. 1996. Strong skolemization.

Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5153–5176. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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 Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824. Association for Computational Linguistics.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. Large language models meet symbolic provers for logical reasoning evaluation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew M. Botvinick. 2018. Machine theory of mind. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4215–4224. PMLR.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. 2023. Minding language models' (lack of) theory of mind: A plug-and-play multi-character belief tracker. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13960–13980. Association for Computational Linguistics.

Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Leyla Isik, Yen-Ling Kuo, and Tianmin Shu. 2025. Muma-tom: Multi-modal multi-agent theory of mind. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 1510–1519. AAAI Press.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2024. Determlr: Augmenting llm-based logical reasoning from indeterminacy to determinacy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9828–9862. Association for Computational Linguistics.

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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics.

Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen-tse Huang, Pinjia He, Wenxiang Jiao, and Michael R. Lyu. 2024. Logicasker: Evaluating and improving the logical reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2124–2155. Association for Computational Linguistics.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Symbolic working memory enhances language models for complex rule application. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17583–17604. Association for Computational Linguistics.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zehao Wang, Lin Yang, Jie Wang, Kehan Wang, Hanzhu Chen, Bin Wang, Jianye Hao, Defu Lian, Bin Li, and Enhong Chen. 2025. Logictree: Improving complex reasoning of llms via instantiated multi-step synthetic logical data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Alex Wilf, Sihyun Shawn Lee, Paul Pu Liang, and Louis-Philippe Morency. 2024. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8292–8308. Association for Computational Linguistics.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025a. Are large language models really good logical reasoners? A comprehensive evaluation and beyond. *IEEE Trans. Knowl. Data Eng.*, 37(4):1620–1634.

Jundong Xu, Hao Fei, Meng Luo, Qian Liu, Liangming Pan, William Yang Wang, Preslav Nakov, Mong-Li Lee, and Wynne Hsu. 2025b. Aristotle: Mastering logical reasoning with A logic-complete decompose-search-resolve framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3052–3075. Association for Computational Linguistics.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13326–13365. Association for Computational Linguistics.

Jundong Xu, Hao Fei, Yuhui Zhang, Liangming Pan, Qijun Huang, Qian Liu, Preslav Nakov, Min-Yen Kan, William Yang Wang, Mong-Li Lee, and Wynne Hsu. 2025c. Muslr: Multimodal symbolic logical reasoning. *CoRR*, abs/2509.25851.

Zhichao Yan, Jiapu Wang, Jiaoyan Chen, Xiaoli Li, Jiye Liang, Ru Li, and Jeff Z Pan. 2025. Atomic fact decomposition helps attributed question answering. *IEEE Transactions on Knowledge and Data Engineering*, 37(12):6959–6972.

Zhichao Yan, Yunxiao Zhao, Jiapu Wang, Jiaoyan Chen, Shaoru Guo, Xiaoli Li, Ru Li, and Jeff Z. Pan. 2026. Logicscore: Fine-grained logic evaluation of conciseness, completeness, and determinateness in attributed question answering. *CoRR*, abs/2601.15050.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025a. Qwen2.5-1m technical report. *CoRR*, abs/2501.15383.

Xiao-Wen Yang, Jie-Jing Shao, Lan-Zhe Guo, Bo-Wen Zhang, Zhi Zhou, Lin-Han Jia, Wang-Zhou Dai, and Yufeng Li. 2025b. Neuro-symbolic artificial intelligence: Towards improving the reasoning abilities of large language models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 10770–10778. ijcai.org.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jun Zhang, Trey Hedden, and Adrian Chia. 2012. Perspective-taking and depth of theory-of-mind reasoning in sequential-move games. *Cognitive science*, 36(3):560–573.

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

Yunxiao Zhao, Zhiqiang Wang, Xiaoli Li, Jiye Liang, and Ru Li. 2024. AGR: reinforced causal agent-guided self-explaining rationalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 510–518. Association for Computational Linguistics.

Yunxiao Zhao, Zhiqiang Wang, Xingtong Yu, Xiaoli Li, Jiye Liang, and Ru Li. 2026. Learnable game-theoretic policy optimization for data-centric self-explanation rationalization. *IEEE Transactions on Knowledge and Data Engineering*, 38(2):1159–1173.

A Logical Symbol Representation

We use first-order logic as the symbolic representation of natural language. The inputs for logical reasoning include premises composed of facts and rules, along with a problem statement.

Facts are assertions about specific objects within a domain and correspond to simple clauses. Their logical form is a predicate applied to constants, describing particular attributes or relations of these objects. For example, `Nice(Bear, True)` asserts that "The bear is nice."

Rules describe logical relationships between predicates, structured as clauses: $F_1 \wedge ... \wedge F_m \rightarrow F_{m+1} \wedge ... \wedge F_n$, where each F_i denotes a fact. The logical connectors used in rules include "and" ($\wedge$), "or" ($\vee$), "not" ($\neg$), and "either...or..." ($\oplus$), which may appear on either side of implication ($\rightarrow$) or equivalence ($\leftrightarrow$) operators. For example, $\forall x\,(\texttt{Kind}(x) \wedge \texttt{Visits}(x, \texttt{Cat})) \rightarrow \texttt{Big}(x)$ asserts that "`If someone is kind and they visit the cat, then they are big.`"

Queries are another fact that needs to be proven based on known facts and rules. The detailed instructions for symbol translation can be found in ***Appendix F***.

B Datasets Details

PrOntoQA is a synthetic dataset designed to assess the deductive reasoning capabilities of large language models. Following prior work, we adopt the most challenging fictional-people version. This version is divided into subsets based on the number of reasoning hops required, and we use the most difficult 5-hop subset for evaluation in our experiments. Each instance in the dataset requires the model to determine whether a given statement is "true" or "false" based on the provided premises.

ProofWriter is another widely used dataset for studying deductive logical reasoning, with problems expressed in forms that are closer to natural language. In our experiments, we adopt the Open World Assumption (OWA) subset. Each instance in this subset consists of a (question, target) pair, and the model is required to determine the truth value of the target proposition. The dataset is divided into five difficulty levels, corresponding to instances requiring 0, $\leqslant 1$, $\leqslant 2$, $\leqslant 3$, and $\leqslant 5$ reasoning hops. We use the most challenging depth-5 subset for evaluation.

FOLIO is a highly challenging, expert-constructed logical reasoning dataset. It covers open-domain scenarios where premises and conclusions primarily integrate real-world commonsense and domain knowledge, expressed in highly natural language. Solving these problems requires models to possess complex first-order logical reasoning capabilities.

The test sizes are 500 for ProntoQA, 600 for ProofWriter, and 204 for FOLIO, respectively. We use data consistent with previous studies for our evaluation (Pan et al., 2023; Xu et al., 2024; Sun et al., 2024; Xu et al., 2025b).

C Baselines Details

We employ three types of methods as our baseline for comparison:

C.1 Linear reasoning

Naive Prompting does not use any explicit reasoning chains or structured prompts. It only provides task instructions and inputs (such as premises and questions) to large language models, and directly causes them to output the final answer. This setting does not impose constraints or guidance on the intermediate reasoning process of the model, which remains implicit.

Chain-of-Thought (CoT) prompting explicitly requires the model to produce a step-by-step reasoning process. Unlike approaches that only output the final answer, CoT asks the model to first generate an intermediate reasoning chain in natural language and then derive the conclusion, so as to better utilize its potential reasoning capabilities in complex, multi-hop reasoning tasks (Wei et al., 2022).

C.2 Aggregative reasoning

Chain-of-Thought with Self-Consistency (CoT-SC) extends the standard CoT approach by guiding large language models to generate multiple distinct chains of reasoning for the same input. It then aggregates these chains' answers (such as through majority voting) to deliver the final answer. By promoting diverse sampling in the solution space and selecting the most consistent answer, CoT-SC mitigates the instability of single-path reasoning and improves overall accuracy and robustness on complex reasoning tasks (Wang et al., 2023).

Cumulative Reasoning (CR) is a reasoning paradigm that progressively accumulates intermediate conclusions. Specifically, the model does not directly provide the final answer in a single reasoning round. Instead, it iteratively generates, refines, and preserves intermediate reasoning results through multiple rounds of execution. It explicitly feeds back key information obtained in previous rounds into the inputs of subsequent rounds, thereby forming a reasoning trajectory that continuously "accumulates evidence" (Zhang et al., 2023).

Tree-of-Thought (ToT) is a reasoning paradigm that explicitly organizes the intermediate reasoning units of models into a tree structure and performs search over this structure. Instead of extending the reasoning chain along a single path at each step, the model generates multiple candidate thoughts, forming a branching structure. Subsequently, high-quality branches are selected for further expansion through strategies such as heuristic scoring, pruning, or backtracking, until the final answer is obtained (Yao et al., 2023).

DetermLR is a logical reasoning paradigm that reformulates the reasoning process as a gradual convergence from uncertainty to certainty. It partitions the premises into certain and uncertain categories, guiding the model to progressively infer new insights. In addition, the method employs a reasoning memory to store and retrieve historical reasoning paths, thereby enabling more accurate reasoning outcomes (Sun et al., 2024).

C.3 Neural Symbolic Reasoning

Logic-LM first converts natural-language inputs into logical symbolic representations and then applies an external symbolic solver to perform rigorous logical reasoning. This approach leverages formal logical rules to process and analyze the symbolic expressions, enabling more structured and precise reasoning (Pan et al., 2023).

SymbCoT integrates logical symbolic representations into the CoT reasoning process. It first maps natural-language inputs into symbolic expressions, enabling the model to reason directly over these formal representations. Building on this, the model applies symbolic reasoning rules to process and analyze the relevant information, thereby strengthening its capability for rigorous logical reasoning (Xu et al., 2024).

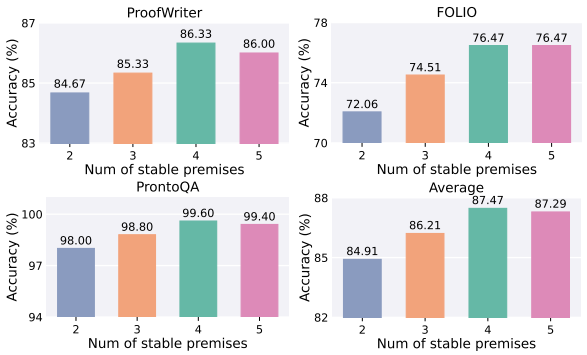

Figure 7: Effect of the number of stable propositions on model performance.

Aristotle constructs a logical reasoning framework composed of a logical decomposer, a logical search router, and a logical solver, explicitly integrating symbolic expressions and logical rules throughout the entire reasoning process. This framework alleviates several key bottlenecks in logical reasoning, including reducing subtask complexity, minimizing search errors, and resolving logical contradictions (Xu et al., 2025b).

D Implementation Details

In principle, our proposed framework imposes no restrictions on the type of LLMs used. To ensure a fair comparison, we use identical in-context examples for all models. To obtain reproducible results, we set the temperature to 0 and select the highest-probability response from the LLM. For experiments on open-source LLMs, all baselines and our method are run on a machine with one NVIDIA Tesla H100 (80GB) GPU.

E Other Experimental Analysis

E.1 Effects of Number of Stability Premises

For the number of stable propositions generated by the Evolution Agent, we set it to 4 in our experiments. It is worth noting that to simultaneously measure the validity and efficiency of reasoning, this parameter is not necessarily better when larger. Results under different settings are shown in Figure 7. When the number of stable propositions is small, the model exhibits lower performance. As the number of generated propositions increases, additional stable propositions help simplify the reasoning process and improve overall performance. When this number is further increased, no additional performance gains are observed. Therefore, considering the trade-off between performance and computa-

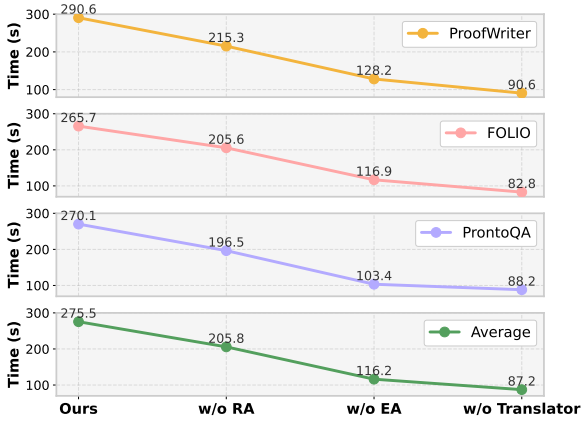

Figure 8: Computational efficiency analysis under different component configurations.

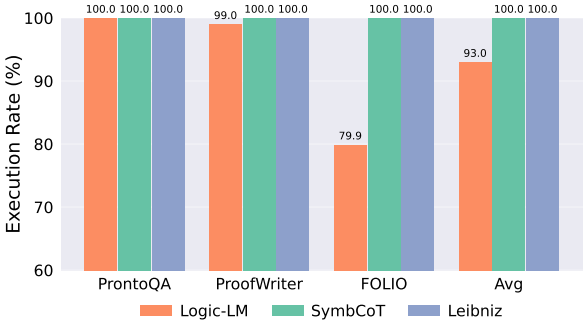

Figure 9: Execution rate for Logic-LM, SymbCoT, and ours.

	ProofWriter		FOLIO	
	Avg. TC	Acc	Avg. TC	Acc
ToT	5216.4	75.83	5413.2	69.12
Leibniz	4548.5	86.33	4973.6	76.47

Table 4: Comparison of average token consumption and accuracy.

tional efficiency, we ultimately set the number of stable propositions to 4.

E.2 Computational Efficiency Analysis

We further analyze the impact of individual components in `Leibniz` on computational efficiency[1]. By sequentially removing key modules, we measure the average processing time per sample under different configurations, as shown in Figure 8. To ensure a fair comparison, all experiments in this analysis are conducted on the same device. We can observe that removing EA yields the greatest acceleration effect, reducing the average processing time by 57.8%. Further removing the Translator reduces the average processing time by 25.0%, reflecting the additional computational overhead introduced by the symbolic representation and transformation process.

Although removing EA or RA can significantly improve computational efficiency, the goal-oriented reasoning trajectories formed through collaborative reasoning by these two agents are crucial for solving complex logical reasoning problems. Similarly, the translator plays a vital role in overall reasoning performance. By bridging natural language and symbolic representations throughout multi-agent reasoning, it enables the model to simultaneously leverage the rigor of symbolic reasoning and the flexibility of neural models, thereby substantially improving reasoning accuracy.

We further analyze the per-sample inference time of `Leibniz` and the baseline methods, with the results shown in Figure 10. We observe that, compared with cumulative reasoning methods, our approach exhibits a clear advantage in reasoning speed. Although our method is slightly slower than the neuro-symbolic approach SymbCoT in terms of inference time, by introducing an Evolution Agent to mine more implicit propositions and combining it with a Reduction Agent for conflict resolution, our approach enables LLMs to more effectively self-monitor and dynamically adjust intermediate reasoning processes, thereby achieving more adaptive and robust logical reasoning.

Moreover, we further report the average token consumption on Qwen2.5-72B, as shown in Table 4. It can be observed that, compared with the baseline methods, our approach achieves superior performance while maintaining a relatively lower average token consumption. This improvement is mainly attributed to the conflict detection mechanism introduced in the RA module: once an explicit contradiction is identified, the system can determine the truth value of the target at an early stage, thereby avoiding redundant subsequent reasoning steps and improving overall reasoning efficiency.

In summary, `Leibniz` achieves an explicit trade-off between reasoning accuracy and computational efficiency. The above experimental results further highlight the value of the proposed multi-agent neuro-symbolic reasoning framework in complex logical reasoning scenarios.

[1]To ensure a fair comparison, we report the inference time of Qwen2.5-72B under different methods on the same device. Note that the runtime may vary across different hardware environments.

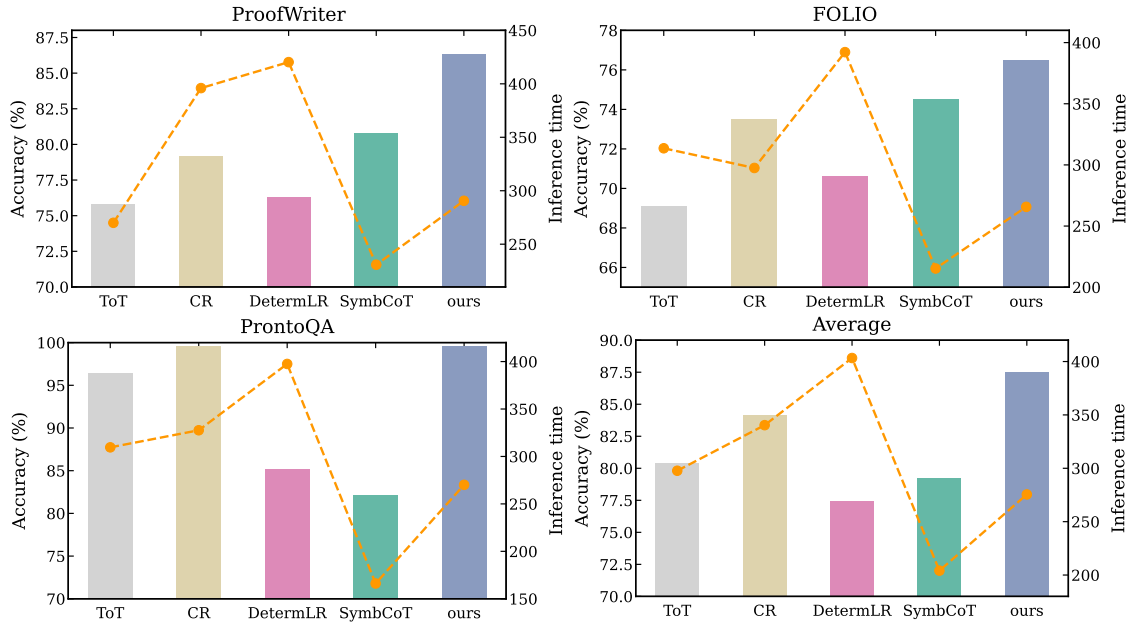

Figure 10: Comparison of per-sample inference time (s) and accuracy (%) of different methods on Qwen2.5-72B, where the line plot represents inference time.

E.3 Robustness of Symbolic Representation

Accurately translating natural language statements into symbolic representations is crucial for building reliable neuro-symbolic logical reasoning systems. To evaluate the robustness of our method with respect to symbolic representations, we conduct a comparative analysis between approaches that rely on external solvers (e.g., Logic-LM) and fully LLM-based reasoners. The evaluation metric focuses on whether the generated symbolic expressions can be successfully parsed and executed, i.e., the execution rate of the symbolic expressions (Execution Rate). For example, if there are 100 questions and the method successfully executes 90 of them, the execution rate is 90%. We use the same external solver as Logic-LM for evaluation.

The results are shown in Figure 9. We can observe that both our method and SymbCoT achieve 100% execution success rates, representing an average improvement of 7.0% over Logic-LM. On the challenging logical reasoning dataset FOLIO, our method outperforms approaches based on external symbolic solvers by 20.1%. Overall, our approach consistently achieves high execution success rates across different datasets and reasoning difficulties, which fully demonstrates its robustness against syntactic errors in symbolic expressions.

It is worth noting that we employ both natural language and symbolic representations throughout the reasoning process. This approach not only enhances the system's flexibility but also effectively alleviates the sensitivity issues that can arise from relying solely on symbolic representations. Moreover, the incorporation of natural language provides richer contextual information for reasoning, thereby strengthening the model's robustness and reliability when solving complex problems. Future research could introduce interpretability analysis mechanisms (Yan et al., 2025; Zhao et al., 2024, 2026) to constrain and optimize the LLM's symbol translation process, thereby yielding symbol representations that are more faithful and consistent at both the semantic and syntactic levels.

E.4 Case Study

We conduct a case study for `Leibniz`, as illustrated in Figure 11. The inputs for logical reasoning include facts F, rules R, and questions Q. `Leibniz` sequentially performs three modules of initialization, bidirectional neuro-symbolic reasoning, and cognitive balance to derive the truth value of the final conclusion.

In the initialization stage, we first translate the facts F, rules R, and query Q expressed in natural language into first-order logic representations, denoted as F_t, R_t, and Q_t, respectively. During subsequent reasoning processes, we simultaneously retain and utilize both the natural language and their corresponding symbolic representations for reasoning. This allows us to ensure the rigor of symbolic expressions while enhancing the overall

robustness of representations, thereby improving the accuracy of logical reasoning.

In the bidirectional neuro-symbolic reasoning stage, we introduce an Evolution Agent and a Reduction Agent to perform collaborative reasoning. Specifically, the Evolution Agent first categorizes the given facts and rules into stable and unstable premises based on belief stability, and then generates additional stable propositions that are conducive to proving the target conclusion through a logical evolution mechanism. In parallel, the Reduction Agent takes the target conclusion as its starting point, continuously searches for potential conflicts within the known premises, and derives new reasoning insights through conflict resolution and reduction operations.

The two agents collaborate closely through continuous updates of their belief states. During logical evolution and logical reduction, both the EA and the RA explicitly take into account each other's current knowledge states and reasoning progress, thereby forming a goal-oriented collaborative search trajectory within the reasoning space. We enable bidirectional collaboration between EA and RA by sharing the belief state space $\mathcal{A}$: EA expands the set of premises for RA based on belief evolution, while RA employs reductive reasoning to indicate effective directions for EA's propositional evolution.

Finally, we align and verify the consistency of the reasoning memories from both agents during **the cognitive balancing stage**, combining their reasoning results to derive a globally consistent final answer. The cognitive balance layer is also implemented as an agent. Specifically, this layer first performs premise–conclusion consistency checking on the reasoning trajectories produced by the evolution agent and the reduction agent. If the reduction agent exhibits an explicit contradiction, the truth value of the target can be directly determined. Otherwise, we combine the stability propositions generated by the evolution agent and leverage the LLM to derive the final answer.

Facts:

Anne is blue. Anne is furry. Anne is green. Anne is quiet. Anne is young. Bob is furry. Bob is round. Erin is green. Erin is young. Harry is furry.

Rules:

Furry people are blue. If someone is furry and young then they are green.
All quiet, round people are rough.
If Anne is green and Anne is blue then Anne is round.
Furry people are quiet. All rough, quiet people are young.
Quiet people are round.

Question:

Harry is not young.

(a) Initialization

Predicates:

Blue(x, bool) ::: Is x blue?
Furry(x, bool) ::: Is x furry?
Green(x, bool) ::: Is x green?
Quiet(x, bool) ::: Is x quiet?
Young(x, bool) ::: Is x young?
Round(x, bool) ::: Is x round?
Rough(x, bool) ::: Is x rough?

Facts:

Blue(Anne, True) ::: Anne is blue.
Furry(Anne, True) ::: Anne is furry.
Green(Anne, True) ::: Anne is green.
Quiet(Anne, True) ::: Anne is quiet.
Young(Anne, True) ::: Anne is young.
Furry(Bob, True) ::: Bob is furry.
Round(Bob, True) ::: Bob is round.
Green(Erin, True) ::: Erin is green.
Young(Erin, True) ::: Erin is young.
Furry(Harry, True) ::: Harry is furry.

Rules:

Furry(x, True) >>> Blue(x, True) ::: Furry people are blue.
Furry(x, True) ∧ Young(x, True) >>> Green(x, True) ::: If someone is furry and young then they are green.
Quiet(x, True) ∨ Round(x, True) >>> Rough(x, True) ::: All quiet, round people are rough.
Green(Anne, True) ∧ Blue(Anne, True)) >>> Round(Anne, True) ::: If Anne is green and Anne is blue then Anne is round.
Furry(x, True) >>> Quiet(x, True) ::: Furry people are quiet.
Rough(x, True) ∨ Quiet(x, True) >>> Young(x, True) ::: All rough, quiet people are young.
Quiet(x, True) >>> Round(x, True) ::: Quiet people are round.

Question:

Young(Harry, False) ::: Harry is not young.

(b) Bidirectional Neuro-Symbolic Reasoning

Evolution Agent

Belief-Stable Premise

Anne is young. Erin is young. Harry is furry. All rough, quiet people are young.

Belief-Unstable Premise

Anne is blue. Anne is furry. Anne is green. Anne is quiet. Bob is furry.
Bob is round. Erin is green. Furry people are blue.
If someone is furry and young then they are green.
All quiet, round people are rough.
If Anne is green and Anne is blue then Anne is round.
Furry people are quiet. Quiet people are round.

Belief Evolution

Round 1:
From [Furry(Harry, True) ::: Harry is furry.] and [Furry(x, True) >>> Blue(x, True) ::: Furry people are blue.] apply ***Modus Ponens*** to infer that [**Blue(Harry, True) ::: Harry is blue.**]

Round 2:
From [Furry(Harry, True) ::: Harry is furry.] and [Furry(x, True) >>> Quiet(x, True) ::: Furry people are quiet.] apply ***Modus Ponens*** to infer that [**Quiet(Harry, True) ::: Harry is quiet.**]

Round 3:
From [Quiet(Harry, True) ::: Harry is quiet.] and [Quiet(x, True) >>> Round(x, True) ::: Quiet people are round.] apply ***Modus Ponens*** to infer that [**Round(Harry, True) ::: Harry is round.**]

Round 4:
From [Round(Harry, True) ::: Harry is round.], [Quiet(Harry, True) ::: Harry is quiet.] and [Quiet(x, True) ∨ Round(x, True) >>> Rough(x, True) ::: All quiet, round people are rough.] apply ***Modus Ponens*** and ***Conjunction Introduction*** to infer that [**Rough(Harry, True) ::: Harry is rough.**]

Reduction Agent

Logical Normalization

Furry(x,False) ∨ Blue(x,True)
Furry(x,False) ∨ Young(x,False) ∨ Green(x,True)
(Rough(x,True) ∨ Quiet(x,False)) ∧ (Rough(x,True) ∨ Round(x,False))
Furry(x,False) ∨ Blue(x,False) ∨ Green(x,False) ∨ Round(x,True)
Furry(x,False) ∨ Quiet(x,True)
(Young(x,True) ∨ Rough(x,False)) ∧ (Young(x,True) ∨ Quiet(x,False))
Quiet(x,False) ∨ Round(x,True)

Contradiction Reduction

Round 1:
Target: Young(Harry, False)
Retrieve contradictory items:
Rough(x,False) ∨ Quiet(x,False) ∨ Young(x,True)
Resolve contradictions: **Rough(Harry, False) ∨ Quiet(Harry, False)**

Round 2:
Target: Rough(Harry, False) ∨ Quiet(Harry, False)
Retrieve contradictory items:
(Quiet(x,False) ∨ Rough(x,True))
Resolve contradictions: **Quiet(Harry, False)**

Round 3:
Target: Quiet(Harry, False)
Retrieve contradictory items:
Furry(x,False) ∨ Quiet(x,True)
Resolve contradictions: **Furry(Harry, False)**

Round 4:
Target: Furry(Harry, False)
Retrieve contradictory items:
Furry(Harry, True)
Contradiction

Shared Belief

(c) Cognitive Balancing

Combining the results of both agents, no new facts can be derived by the Evolution Agent and no new sub-goals remain for the Reduction Agent. Thus, they jointly conclude that the statement "*Harry is not young*" is **False**.

Figure 11: An illustration of the reasoning process in Leibniz. The **green box** contains the input for logical reasoning, while the **blue box** shows the problem-solving process of our method.

F Full Prompting

Below is the detailed prompt of our method.

Step-1: Translation

Task Description:
You are given a problem description and a question. The task is to:

1. Define all the predicates in the problem.
2. Parse the problem into logic rules based on the defined predicates.
3. Write all the facts mentioned in the problem.
4. Parse the question into the logic form.

Problem:

Facts (F):

- The cat needs the mouse. The cat sees the squirrel. The cat does not see the tiger. The mouse is red. The mouse does not visit the cat. The squirrel is big. The squirrel visits the tiger. The tiger is big. The tiger sees the mouse. The tiger visits the squirrel.

Rules (R):

- If someone visits the tiger and the tiger needs the mouse then the mouse sees the squirrel. If someone is blue and red then they need the tiger. If someone needs the tiger then the tiger needs the mouse. If someone visits the squirrel then they are not blue. Red people are blue. If someone needs the cat and they are not big then they do not visit the squirrel. If someone sees the squirrel then they need the mouse.

Question:

- True or false: The mouse visits the cat.

Predicates:

- `Needs(x, y)`: Does x need y?
- `Red(x)`: Is x red?
- `Visits(x, y)` ::: Does x visit y?
- (... More predicates ...)

Facts (F_t ::: F):

- `Needs(Cat, Mouse)` ::: The cat needs the mouse.
- `Sees(Cat, Squirrel)` ::: The cat sees the squirrel.
- `Red(Mouse)` ::: The mouse is red.
- (... More facts ...)

Rules (R_t ::: R):

- `Visits(x, Tiger)` $\wedge$ `Needs(Tiger, Mouse)` $\rightarrow$ `Sees(Mouse, Squirrel)` ::: If someone visits the tiger and the tiger needs the mouse then the mouse sees the squirrel.
- `Blue(x)` $\wedge$ `Red(x)` $\rightarrow$ `Needs(x, Tiger)` ::: If someone is blue and red then they need the tiger.
- `Visits(x, Squirrel)` $\rightarrow$ `Blue(x, False)` ::: If someone visits the squirrel then they are not blue.
- `Needs(x, Cat)` $\wedge$ `Big(x, False)` $\rightarrow$ `Visits(x, Squirrel, False)` ::: If someone needs the cat and they are not big then they do not visit the squirrel.
- (... More rules ...)

Query (Q_t ::: Q):

- `Visits(Mouse, Cat, False)` ::: The mouse does not visit the cat.

Step-2.1: Evolution Agent - Stability Premise Identification

Task Description:
Assume you are a leading AI scientist, logician, and mathematician. Perform the task using explicit, step-by-step reasoning.

1. Carefully read and internalize the following definition: A premise is considered a stable premise if (a) it contains the same noun or adjective as the Conclusion, and (b) it is expressed as a declarative statement, not in a conditional form such as "if . . . " or "if . . . then . . . ".
2. Carefully examine the given symbolic Premise and Conclusion. Determine whether the Premise qualifies as a stable premise according to the definition above.
3. Ensure that your classification decision is derived strictly and explicitly from the given definition. Do not rely on background knowledge, world knowledge, or unstated assumptions.

Premises (P_t ::: P):

- `Blue(Anne)` ::: Anne is blue.
- `Furry(Anne)` ::: Anne is furry.
- `Green(Anne)` ::: Anne is green.
- `Quiet(Anne)` ::: Anne is quiet.
- `Young(Anne)` ::: Anne is young.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- `Furry(x)` $\vee$ `Young(x)` $\rightarrow$ `Green(x)` ::: If someone is furry and young then they are green.
- `Quiet(x)` $\wedge$ `Round(x)` $\rightarrow$ `Rough(x)` ::: All quiet, round people are rough.
- `Blue(Anne)` $\vee$ `Green(Anne)` $\rightarrow$ `Round(Anne)` ::: If Anne is green and Anne is blue then Anne is round.
- `Furry(x)` $\rightarrow$ `Quiet(x)` ::: Furry people are quiet.
- `Rough(x)` $\vee$ `Quiet(x)` $\rightarrow$ `Young(x)` ::: All rough, quiet people are young.
- `Quiet(x)` $\rightarrow$ `Round(x)` ::: Quiet people are round.
- (... More premises ...)

Conclusion (Q_t ::: Q):

- `Young(Harry, False)` ::: Harry is not young.

Stable premises ($\mathcal{S}$):

- `Young(Anne)` ::: Anne is young.
- `Young(Erin)` ::: Erin is young.
- `Furry(Harry)` ::: Harry is furry.
- `Rough(x)` $\vee$ `Quiet(x)` $\rightarrow$ `Young(x)` ::: All rough, quiet people are young.

Unstable premises ($\mathcal{U}$):

- `Blue(Anne)` ::: Anne is blue.
- `Furry(Anne)` ::: Anne is furry.
- `Green(Anne)` ::: Anne is green.
- `Quiet(Anne)` ::: Anne is quiet.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- `Furry(x)` $\vee$ `Young(x)` $\rightarrow$ `Green(x)` ::: If someone is furry and young then they are green.
- (... More other unstable premises ...)

Step-2.2: Evolution Agent - Symbolic Rule Grounding

Task Description:
Suppose you are one of the greatest AI scientists, logicians, and mathematicians. Let us reason step by step.

1. First, read and analyze the stable premise and unstable premise sets, and then select relevant premises from them.
2. Next, read the reasoning history of the Evolution Agent ($\mathcal{M}_E$) and the Reduction Agent ($\mathcal{M}_R$), along with the belief state space $\mathcal{A}$ (including stable and unstable propositions). If a false proposition appears in the history, and a specific Optimal Premise has already been used there, do not select the same premise again as your answer.
3. From the stable premise set, select the Optimal Premise that shares the same subject as the Conclusion, and assign it a relevance score between 0 and 1. Assess how this Optimal Premise relates to all other stable premises and unstable premises, according to the Relevance Scoring Rules. All stable premises and unstable premises with relevance scores greater than 0.25 are included as the final results, together with the Optimal Premise.
4. Relevance Scoring Rules:
   (a) When computing relevance, add 0.25 for each shared noun and 0.3 for each shared adjective between two sentences.
   (b) Relevance scores start from 0 and accumulate, with an upper limit of 1.0.
   (c) If sentence p1 is a hypothetical or conditional premise of sentence p2, add 0.25 points to p1.

Mental variables:

Goal (Q_t ::: Q):

- `Young(Harry, False)` ::: Harry is not young.

Shared belief ($\mathcal{A}$) – Stable premises ($\mathcal{S}$):

- `Young(Anne)` ::: Anne is young.
- `Young(Erin)` ::: Erin is young.
- `Furry(Harry)` ::: Harry is furry.
- `Rough(x)` $\vee$ `Quiet(x)` $\rightarrow$ `Young(x)` ::: All rough, quiet people are young.

Shared belief ($\mathcal{A}$) – Unstable premises ($\mathcal{U}$):

- `Blue(Anne)` ::: Anne is blue.
- `Furry(Anne)` ::: Anne is furry.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- (... More other unstable premises ...)

RA's goal belief: $\mathcal{M}_R$

Belief State Memory of EA: $\mathcal{M}_E$

Optimal premise p^*:

- `Furry(Harry)` ::: Harry is furry.

Candidate premise combinations $\mathcal{H}$:

- `Furry(Harry)` ::: Harry is furry.
- `Furry(Anne)` ::: Anne is furry.
- `Furry(Bob)` ::: Bob is furry.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- `Furry(x)` $\vee$ `Young(x)` $\rightarrow$ `Green(x)` ::: If someone is furry and young then they are green.
- `Furry(x)` $\rightarrow$ `Quiet(x)` ::: Furry people are quiet.

Step-2.3: Evolution Agent - Symbolic Rule Implementation

Task Description:
Assume the role of a highly capable AI system specialized in artificial intelligence, formal logic, and mathematical reasoning. Perform step-by-step logical reasoning. Your task is to derive one new "Proposition" from the given "Premises" by strictly applying first-order logic (FOL) reasoning rules.

1. The derived Proposition must not contain conditional operators such as "if".
2. The Proposition must be logically valid and soundly entailed by the given Premises.
3. The proposition must be neither identical to nor a trivial restatement of any given premise.
4. The Proposition must be derived only from the provided Premises, without introducing: external background knowledge, commonsense assumptions, or any unsourced information.
5. The first-order-logic inference rules include but are not limited to: Modus Ponens, Modus Tollens, Generalization, Specialization, Conjunction, Elimination, Transitivity, Proof By Division Into Cases, Contradiction Rule, and etc.
6. Logical Reasoning Rules Examples:
   (a) Modus Ponens: If `Cold(x)` $\rightarrow$ `Likes(x, cat)` and `Cold(mouse)`, then new Proposition: `Likes(mouse, cat)`.
   (b) Modus Tollens: If `Blue(x)` $\rightarrow$ `Eats(x, Rabbit)` and `Eats(Tiger, Rabbit, False)`, then new Proposition: `Blue(Tiger, False)`.
   (c) Disjunctive Syllogism: If `Plant(x)` $\oplus$ `Animal(x)` and `Plant(eel, False)`, then new Proposition: `Animal(eel, True)`.
   (d) Transitivity: If `Eel(x)` $\rightarrow$ `Fish(x)` and `Fish(x)` $\rightarrow$ `Plant(x, False)`, then new Proposition: `Eel(x)` $\rightarrow$ `Plant(x, False)`.

Conclusion (Q_t ::: Q):

- `Young(Harry, False)` ::: Harry is not young.

Candidate premise combinations $\mathcal{H}$:

- `Furry(Harry)` ::: Harry is furry.
- `Furry(Anne)` ::: Anne is furry.
- `Furry(Bob)` ::: Bob is furry.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- `Furry(x)` $\vee$ `Young(x)` $\rightarrow$ `Green(x)` ::: If someone is furry and young then they are green.
- `Furry(x)` $\rightarrow$ `Quiet(x)` ::: Furry people are quiet.

Symbolic Cognitive Deduction:

- From `Furry(Harry)` ::: Harry is furry, `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue, applying Modus Ponens to infer that "`Blue(Harry)` ::: Harry is blue".

Proposition p_{new}:

- `Blue(Harry)` ::: Harry is blue.

Step-2.4: Evolution Agent - Logical Verification

▷ **Logical Validity Verification:**

Task Description:

1. Given a set of Premises and a derived Proposition, evaluate the logical validity of the derivation using the Logical Reasoning Rules.
2. Specifically, verify whether the Proposition is soundly inferred from the Premises by strictly following the corresponding logical inference rules, and whether the entire reasoning process is formally correct and internally consistent.
3. Output True if the derivation conforms to the Logical Reasoning Rules and constitutes a valid logical inference; otherwise, output False.
4. Logical Reasoning Rules Examples:
   (a) Modus Ponens: If `Cold(x)` $\rightarrow$ `Likes(x, cat)` and `Cold(mouse)`, then new Proposition: `Likes(mouse, cat)`.
   (b) Modus Tollens: If `Blue(x)` $\rightarrow$ `Eats(x, Rabbit)` and `Eats(Tiger, Rabbit, False)`, then new Proposition: `Blue(Tiger, False)`.
   (c) Disjunctive Syllogism: If `Plant(x)` $\oplus$ `Animal(x)` and `Plant(eel, False)`, then new Proposition: `Animal(eel, True)`.
   (d) Transitivity: If `Eel(x)` $\rightarrow$ `Fish(x)` and `Fish(x)` $\rightarrow$ `Plant(x, False)`, then new Proposition: `Eel(x)` $\rightarrow$ `Plant(x, False)`.

Premise:

- `Furry(Harry)` ::: Harry is furry.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.

Proposition p_{new}:

- `Blue(Harry)` ::: Harry is blue.

Logical Judgment: The reasoning step is logically valid. {True}.

▷ **Target Contribution Verification:**

Task Description:

1. Given the premises and the conclusion, assess whether the premises qualify as stable premises and thus provide support for the target conclusion.
2. Carefully read and internalize the following definition: A stable premise is defined as a premise that shares the same noun or adjective with the conclusion and is not expressed in a conditional form, such as "if ..." or "if ... then ...". Analyze the given premises and conclusion accordingly, and determine whether the new proposition satisfies this definition.
3. Ensure that your judgment is based strictly on the stated definition and the given texts, without relying on external knowledge, implicit assumptions, or common-sense reasoning.

Premise:

- `Furry(Harry)` ::: Harry is furry.
- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.

Proposition p_{new}:

- `Blue(Harry)` ::: Harry is blue.

Conclusion (Q_t):

- `Young(Harry, False)` ::: Harry is not young.

Logical Judgment: The proposition provides support for the target conclusion. {True}.

Step-3: Reduction Agent

▷ **Logical Normalization:**

Task Description:

1. Given a set of premises and a conclusion expressed in formal logic, systematically transform the logical formulas through Normalization and Skolemization.
2. Normalization: Convert each premise and the conclusion into Prenex Normal Form, and subsequently transform them into Conjunctive Normal Form.
3. Skolemization: Eliminate existential quantifiers by introducing appropriate Skolem constants or Skolem functions, while preserving satisfiability and logical equivalence under universal quantification.

Premise p_t (Mental variables – shared belief $\mathcal{A}$):

- `Furry(x)` $\rightarrow$ `Blue(x)` ::: Furry people are blue.
- `Quiet(x)` $\rightarrow$ `Round(x)` ::: Quiet people are round.
- `Furry(x)` $\vee$ `Young(x)` $\rightarrow$ `Green(x)` ::: If someone is furry and young then they are green.
- (... More premises ...)

Conclusion (Q_t):

- `Young(Harry, False)` ::: Harry is not young.

Normalized Premise:

- `Furry(x,False)` $\vee$ `Blue(x,True)`.
- `Quiet(x,False)` $\vee$ `Round(x,True)`.
- `Furry(x,False)` $\vee$ `Young(x,False)` $\vee$ `Green(x,True)`.
- (... More normalized premises ...)

Normalized Conclusion:

- `Young(Harry, False)`.

▷ **Conflict Resolution:**

Task Description:

1. Given two logical clauses, determine whether they can be resolved by identifying conflicting terms and applying the resolution principle.
2. Two terms are considered conflicting if they have the same predicate symbol and identical arguments, but differ either in polarity (one is negated while the other is not) or in their boolean truth value (True versus False).
3. If a pair of conflicting terms is identified across the two clauses, apply the resolution principle to eliminate the conflicting literals and derive a new clause. For instance, resolving $P(x, True) \vee Q(x, True)$ with $P(x, False) \vee M(x, True)$ yields the resolvent $Q(x, True) \vee M(x, True)$.

Current subgoal (Mental variables – own goal p_{res}):

- `Young(Harry, False)`.

Conflicting terms:

- `Rough(x,False)` $\vee$ `Quiet(x,False)` $\vee$ `Young(x,True)`.

New subgoal (p'_{res}):

- `Rough(Harry, False)` $\vee$ `Quiet(Harry, False)`.