MultiLogicNMR(er): A Benchmark and Neural-Symbolic Framework for Non-monotonic Reasoning with Multiple Extensions

Yeliang Xiu, Yongmei Liu*
Dept. of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
xiuyliang@mail2.sysu.edu.cn; ymliu@mail.sysu.edu.cn

Abstract

Non-monotonic reasoning (NMR) refers to the fact that conclusions may be invalidated by new information. It is widely used in daily life and legal reasoning. An NMR task usually has multiple extensions, which are sets of plausible conclusions. There are two reasoning modes – skeptical and credulous reasoning, depending on whether to believe facts in all extensions or any one extension. Despite some preliminary work exploring the NMR abilities of LLMs, the multi-extension NMR capabilities of LLMs remain underexplored. In this paper, we synthesize a multi-extension NMR dataset MultiLogicNMR, and construct two variants of the dataset with more extensions or text diversity. We propose a neural-symbolic framework MultiLogicNMRer for multi-extension NMR. Experimental evaluation with the datasets shows that LLMs still face significant challenges in NMR abilities, and reveal the effectiveness of our neural-symbolic framework, with an average accuracy gain of about 15% compared to prompt-based methods, and even outperforming some fine-tuning methods. All code and data are publicly available[1].

1 Introduction

Logical reasoning is a fundamental aspect of human intelligence and plays a central role in intelligent systems for problem solving and decision making (Brachman and Levesque, 2004). Large language models (LLMs) have recently made remarkable progress in NLP and related fields, and demonstrated certain reasoning abilities (Wei et al., 2022). Naturally, logical reasoning over natural language has received much attention. Various datasets have been proposed that cover deductive, inductive, and abductive reasoning. Various methods have been explored, including prompting, fine-tuning, and neural-symbolic methods.

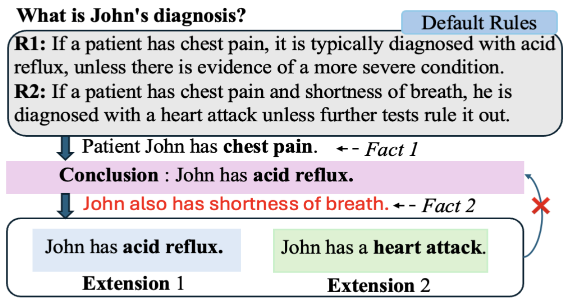

Figure 1: An example of multi-extension NMR.

Non-monotonic reasoning (NMR) is one of the important reasoning modes in logic (Lukaszewicz, 1990), and has applications in daily decision making (Szalas, 2019), medical diagnosis (Jackson, 1989), and legal reasoning (Calegari et al., 2019). Generally, NMR refers to the fact that conclusions may be invalidated by new information. Most of what we learn about the world is in terms of generics, properties that hold "in general", but with exceptional cases. Such generics are called default rules. Figure 1 shows an example of medical diagnosis with two default rules. When we first learn that patient John has chest pain, we conclude that he has acid reflux. Suppose that we later know that John also has shortness of breath. Now, two competing rules apply. This leads to two possible extensions, *i.e.*, sets of plausible conclusions: John has acid reflux, assuming that Rule 1 overrides Rule 2; John has heart attack, assuming that Rule 2 overrides Rule 1. What conclusions should we draw? There are two reasoning modes: skeptical reasoning only believes in common facts in all extensions, while credulous reasoning believes in facts in any one extension. For instance, by applying skeptical reasoning, we withdraw the conclusion that John has acid reflux, because it is not a fact present in every possible extension. Skeptical reasoning is suitable for domains like legal reasoning, which require a high degree of certainty. In

*Corresponding author.
[1]https://github.com/sysulic/MultiLogicNMRer

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18372–18405
November 4-9, 2025 ©2025 Association for Computational Linguistics

contrast, credulous reasoning is better suited for scenarios such as creative problem-solving, where it is valuable to explore possibilities. The choice between these two reasoning modes is therefore critical, as each corresponds to different practical application needs.

NMR has been thoroughly studied in symbolic AI. Various logics have been proposed to formalize NMR, and one of the most popular ones is the default logic proposed by Reiter (1980). In default logic, a set of facts and default rules is called a default theory. The complexities of skeptical and credulous reasoning in default logic are harder than satisfiability and validity-testing in propositional logic, respectively. NMR can be implemented with ASP (Answer set programming) (Lifschitz, 2008), a declarative programming paradigm where the solutions of a program are represented as answer sets. An ASP program is a set of logic rules, and an answer set of a program is a minimal consistent set of facts that satisfy all the rules in the program. When a default theory satisfies a certain syntactic constraint, it can be conveniently transformed to an ASP program so that the answer sets of the program are exactly extensions of the theory. The bound-based algorithm is a powerful technique in ASP to efficiently compute answer sets avoiding exhaustive search (Gebser et al., 2012).

In recent years, some work has been done exploring the NMR abilities of LLMs. Rudinger et al. (2020) used NLI datasets to build an NMR dataset δ-NLI through crowdsourcing. However, δ-NLI entangles NMR with commonsense reasoning. To disentangle the two, Xiu et al. (2022) synthesized a NMR benchmark LogicNMR, with explicit facts and rules. However, LogicNMR only involves NMR with a single extension. Recently, by combining LLM-generated sentences and rule combination templates, Parmar et al. (2024) and Patel et al. (2024), respectively, constructed comprehensive benchmarks LogicBench and Multi-LogiEval, covering NMR. However, the multi-extension NMR capabilities of LLMs remain underexplored.

In this paper, we first construct the MultiLogicNMR dataset for multi-extension NMR in both skeptical and credulous reasoning modes. We follow the synthesis method for building deductive reasoning datasets such as RuleTaker (Clark et al.), ProntoQA (Boratko et al., 2020) and LogicNLI (Tian et al., 2021). In particular, we first synthesize samples in default logic, solve them with an ASP solver, and then translate the samples into natural language using templates. In addition, we construct two dataset variants MultiLogicNMR_OOD and MultiLogicNMR_NL to evaluate models' generalization to NMR with more extensions and the robustness of models to text diversity. We propose a neural-symbolic framework MultiLogicNMRer for multi-extension NMR, by plugging into the framework of the bound-based algorithm for ASP solving various LLM-based modules to perform single-step operations. Experimental results show that the multi-extension NMR capabilities of prompt-based models are limited, with an average accuracy of about 50%. However, our framework can enhance the NMR capabilities of LLMs, with an average accuracy gain of about 15% over prompting methods, and even outperforming some fine-tuning methods.

2 Related Work

Preliminaries: Similarly to LogicNMR (Xiu et al., 2022), in this work, we use Reiter's default logic as the logic underlying MultiLogicNMR. A default rule is of the form: $\alpha : \beta_1, \beta_2, \ldots, \beta_m / \gamma$, where α, β_i and γ are formulas in first-order logic (FOL), α is called the prerequisite, $\beta_1, \beta_2, \ldots, \beta_m$ the justifications, and γ the conclusion. The default rule can be interpreted as: if α can be inferred and $\beta_1, \beta_2, \ldots, \beta_m$ are consistent, then γ can be deduced. A default theory is a pair $\Gamma = \langle D, W \rangle$, where D is a set of default rules, and W is a set of facts, which are first-order sentences. For example, a default theory Γ_0 consists of $D_0 = \{prof(x) : teaches(x) \,/\, teaches(x),\ chair(x) : \neg\, teaches(x) \,/\, \neg\, teaches(x)\}$ and $W_0 = \{prof(J),\ chair(J)\}$, where J is a constant. A set of sentences E is an extension of $\Gamma = \langle D, W \rangle$ iff for every sentence π, $\pi \in E$ iff $W \cup \Delta \models \pi$, where $\Delta = \{\gamma \mid \alpha : \beta_1, \ldots, \beta_m/\gamma \in D, \alpha \in E, \neg\, \beta_1, \ldots, \neg\, \beta_m \notin E\}$, and $\models$ is the logic entailment relation of FOL. So, an extension E is the set of entailments of $W \cup \Delta$, where Δ is a set of conclusions of default rules from D. For example, Γ_0 has two extensions $E_1 = W_0 \cup \{teaches(J)\}$, $E_2 = W_0 \cup \{\neg\, teaches(J)\}$.

NMR over Natural Language: In addition to the four works discussed in the introduction, there are some other preliminary explorations of the NMR capabilities of LLMs. Antoniou and Batsakis (2023) conducted a preliminary evaluation of the NMR capabilities of GPT3.5 on ten classic examples, showing a large gap compared to symbolic solvers. Leidinger et al. (2024) evaluated the

NMR capabilities of LLMs across two datasets, showing that LLMs fail to reason consistently and robustly when adding supporting or unrelated facts. In addition, Kazemi et al. (2023b) proposed a defeasible reasoning benchmark BoardGameQA for reasoning with contradictory information guided by preferences over sources, illustrating a significant gap in LLMs' abilities in such reasoning.

Datasets for Logical Reasoning over Natural Language: Many datasets have been proposed for logical reasoning over natural language. Deductive reasoning datasets include RuleTaker (Clark et al.), ProntoQA (Saparov and He, 2023), EntailmentBank (Dalvi et al., 2021), FOLIO (Han et al., 2024). Inductive reasoning is covered by datasets such as bAbI (Weston et al., 2016) and CLUTRR (Sinha et al., 2019). For abductive reasoning, AlphaNLI (Bhagavatula et al., 2020) is a key dataset.

Neural-symbolic Methods for Logical Reasoning over Natural Language: Neural-symbolic methods have been widely explored for logical reasoning over natural language, and can be categorized into two types: search-based and autoformalization-based. Search-based approaches typically integrate LLMs to perform single-step reasoning operations within classical search frameworks. For example, Hao et al. (2023) proposed the RAP framework based on Monte Carlo tree search (Browne and et. al., 2012), and Kazemi et al. (2023a) proposed LAMBADA based on the backward chaining algorithm. Autoformalization-based approaches combine translation with LLMs from natural languages to formal languages and rigorous reasoning of symbolic solvers. Typical works are Logic-LM (Pan et al., 2023) and LINC (Olausson et al., 2023). In addition, Ishay et al. (2023) and Coppolillo et al. (2024) explored autoformalization into ASP and reasoning with ASP solvers. However, auto-formalization is prone to grammatical and semantic errors and information loss. In this paper, we propose a search-based neural-symbolic framework for multi-extension NMR.

3 Dataset Generation

Figure 2 illustrates the process of generating the dataset. The main idea is to first synthesize samples formalized with default logic, solve them using an ASP solver, and then translate the samples into natural language using templates. To enhance text diversity, we employ an LLM to rewrite the sentences generated with templates.

Generating Default Theories: In this paper, we generate default theories with a syntactic constraint, which we first introduce. A term is a constant or a variable. An atom has the form $P(t_1, \ldots, t_n)$, where P is an n-ary predicate symbol, and $t_1, \ldots, t_n$ are terms. A literal is an atom or the negation of an atom. A default rule $\alpha : \beta_1, \beta_2, \ldots, \beta_m/\gamma$ is literal-based if α is a conjunction of literals, $\beta_1, \ldots, \beta_m$ and γ are all literals. A default theory is literal-based if each default rule is literal-based and each fact is a literal. Baral and Gelfond (1994) showed that a literal-based default theory can be conveniently transformed to an ASP program so that the answer sets of the program are exactly extensions of the theory, where a default rule $\alpha : \beta_1, \beta_2, \ldots, \beta_m/\gamma$ is translated into the ASP rule γ :- $\alpha, not\,\neg\beta_1, \ldots, not\,\neg\beta_m$, where not is negation-as-failure in logic programming. In this paper, we focus on literal-based default rules where α is a conjunction of at most two literals, and there are at most two justifications.

We now describe how to generate a default theory. LogicNLI (Tian et al., 2021) is a synthesized dataset for first-order reasoning in natural language. We use the predicate list and the constant list in constructing LogicNLI: the predicate pool contains unary and binary predicates of adjectives such as "intelligent", and the constant pool consists of names such as "Toby". The default rules are generated in sequence. Each rule is generated by generating the sequence of literals in the rule. Each literal is generated by first generating an atom of the form $P(x)$ for a unary-predicate P or $Q(x, y)$ for a binary-predicate Q, and then negating the atom with a probability of 50%. The predicate of a prerequisite literal is selected from the conclusion predicates in previous rules with a probability of 50%. Similarly, the predicate of a prerequisite (resp. conclusion) literal is selected from the prerequisite (resp. justification) predicates in previous rules with a probability of 50%. To generate facts and questions, we randomly pick two different constants C_1 and C_2 from the constant pool. Each fact is generated by first randomly picking a literal from literals that appear in a prerequisite literal but no conclusion literal, and then instantiating it with C_1 or C_1 and C_2. Each question is generated by first generating an atom of the form $P(C_1)$, and then negating the atom with a probability of 50%. The predicate of a question is randomly picked from predicates that appear in a conclusion literal but no prerequisite literal. Finally, we filter out those

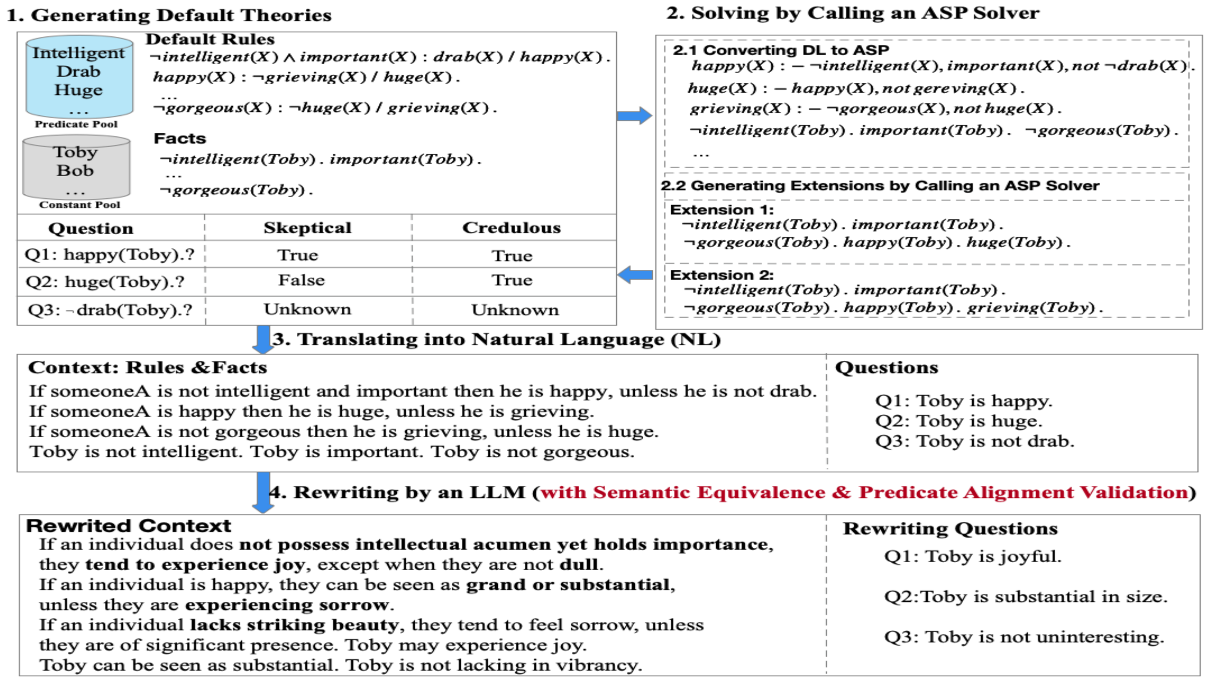

Figure 2: The framework for automatically constructing multi-extension NMR datasets.

theories with no extension. This is done using the method we describe next.

Solving by Calling an ASP Solver: Given a literal-based default theory Γ and a question Q, we now explain how to automatically compute the answers in both skeptical and credulous reasoning modes. The most widely used ASP solver is clingo[2](Gebser and et al., 2019). We first convert Γ into an ASP program using the afore-mentioned translation, and then call clingo to compute all extensions of Γ. Equation 1 (resp. 2) shows the answer for skeptical (resp. credulous) reasoning, where $E \vdash Q$ means $Q \in E$. The answers may be True (T), False (F), or Unknown (M). Note that for the credulous reasoning mode, we filter out those theories such that both a question and its negation belong to some extension.

$$A_S(\Gamma, Q) = \begin{cases} T, \text{ if } \forall_E \, E \vdash Q\,; \\ F, \text{ if } \forall_E \, E \vdash \neg Q\,; \\ M, \text{ if } \exists_E \, E \nvdash Q, \exists_E \, E \nvdash \neg Q\,. \end{cases} \tag{1}$$

$$A_C(\Gamma, Q) = \begin{cases} T, \text{ if } \exists_E \, E \vdash Q\,; \\ F, \text{ if } \exists_E \, E \vdash \neg Q\,; \\ M, \text{ if } \forall_E \, E \nvdash Q, E \nvdash \neg Q\,. \end{cases} \tag{2}$$

Translating into Natural Language: In this step, we translate all the ASP rules, facts, and questions into natural language using templates. For example, the ASP rule "$grieving(x)$:- $\neg\, gorgeous(x), not\, huge(x)$." is translated into "If someoneA is not gorgeous, then he is grieving, unless he is huge.".

Rewriting Samples by an LLM: To enhance the text diversity of the synthesized samples, we employ GPT-4o-mini to rewrite the generated sentences. To ensure the semantic and logical correctness of the rewritten samples, we use two validation mechanisms to ensure the correctness of the rewritten samples, and when a rewritten sample fails the two validations, the rewriting process is repeated. Semantic equivalence is used to measure whether the rewritten sample is semantically equivalent to the original sample. We use a prompt-based LLM to generate a label "True" or "False" to judge semantic equivalence. For each sample, we generate labels four times, and consider semantic equivalence holds when the model outputs "True" at least three times. Secondly, predicate alignment mandates that distinct predicates in the original sample must not be translated into the same words. This can effectively filter out rewritten samples with logical errors generated by incorrect predicate mapping. Specifically, a prompt-based LLM is first used to extract the predicates in the samples, and then it is judged whether the predicates are aligned based on semantic similarity. Figures 7, 8, and 9 in Appendix A.2 present the prompts to rewrite samples, evaluate semantic equivalence, and extract predicates from the samples.

We first generate a basic dataset **MultiLogic-**

[2]https://pypi.org/project/clyngor/

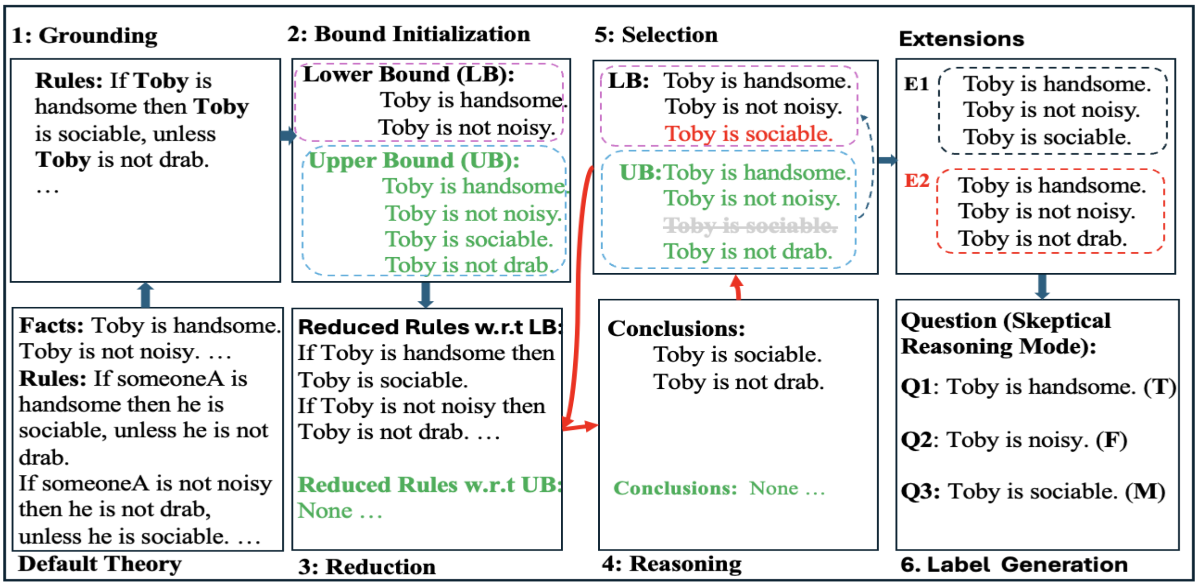

Figure 3: The proposed neural-symbolic framework, MultiLogicNMRer.

NMR and a dataset **MultiLogicNMR_OOD** with more extensions without using LLM-rewriting, and then construct **MultiLogicNMR_NL** based on MultiLogicNMR using LLM-rewriting. The statistical information for all datasets is shown in Table 3 in Appendix A.1. Each sample contains three questions. The number of extensions in each sample of MultiLogicNMR belongs to $R = \{1, 2, 3, 4, 5\}$, and the number of samples with each number of extensions is equal. **MultiLogicNMR_OOD** is aimed at measuring the generalization of LLMs in the number of extensions. The number of extensions in each sample of MultiLogicNMR_OOD belongs to $R = \{6, 8, 10, 12, 16\}$. **MultiLogicNMR_NL** is aimed for evaluating the robustness of models in the diversity of natural languages. We use the Remote Clique and Chamfer Distance indicators (Li et al., 2023; Cox et al., 2021) as well as LLM-based scoring to evaluate the diversity of samples. Tables 4 and 5 in Appendix A.2 show the diversity results of MultiLogicNMR and MultiLogicNMR_NL in terms of evaluation metrics and LLM evaluation, respectively. The evaluation results show that the diversity of the rewritten MultiLogicNMR_NL is higher than that of MultiLogicNMR. Figures 5 and 6 in Appendix A.1 show an example of MultiLogicNMR and the LLM-rewritten example in MultiLogicNMR_NL, respectively.

4 A Neural Symbolic Framework

We propose a neural-symbolic framework MultiLogicNMRer for multi-extension NMR based on the bound-based algorithm for ASP solving, by plugging into the algorithm framework various LLM-based modules to perform single-step operations. The bound-based algorithm (Algorithm 1) is a powerful technique in ASP solving, and it is part of the core solving strategy of clingo. It computes all answer sets by iteratively refining a lower bound (definitely true atoms, initialized as the empty set) and an upper bound (possibly true atoms, initialized as the set of all atoms) and branching on an atom in the difference of the two bounds. The refinement of the lower (resp. upper) bound is done by reducing the logic program w.r.t. the upper (resp. upper) bound and then draw conclusions using the reduced program. Note that the algorithm applies to normal logic programs, *i.e.*, logic programs without classical negation. However, it can be easily extended to logic programs with classical negation, which forms the foundation of our implementation.

Figure 3 shows the framework of MultiLogicNMRer, which includes six modules. The bottom left shows the input of the framework, which is a default theory in natural language; the top right shows the set of extensions computed with the framework. The solving process starts with grounding and bound initialization, and proceeds to reduction and reasoning. When the lower bound is not a subset of the upper bound, which signifies a conflict, no extension is found; when the two bounds are the same, an extension is found; otherwise, the selection module selects from the difference of the two bounds, and the process continues with two

Algorithm 1: Bound-based ASP Solving

Input: Logic Program **P**.

$expand_p$(L, U)
    repeat
        $L' \leftarrow L, U' \leftarrow U$
        $L \leftarrow L' \cup Cn(P^{U'})$
        $U \leftarrow U' \cap Cn(P^{L'})$
    Until ($L = L'$ and $U = U'$) or $L \not\subseteq U$
$solver_p$(L, U)
    $(L, U) \leftarrow expand_p(L, U)$
    if $L \not\subseteq U$ **then** failure
    if $L = U$ **then** output L
    else $a \leftarrow choose(U \setminus L)$
        $solve_p(L \cup \{a\}$, U)
        $solve_p(L, U \setminus \{a\}$)
$P \leftarrow ground(\mathbf{P})$
$L, U \leftarrow init(P)$
$solve_p$(L, U)

branches and moves to reduction. When all extensions are computed, the solving process ends with label generation. All modules except the selection module are implemented with prompt-based LLMs, and the prompts are shown in Appendix A.3.

Grounding Module replaces pronouns in rules with constants. For example, the rule "If someoneA is handsome then he is delicious, unless he is not drab." is instantiated as "If Toby is handsome then Toby is delicious, unless Toby is not drab.".

Bound Initialization Module extracts literal statements from the ground rules and initializes the upper and lower bounds. The lower bound (LB) is initialized as the set of facts of the default theory in natural language, and the upper bound (UB) is initialized as the set of extracted literal statements.

Reduction Module obtains the reduced rules w.r.t. the lower and upper bounds. For a rule γ :- $\alpha, not\, \neg\beta_1, \ldots, not\, \neg\beta_m$, if none justification β_i appears in the lower or upper bound, it can be reduced to γ :- α; otherwise, it cannot be reduced.

Reasoning Module draws conclusions from the facts and reduced rules, and updates the lower and upper bounds. The idea of drawing conclusions is to iteratively reason through each reduced rule until no new conclusions can be generated. In each iteration, to reason through all the reduced rules, there are two choices. The first is to use a single LLM call; the second is to use a LLM call for each reduced rule. For performance reason, we chose the second approach, and in the experiments section, we will do a performance comparison with the first. The lower bound is updated to include conclusions from the reduced rules w.r.t. the upper bound; the upper bound is updated to exclude atomic statements that cannot be concluded from the reduced rules w.r.t. the lower bound.

Selection Module selects a literal statement that is in the upper bound but not in the lower bound. The selection is done using similarity based on semantic vectors[3]. Then the computation is continued on two branches: On one, the lower bound is updated by including the literal statement; on the other, the upper bound is updated by excluding the literal statement.

Label Generation Module labels the questions based on all extensions found. We first determine whether the question statement is contained in each extension, and then label the question according to skeptical reasoning or credulous reasoning.

5 Experiments

5.1 Experimental Settings

We systematically evaluate different methods for NMR across different models on the three datasets MultiLogicNMR, MultiLogicNMR_OOD and MultiLogicNMR_NL. The models include closed-source LLMs (GPT-3.5-turbo (Brown et al., 2020), GPT-4o-mini (OpenAI, 2023), o3-mini[4], and open-source LLMs (DeepSeek-R1-32B (DeepSeek-AI et al., 2025), Gemma3-27B (Kamath et al., 2025)). The methods include prompting, fine-tuning, and autoformalization-based baselines, and our neural-symbolic framework MultiLogicNMRer. The prompting strategies include standard zero/few-shot and symbolic algorithm-based prompting (AlgCoT). AlgCoT prompts the model to solve according to the bound-based algorithm. Appendix A.6 shows the prompts of all the methods. The autoformalization-based method (NL2ASP) first uses the model to translate natural language into ASP and then calls clingo to solve. The LowRank Adaptation (LoRA) is used to fine-tune open-source LLMs on the training set. Table 6 in Appendix A.4 gives the hyperparameters of the fine-tuned models. Finally, we do human evaluation by selecting 100 MultiLogicNMR samples and asking three computer science graduate students to solve them. Human evaluation guidelines are described in Appendix A.5. To ensure the reproducibility of

[3]distilbert-base-nli-stsb-mean-tokens
[4]https://openai.com/index/openai-o3-mini/

the experimental results, we set the temperature parameters of all models to 0. We use accuracy as the evaluation metric.

5.2 Main Results

Table 1 shows the accuracy of the methods in the models in the three datasets in the skeptical and credulous reasoning modes. The main observations are: The NMR capabilities of prompt-based models are limited, with an average accuracy of about 50%. However, our framework MultiLogicNMRer can enhance the NMR capabilities of LLMs, with an average accuracy gain of about 15% compared to prompt-based models, and even outperforming some fine-tuned models. In the following, we analyze the performance of the methods on each of the three datasets.

The above observations are most obvious on MultiLogicNMR. First, the NMR capabilities of the prompt-based models are relatively limited. Among these, o3-mini generally performs better than GPT3.5-turbo and GPT4o-mini, indicating that o3-mini has stronger abilities to solve complex logical reasoning tasks. In addition, AlgCoT can effectively enhance the reasoning abilities of the model. For example, AlgCoT on Gemma3-27B achieves accuracies of 61.0% and 61.5% in skeptical and credulous reasoning, respectively. Second, the performance of the autoformalization-based method varies across models. For example, on o3-mini it achieves accuracies of 71.3% and 69.3% in the two reasoning modes, but on DeepSeek-R1-32B only 37.8% and 45.1%. Third, fine-tuning can significantly improve the NMR abilities of the models. For example, the fine-tuned DeepSeek-R1-32B achieves 76.5% and 79.8% accuracies. Finally, our framework achieves the best results across all methods with only one exception.

MultiLogicNMR_OOD is used to evaluate the generalization of the method's NMR abilities with respect to the number of extensions. Although the number of extensions in MultiLogicNMR_OOD is higher than that in MultiLogicNMR, we can observe results and trends similar to those in MultiLogicNMR. This shows that increasing the number of extensions does not significantly increase the difficulty of reasoning. However, it is worth noting that since the number of rules and facts in MultiLogicNMR_OOD samples is higher than that in MultiLogicNMR, the autoformalization-based method usually performs worse on MultiLogicNMR_OOD than on MultiLogicNMR.

MultiLogicNMR_NL is used to evaluate the robustness of the methods against text diversity. Clearly, each method performs worse on MultiLogicNMR_NL than on MultiLogicNMR. For example, in skeptical reasoning, the accuracy of fine-tuned Gemma3-27B drops from 70.1% to 44.7%, and the accuracy of the autoformalization-based method on o3-mini drops from 71.3% to 36%. This fully illustrates that text diversity increases the difficulty of reasoning. Nonetheless, our framework MultiLogicNMRer still beats the other methods with a possible exception of the fine-tuning method.

5.3 Other NMR Datasets

To further verify the effectiveness of our framework, we evaluate MultiLogicNMRer (using the credulous reasoning mode) on LogicNMR (Xiu et al., 2022) and Multi-LogiEval(nm) (Patel et al., 2024) datasets, where nm shows the NMR subdatset. As shown in Table 2, on GPT4o-mini, MultiLogicNMRer clearly outperforms prompt-based methods with the only exception of d1 for Multi-LogiEval(nm). Specifically, MultiLogicNMRer achieves an accuracy of 74.3% on LogicNMR, much higher than those of the prompt-based methods. On Multi-LogiEval(nm), although the increase in reasoning depth greatly challenges the prompt-based methods, MultiLogicNMRer still achieves high accuracies, revealing that the effectiveness of MultiLogicNMRer is not affected by reasoning depth. It should be noted that MultiLogicNMRer performs poorly on samples with a reasoning depth of 1, a possible explanation is that such samples require more commonsense reasoning.

5.4 Analysis

5.4.1 Label Analysis on MultiLogicNMR

To analyze the challenges of LLMs on MultiLogicNMR, Figure 28 in Appendix A.7 shows the label distribution generated by methods in skeptical reasoning. The results show that few-shot prompting on GPT4o-mini has the lowest accuracy for questions with "Unknown" labels, which is only 116/500. Although fine-tuning can improve the model's accuracy on questions with "True" and "False" labels, it still performs poorly on questions with "Unknown" labels. A possible reason is that it is challenging for LLMs to find all extensions to answer questions with "Unknown" labels correctly. Finally, MultiLogicNMRer on DeepSeek-R1-32B can correctly answer 381/500 questions with "Unknown" labels while maintaining high accuracies

Table 1: Accuracy (%) of methods across models on the three datasets. Within the same model, the best result is **bold**, and the second best result is underlined.

Model	Method	MultiLogicNMR		MultiLogicNMR_OOD		MultiLogicNMR_NL	
		skeptical	credulous	skeptical	credulous	skeptical	credulous
o3-mini	Few-Shot	51.9	67.6	47.3	67.9	55.5	63.1
	Few-Shot AlgCoT	51.3	63.3	51.1	50.4	53.5	50.7
	NL2ASP	71.3	69.3	52.3	50.3	36.0	38.0
GPT3.5 -turbo	Zero-Shot	42.2	38.0	39.3	42.0	41.2	43.7
	Few-Shot	43.4	42.7	38.0	46.0	36.3	45.0
	Zero-Shot AlgCoT	43.7	36.7	39.7	39.3	39.2	39.2
	Few-Shot AlgCoT	47.8	48.9	47.3	49.9	40.9	47.5
	NL2ASP	38.8	37.5	36.7	35.0	34.0	33.6
	MultiLogicNMRer	**64.5**	**65.1**	**59.5**	**60.7**	**48.5**	**55.3**
GPT4o -mini	Zero-Shot	43.2	43.9	41.3	40.1	37.9	44.9
	Few-Shot	52.6	52.6	42.5	44.9	38.8	47.7
	Zero-Shot AlgCoT	42.4	41.8	38.9	37.6	37.9	43.8
	Few-Shot AlgCoT	49.2	48.5	40.3	44.9	39.4	46.1
	NL2ASP	64.3	59.7	57.7	46.8	35.7	37.8
	MultiLogicNMRer	**74.9**	**82.8**	**79.8**	**77.3**	**57.5**	**62.9**
DeepSeek -R1-32B	Zero-Shot	45.0	46.5	42.5	41.1	41.7	44.5
	Few-Shot	43.1	57.5	43.0	44.4	40.7	44.4
	Zero-Shot AlgCoT	47.0	51.2	44.3	50.7	44.2	49.0
	Few-Shot AlgCoT	47.8	51.1	42.1	49.5	39.7	43.1
	NL2ASP	37.8	45.1	28.1	32.6	33.3	33.7
	Fine-tuning	**76.5**	79.8	63.0	**75.9**	**59.1**	**68.7**
	MultiLogicNMRer	75.3	**80.7**	**71.7**	73.5	53.3	59.9
Gemma3 -27B	Zero-Shot	46.7	53.6	41.1	49.3	38.6	47.8
	Few-Shot	53.6	60.6	49.9	62.1	45.9	56.3
	Zero-Shot AlgCoT	50.5	67.3	48.3	63.3	44.0	57.5
	Few-Shot AlgCoT	61.0	61.5	50.7	63.2	52.8	56.8
	NL2ASP	38.7	48.0	27.7	31.7	36.0	36.0
	Fine-tuning	70.1	81.0	61.3	73.6	44.7	**69.7**
	MultiLogicNMRer	**82.0**	**82.8**	**80.7**	**81.9**	**55.8**	60.0
Human		89.3	95.4	-	-	-	-

Table 2: Results on GPT4o-mini on LogicNMR and Multi-LogiEval(nm). We select 100 samples from LogicNMR. The d_i indicates the reasoning depth is i.

Method	Accuracy (%)					
	Logic NMR	Multi-LogiEval(nm)				
		d1	d2	d3	d4	d5
Zero-Shot	27.3	42.5	70.2	50.0	52.5	40.0
Few-Shot	31.9	**46.8**	74.8	42.5	55.0	35.0
MultiLogic NMRer	**74.3**	36.2	**75.7**	**64.2**	**83.6**	**95.0**

for questions with other labels. These results further illustrate that MultiLogicNMRer is more effective than the prompting and fine-tuning methods. Appendix A.7 shows the label distribution.

5.4.2 Ablation Analysis for MultiLogicNMRer

We conduct an ablation study to verify the contributions of the modules in MultiLogicNMRer. We consider four variant methods. The first, denoted by MultiLogicNMRer(allatonce), uses a single LLM call to reason through all reduced rules, as described about the reasoning module in Section 4. The rest are obtained by omitting the grounding module, the reduction module, and both the grounding and reduction modules, respectively. Figure 4 illustrates the results of the ablation study, where we observe a clear drop in performance for the four variant methods. In particular, the allatonce variant exhibits a substantial drop in performance because here LLM calls are used for more complex reasoning tasks in contrast to MultiLogicNMRer where such a reasoning task is decomposed into a series of simpler and more manageable subtasks, accomplished by LLM calls.

5.4.3 Error Analysis on MultiLogicNMR_NL

As the text diversity increases, the correctness of the grounding, reduction, and reasoning modules decreases. Table 8 in Appendix A.7 shows cor-

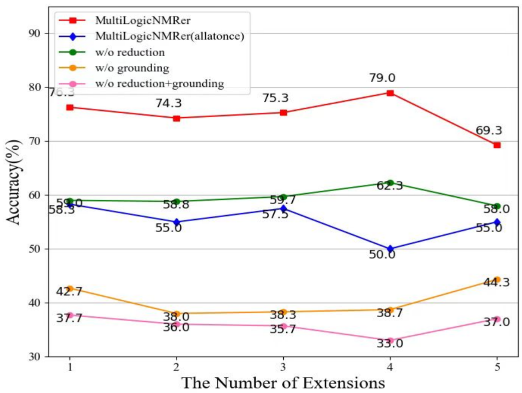

Figure 4: Results of Ablation on MultiLogicNMR dataset in skeptical reasoning.

rect and incorrect outputs generated by different modules in MultiLogicNMRer on the MultiLogicNMR_NL dataset. When faced with complex sentences, the grounding module can confuse justification facts with prerequisite facts, and the reasoning module may fail to draw straightforward conclusions. For example, for justification "except when they are unknown or feel relieved", the grounding module can mistake justification facts as prerequisite facts. In the presence of the fact "Sinclair exhibits an essence of purity" and the rule "If Sinclair embodies purity then Sinclair is prosperous", the reasoning module fails to draw the conclusion "Sinclair is prosperous". Our analysis reveals that the reduction and reasoning modules are the most error-prone components of the MultiLogicNMRer.

6 Discussions

Limitations of the Synthetic Dataset. Our work relies on a synthetic dataset, which, while enabling controlled evaluation, may not fully capture the complexity of real-world scenarios. This limitation is shared by many logical reasoning benchmarks such as such as RuleTaker (Clark et al.), LogicNLI (Tian et al., 2021) and PrOntoQA (Saparov and He, 2023). To mitigate this, we created the MultiLogicNMR_NL dataset by rewriting samples with an LLM, enhancing textual diversity while protecting logical correctness through semantic equivalence and predicate alignment verifiers. For further improvements, we are interested in exploring hybrid methods that combine symbolic templates with LLM calls to create linguistically more natural and complex datasets. However, a challenge is to balance complexity with correctness.

Computational Cost and Scalability. The proposed MultiLogicNMRer is computationally intensive, requiring approximately between 100 and 200 LLM calls to solve a single sample. This reflects a common trade-off between accuracy and computational cost in advanced neural-symbolic systems, similar to those seen in deductive reasoning frameworks such as LAMBADA (Kazemi et al., 2023a) and RAP (Hao et al., 2023). Future work will focus on designing more computationally efficient neural-symbolic solvers, potentially by optimizing the number of required LLM interactions without compromising reasoning performance.

Evaluation Metrics. We have only used accuracy as the evaluation metric, which is standard for evaluation of logical reasoning on datasets such as RuleTaker (Clark et al.) and FOLIO (Han et al., 2024). However, accuracy alone may not fully capture the correctness of the underlying reasoning process. In a very recent work, (Ren et al., 2025) proposed the Exact Match metric to evaluate the correctness of derived extensions. Thus developing more fine-grained metrics for multi-extention non-monotonic reasoning remains a challenging but important area for future research.

7 Conclusions

NMR is an important mode of logical reasoning, and an NMR task usually has multiple extensions. This paper explores and improves the multi-extension NMR capabilities of LLMs: we not only construct three datasets (MultiLogicNMR, MultiLogicNMR_OOD, and MultiLogicNMR_NL), but also propose a neural-symbolic framework MultiLogicNMRer, for multi-extension NMR. Using the datasets, we systematically evaluate the multi-extension NMR abilities of LLMs in both skeptical and credulous reasoning modes. Our evaluation shows that LLMs still face great challenges in NMR, and MultiLogicNMRer significantly improves the NMR capabilities of LLMs. Our work reveals the potential of neural-symbolic approaches for NMR on natural language. In the future, we are interested in building more reliable, general, and computationally more efficient NMR solvers.

Acknowledgements

We acknowledge support from the Natural Science Foundation of China under Grant No. 62476295.

Limitations

Below we outline the limitations of our datasets and neural-symbolic framework. First, our primary dataset MultiLogicNMR is synthetically generated. Although this approach allows for a controlled evaluation of logical reasoning, the samples lack the linguistic complexity and real-world semantic grounding found in naturally occurring text. Second, our MultiLogicNMRer framework is computationally intensive, requiring 100 to 200 sequential LLM calls to solve a single sample. This high cost currently limits its practical applicability in resource-constrained scenarios. Furthermore, our evaluation relies primarily on accuracy, which measures the correctness of the final answer but does not directly assess the validity of the intermediate reasoning steps or the generated extensions. Finally, our fine-tuning experiments were constrained by available hardware, necessitating the use of quantized versions of the DeepSeek-R1-32B and Gemma3-27B models on a single 4090 GPU. The performance of these quantized models may not fully represent that of their original, unquantized models.

References

Grigoris Antoniou and Sotiris Batsakis. 2023. Defeasible reasoning with large language models - initial experiments and future directions. In *RuleML+RR 2023, Oslo, Norway, 18 - 20 September, 2023*, volume 3485.

Chitta Baral and Michael Gelfond. 1994. Logic programming and knowledge representation. *J. Log. Program.*, 19/20:73–148.

Chandra Bhagavatula, Ronan Le Bras, and et. al. 2020. Abductive commonsense reasoning. In *ICLR 2020*.

Michael Boratko, Xiang Li, and et al. 2020. Protoqa: A question answering dataset for prototypical commonsense reasoning. In *EMNLP 2020, November 16-20, 2020*, pages 1122–1136.

Ronald J. Brachman and Hector J. Levesque. 2004. *Knowledge Representation and Reasoning*.

Tom B. Brown, Benjamin Mann, and Nick Ryder et al. 2020. Language models are few-shot learners. In *NeurIPS 2020, December 6-12, 2020, virtual*.

Cameron B Browne and et. al. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Roberta Calegari, Giuseppe Contissa, and et al. 2019. Defeasible systems in legal reasoning: A comparative assessment. In *JURIX 2019, Madrid, Spain, December 11-13, 2019*, volume 322 of *Frontiers in Artificial Intelligence and Applications*, pages 169–174. IOS Press.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3882–3890.

Erica Coppolillo, Francesco Calimeri, and et al. 2024. LLASP: fine-tuning large language models for answer set programming. In *KR 2024, Hanoi, Vietnam. November 2-8, 2024*.

Samuel Rhys Cox, Yunlong Wang, and et al. 2021. Directed diversity: Leveraging language embedding distances for collective creativity in crowd ideation. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 393:1–393:35.

Bhavana Dalvi, Peter Jansen, and et al. 2021. Explaining answers with entailment trees. In *EMNLP 2021*, pages 7358–7370.

DeepSeek-AI, Daya Guo, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Martin Gebser and Roland Kaminski et al. 2019. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1):27–82.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. 2012. *Answer Set Solving in Practice*.

Simeng Han, Hailey Schoelkopf, and et al. 2024. FOLIO: natural language reasoning with first-order logic. In *EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22017–22031.

Shibo Hao, Yi Gu, and et al. 2023. Reasoning with language model is planning with world model. In *EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173.

Adam Ishay, Zhun Yang, and Joohyung Lee. 2023. Leveraging large language models to generate answer set programs. In *KR 2023, Rhodes, Greece, September 2-8, 2023*, pages 374–383.

Peter Jackson. 1989. Applications of nonmonotonic logic to diagnosis. *Knowl. Eng. Rev.*, 4(2):97–117.

Aishwarya Kamath, Johan Ferret, and et al. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Mehran Kazemi, Najoung Kim, and et al. 2023a. LAMBADA: backward chaining for automated reasoning in natural language. In *ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6547–6568.

Mehran Kazemi, Quan Yuan, and et al. 2023b. Boardgameqa: A dataset for natural language reasoning with contradictory information. In *NeurIPS 2023*.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. 2024. Are LLMs classical or nonmonotonic reasoners? lessons from generics. In *ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 558–573.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *EMNLP 2023*, pages 10443–10461.

Vladimir Lifschitz. 2008. What is answer set programming? In *AAAI 2008, July 13-17, 2008*, pages 1594–1597. AAAI Press.

Witold Lukaszewicz. 1990. *Non-monotonic reasoning - formalization of commonsense reasoning*. Ellis Horwood.

Theo Olausson, Alex Gu, and et al. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *EMNLP 2023*, pages 5153–5176.

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

Liangming Pan, Alon Albalak, Xinyi Wang, and William 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*, pages 3806–3824.

Mihir Parmar, Nisarg Patel, and et al. 2024. Towards systematic evaluation of logical reasoning ability of large language models. *arXiv preprint arXiv:2404.15522*.

Nisarg Patel, Mohith Kulkarni, and et al. 2024. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. In *EMNLP 2024*, pages 20856–20879.

Raymond Reiter. 1980. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132.

Lin Ren, Guohui Xiao, Guilin Qi, Yishuai Geng, and Haohan Xue. 2025. Can llms solve ASP problems? insights from a benchmarking study (extended version). *CoRR*, abs/2507.19749.

Rachel Rudinger, Vered Shwartz, and et al. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4661–4675.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *ICLR 2023*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *EMNLP-IJCNLP 2019*, pages 4505–4514.

Andrzej Szalas. 2019. Decision-making support using nonmonotonic probabilistic reasoning. In *KES-IDT 2019, Volume 1, Malta, June 17-19, 2019*, volume 142 of *Smart Innovation, Systems and Technologies*, pages 39–51.

Jidong Tian, Yitian Li, and et al. 2021. Diagnosing the first-order logical reasoning ability through logicnli. In *EMNLP 2021, 7-11 November, 2021*, pages 3738–3747.

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

Jason Weston, Antoine Bordes, and et al. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *ICLR 2016, May 2-4, 2016*.

Yeliang Xiu, Zhanhao Xiao, and Yongmei Liu. 2022. Logicnmr: Probing the non-monotonic reasoning ability of pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022, December 7-11, 2022*, pages 3616–3626.

A Appendix

A.1 Dataset Samples and Statistics

Table 3 gives the statistical information of the generated datasets. We generated a data subset for both skeptical and credulous reasoning, so the benchmark contains a total of 6 datasets. Each sample contains three questions, so the number of questions is three times the number of samples. In addition, since the number of extensions in the MultiLogicNMR_OOD dataset is higher than that in the MultiLogicNMR, the number of facts and rules in the samples in MultiLogicNMR_OOD is about twice that in MultiLogicNMR.

Table 3: Statistical information for proposed datasets.

<table>
<tr><th>Dataset</th><th colspan="2">Mode</th><th>#Num.</th><th>#Ques.</th><th>#F.Avg</th><th>#R.Avg</th><th>#Extensions (1:1:1:1:1)</th><th>Label (T:F:M)</th></tr>
<tr><td rowspan="6">MultiLogicNMR</td><td rowspan="3">Skeptical</td><td>Train</td><td>5000</td><td>15000</td><td rowspan="6">12</td><td rowspan="6">10</td><td rowspan="6">[1,2,3,4,5]</td><td rowspan="10">≈1:1:1</td></tr>
<tr><td>Dev</td><td>500</td><td>1500</td></tr>
<tr><td>Test</td><td>500</td><td>1500</td></tr>
<tr><td rowspan="3">Credulous</td><td>Train</td><td>5000</td><td>15000</td></tr>
<tr><td>Dev</td><td>500</td><td>1500</td></tr>
<tr><td>Test</td><td>500</td><td>1500</td></tr>
<tr><td rowspan="2">MultiLogicNMR_OOD</td><td>Skeptical</td><td>Test</td><td>500</td><td>1500</td><td rowspan="2">22</td><td rowspan="2">20</td><td rowspan="2">[6,8,10,12,16]</td></tr>
<tr><td>Credulous</td><td>Test</td><td>500</td><td>1500</td></tr>
<tr><td rowspan="2">MultiLogicNMR_NL</td><td>Skeptical</td><td>Test</td><td>500</td><td>1500</td><td rowspan="2">12</td><td rowspan="2">10</td><td rowspan="2">[1,2,3,4,5]</td></tr>
<tr><td>Credulous</td><td>Test</td><td>500</td><td>1500</td></tr>
</table>

The #Num. represents the number of samples in the generated dataset. The #Ques. represents the number of questions in generated dataset. #F. Avg represents the average number of facts in the dataset. The #R. Avg represents the average number of rules in the dataset, and #Extension represents the number of extensions.

An Example from MultiLogicNMR in Skeptical Reasoning Mode.

Facts:
Sinclair is not confused.
Sinclair is pure.
Sinclair is not scared.
Sinclair is glamorous.
Sinclair does not respect Grover.
Sinclair is fierce.
Sinclair is shy.
Sinclair is not pessimistic.
Sinclair is not unlikely.
Sinclair is not super.
Sinclair is calm.
Sinclair is dramatic.
Sinclair does not sneer Grover.
Sinclair is passionate.
Default Rules:
If someoneA is not confused and not pessimistic then he is not sorry, unless he is not inexpensive or he is not relieved.
If someoneA is pure then he is financial, unless he is actual.
If someoneA is not unlikely and glamorous then he is not weary, unless he is not eager.
If someoneA is not sorry then he is not inexpensive, unless he is not sorry or he is numerous.
If someoneA is passionate and financial then he is numerous, unless he is not inexpensive.
If someoneA does not sneer someoneB and someoneA is calm then he is not nervous, unless he is not envious or he is numerous.
If someoneA is not super and not pessimistic then he is not relieved, unless he is not lonely or he is not sorry.
If someoneA is not scared then he is not lonely, unless he is not known or he is not relieved.
If someoneA does not respect someoneB and someoneA is dramatic then he is shrill, unless he is not inexpensive.
If someoneA is fierce and shy then he is bad tempered, unless he is passionate or he is not inexpensive.

Question 1: Sinclair is not sorry. **Answer is:** Unknown.

Question 2: Sinclair is not financial. **Answer is:** False.

Question 3: Sinclair is not weary. **Answer is:** True.

Figure 5: An Example from MultiLogicNMR in Skeptical Reasoning Mode.

An Example from MultiLogicNMR_NL in Skeptical Reasoning Mode

Facts:
Sinclair is clear-headed and focused.
Sinclair exhibits an essence of purity.
Sinclair is courageous and unafraid.
Sinclair exudes an aura of allure and sophistication.
Sinclair does not hold Grover in esteem. Sinclair displays a bold and intense nature.
Sinclair is reserved and reluctant to engage in social interactions.
Sinclair has an optimistic outlook on life.
Sinclair is open to possibilities.
Sinclair does not possess extraordinary qualities.
Sinclair maintains a tranquil demeanor.
Sinclair possesses a flair for dramatics.
Sinclair does not express disdain towards Grover.
Sinclair exhibits intense enthusiasm and fervor.
Default Rules:
If an individual is clear-headed and holds an optimistic outlook, they are likely not to express regret, except in cases where they lack accessibility or do not feel a sense of relief.
If an individual embodies purity, then they are prosperous, unless they are genuine.
If an individual is open to possibilities and exudes allure, then they are not fatigued, unless they lack eagerness.
If an individual does not express regret, then they do not lack accessibility, unless they experience regret or are abundant.
If an individual is fervent and prosperous, then they are abundant, unless they lack accessibility.
If an individual refrains from expressing disdain towards another and maintains a tranquil demeanor, they are not likely to experience anxiety, except when they feel jealousy or are abundant.
If an individual lacks extraordinary qualities and holds an optimistic outlook, they are likely to feel relieved, unless they feel solitude or experience regret.
If an individual is courageous, they are not likely to feel isolated, except when they are unknown or feel relieved.
If an individual does not hold another in esteem and possesses a flair for dramatics, they exhibit a tendency toward loudness, except when they lack accessibility.
If an individual displays bold intensity and reservations, they are prone to irritability, unless they are fervent or lack accessibility.

Question 1: Sinclair does not express regret. **Answer is:** Unknown.

Question 2: Sinclair is not prosperous. **Answer is:** False.

Question 3: Sinclair is not fatigued. **Answer is:** True.

Figure 6: An Example from MultiLogicNMR_NL in Skeptical Reasoning Mode.

A.2 Prompts for MultiLogicNMR_NL Generation and Evaluation

We use the Remote Clique and Chamfer Distance indicators (Li et al., 2023; Cox et al., 2021) as well as LLM-based scoring to evaluate the diversity of samples. The Remote Clique score is the average mean distance of an instance from other instances. In addition, the Chamfer Distance Score is the average minimum distance of an example from other instances. Higher scores mean higher diversity in the dataset. Figure 10 shows the prompt to score sample diversity based on an LLM. We require a few-shot prompt-based model to score the diversity of samples between 0 and 5. The larger the score, the higher the diversity of the samples considered by the model. As shown in Tables 4 and 5, the diversity of the rewritten MultiLogicNMR_NL dataset is higher than that of the MultiLogicNMR dataset. An example of the MultiLogicNMR_NL dataset is shown in Figure 6.

Table 4: Comparison results between MultiLogicNMR_NL and MuLtiLogicNMR under diversity indicators.

Dataset	Mode	Remote Clique Score	Chamfer Distance Score
MultiLogicNMR	Skeptical	0.207	0.075
	Credulous	0.217	0.094
MultiLogicNMR_NL	Skeptical	0.234	0.101
	Credulous	0.246	0.106

Table 5: Diversity results from large language model evaluations.

Model	MultiLogicNMR		MultiLogicNMR_NL	
	credulous	skeptical	credulous	skeptical
GPT4o-mini	2.0	2.01	2.87	2.82
o3-mini	3.0	2.15	4.19	3.20
DeepSeek-R1-32B	2.74	2.63	3.07	3.08
Gemma3-27B	3.0	3.0	4.45	4.19
Avg	2.69	2.73	3.65	3.32

Few-shot Prompt for the Generating MultiLogicNMR_NL Dataset

Task Description: Given natural language sentences, You need to regenerate new sentences based on the given sentences. We require that different predicates in the sentences be rewritten into different words that are semantically equivalent. That is newly generated sentences are required not to change the semantics of the original sentences.
The input and output format are: The input facts sentences are:. The input rules sentences are:. The input questions sentences are: .
The output facts sentences are: . The output rules sentences are: . The output questions sentences are: .
For example: The input facts sentences are: Hanna is nervous. Hanna is not fuzzy. Hanna is not quiet. Hanna is rainy. Hanna is not inquisitive. Hanna is dizzy. Hanna is courteous.
The input rules sentences are: If someoneA is nervous and gifted then he is boring, unless he is important. If someoneA is courteous and boring then he is oak, unless he is not historical or he is embarrassed. If someoneA is not inquisitive then he is practical, unless he is excited or happy. If someoneA is tense and excited then he is important, unless he is boring or he is not mean.
The input questions sentences are: Hanna is not boring. Hanna is oak. Hanna is not magnificent.
The output facts sentences are: Hanna is experiencing a sense of unease and trepidation. Hanna is not particular or finicky in her preferences. Hanna is a lively and expressive person. Hanna is experiencing a spell of inclement weather. Hanna exhibits a lack of intellectual curiosity and a dearth of inquisitiveness. Hanna is experiencing some lightheadedness and feeling a bit disoriented. Hanna displays a charming and gracious demeanor in her interactions with others.
The output rules sentences are: If an individual possesses a combination of nervousness and exceptional gifts, they may come across as uninteresting, unless their presence holds significant importance. If an individual exhibits both courtesy and a tendency toward dullness, they can be likened to an oak tree, unless they lack any historical significance or feel ashamed in certain situations. If an individual is not inclined towards curiosity, then they are likely to embrace practicality, except in cases where they are filled with excitement or contentment. An individual experiences a combination of tension and excitement, which denotes their perceived significance, unless their demeanor is perceived as dull or unpleasant.
The output questions sentences are: Hanna may across as interesting. Hanna can be likened to an oak tree. Hanna is not magnificent.
Note that you only need to output the modified sentence.

Figure 7: The few-shot prompt for generating the MultiLogicNMR_NL dataset.

Prompt for evaluating semantic equivalence

You are an excellent language expert. You need to match the equivalence between the two given sentences. If the semantics of the two sentences are the same, the two sentences are considered to be consistent, otherwise they are inconsistent. // Note that you only need the output to be consistent or inconsistent.

Figure 8: Prompt for evaluating semantic equivalence

Prompt for Extracting Predicates from Sentences

You are an excellent language expert. You need to extract the corresponding predicates from the formal sentence and the natural language sentence.
The input format is: . The formal logic program sentence is: .The natural language sentence is: .
The output is: The extracted predicate pairs are:.
For example 1: The formal logic program sentence is: **-noisy(skyla)**. The natural language sentence is: **Skyla is not loud.**. The extracted predicate pairs are: **[[noisy, loud]]**.
For example 2: The formal logic program sentence is: **-nervous(X):-calm(X), not efficient(X), not -relieved(X).**. The natural language sentence is: **If an individual is serene, they are generally not anxious unless they are productive or lack consolation.**. The extracted predicate pairs are: **[[nervous, anxious],[calm, serene],[efficient, productive],[relieved, consolation]]**.

Figure 9: Prompt for extracting predicates from sentences.

Prompt for Evaluating Sample Diversity by LLM.

You are an excellent language expert. You need to evaluate the diversity of the given sentences. Diversity captures the variation among the generated data, reflecting differences in text length, topic, or even writing style. The sentence diversity score ranges from 1,2,3,4,5.
If the diversity of the sentences is high, the diversity score is 1.
If the diversity of the sentences is very poor, the diversity score is 0.
The input and output format are: The input sentence is: .
The diversity score of the sentence is: .
For example 1: The input sentence is: Skyla revere Hanley.Skyla is not noisy.Skyla is not bossy.Skyla is not important. Skyla is colorful.Skyla is not grieving.Skyla is calm.Skyla is not obvious.If someoneA is not alert then he is cheerful, unless he is wide eyed. If someoneA is not stupid then he is wide eyed ,unless he is cheerful.If someoneA is not muddy and wide eyed then he is not bitter ,unless he is cheerful.If someoneA revere someoneB then he is modest, unless he is not bitter or he is brainy.
The diversity score of the sentence is: **2**.
For example 2: The input sentence is: Skyla holds Hanley in high regard. Skyla is not loud. Skyla is not domineering. Skyla is not significant. Skyla lacks vigilance. If an individual is serene, they are generally not anxious unless they are productive or lack consolation. An individual who is neither mourning nor lacking vigilance is generally not alluring, unless they are sensible or attractive. If an individual is neither evident nor highly attentive, they are productive, unless they are enigmatic or not anxious.
The diversity score of the sentence is: **4**.
Note that you only need to generate the diversity score for the sentence. Do not output your reasoning or thinking process.

Figure 10: Prompt for evaluating sample diversity by LLM.

A.3 Prompts for MultiLogicNMRer Framework

Prompt for Grounding Module

Task Description: Given a set of facts and a rule, you need to instantiate the rules based on given facts. Instantiation requires the replacement of pronouns in rules with individuals from the fact.

Example 1: Godwin is not sour. Godwin is short. Godwin is scared. Godwin is wild. Godwin is expensive. Godwin is not bad. Godwin is not straightforward. Godwin is anxious. Godwin is not stubborn. Godwin is not zany. Godwin laugh Connor. Godwin esteem Connor. Godwin is not immediate. Godwin is persistent. The rule is: If someoneA laugh someoneB and he is not stubborn then he is old, unless he is not poor.
The output is: If Godwin laugh Connor and Godwin is not stubborn then Godwin is old, unless Godwin is not poor.

The output format is: The output is:“ ”.

Note that you need to output all instantiation rules.

Figure 11: Prompt for the Grounding Module

Prompt for Upper and Lower Bound Initialization Module

Task Description: Given a set of facts and a rule, you need to extract all instantiated facts in the rule.

Example 1: The rule is: If Godwin laugh Connor and Godwin is not stubborn then Godwin is old, unless Godwin is not poor or Godwin is unhappy.
The output is: Godwin laugh Connor. Godwin is not stubborn. Godwin is old. Godwin is not poor. Godwin is unhappy.

The output format is: The output is: “ ”.

Note that you only need to output all instantiated facts in the rule, do not print the contents of the prompt, and don’t output the same facts repeatedly.

Figure 12: Prompt for the Upper and Lower Bound Initialization Module

Rule decomposition prompting in Reduction Module

Task Description: Given a rule, The rule format is: If A then B, unless C. The A is the prerequisite, the B is the conclusion, and the C is the justification. You need to output all prerequisite, conclusions, and justifications in this rule.

Example 1: The rule is: If Brice is emotional then Brice is beige, unless Brice is sufficient.
The output is: prerequisite: "Brice is emotional.", conclusion:"Brice is beige. ", justification: "Brice is sufficient.".

Example 2: The rule is: If Cadman is historical and Cadman is emotional then Cadman is swift, unless Cadman is smart or Cadman is happy.
The output is: prerequisite: "Cadman is historical. Cadman is emotional.", conclusion: "Cadman is swift."; justification: "Cadman is smart. Cadman is happy. ".

The output format is: The output is: prerequisite: " ", conclusion:" ", justification: " ".

Figure 13: Rule decomposition prompting in Reduction Module

Prompt for Reasoning Module

Task Description: Given facts and a rule. You need to reason about the rules based on facts. The rule format is usually: If A then B. The A is the prerequisite, the B is the conclusion. If the prerequisite A is in the facts, you can deduce conclusion B. If the prerequisite A is not in the facts, then you can not deduce the conclusion B, so your output is: None.

Example 1: The input facts are: Godwin is not sour.Godwin is short.Godwin is scared. Godwin is wild. Godwin is expensive. Godwin is not bad. Godwin is not straightforward. Godwin is anxious. Godwin is not sour. Godwin is not zany. Godwin laugh Connor. Godwin esteem Connor. Godwin is immediate. Godwin is persistent. The rules are: If Godwin is not sour and immediate then Godwin is not lovely.
The output is: Godwin is not lovely.

Example 2: The facts are: Juliana is not old. Juliana is not anxious. Juliana is asleep. Juliana is giant. Juliana is not short. Juliana is comfortable. Juliana is not fearless. Juliana is aggressive. Juliana is not hot. Juliana is not southern. Juliana is not technical. Juliana is not educational. Juliana is not octagonal. Juliana is low. Juliana is not poor. The rule is: If someoneA is not short and not low then Juliana is persistent.
The output is: None.

The output format is: The output is:" ".

Note that you only need to output rule conclusions that can be inferred, not facts and reasoning processes.

Figure 14: Prompt for Reasoning Module

Answer Extraction prompting in Skeptical Reasoning Mode

Task Description: Given the extensions and question, and each extension consists of facts. you need to answer the questions according to the given extensions.
If the question can be inferred under all extensions, and the negation of the question cannot be inferred under all extensions, the answer label of the question is: "True".
If the negation of the question can be inferred under all extensions, and the question cannot be inferred under all extensions, the answer label of the question is: "False".
If the question and the negation of the question cannot be deduced under a certain extension, the answer label of the question is: "Unknown".
The input format is: The extension 1 are:"". The extension 2 are:"". The question is:"". The output format is: The answer is:"".
For example: The extension 1 are: "Magnus and Malcolm is spurn. Magnus is difficult. Magnus is arrogant. Magnus is nasty. Magnus is dangerous. Magnus is important. Magnus is vast. Magnus is not handsome.". The extension 2 are: "Magnus is not important. Magnus is vast. Magnus is dramatic. Magnus is not handsome. Magnus is poor. Magnus is sensitive.".
The question is: "Magnus is not important". The answer is: "Unknown".
The question is: "Magnus is vast". The answer is: "True".
The question is: "Magnus is handsome.". The answer is: "False".
Note that you only need to generate the answer label for the question without giving an explanation or justification. Please read all extensions carefully and answer the question.

Figure 15: Answer Extraction prompting in Skeptical Reasoning Mode

Answer Extraction prompting in Credulous Reasoning Mode

Task Description: Given the extensions and question, and each extension consists of facts. you need to answer the questions according to the given extensions.
If the question can be inferred based on a certain facts set, the answer label of the question is "True";
If the negation of the question can be inferred based on a certain facts set, the answer label of the question is "False";
If the question and the negation of the question both cannot be deduced under all facts set, the answer label of the question is "Unknown".
The input format is: The extension 1 are:"". The extension 2 are:"". The question is:"". The output format is: The answer is:"".
For example: The extension 1 are: "Magnus and Malcolm is spurn. Magnus is difficult. Magnus is arrogant. Magnus is nasty. Magnus is dangerous. Magnus is important. Magnus is vast. Magnus is handsome.". The extension 2 are: "Magnus is not important. Magnus is vast. Magnus is dramatic. Magnus is not handsome. Magnus is poor. Magnus is sensitive.".
The question is: "Magnus is happiness.". The answer is: "Unknown".
The question is: "Magnus is important". The answer is: "True". The question is: "Magnus is not dramatic.". The answer is: "False".
Note that you only need to generate the answer label for the question without giving an explanation or justification. Please read all extensions carefully and answer the question.

Figure 16: Answer Extraction prompting in Credulous Reasoning Mode

Prompt for Training the Model under Skeptical Reasoning Mode

Task Description: Given contexts and question, and the context consists of facts and default rules. You need first to find all extensions based on the given context and then to answer the question according to the extensions. An extension is a set of non-contradictory conclusions generated by rule-based reasoning. There may be multi-extensions in the context, and you need to generate all the extensions. Next, the answer to the question is generated based on the generated extension.
If the question can be inferred under all extensions, and the negation of the question cannot be inferred under all extensions, the answer label of the question is: “True”;
If the negation of the question can be inferred under all extensions, and the question cannot be inferred under all extensions, the answer label of the question is: “False”;
If the question and the negation of the question cannot be deduced under a certain extension, the answer label of the question is: “Unknown”.
The input format is: The context are:“”.
The output format is: The answer of the question is:“”.
You must find all extensions and the answer of the question.
Please read the context carefully. Letś think step by step.

Figure 17: Prompt for fine-tuning the Model under Skeptical Reasoning Mode

A.4 The Detailed Description for Model Fine-tuning

We use the LoRA fine-tuning method to fine-tune the open-source LLMs DeepSeek-R1-32B and Gemma3-27B, respectively. The parameters of the fine-tuned model are shown in Table 6. All fine-tuning experiments are completed on a single NVIDIA 4090 GPU based on the unsloth[5] framework.

In the training phase, we use the context consisting of facts and rules, questions, and labels as training samples. However, in the test phase, in order to truly evaluate the multi-extension non-monotonic reasoning abilities of the fine-tuned model, we require the model to generate extensions and questions at the same time. Figure 17 and 18 respectively show the prompt content of the fine-tuned model in the training phase and the test phase under Skeptical Reasoning Mode.

Table 6: Fine-tuning parameters of open-source LLMs.

Parameter	Value
per_device_train_batch_size	4
gradient_accumulation_steps	4
warmup_steps	10
max_steps	200
weight_decay	0.01
optim	Adamw_8bit
seed	3407

[5]https://github.com/unslothai/unsloth

Prompt for Testing the Model under Skeptical Reasonging

Task Description: Given contexts and question, and the context consists of facts and default rules. You need first to generate all extensions based on the given context and then to answer the question according to the extensions. An extension is a set of non-contradictory conclusions generated by rule-based reasoning. There may be multi-extensions in the context, and you need to generate all the extensions. Next, the answer to the question is generated based on the generated extension.
If the question can be inferred under all extensions, and the negation of the question cannot be inferred under all extensions, the answer label of the question is: “True”.
If the negation of the question can be inferred under all extensions, and the question cannot be inferred under all extensions, the answer label of the question is: “False”.
If the question and the negation of the question cannot be deduced under a certain extension, the answer label of the question is: “Unknown”.
The input format is: The context are: “”.
The output format is: The answer of the question is:“”.
The extensions are:“”.
For example: The context are: Connor is not lovely. Connor is poor. Connor is not wonderful.Connor is not former.Connor is wicked. Connor is not comfortable. Connor is not oval. Connor is not cultural.Connor is old. Connor is not hungry. Connor does not honour Godfrey. Connor is awful.Connor is not long. If someoneA is not long then he is not immediate, unless he is not giant or he is sparkling. If someoneA is awful and not alert then he is not creative, unless he is decent.If someoneA is not wonderful then he is civil, unless he is not substantial or he is not zany. If someoneA is not immediate and he does not honour someoneB then he is not giant, unless he is not immediate.If someoneA is wicked then he is decent, unless he is not creative or he is significant. If someoneA is not comfortable and not giant then he is sparkling, unless he is not decent or he is not immediate.If someoneA is old and not lovely then he is not hollow, unless he is not giant or he is not crowded.If someoneA is not former and not hungry then he is not alert, unless he is hard or he is not immediate.If someoneA is not oval then he is cheap, unless he is not energetic or he is not technical.If someoneA is poor and not cultural then he is not crowded, unless he is oval or he is not hollow.
The question is:“Connor is civil.”.
The answer of the question is:“True”.
The extensions are:[[Connor is civil. Connor is decent. Connor is cheap. Connor is not crowded. Connor is not immediate.], [Connor is civil. Connor is decent. Connor is cheap. Connor is not hollow. Connor is not immediate.]].
You must generate all extensions and the answer of the question. Please read the context carefully. Lets̀ think step by step.

Figure 18: Prompt for Testing the Model under Skeptical Reasoning Mode.

A.5 Human Evaluation Guidelines and Results

We selected 100 samples from the MultiLogicNMR dataset for manual annotation in the skeptical reasoning and credulous reasoning mode. Figure 19 shows an example of human evaluation. The following are the instructions for human evaluation:

A.5.1 Human Evaluation Guidelines

You are given a context, which specifies a default theory, a reasoning mode (skeptical or credulous), and three questions. Your goal is to answer each question with a label ("T" for True, "F" for False, or "M" for Unknown).

A default theory consists of facts and default rules. Each fact is a true statement. Each default rule is of the form "If A (the prerequisite), then B (the conclusion), unless C (the justification)". To annotate correctly, you must first understand the concept of an "Extension". An "Extension" represents a complete self-consistent set of conclusions that can be derived from the facts and default rules. Note that a default theory may have more than one extension. The following is an example.

Fact 1*: John is a professor.*

Fact 2*: John is the chair.*

Rule 1*: Professors usually teach.*

Rule 2*: Chairs usually do not teach.*

Rule 3*: If someone teaches, he usually has a teaching assistant.*

Analysis*: The facts trigger both Rule 1 and Rule 2, but their conclusions (teach vs. not teach) are contradictory. This results in two possible extensions:*

Extension 1*: John is a professor, John is the chair, John teaches, John has a teaching assistant.*

Extension 2*: John is a professor, John is the chair, John does not teach.*

There are two reasoning modes:

Mode 1: Skeptical Reasoning

This mode is highly cautious and requires a high degree of certainty. We only believe a conclusion if it is true across all possible extensions.

- Label as "T" (True) – If and only if the question is true in every extension.
- Label as "F" (False) – If and only if the question is false in every extension.
- Label as "M" (Unknown) – In all other cases. This includes when a statement is true in some extensions but not others, or when it is neither true or false in some extension.

(Continuing the above Example) Under the skeptical reasoning mode, for the question "John does not teach", the label is "M" (Unknown) because the statement is true only in Extension 2 and does not hold true across all extensions.

Mode 2: Credulous Reasoning This mode is more open and believes a conclusion as long as it holds in some extension.

- Label as "T" (True) – If and only if the question is true in some extension.
- Label as "F" (False) – If and only if the question is false in some extension.
- Label as "M" (Unknown) – If and only if the question is neither true nor false in any extension.

(Continuing the above Example) In credulous reason mode, the question "Does John have a teaching assistant" is labeled "T" (True) because the statement is true in Extension 1.

Please read the context carefully and answer each question according to the specified reasoning mode. Thank you for your work!

Human	Accuracy (%)		
	Skeptical	Credulous	Avg.
Human1	88.9	95.9	92.4
Human2	91.2	95.9	93.6
Human3	87.8	94.4	91.1
Avg.	89.3	95.4	-

Table 7: Detailed Human Evaluation Results.

A.5.2 Human Evaluation Results

We selected 50 MultiLogicNMR samples for each of the skeptical and credulous reasoning modes and asked three computer science students to solve them. All three of them possess a good knowledge of non-monotonic reasoning and have a solid understanding of the concept of extensions. Table 7 shows the accuracy of their answers.

An Example for Human Evaluaton.

Facts:
Sinclair is not confused.
Sinclair is pure.
Sinclair is not scared.
Sinclair is glamorous.
Sinclair does not respect Grover.
Sinclair is fierce.
Sinclair is shy.
Sinclair is not pessimistic.
Sinclair is not unlikely.
Sinclair is not super.
Sinclair is calm.
Sinclair is dramatic.
Sinclair does not sneer Grover.
Sinclair is passionate.
Default Rules:
If someoneA is not confused and not pessimistic then he is not sorry, unless he is not inexpensive or he is not relieved.
If someoneA is pure then he is financial, unless he is actual.
If someoneA is not unlikely and glamorous then he is not weary, unless he is not eager.
If someoneA is not sorry then he is not inexpensive, unless he is not sorry or he is numerous.
If someoneA is passionate and financial then he is numerous, unless he is not inexpensive.
If someoneA does not sneer someoneB and someoneA is calm then he is not nervous, unless he is not envious or he is numerous.
If someoneA is not super and not pessimistic then he is not relieved, unless he is not lonely or he is not sorry.
If someoneA is not scared then he is not lonely, unless he is not known or he is not relieved.
If someoneA does not respect someoneB and someoneA is dramatic then he is shrill, unless he is not inexpensive.
If someoneA is fierce and shy then he is bad tempered, unless he is passionate or he is not inexpensive.
Reasoning Mode: Skeptical Reasoning

Question 1: Sinclair is not sorry. **Answer is:** ?.

Question 2: Sinclair is not financial. **Answer is:** ?.

Question 3: Sinclair is not weary. **Answer is:** ?.

Figure 19: An Example for Human Evaluation.

A.6 Prompts for the Baseline Prompt-based Methods

Zero-Shot Prompt for Skeptical Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under all reasoning paths based on the context, and the negation of the question cannot be inferred under all reasoning paths based on the context, the answer label of the question is: “True”;
If the negation of the question can be inferred under all reasoning path based on the context, and the question cannot be inferred under all reasoning path based on the context, the answer label of the question is: “False”;
If the question and the negation of the question cannot be deduced under a certain reasoning path based on the context, the answer label of the question is: “Unknown”. You must generate answer labels for the question.
The input format is: Context: “ ”. Question:“ ”.
The output format is: The answer label of the question is:“ ”.
Note that you only need to generate the answer label for the question without giving an explanation or justification. Please read the context carefully and answer the questions.

Figure 20: Zero-Shot Prompt for Skeptical Reasoning

Zero-Shot Prompt for Credulous Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “True”;
If the negation of the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “False”;
If the question and the negation of the question both cannot be deduced under all reasoning path based on the context, the answer label of the question is: “Unknown”. You must generate answer labels for the question.
The input format is: Context: “ ”. Question: “ ”.
The output format is: The answer label of the question is: “ ”.
Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

Figure 21: Zero-Shot Prompt for Credulous Reasoning

Few-Shot Prompt for Skeptical Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled "True", "False" and "Unknown".
If the question can be inferred under all reasoning paths based on the context, and the negation of the question cannot be inferred under all reasoning paths based on the context, the answer label of the question is: "True";
If the negation of the question can be inferred under all reasoning path based on the context, and the question cannot be inferred under all reasoning path based on the context, the answer label of the question is: "False";
If the question and the negation of the question cannot be deduced under a certain reasoning path based on the context, the answer label of the question is: "Unknown". Each context has a question, you must generate answer labels for each question.
The input format is: Context: " ". Question:" ".
The output format is: The answer label of the question is:" ".
Example 1: Context: Basil is not innocent. Basil is not wooden. Basil is discreet. Basil is not petite. Basil is comprehensive. Basil is nutty. Basil is historical. ... If someoneA is historical then he is red, unless he is not lively or he is not big. If someoneA is nutty and steep then he is miniscule, unless he is not weary or he is outstanding. If someoneA is not petite then he is brave, unless he is sticky or he is psychological. If someoneA is not wooden and miniscule then he is psychological, unless he is brave. ...
If the question is: Basil is red. **Then the answer label for the question is:** "True";
If the question is: Basil is miniscule. **Then the answer label for the question is:** "Unknown";
If the question is: Basil is not ashamed. **Then the answer label for the question is:** "False".
Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

Figure 22: Few-Shot Prompt for Skeptical Reasoning

Few-Shot Prompt for Credulous Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled "True", "False" and "Unknown".
If the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: "True"; If the negation of the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: "False"; If the question and the negation of the question both cannot be deduced under all reasoning path based on the context, the answer label of the question is: "Unknown". Each context has three questions, You must generate answer labels for each question.
The input format is: Context: " ". Question:" ".
The output format is: The answer label of question is: " ".
Example 1: Context: Cecil is acceptable. Cecil is uptight. Cecil is not good tempered. Cecil is not severe. Cecil is not messy. Cecil is not self disciplined. Cecil is not logical. Cecil is not right. Cecil is careful.... If someoneA is not logical then he is not visible, unless he is not harsh. If someoneA is not messy and careful then he is not outstanding, unless he is not uptight. If someoneA is uptight and not severe then he is not successful, unless he is similar or he is not good. If someoneA is not visible then he is serious, unless he is not outstanding. If someoneA is not self disciplined then he is not fantastic, unless he is emotional or he is serious. ...
If the question is: Cecil is good. **Then the answer label for the question is:** "False"; **If the question is:** Cecil is not visible. **Then the answer label for the question is:** "Unknown"; **If the question is:** Cecil is similar. **Then the answer label for the question is:** "True".
Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

Figure 23: Few-Shot Prompt for Credulous Reasoning

Zero-Shot AlgCoT Prompt for Skeptical Reasoning

Task Description: Given contexts and question, and the context consists of facts and default rules. You need to first generate all extensions based on the given context, then answer the question according to the extensions.
To generate an extension in the context:
Firstly, to generate the initial upper and lower bounds fact sets of extension based on the instantiated default rules, the lower bound fact set is initialized with the original facts. The upper bound fact set should also include all the facts extracted from the rules.
Then, the upper and lower bounds facts are updated using the conclusions generated by the rules. If the updated lower bound fact set is still a subset of the upper bound fact set, a fact is selected from the upper bound fact set and added to the lower bound set.
Next, the rules are inferred based on the updated upper and lower bound fact sets, and the upper and lower bound fact sets are updated again using the generated conclusions. Specifically, the conclusions generated by the rules under the upper bound fact set should be included in the lower bound fact set. In comparison, the upper bound fact set should only contain the conclusions generated by the reduced rules in the lower bound fact set.
Then, the process is iterated until the upper and lower bounds facts are consistent and an extension is found. There may be multi-extensions in the context, and you need to find all the extensions according to the above steps.
Finally, the answer to the question is generated based on the generated extension.
If the question can be inferred under all extensions, and the negation of the question cannot be inferred under all extensions, the answer label of the question is: "True".
If the negation of the question can be inferred under all extensions, and the question cannot be inferred under all extensions, the answer label of the question is: "False".
If the question and the negation of the question cannot be deduced under a certain extension, the answer label of the question is: "Unknown".
The input format is: Facts:"". Default Rules:"". Question:"".
The output format is: The answer label of the question is: "".
Note that you only need to generate the answer label for the question. Do not output your reasoning or thinking process. Please read the context carefully and answer the questions.
Letś think step by step.

Figure 24: Zero-Shot AlgCoT Prompt for Skeptical Reasoning

Zero-Shot AlgCoT Prompt for Credulous Reasoning

Task Description: Given contexts and question, and the context consists of facts and default rules. You need to first generate all extensions based on given context, then to answer the question according to the extensions.
To generate an extension in the context:
Firstly, to generate the initial upper and lower bounds fact sets of extension based on the instantiated default rules, the lower bound fact set is initialized with the original facts. The upper bound fact set should also include all the facts extracted from the rules.
Then, the upper and lower bounds facts are updated using the conclusions generated by the rules. If the updated lower bound fact set is still a subset of the upper bound fact set, a fact is selected from the upper bound fact set and added to the lower bound set.
Next, the rules are inferred based on the updated upper and lower bound fact sets, and the upper and lower bound fact sets are updated again using the generated conclusions. Specifically, the conclusions generated by the rules under the upper bound fact set should be included in the lower bound fact set. In comparison, the upper bound fact set should only contain the conclusions generated by the reduced rules in the lower bound fact set.
Then, the process is iterated until the upper and lower bounds facts are consistent and an extension is found. There may be multi-extensions in the context, and you need to find all the extensions according to the above steps.
Finally, the answer to the question is generated based on the generated extension.
If the question can be inferred under a certain extension, the answer label of the question is: "True".
If the negation of the question can be inferred under a certain extension, the answer label of the question is: "False".
If the question and the negation of the question both cannot be deduced under all extension, the answer label of the question is: "Unknown".
The input format is: Facts:"", Default Rules:"". Question:"".
The output format is: The answer label of the question is:"".
Note that you only need to generate the answer label for the question. Do not output your reasoning or thinking process. Please read the context carefully and answer the questions.
"Letś think step by step."

Figure 25: Zero-Shot AlgCoT Prompt for Credulous Reasoning

Few-Shot AlgCoT Prompt for Skeptical Reasoning

Task Description: Given contexts and question, and the context consists of facts and default rules. You need to first generate all extensions based on the given context, then answer the question according to the extensions. To generate an extension in the context: **Firstly**, to generate the initial upper and lower bounds fact sets of extension based on the instantiated default rules, the lower bound fact set is initialized with the original facts. The upper bound fact set should also include all the facts extracted from the rules. **Then**, the upper and lower bounds facts are updated using the conclusions generated by the rules. If the updated lower bound fact set is still a subset of the upper bound fact set, a fact is selected from the upper bound fact set and added to the lower bound set. **Next**, the rules are inferred based on the updated upper and lower bound fact sets, and the upper and lower bound fact sets are updated again using the generated conclusions. . . . **Then**, the process is iterated until the upper and lower bounds facts are consistent and an extension is found. There may be multi-extensions in the context, and you need to find all the extensions according to the above steps. **Finally**, the answer to the question is generated based on the generated extension. If the question can be inferred under all extensions, and the negation of the question cannot be inferred under all extensions, the answer label of the question is: "True". If the negation of the question can be inferred under all extensions, and the question cannot be inferred under all extensions, the answer label of the question is: "False". If the question and the negation of the question cannot be deduced under a certain extension, the answer label of the question is: "Unknown".
The input format is: Facts:"". Default Rules:"". Question:"".
The output format is: The answer label of the question is: "".
For example: Context: Facts: Toby is not noisy. Toby is handsome. Default Rules: If someoneA is handsome then he is delicious, unless he is not drab. If someoneA is not noisy then he is not drab, unless he is delicious. The first step is to generate the upper and lower bounds of the extension. The lower bound fact set is "Toby is not noisy. Toby is handsome.", and the upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is delicious. Toby is not drab.". Then, the upper bound facts are updated using the conclusion "Toby is delicious. Toby is not drab." generated by the lower bound fact set based on the default rule. The new upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is delicious. Toby is not drab.". Similarly, since the default rule does not generate new facts when reasoning on the upper bound fact set, the lower bound fact set remains unchanged. At this time, since the lower bound fact set is a subset of the upper bound fact set, a fact is randomly selected from the upper bound fact set and added to the lower bound fact set. The new lower bound fact set is updated to "Toby is not noisy. Toby is handsome. Toby is delicious." The new upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is not drab.". Then the default rule is inferred based on the new upper and lower bound fact sets, and the process is iterated until the upper and lower bounds of the extension are consistent. Finally, the context can generate two extensions: The extension 1 are: "Toby is not noisy. Toby is handsome. Toby is delicious.". The extension 2 is "Toby is not noisy. Toby is handsome. Toby is not drab.". If the Question is: Toby is not noisy. Since all expansions can infer the query "Toby is not noisy.", so the answer label for the question is: True. If the question is: Toby is not drab. Although extension 2 can deduce the query "Toby is not drab", extension 1 cannot deduce this query, so the answer label for the question is: Unknown; If the question is: Toby is not handsome. Since all three extensions can lead to the query "Toby is handsome", so the answer label for the question is: False.
Note that you only need to generate the answer label for the question. Do not output your reasoning or thinking process. Please read the context carefully and answer the questions.
Letś think step by step.

Figure 26: Few-Shot AlgCoT Prompt for Skeptical Reasoning

Few-Shot AlgCoT Prompt for Credulous Reasoning

Task Description: Given contexts and question, and the context consists of facts and default rules. You need to first generate all extensions based on given context, then to answer the question according to the extensions. To generate an extension in the context: **Firstly**, to generate the initial upper and lower bounds fact sets of extension based on the instantiated default rules, the lower bound fact set is initialized with the original facts. The upper bound fact set should also include all the facts extracted from the rules. **Then**, the upper and lower bounds facts are updated using the conclusions generated by the rules. If the updated lower bound fact set is still a subset of the upper bound fact set, a fact is selected from the upper bound fact set and added to the lower bound set. **Next**, the rules are inferred based on the updated upper and lower bound fact sets, and the upper and lower bound fact sets are updated again using the generated conclusions. . . . **Then**, the process is iterated until the upper and lower bounds facts are consistent and an extension is found. There may be multi-extensions in the context, and you need to find all the extensions according to the above steps. **Finally**, the answer to the question is generated based on the generated extension. If the question can be inferred under a certain extension, the answer label of the question is: "True". If the negation of the question can be inferred under a certain extension, the answer label of the question is: "False".
If the question and the negation of the question both cannot be deduced under all extension, the answer label of the question is: "Unknown".
The input format is: Facts:"", Default Rules:"". Question:"".
The output format is: The answer label of the question is:"". For example: Facts: Toby is not noisy. Toby is handsome.. . . Default Rules: If someoneA is handsome then he is delicious, unless he is not drab. If someoneA is not noisy then he is not drab, unless he is delicious. The first step is to generate the upper and lower bounds of the extension. The lower bound fact set is "Toby is not noisy. Toby is handsome.", and the upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is delicious. Toby is not drab.". Then, the upper bound facts are updated using the conclusion "Toby is delicious. Toby is not drab." generated by the lower bound fact set based on the default rule. The new upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is delicious. Toby is not drab.". Similarly, since the default rule does not generate new facts when reasoning on the upper bound fact set, the lower bound fact set remains unchanged. At this time, since the lower bound fact set is a subset of the upper bound fact set, a fact is randomly selected from the upper bound fact set and added to the lower bound fact set. The new lower bound fact set is updated to "Toby is not noisy. Toby is handsome. Toby is delicious." The new upper bound fact set is "Toby is not noisy. Toby is handsome. Toby is not drab.". Then the default rule is inferred based on the new upper and lower bound fact sets, and the process is iterated until the upper and lower bounds of the extension are consistent. Finally, the context can generate two extensions: The extension 1 are: "Toby is not noisy. Toby is handsome. Toby is delicious."
The extension 2 are "Toby is not noisy. Toby is handsome. Toby is not drab.". If the question is: Toby is not drab. Since extension 2 can lead to the query "Toby is not drab.", so the answer label for the question is: True. If the question is: Toby is intelligent. Since extensions 1 and 2 cannot deduce the query and the negation of the query, the answer label for the question is: Unknown. If the question is: Toby is not handsome. Since the negation of query fact "Toby is handsome." can be derived in extensions, the answer label for the question is: False.
Note that you only need to generate the answer label for the question. Do not output your reasoning or thinking process. Please read the context carefully and answer the questions.
"Letś think step by step."

Figure 27: Few-Shot AlgCoT Prompt for Credulous Reasoning

A.7 Label Analysis and Error Examples

Figure 28 and 29 show the distribution of answers generated by different methods in skeptical reasoning and credulous reasoning, respectively. Specifically, in skeptical reasoning mode, questions with Unknown answers are still very challenging to prompting and fine-tuning methods, and our MultiLogicNMRer framework has significantly improved the performance.

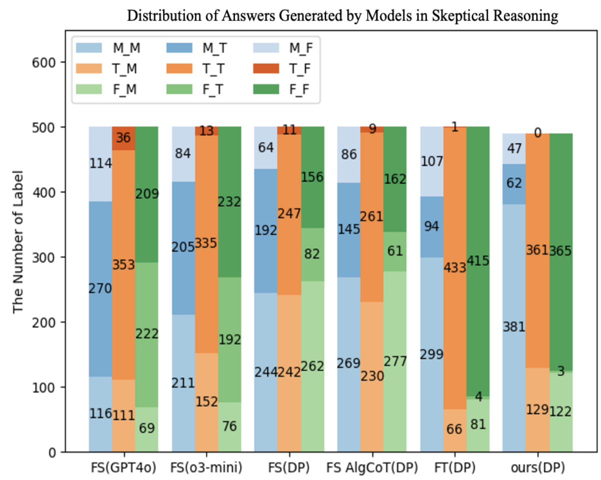

Figure 28: The distribution of answers generated by models in skeptical reasoning. FS and FT represent few-shot prompting and fine-tuning, respectively. DP represents the DeepSeek-R1-32B model. The legend at the top explains the meaning of the different colors; for example, M_M represents cases where both the correct and predicted labels are M.

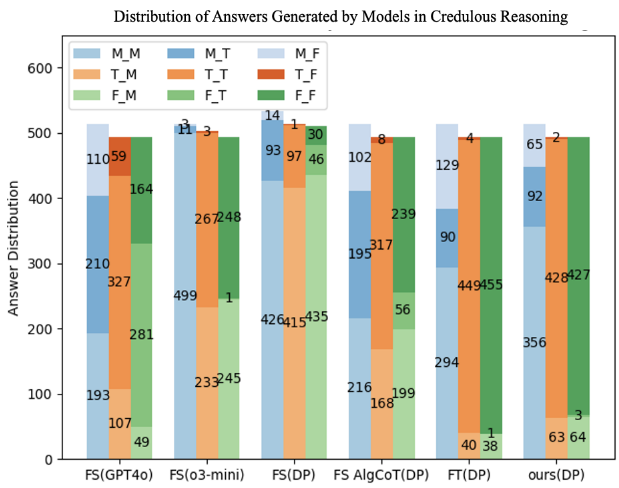

Figure 29: The distribution of answers generated by models in credulous reasoning. FS and FT represent few-shot prompting and fine-tuning, respectively. DP represents the DeepSeek-R1-32B model. The legend at the top explains the meaning of the different colors; for example, M_M represents cases where both the correct and predicted labels are M.

Table 8: Correct and incorrect outputs generated by different modules in MultiLogicNMRer on MultiLogicNMR_NL samples. The red text indicates the incorrect parts of the outputs. The ✓ and ✗ indicate correct and incorrect outputs, respectively.

<table>
<tr><th>Dataset</th><th>Module</th><th>Input</th><th>Output</th><th>Label</th></tr>
<tr><td rowspan="6">MultiLogic NMR_NL</td><td rowspan="2">Grounding</td><td>If an individual possesses an optimistic outlook and feels afraid, then they are familiar, unless they are insensitive or not domineering.</td><td>If Toby does not have a negative outlook and Toby feels afraid, then Toby is familiar, unless Toby lacks intelligence or Toby is not domineering.</td><td>✓</td></tr>
<tr><td>If an individual is courageous, they are not likely to feel isolated, except when they are unknown or feel relieved.</td><td>If Sinclair is courageous and Sinclair is not unknown and Sinclair does not feel relieved, then Sinclair is not isolated.</td><td>✗</td></tr>
<tr><td rowspan="2">Reduction</td><td>Sinclair does not hold Grover in esteem.
Sinclair is reserved and reluctant to engage in social interactions.
If Sinclair embodies purity, then Sinclair is prosperous, unless Sinclair is genuine.</td><td>If Sinclair embodies purity then Sinclair is prosperous.</td><td>✓</td></tr>
<tr><td>Beatrice is likened to a woolen material. Beatrice is youthful.
Beatrice have a woolen disposition.
If Beatrice lacks enthusiasm, Beatrice maintain cleanliness, unless Beatrice have a woolen disposition.</td><td>If Beatrice lacks enthusiasm and Beatrice maintain cleanliness then Beatrice is swift.</td><td>✗</td></tr>
<tr><td rowspan="2">Reasoning</td><td>Sinclair is not poor.
Sinclair is open to possibilities.
Sinclair exudes an aura of allure and sophistication.
If Sinclair is open to possibilities and Sinclair exudes an aura of allure and sophistication then Sinclair is not fatigued.</td><td>Sinclair is not fatigued.</td><td>✓</td></tr>
<tr><td>Sinclair does not hold Grover in esteem.
Sinclair exhibits an essence of purity.
Sinclair does not express disdain towards Grover.
If Sinclair embodies purity then Sinclair is prosperous.</td><td>Sinclair is likely to feel relieved.</td><td>✗</td></tr>
</table>