AS-NMRer: Improving Autoformalization for Non-monotonic Reasoning via Abstraction, Search, and Fine-tuning

Anonymous ACL submission

Abstract

LLMs often struggle with complex logical reasoning, particularly in non-monotonic reasoning (NMR), where conclusions may be retracted in light of new evidence. While autoformalization-based neuro-symbolic frameworks offer a promising solution by translating text into logic programs, they often fail to handle the ambiguity and noise present in realistic linguistic contexts. To address these challenges, we propose AS-NMRer, a framework that enhances autoformalization through abstraction, search, and fine-tuning. First, we introduce an abstraction module to extract atomic facts and rules from noisy contexts. Next, a step-wise Best-of-N search algorithm incrementally maps these facts and rules into logic programs with dual verification that ensures both syntactic correctness and semantic fidelity. Finally, we design an expert iteration loop that leverages solver-verified examples to fine-tune the model, enabling iterative self-improvement. Extensive experiments on four NMR benchmarks show that AS-NMRer significantly outperforms competitive baselines. AS-NMRer with gemma3-27B improves the F1 score by 7% over the prompt-based DeepSeek-V3.2-671B on the challenging LogicBench.

1 Introduction

Logical reasoning over natural language (NL) leveraging LLMs has attracted significant attention. To benchmark this capability, diverse datasets have been established, ranging from synthetic rule-based corpora like RuleTaker (Clark et al., 2020) to more complex benchmarks such as FOLIO (Han et al., 2024) and ProverQA (Qi et al., 2025). Prior works have explored prompting strategies (Wei et al., 2022; Kojima et al., 2022), fine-tuning (Lewkowycz et al., 2022), and search-based neuro-symbolic frameworks (Hao et al., 2023).

Recently, autoformalization-based neuro-symbolic approaches integrate LLM-based translation from NLs to formal languages with the rigorous reasoning of symbolic solvers. Representative frameworks include Logic-LM (Pan et al., 2023) which introduces self-refinement to use the symbolic solver's error messages to refine the formalization, and LINC (Olausson et al., 2023), which uses majority voting to determine the result across multiple formulations. Recently, Fang et al. (2025) proposed LLM-ASPIC$^+$, which integrates neural grounding with formal argumentation to resolve contradictions via argumentation theory. Additionally, Hu et al. (2025) introduced a thought-guided retrieval-augmented generation framework to enhance the precision of autoformalization and enable self-refinement.

Raw Context: In a room filled with various objects, two heavy blocks, Block A and Block B, stand out. Normally, heavy blocks like these are placed on the table, but surprisingly, Block A is not found on the table. On the other hand, Block B grabs attention with its vibrant red color.

Question: Does the context imply that A is on the table?

ASP Context: standout(blockA). standout(blockB). placeOn(blockB, table). color(blockB, red).
ASP Question: placedOn(blockA, table).

Predicted Answer: Unknown ✗

Figure 1: A LogicBench example where DeepSeek-R1-32B is distracted by irrelevant noise (red text). The model formalizes the color details but misses the critical fact and rule, leading to a logical reasoning failure.

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 (Pertierra et al., 2017). NMR refers to the fact that conclusions may be invalidated by new information. Recently, the evaluation of NMR in LLMs has garnered significant attention. Various NMR datasets have been proposed, and they can be broadly put into three cat-

egories. The first is built on existing datasets, including δ-NLI (Rudinger et al., 2020), and datasets built upon large-scale generics corpora (Calderón et al., 2025) as introduced by Leidinger et al. (2024) and Kirkpatrick and Sterken (2025). The second category employs template-based synthesis, exemplified by LogicNMR (Xiu et al., 2022), and ASP-Bench (Ren et al., 2025). Finally, to balance logical rigor with realistic linguistic complexity, a third category combines template synthesis with LLM-based rewriting. This includes LogicBench (Parmar et al., 2024) and Multi-LogiEval (Patel et al., 2024). Results on these benchmarks consistently show LLMs' deficiencies in NMR. To address the issue, Xiu and Liu (2025) proposed MultiLogicNMRer, a search-based neuro-symbolic framework for multiple-extension NMR tasks that is designed based on symbolic solving strategies.

This paper explores the autoformalization-based neural-symbolic approach for NMR. Most existing work on logic reasoning adopts a monolithic formalization strategy, attempting to convert the entire context into a logic program in a single pass. This approach faces two primary challenges. First, it diverges from human cognitive processes, where humans typically perform abstraction first: filtering out irrelevant information and simplifying relevant details. Without abstraction, the inherent noise in natural language makes direct formalization extremely challenging. As illustrated in Figure 1, the raw context contains information irrelevant to answering the question (marked in red). DeepSeek-R1-32B is distracted by such salient but irrelevant details, explicitly formalizing the "vibrant red color" into "color(blockB, red).", yet it fails to capture the critical fact "blockA is heavy" and rule "heavy blocks are usually on the table", leading to an incorrect answer. This demonstrates that linguistic noise misguides the model's attention, causing it to prioritize superficial details over essential logical structures. Secondly, this monolithic paradigm lacks the granularity to verify intermediate steps, meaning that a single misinterpretation of a noisy sentence can cascade into a completely invalid logic program without any mechanism for error localization.

To address these challenges, we propose AS-NMRer, a framework to improve autoformalization for NMR through abstraction, search, and fine-tuning. First, we employ a novel LLM-based abstraction mechanism to leverage questions to distill raw contexts into concise facts and rules. By filtering out redundant information, the abstraction module significantly reduces the input complexity and mitigates the interference of irrelevant information on the autoformalization process. Second, a step-wise Best-of-N (BoN) search algorithm (Zhu et al., 2025) is utilized to incrementally formalize these abstracted sentences into ASP (Answer Set Programming) (Gelfond and Lifschitz, 1988), a declarative programming paradigm. The best candidate is selected by a dual-verification mechanism: a symbolic solver-based syntax verifier ensures syntactic correctness, while an LLM-based semantic verifier rates semantic fidelity to the original context. Third, to iteratively improve the model's abilities of abstraction and formalization, we design a self-improvement loop to finetune the model on high-quality samples generated from training set.

Our contributions are as follows: First, we propose AS-NMRer, a framework utilizing an abstraction module to distill core facts and rules. Second, we design a verification-guided step-wise search strategy to ensure fine-grained formal sentence syntactic and semantic correctness. Third, we design an expert iteration mechanism to enable model self-evolution through solver-verified samples. Finally, extensive experiments show that AS-NMRer consistently outperforms the strongest baselines, notably enabling Gemma3-27B to improve the F1 score by 7% over prompt-based DeepSeek-V3.2-671B on LogicBench.

2 Related Work

Autoformalization in Mathematical Theorem Proving. Progress in Mathematical Theorem Proving has impacted logical reasoning research (Jiang et al., 2023; Li et al., 2024; Lu et al., 2025). Current approaches in Mathematical Theorem Proving integrate LLMs with rigorous formal environments (e.g., Lean), providing deterministic feedback to guide the reasoning process (Wu et al., 2022; Zhang et al., 2024). For instance, Trinh et al. (2024) proposed the neuro-symbolic system AlphaGeometry, which achieves near-human performance on international mathematical olympiad tasks. In addition, test-time search has proven critical for enhancing performance. Xin et al. (2025) demonstrated that a scalable best-first tree search can achieve State-of-the-Art results on MiniF2F by effectively filtering reasoning paths. Moreover, to address data scarcity, Lin et al. (2025a) trained a model on large-scale autoformalized proofs. Their

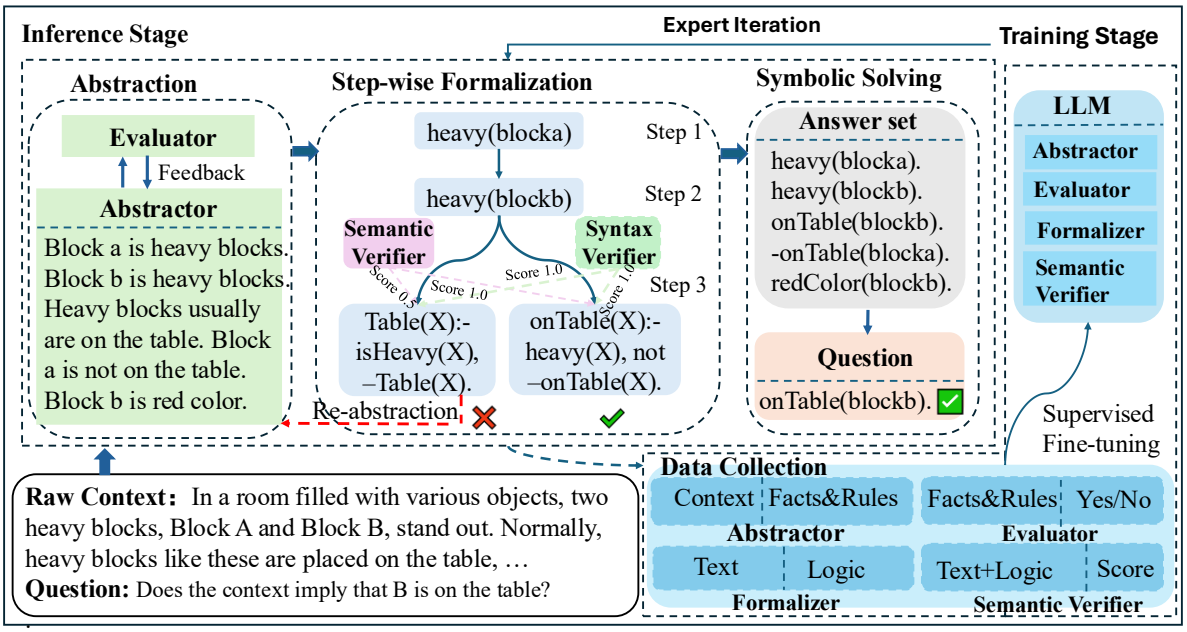


Figure 2: Overview of the AS-NMRer framework.

subsequent work (Lin et al., 2025b) introduced verifier-guided self-correction, enabling the model to master complex theorems through iterative refinement. Drawing inspiration from these advancements, our work AS-NMRer adapts the paradigms of verification-guided search and iterative refinement to the domain of non-monotonic reasoning. However, to minimize the interference of noise in the original context, AS-NMRer introduces a novel abstraction mechanism to distill core facts and rules from raw context.

3 The AS-NMRer Framework

We present AS-NMRer, a neuro-symbolic framework that integrates abstraction, step-wise formalization, and symbolic solving, while utilizing verified samples for continuous self-evolution.

3.1 Preliminaries

Default logic (Reiter, 1980) and ASP (Gelfond and Lifschitz, 1988) serve as the theoretical foundation of our framework. A default theory $\Gamma = (W, D)$ consists of facts W and default rules D, where a default rule $\frac{\alpha:\beta_1,\ldots,\beta_k}{\gamma}$ permits inferring γ if the prerequisite α holds and $\beta_1, \ldots, \beta_k$ are consistent with the logical context. Baral and Gelfond (1994) showed that a literal-based default theory can be conveniently transformed to an ASP program, where such a default rule is translated into the ASP rule $\gamma \leftarrow \alpha, \text{not } \neg\beta_1, \ldots, \text{not } \neg\beta_k$, where *not* denotes negation-as-failure in logic programming (Gelfond and Lifschitz, 1991).

3.2 Problem Formulation and Overview

We formulate the NMR task as an autoformalization problem. Formally, given a raw natural language context C_{raw} and a question q, the objective is to translate them into an ASP logic program Π and a query literal l_q. The answer y is then derived from the answer sets produced by executing Π with a symbolic solver, determining the status of l_q as True, False, or Unknown. As illustrated in Figure 2, the proposed framework, AS-NMRer, consists of three distinct stages: (1) *Abstraction*, which filters irrelevant distractors to distill core facts and rules; (2) *Step-wise Formalization*, which generates Π by incrementally generating formal sentences through a verification-guided Best-of-N search; and (3) *Symbolic Solving*, which executes Π to derive the answer based on the computed answer sets. Finally, the model iteratively evolves by collecting examples validated during the inference phase. The complete set of prompts employed across all these modules is detailed in Appendix A.

3.3 Abstraction Module

To reduce linguistic noise, the abstraction module distills raw contexts and questions into atomic facts and rules. Conditioned on the question q, this module uses an LLM-based abstractor to jointly map the raw context C_{raw} and q into a set of atomic sen-

tences $S_{abs} = \{s_1, s_2, \ldots, s_m\}$ and a refined query q_{abs} in a unified pass, where m represents the total number of extracted sentences. To guarantee reliability, an auxiliary evaluator assesses S_{abs} against two criteria: *soundness*, which ensures fidelity to the source, and *completeness*, which verifies sufficiency for reasoning. If the assessment fails, a feedback loop triggers the abstractor to regenerate S_{abs}, allowing for a maximum of three attempts.

3.4 Step-wise Formalization

To improve autoformalization reliability, we employ a step-wise Best-of-N search strategy that incrementally maps abstracted sentences and the question into ASP formulas. For each sentence $s_i \in S_{abs}$, the formalizer first leverages the LLM to generate a set of candidate default logic formulas $\{d_i^1, d_i^2, ..., d_i^N\}$, where N denotes the search size. Subsequently, these candidates are automatically converted into executable ASP rules $\{r_i^1, r_i^2, ..., r_i^N\}$. These candidates are then validated via a dual-verification mechanism. A symbolic solver acts as a syntax verifier, assigning a binary syntax score of 1 to candidates that execute without error and 0 to those that trigger solver failures. Concurrently, an LLM-based semantic verifier checks the consistency between the ASP formula and the abstracted sentence, assigning a score ranging from 0 to 1; candidates with a semantic score at or below 0.1 are discarded. The optimal candidate is then determined by calculating the equally weighted average of the syntax and semantic scores and selecting the highest-ranking formula. If no candidate satisfies the verification thresholds, the framework activates a Re-abstraction mechanism, prompting the abstraction module to refine the abstracted sentence s_i,allowing for a maximum of three sampling attempts to resolve the formalization failure.

3.5 Symbolic Solving

The symbolic solving module leverages a deterministic ASP solver (Clingo) to execute the verified logic program Π and derive the label of the question l_q from generated answer sets. As illustrated in Figure 2, the solver derives an answer set containing facts such as "heavy(blockb)" and "onTable(blockb)". Since the question "onTable(blockb)" is explicitly present in the answer set, the label of the question is True. In addition, to address scenarios where the logic program yields multiple answer sets, there are two reasoning modes: skeptical and credulous reasoning, depending on whether to consider facts in all answer sets or any one answer set. Thus, the symbolic solving ensures the logical reliability and correctness of the reasoning process.

3.6 Self-Improvement via Expert Iteration

To enhance autoformalization, we design an expert iteration mechanism using supervised fine-tuning (SFT) on high-quality examples accumulated during the inference stage. Within our framework, a single LLM serves as the unified backbone, concurrently performing the roles of the abstractor, evaluator, formalizer, and semantic verifier. To ensure the correctness of the training samples, we retain only the instances where the framework successfully derives the correct final answer. As illustrated in the data collection component of Figure 2, this mechanism constructs distinct training samples for each module; for instance, a training instance for the formalizer is constructed by pairing an abstracted sentence "Block b is heavy blocks" with its corresponding logical program "heavy(blockb)". By fine-tuning the unified backbone on these validated corpora, the model progressively internalizes the syntactic and semantic rules required for NMR.

4 Experiments

We evaluate AS-NMRer on four diverse benchmarks. Beyond performance comparisons, we analyze the impact of search size and the expert iteration strategy on reasoning performance. We then perform ablation studies to verify the specific contribution of each component. Finally, we present qualitative case studies to investigate specific error patterns in abstraction and formalization.

4.1 Experimental Settings

To evaluate the NMR capabilities of the AS-NMRer across realistic contexts and complex logical structures, we employ four diverse benchmarks. We use LogicBench (BQA task) (Parmar et al., 2024) and Multi-LogiEval (Patel et al., 2024) to assess the model's robustness against realistic linguistic structures, the latter specifically focusing on multi-step reasoning chains ranging from depth 1 to 5. Furthermore, to isolate pure NMR from commonsense priors, we use two synthetic benchmarks: Logic-NMR (Xiu et al., 2022) to probe NMR capability within dynamic knowledge bases, and MultiLogic-NMR (Xiu and Liu, 2025) to assess reasoning in scenarios with multiple answer sets.

Table 1: Main results of AS-NMRer compared with baselines across multiple datasets (F1 score %). **Bold** indicates the best results, and underlined. Results for MultiLogicNMRer are taken from the original paper and the symbol '-' indicates that the result is not reported.

Base Model	Method	LogicBench	Multi-LogiEval	LogicNMR	MultiLogicNMR	
					Credulous	Skeptical
o3-mini	Zero-Shot	85.2	55.6	27.8	39.7	36.8
	Few-Shot	**86.4**	58.3	33.9	67.2	71.7
	LLM2ASP	81.6	**72.9**	**58.0**	**76.9**	**76.9**
GPT-4o-mini	Zero-Shot	78.8	38.8	32.7	45.9	38.1
	Few-Shot	**80.9**	37.1	28.8	55.7	40.4
	MultiLogicNMRer	-	**70.9**	74.3	82.8	74.9
	LLM2ASP	57.2	50.3	52.8	59.6	63.8
	AS-NMRer	76.5	65.5	**95.8**	**88.4**	**88.6**
DeepSeek-V3.2 -671B	Zero-Shot	73.5	51.7	37.3	40.4	39.2
	Few-Shot	77.2	55.1	47.6	49.8	56.3
	LLM2ASP	70.4	**73.1**	53.6	70.8	66.8
	AS-NMRer	**79.1**	64.1	**95.7**	**92.1**	**91.3**
DeepSeek-R1 -32B	Zero-Shot	62.1	36.2	36.4	47.6	51.6
	Few-Shot	76.2	57.3	42.9	39.8	49.2
	Fine-Tuned	80.1	64.5	**94.0**	**79.8**	76.1
	LLM2ASP	68.7	60.8	47.3	52.1	52.1
	AS-NMRer	**85.1**	**72.8**	92.8	79.2	**78.6**
Gemma3-27B	Zero-Shot	60.9	41.0	38.2	61.1	41.1
	Few-Shot	75.2	47.1	42.2	63.7	51.5
	Fine-Tuned	80.2	61.7	49.6	**81.8**	63.5
	LLM2ASP	58.8	63.5	54.5	46.2	54.9
	AS-NMRer	**87.8**	**72.2**	**93.6**	76.3	**78.5**

To verify the effectiveness of AS-NMRer, we conduct a comprehensive evaluation against a diverse suite of backbone models and competitive baselines. Specifically, we select representative closed-source LLMs, including GPT-4o-mini (Hurst et al., 2024) and o3-mini (Jaech et al., 2024), alongside leading open-source models such as DeepSeek-R1-32B (DeepSeek-AI, 2025) and Gemma3-27B (Team, 2025). For comparison, we employ standard zero-shot and few-shot prompting strategies, as well as supervised fine-tuning (SFT). Furthermore, we benchmark against two neuro-symbolic approaches, including MultiLogicNMRer (Xiu and Liu, 2025), a search-based framework for multi-extension NMR, and LLM2ASP(Ishay et al., 2023), a direct translation baseline which we implemented by excluding constraint generation. Implementation details are provided in Appendix B. Following prior research, we report the F1-score as the primary metric to evaluate reasoning accuracy.[1]

[1]The core data and code are available at https://anonymous.4open.science/r/AS-NMRer.

4.2 Main Results

Table 1 summarizes the comparative performance of AS-NMRer against baseline methods across four diverse benchmarks. These results validate the effectiveness and robustness of the proposed AS-NMRer. Due to space constraints, we report the averaged F1 scores in the main text, while detailed results across specific reasoning patterns are provided in Appendix C.

Overall, AS-NMRer demonstrates superior performance across the majority of experimental settings. On LogicBench, AS-NMRer with Gemma3-27B achieves an average F1 score of 87.8%, significantly outperforming both the fine-tuned baseline at 80.2% and the few-shot prompting approach at 75.2%. Notably, AS-NMRer consistently surpasses the LLM2ASP baseline across most datasets. This performance gap highlights the limitations of directly translating natural language into ASP, validating the effectiveness of our stepwise formalization strategy. Furthermore, We observe a notable difference when comparing

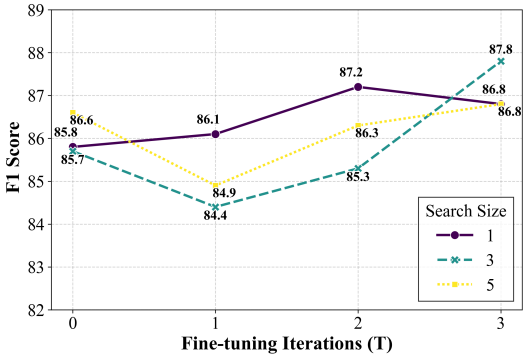
Table 2: Results of the AS-NMRer on the MultiLogicNMR at different search sizes, where E represents the number of answer sets. Avg. denotes the average F1 score across different numbers of answer sets.

Dataset	Model	Search Size	Mode	F1 (%)					
				E=1	E=2	E=3	E=4	E=5	Avg.
MultiLogic -NMR	DeepSeek -R1- 32B	1	credulous	52.7	65.5	64.6	61.1	63.9	61.8
		3		76.6	**75.4**	70.1	**88.5**	63.6	75.0
		5		**82.2**	73.6	**75.2**	86.7	**77.3**	**79.2**
		1	skeptical	77.2	68.8	63.5	79.8	78.8	75.7
		3		**86.0**	70.0	**81.1**	**84.6**	65.6	78.5
		5		78.9	**79.4**	78.5	81.0	**75.6**	**78.9**

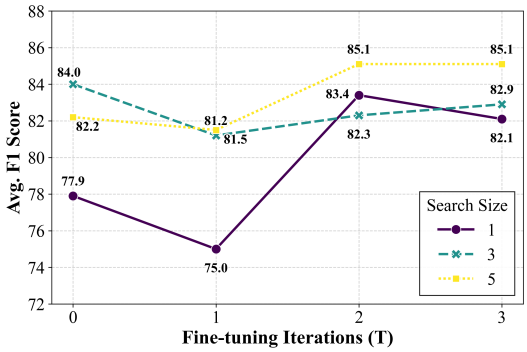
our approach with MultiLogicNMRer. On MultiLogiEval, MultiLogicNMRer with GPT-4o-mini achieves a score of 70.9%, slightly outperforming AS-NMRer, which attains 65.5%. This suggests that the search-based MultiLogicNMRer exhibits greater robustness in handling complex linguistic patterns, whereas autoformalization-based approaches still encounter inherent challenges. However, on synthetic datasets, AS-NMRer with GPT-4o-mini demonstrates a substantial advantage, notably achieving 95.8% on LogicNMR compared to 74.3% for MultiLogicNMRer. Additionally, on MultiLogicNMR, AS-NMRer with Gemma3-27B achieves an F1 score of 78.5% in skeptical reasoning, marking a significant improvement over the zero-shot baseline of 41.1% and the much larger DeepSeek-V3.2-671B of 39.2%.

4.3 Analysis

Impact of Search Size. We first analyze the impact of the search size N in the Best-of-N strategy on reasoning performance. Table 2 presents the performance on the MultiLogicNMR dataset across varying search sizes. As observed, increasing the search width generally yields performance gains, though these gains exhibit diminishing returns as N increases. Notably, for DeepSeek-R1-32B in the credulous setting, expanding N from 1 to 3 results in a substantial improvement, boosting the F1 score from 61.8% to 79.2%. A similar trend is observed on the LogicBench dataset, as detailed in Table 3. Specifically, AS-NMRer with Gemma3-27B improves from 85.8% at $N = 1$ to 86.6% at $N = 5$. However, the gains from $N = 3$ to $N = 5$ are marginal, indicating that our dual-verification mechanism is highly efficient at identifying correct formalizations early in the search process. This suggests that a moderate search width (e.g., $N = 3$) offers an optimal trade-off between computational overhead and reasoning accuracy.



(a) Results of AS-NMRer (Gemma3-27B)



(b) Results of AS-NMRer (DeepSeek-R1-32B)

Figure 3: Results of AS-NMRer over fine-tuning iterations with different search sizes on LogicBench Dataset.

Efficacy of Expert Iteration. To evaluate the expert iteration strategy, we track the performance of AS-NMRer on LogicBench from the base stage $T = 0$ through three subsequent fine-tuning iterations. For each iteration, we construct the training set using approximately 2,300 samples where the framework correctly derives the answer. As shown in Figure 3, AS-NMRer with Gemma3-27B exhibits a steady improvement, confirming that learning from verified examples enhances reasoning capabilities. Notably, the benefits of larger search sizes diminish as the fine-tuning progresses. While

Table 3: Results of AS-NMRer on LogicBench at different search sizes. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Model	Search Size	F1 (%)								
			DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
LogicBench	DeepSeek -R1- 32B	1	92.1	92.1	81.5	53.7	72.7	75.9	77.9	77.5	77.9
		3	92.1	**94.7**	**94.7**	**62.9**	78.9	**86.6**	**81.5**	**84.1**	**84.0**
		5	**94.7**	89.4	94.7	56.2	**81.8**	78.4	80.9	81.3	82.2
	Gemma 3-27B	1	94.7	**92.1**	94.7	64.4	84.8	81.3	84.0	90.7	85.8
		3	94.7	89.4	94.7	**73.6**	81.8	83.8	78.7	92.1	85.7
		5	**94.7**	83.8	**97.4**	69.6	**84.8**	**86.6**	**84.0**	**92.1**	**86.6**

the initial model ($T = 0$) benefits from search (rising from 85.8% at $N = 1$ to 86.6% at $N = 5$), the final model ($T = 3$) converges, achieving 86.8% with both $N = 1$ and $N = 5$. This indicates that the model internalizes formal language constraints, transforming the cost of test-time search into intrinsic proficiency. In contrast, DeepSeek-R1-32B experiences a transient dip at $T = 1$ (e.g., $N = 1$ drops from 77.9% to 75.0%). This fluctuation is likely due to the suboptimal quality of initial training samples. However, the model rapidly recovers in subsequent iterations, eventually surpassing the baseline to achieve a peak F1 score of 85.1%.

Ablation Analysis. To verify the contribution of each component, we conducted an ablation study on LogicBench, as shown in Table 4. First, the abstraction mechanism is essential. Removing this module (w/o Abstraction) causes the most significant drop in performance, falling to 49.2%. This confirms that filtering linguistic noise is essential for the formalizer to map raw narratives into valid logical rules. Second, regarding the search strategy, the performance of the monolithic variant (w/o Step-wise BoN) drops from 84.0% to 72.8% on DeepSeek-R1-32B. This indicates that single-pass translation often misses critical details, whereas our step-wise decomposition ensures granular correctness. Third, the re-abstraction mechanism acts as a critical refinement loop. Removing re-abstraction (w/o Re-abstraction) leads to a moderate decline to 81.8%, suggesting that dynamic refinement based on feedback enhances robustness against extraction errors. Finally, the dual-verification protocol ensures program validity. Disabling the syntax verifier (w/o Syn. Verifier) yields 65.9%, as the solver cannot execute malformed ASP sentences. Similarly, removing the semantic verifier (w/o Sem. Verifier) yields 66.4%, indicating that syntactically correct programs may still lack semantic fidelity.

4.4 Case Study

Analysis of Abstraction Failures. The efficacy of the abstraction module hinges on a trade-off between noise reduction and information retention, presenting a significant challenge for the model. As illustrated in Figure 9 in Appendix D, we identify two primary failure modes arising from this trade-off. First, Information Loss occurs when the abstractor is overly aggressive, discarding critical default rules, such as "Parents are usually supportive", thereby severing the logical chain. Conversely, Information Redundancy arises from excessive conservatism, where irrelevant narrative details like "expecting a child" are retained. These redundant distractors significantly increase the complexity of the downstream formalization.

Formalization Errors. We analyze the distinct error patterns arising from monolithic versus step-wise formalization. As shown in Figure 10 in Appendix D, monolithic approaches struggle with information density, often failing to instantiate essential atomic facts such as "heavy(blockb)". Such omissions result in structural information loss that breaks the reasoning chain, demonstrating that monolithic formalization lacks the granularity required for robust NMR. While our step-wise decomposition mitigates these global failures, fine-grained semantic misalignments persist at the sentence level. Table 16 in Appendix D categorizes these residual errors into three types: Logical Structure Errors, Predicate Mapping Errors, and Conflation of Facts and Rules. A notable instance of conflation involves the mistranslation of strict facts, such as "Dogs are animals", into defeasible rules. These findings indicate that, despite architectural improvements, achieving precise semantic alignment in autoformalization remains a significant challenge for LLMs.

Table 4: Ablation study results of AS-NMRer on LogicBench with search size of 3. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Syn. and Sem. denote the Syntax and Semantic verifiers. Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Model	Search Size	Method	F1 (%)								
				DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
Logic Bench	DeepSeek -R1- 32B	3	**AS-NMRer**	**92.1**	**94.7**	**94.7**	**62.9**	**78.9**	**86.6**	81.5	84.1	**84.0**
			w/o Abstraction	44.1	32.1	57.4	46.2	49.7	45.2	37.4	81.8	49.2
			w/o Step-wise BoN	69.3	55.3	44.1	61.7	67.5	67.6	63.7	53.2	72.8
			w/o Syn. Verifier	86.6	78.4	74.9	50.9	72.1	83.8	72.5	91.6	65.9
			w/o Sem. Verifier	80.9	74.9	68.4	61.5	77.7	73.6	75.8	**96.4**	66.4
			w/o Reabstraction	86.6	94.7	86.6	54.2	77.3	84.0	83.8	85.4	81.8
	Gemma 3-27B		**AS-NMRer**	94.7	**89.4**	94.7	**73.6**	81.8	83.8	78.7	92.1	**85.7**
			w/o Abstraction	48.8	44.3	68.4	50.9	58.5	55.2	45.2	80.9	56.5
			w/o Step-wise BoN	**97.4**	72.5	74.9	61.5	84.2	63.8	74.9	77.9	75.9
			w/o Syn. Verifier	84.0	84.0	89.4	60.4	**92.4**	78.4	**83.8**	**94.4**	83.3
			w/o Sem. Verifier	84.0	84.0	89.4	66.8	88.6	80.9	78.4	93.4	83.2
			w/o Reabstraction	94.7	83.8	**100**	72.2	79.6	**83.8**	75.9	92.1	85.3

5 Discussion

Importance of Abstraction. Our findings suggest that abstraction is not merely an optimization but a prerequisite for NMR in realistic settings. By prioritizing the extraction of core atomic facts and default rules, AS-NMRer significantly reduces the input complexity for the formalizer. This ensures that the subsequent translation focuses solely on essential reasoning components, effectively mitigating the interference of irrelevant linguistic details.

Granularity in Formalization. The superior performance of our step-wise strategy over monolithic baselines highlights the importance of verification granularity. Monolithic translation suffers from inherent brittleness, where a single syntax error or semantic hallucination invalidates the entire program. In contrast, our sentence-level dual-verification acts as a fine-grained filter, localizing errors to individual rules. This ensures that the final logic program is constructed only from constituents that are both syntactically executable and semantically faithful.

Limitations of Supervised Fine-Tuning. Although expert iteration successfully accumulates validated examples, the performance gains from SFT eventually reach a plateau. This saturation suggests that the model tends to exploit surface-level syntactic heuristics rather than internalizing the underlying logical semantics, fundamentally limiting its ability to achieve genuine mastery of the logical language.

6 Conclusions

Non-monotonic reasoning (NMR) is a fundamental mode of logical reasoning, while autoformalization has emerged as a vital neural-symbolic approach for logical reasoning. This work addresses the fragility of autoformalization in NMR, highlighting narrative noise and monolithic translation errors as critical bottlenecks. We propose AS-NMRer, which enhances autoformalization for NMR through abstraction, search, and fine-tuning. First, to mitigate linguistic noise, we employ an abstraction module to distill raw contexts into structured atomic facts and default rules before formalization. Second, this approach significantly enhances the model's autoformalization capability through a verification-guided step-wise search that ensures syntactic and semantic correctness. Furthermore, we introduce an expert iteration loop that fine-tunes the model to internalize the syntax and semantics of the formal language. Empirical evaluations across four benchmarks confirm that AS-NMRer significantly enhances reasoning performance, demonstrating that equipping smaller open-source models with rigorous fine-grained formalization strategies enables them to offset the scale advantage of massive proprietary baselines. Future work could investigate integrating reinforcement learning with structured search to address the bottlenecks of supervised fine-tuning. Building on this, we aim to extend the framework's capability to robustly solve non-monotonic reasoning tasks in real-world scenarios like legal reasoning.

Limitations

While AS-NMRer demonstrates robust performance across various NMR benchmarks, we identify several limitations inherent to the current framework. First, regarding error propagation, the framework operates as a pipeline, rendering the downstream formalization quality highly dependent on that of the initial abstraction module. While the evaluator and re-abstraction mechanism mitigate explicit formalization failures, subtle semantic deviations or omitted premises may evade detection and proceed to the symbolic solver. Second, concerning computational efficiency, the step-wise Best-of-N search strategy introduces significant inference overhead. Specifically, the necessity for multiple invocations of both the ASP solver and the semantic verifier at every reasoning step necessitates a trade-off between accuracy and speed, thereby restricting the framework's deployability in real-time or resource-constrained environments. Finally, regarding the expert iteration mechanism, the self-improvement loop exhibits sensitivity to both the base model's initial capabilities and the scale of accumulated data. We observe that effective self-evolution relies on accumulating a critical mass of high-quality, solver-verified samples; insufficient initial data can lead to training instability or slow convergence during the early phases of the iteration loop.

References

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

Gustavo Cilleruelo Calderón, Emily Allaway, and et al. 2025. Generics are puzzling. can language models find the missing piece? In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 6571–6588.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *IJCAI 2020*, pages 3882–3890. ijcai.org.

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

Xiaotong Fang, Zhaoqun Li, Chen Chen, and Beishui Liao. 2025. Llm-aspic+: A neuro-symbolic framework for defeasible reasoning. In *ECAI 2025*, pages 1567–1574. IOS Press.

Michael Gelfond and Vladimir Lifschitz. 1988. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, pages 1070–1080. MIT Press.

Michael Gelfond and Vladimir Lifschitz. 1991. Classical negation in logic programs and disjunctive databases. *New Gener. Comput.*, 9(3/4):365–386.

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.

Ruikang Hu, Shaoyu Lin, and et al. 2025. LTRAG: enhancing autoformalization and self-refinement for logical reasoning with thought-guided RAG. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 2483–2493.

Aaron Hurst, Adam Lerer, and et al. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

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.

Aaron Jaech, Adam Kalai, and et al. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

Albert Qiaochu Jiang, Sean Welleck, and et al. 2023. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

James Ravi Kirkpatrick and Rachel Katharine Sterken. 2025. Generics and default reasoning in large language models. *CoRR*, abs/2508.13718.

Takeshi Kojima, Shixiang Shane Gu, and et al. 2022. Large language models are zero-shot reasoners. In *NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Aitor Lewkowycz, Anders Andreassen, and et al. 2022. Solving quantitative reasoning problems with language models. In *NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zenan Li, Yifan Wu, and et al. 2024. Autoformalize mathematical statements by symbolic equivalence and semantic consistency. In *NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yong Lin, Shange Tang, and et al. 2025a. Goedel-prover: A frontier model for open-source automated theorem proving. *CoRR*, abs/2502.07640.

Yong Lin, Shange Tang, and et al. 2025b. Goedel-prover-v2: Scaling formal theorem proving with scaffolded data synthesis and self-correction. *CoRR*, abs/2508.03613.

Jianqiao Lu, Yingjia Wan, and et al. 2025. Formalalign: Automated alignment evaluation for autoformalization. In *ICLR 2025, Singapore, April 24-28, 2025*.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5153–5176.

Liangming Pan, Alon Albalak, and et al. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824.

Mihir Parmar, Nisarg Patel, and et al. 2024. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13679–13707.

Nisarg Patel, Mohith Kulkarni, and et al. 2024. Multilogieval: Towards evaluating multi-step logical reasoning ability of large language models. In *EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 20856–20879.

Marcos A. Pertierra, Sarah Lawsky, Erik Hemberg, and Una-May O'Reilly. 2017. Towards formalizing statute law as default logic through automatic semantic parsing. In *(ICAIL 2017), London, UK, June 16, 2017*, volume 2143 of *CEUR Workshop Proceedings*.

Chengwen Qi, Ren Ma, and et al. 2025. Large language models meet symbolic provers for logical reasoning evaluation. In *ICLR 2025, Singapore, April 24-28, 2025*.

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

Lin Ren, Guohui Xiao, and et al. 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, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4661–4675.

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. Springer.

Gemma Team. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Trieu H. Trinh, Yuhuai Wu, and et al. 2024. Solving olympiad geometry without human demonstrations. *Nat.*, 625(7995):476–482.

Jason Wei, Xuezhi Wang, and et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yuhuai Wu, Albert Qiaochu Jiang, and et al. 2022. Autoformalization with large language models. In *NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ran Xin, Chenguang Xi, and et al. 2025. Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 32588–32599.

Yeliang Xiu and Yongmei Liu. 2025. Multilogicnmr (er): A benchmark and neural-symbolic framework for non-monotonic reasoning with multiple extensions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18383–18416.

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, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3616–3626.

Lan Zhang, Xin Quan, and André Freitas. 2024. Consistent autoformalization for constructing mathematical libraries. In *EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 4020–4033.

King Zhu, Hanhao Li, and et al. 2025. Scaling test-time compute for LLM agents. *CoRR*, abs/2506.12928.

A Prompts for AS-NMRer Framework

This section presents the detailed prompts used in the AS-NMRer framework, including prompts for the Abstractor (Figure 4), Evaluator (Figure 5), Re-abstraction (Figure 6), Formalizer (Figure 7), and Semantic Verifier (Figure 8)

Prompt for Abstractor in Abstraction Module

Task Description: Given the natural language context and the question, you must first extract predicates and entities. Then, extract atomic facts and default rules regarding these predicates and entities from the context. The generated facts and rules are necessary for reasoning and answering the question. A **Fact** is a specific, unconditional, and ground statement about particular entities, their properties, or their relationships (e.g., entityA is predicateB). A **Default Rule** is a general rule that applies to a class of entities but allows for exceptions (e.g., predicateA is usually predicateB). Information in the context that is irrelevant to reasoning and answering the question should be simplified or omitted. Furthermore, exceptions in facts and rules need to be represented by corresponding predicates. Regarding the extraction order, you must first extract all facts necessary for reasoning about the question, and then extract all default rules. The extracted facts must not contradict each other. Individual facts and rules must be separated by periods.
The input format is: The natural language context sentences are: “”. The questions are: “”.
The output format is: The extracted entities are: “”. The extracted predicates are: “”. The extracted facts and rules from the context are: “”. The extracted questions are: “”.

For Example: [Insert a manually annotated example specific to the current dataset here. This example must demonstrate the abstraction process from the original natural language context to the corresponding abstraction facts and rules.]
Do not output your reasoning or thinking process.

Figure 4: Prompt for Abstractor in Abstraction Module

Prompts for Evaluator in Abstraction Module

Task Description: You are a rigorous logical analyst. Your task is to evaluate the correctness of a set of facts and default rules extracted from a context, specifically for the purpose of answering reasoning-based questions.
Note that the extracted facts and default rules must be necessary to answer the question; irrelevant information should not be included. A **Fact** is a specific, unconditional, ground statement about particular entities, their properties, or their relationships (often using copulas like 'is'). A **Default Rule** is a general rule applying to a class of entities but allowing for exceptions (often signaled by adverbs like 'typically' or 'usually').
You must verify the input along two dimensions: 1. **Soundness**: All facts and rules must be entirely derived from the context and faithful to its semantics. 2. **Completeness**: The facts and rules must be sufficient to form a complete reasoning chain to definitively answer the question.
Please conduct your analysis and adhere to the following output rules:
If both soundness and completeness are met, your verification result must be 'yes' and the suggestion must be 'None'.
If either criterion is not met, the verification result must be 'no'. You must then generate specific suggestions for missing facts or default rules necessary to complete the reasoning. Do not repeat facts and rules that are already correctly extracted.
The input format is: The natural language context sentences are: "". The extracted facts and default rules are: "". The questions are: "".
The output format is: The verification result is: "". The suggestion of extracted facts and default rules is: "". The extracted questions are: "".
Do not output your reasoning or thinking process.

Figure 5: Prompts for Evaluator in Abstraction Module

Prompts for Sentence Re-abstraction

Task Description: A sentence extracted from the context has failed during the formalization process. Your task is to re-extract or refine this natural language sentence based on the original context and the provided error message. The goal is to generate a sentence that preserves the original meaning but resolves the formalization error. A **Fact** is a specific, unconditional, ground statement about particular entities, their properties, or their relationships (e.g., entityA is predicateB). A **Default Rule** is a general rule that applies to a class of entities but allows for exceptions (e.g., predicateA is usually predicateB).
Instructions: 1. Simplify or omit information in the context not related to reasoning or answering the question. 2. Replace exceptions in facts and rules with corresponding predicates. 3. Focus on fixing the specific sentence that caused the error.
The input format is: The natural language context sentences are: "". The questions are: "". The previously extracted sentences are: "". The incorrectly formalized natural language sentence is: "". The error message is: "".
The output format is: The re-extracted sentence in the context is: "".
Do not output your reasoning or thinking process.

Figure 6: Prompts for Sentence Re-abstraction

Prompts for Formalizer

Task Description: You are a Default Logic expert. Your task is to translate a given natural language sentence into Default Logic formalism. First, determine whether the sentence is a Fact or a Default Rule, and then translate it accordingly.
A **Fact** in Default Logic is represented as an atomic formula composed of predicates and entities, such as "predicate1(entity1).", stating that entity1 has the property predicate1. A **Default Rule** is represented as a logical formula composed of predicates and variables. For example, the rule "predicate1(X) : predicate2(X) / predicate2(X)." signifies that if X is predicate1, then X is typically inferred to be predicate2.
Constraints: 1. Extract entities (or variables) and predicates from the sentence first. 2. No spaces are allowed in entity names and predicates. 3. To ensure consistency, you must prioritize using predicates and entities already extracted from the provided context.
The input format is: The natural language context sentences are: "". The extracted predicates from context are: "". The extracted entities from context are: "". The translated formalized context sentences are: "". The natural language sentence to be translated is: "".
The output format is: The predicate of the sentence is: "". The entities of the sentence are: "". The translated formal logic program sentence is: "".
For Example: [Insert a manually annotated example specific to the current dataset here. This example must demonstrate the formalization process from the dataset's natural language to the corresponding Default Logic format.]
Do not output your reasoning or thinking process.

Figure 7: Prompts for Formalizer

Prompts for Semantic Verifier

Task Description: You are an expert in Answer Set Programming (ASP) logic and linguistics. In ASP, a **Fact** is represented as an atomic formula composed of predicates and entities, such as "predicate1(entity1).", stating that entity1 has the property predicate1. An **ASP Rule** is represented as a logical formula composed of predicates and variables. For example, the rule "predicate2(X) :- predicate1(X), not -predicate2(X)." signifies a default inference: if X is predicate1, then X is typically inferred to be predicate2 (assuming there is no evidence to the contrary).
Your task is to evaluate the correctness of the formalization from a natural language sentence to an ASP statement based on the following criteria: 1. Syntactic Correctness 2. Semantic Equivalence 3. Consistency of Atomic Propositions 4. Conciseness You are required to provide a final overall score ranging from 0 to 1, where 1 means the translated ASP statement is completely correct, and 0 means it is completely wrong. **The input format is:** The natural language sentences are: "". The translated logic program sentences are: "".
The output format is: The score is:
For example: [Insert a manually annotated example specific to the current dataset here. This example must demonstrate the scoring based on the criteria above.]
Note that you only need to output the final overall score. Do not output your reasoning or thinking process.

Figure 8: Prompts for Semantic Verifier

B The Detailed Description for Model Fine-tuning

We employed the LoRA method to fine-tune the open-source LLMs, DeepSeek-R1-32B and Gemma3-27B. The hyperparameters used for fine-tuning are detailed in Table 5. All experiments were conducted on a single NVIDIA RTX 4090 GPU utilizing the Unsloth[2] framework."

Table 5: Fine-tuning parameters of open-source LLMs.

Parameter	Value
per_device_train_batch_size	2
gradient_accumulation_steps	8
warmup_steps	10
max_steps	100
weight_decay	0.01
optim	Adamw_8bit
seed	3407

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

C Detailed Experimental Results

LogicBench serves as the primary testbed for evaluating robustness against linguistic noise in Binary Question Answering (BQA) tasks, encompassing eight distinct deductive reasoning patterns. Table 6 details the comparative performance of AS-NMRer against Zero-Shot, Few-Shot, and Fine-Tuned baselines, stratifying results by specific logic types: Disjunction Resolution (DRD), Disjunction Introduction (DRI), Double Negation Elimination (DRO), Disjunctive Syllogism (DRS), Reductio ad Absurdum (RAP), and three variants of Reasoning by Cases (REI, RE2, RE3). To isolate the contributions of inference-time mechanisms, Tables 7 and 8 present ablation studies for DeepSeek-R1-32B and Gemma3-27B, respectively, quantifying the performance gains achieved by scaling the Search Size (N) and increasing Fine-tuning Iterations (T).

Table 6: Results on LogicBench (BQA) across different methods and models. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Method	Base Model	F1 (%)								
			DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
LogicBench (BQA)	Zero-Shot	DeepSeek-R1-32B	87.3	70.3	84.9	19.1	28.7	76.7	81.9	48.1	62.1
		DeepSeek-V3.2-671B	97.4	87.4	70.3	41.6	56.1	84.6	79.9	70.6	73.5
		Gemma3-27B	87.3	68.0	89.9	23.8	25.4	73.3	73.9	45.8	60.9
		o3-mini	100	92.5	90.0	62.1	72.5	84.6	84.9	95.0	85.2
		GPT-4o-mini	95.0	85.0	84.6	39.8	73.2	87.3	79.8	86.1	78.8
	Few-Shot	DeepSeek-R1-32B	97.5	87.5	73.3	35.4	40.7	89.9	87.5	97.5	76.2
		DeepSeek-V3.2-671B	100	89.7	73.9	38.9	60.9	79.5	82.2	92.5	77.2
		Gemma3-27B	97.5	87.5	76.3	34.2	40.6	92.5	82.4	91.2	75.2
		o3-mini	100	95.0	87.3	69.7	73.2	84.6	87.5	93.7	86.4
		GPT-4o-mini	100	85.0	95.0	35.5	69.3	85.0	82.4	95.0	80.9
	Fine-Tuned	DeepSeek-R1-32B	100	92.5	97.5	67.5	66.1	77.1	89.9	50.0	80.1
		Gemma3-27B	100	89.9	95.0	66.4	69.1	84.8	87.4	48.6	80.2
	LLM2ASP	DeepSeek-V3.2-671B	74.9	83.8	77.9	70.9	69.6	64.9	71.7	49.1	70.4
		GPT-4o-mini	61.3	68.4	61.3	58.2	56.2	42.4	66.0	43.8	57.2
		o3-mini	92.1	94.7	83.8	89.1	70.9	89.4	86.8	46.4	81.6
		DeepSeek-R1-32B	74.9	72.5	68.4	68.0	62.5	62.7	81.3	58.6	68.7
		Gemma3-27B	71.7	60.4	68.4	62.6	45.3	54.8	64.9	42.2	58.8
	AS-NMRer (ours)	DeepSeek-R1-32B	92.1	75.5	94.7	68.3	83.3	86.8	89.4	90.7	85.1
		DeepSeek-V3.2-671B	86.6	77.9	92.1	55.3	77.3	77.9	75.4	90.4	79.1
		Gemma3-27B	89.4	86.8	92.1	65.9	92.4	100	83.8	92.1	87.8
		GPT-4o-mini	94.7	63.8	94.7	66.8	70.9	75.5	78.4	66.9	76.5

Table 7: Results of DeepSeek-R1-32B on LogicBench with different iterations and search sizes. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Model	Iter	Search Size	F1(%)								
				DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
Logic Bench	DeepSeek-R1 -32B	0	1	92.1	92.1	81.5	53.7	72.7	75.9	77.9	77.5	77.9
			3	92.1	94.7	94.7	62.9	78.9	86.6	81.5	84.1	84.0
			5	94.7	89.4	94.7	56.2	81.8	78.4	80.9	81.3	82.2
		1	1	89.4	80.9	83.8	48.2	69.4	75.5	75.5	77.4	75.0
			3	91.9	92.1	86.6	61.7	76.8	81.3	77.9	81.3	81.2
			5	97.4	81.3	83.0	64.1	75.8	83.8	80.9	84.1	81.5
		2	1	94.7	75.5	92.1	57.5	79.2	92.1	84.0	92.1	83.4
			3	89.4	86.6	97.4	58.2	79.6	86.8	80.9	79.8	82.3
			5	97.4	86.6	86.8	68.3	78.7	84.2	89.4	89.4	85.1
		3	1	92.1	77.9	92.1	60.6	79.6	81.5	81.3	90.7	82.1
			3	94.7	72.5	94.7	54.3	78.7	94.7	84.0	89.4	82.9
			5	92.1	75.5	94.7	68.3	83.3	86.8	89.4	90.7	85.1

Table 8: Results of Gemma3-27B on LogicBench with varying iterations and search sizes. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Model	Iter	Search Size	F1(%)								
				DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
Logic Bench	Gemma3 -27B	0	1	94.7	92.1	94.7	64.4	84.8	81.3	84.0	90.7	85.8
			3	94.7	89.4	94.7	73.6	81.8	83.8	78.7	92.1	85.7
			5	94.7	83.8	97.4	69.6	84.8	86.6	84.0	92.1	86.6
		1	1	94.7	83.8	94.7	68.3	87.0	86.6	81.5	92.1	86.1
			3	97.4	86.6	89.4	65.6	85.4	83.8	86.8	92.1	85.9
			5	92.1	83.5	100	65.5	84.8	83.8	81.3	88.0	84.8
		2	1	86.8	84.0	94.7	65.9	92.4	100	78.7	94.7	87.2
			3	86.8	89.4	86.8	70.9	90.3	92.1	72.5	93.4	85.3
			5	94.7	86.6	89.4	64.4	86.4	100	75.9	92.9	86.3
		3	1	89.4	86.4	92.1	65.9	90.3	100	75.5	94.7	86.8
			3	89.4	86.8	92.1	65.9	92.4	100	83.8	92.1	87.8
			5	92.1	86.8	97.4	62.1	88.1	94.7	78.4	94.7	86.8

To assess the framework's capacity for maintaining logical coherence over extended deduction chains, we employ the Multi-LogiEval benchmark across increasing reasoning depths ranging from one to five. Table 9 reports performance on the fundamental depth-1 subset (D1), establishing a baseline for single-step deduction capabilities. A more granular analysis is provided in Table 10, which decomposes depth-2 (D2) results into Modus Ponens (MP) and Modus Tollens (MT) categories to reveal specific deductive strengths. Finally, Table 11 extends this evaluation to high-complexity scenarios, summarizing F1 scores for deep reasoning chains (depths 3 through 5) where error propagation typically degrades performance.

Table 9: Results on Multi-LogiEval (d1) across different methods and models. The abbreviations DRD, DRI, DRS, DRO, RE1-3, and RAP correspond to the reasoning patterns defined in LogicBench (Parmar et al., 2024). Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Method	Base Model	F1 (%)								
			DRD	DRI	DRO	DRS	RE1	RE2	RE3	RAP	Avg.
Multi-LogiEval (d1)	Zero-Shot	DeepSeek-R1-32B	100	48.7	47.4	47.4	45.9	27.5	25.0	100	55.2
		DeepSeek-V3.2-671B	39.4	28.5	13.0	4.8	31.0	33.5	78.0	39.0	33.4
		Gemma3-27B	47.3	20.0	44.4	0.0	19.1	19.1	43.5	26.3	27.5
		o3-mini	45.9	23.1	4.7	4.8	16.7	33.5	64.3	45.9	29.8
		GPT-4o-mini	37.5	20.0	20.0	0.0	25.9	30.7	79.8	37.5	31.4
	Few-Shot	DeepSeek-R1-32B	48.7	41.1	37.5	13.3	23.3	37.3	37.5	47.4	35.7
		DeepSeek-V3.2-671B	47.4	23.1	48.7	9.1	28.6	45.1	78.0	47.4	40.9
		Gemma3-27B	100	23.6	45.9	27.5	100	44.4	25.3	45.9	51.6
		o3-mini	48.7	31.0	41.1	9.0	16.7	37.5	64.3	48.7	37.1
		GPT-4o-mini	39.4	20.0	13.0	0.0	33.3	31.0	89.5	47.3	34.2
	Fine-Tuned	DeepSeek-R1-32B	100	48.7	100	23.0	44.4	25.9	39.4	47.4	53.6
		Gemma3-27B	100	42.8	48.7	37.5	42.8	25.9	37.5	45.9	47.6
	LLM2ASP	DeepSeek-V3.2-671B	40.6	40.6	5.0	0.0	17.4	9.5	36.7	20.8	21.3
		GPT-4o-mini	29.6	32.1	13.6	0.0	13.6	10.5	36.7	17.4	19.2
		o3-mini	100	44.1	0.0	29.6	26.9	15.6	29.0	17.4	32.9
		DeepSeek-R1-32B	47.2	42.4	13.6	0.0	29.6	15.6	35.2	13.7	24.7
		Gemma3-27B	44.1	45.7	9.5	0.0	5.0	5.0	39.1	9.5	19.7
	AS-NMRer (ours)	DeepSeek-R1-32B	36.7	32.1	24.0	36.7	20.8	35.2	52.7	24.0	32.8
		DeepSeek-V3.2-671B	36.7	52.4	0.0	40.8	20.8	15.6	57.8	33.3	32.1
		Gemma3-27B	47.2	48.6	20.8	42.4	40.6	13.6	62.7	36.7	39.1
		GPT-4o-mini	36.7	32.1	5.0	32.1	26.9	15.6	62.7	36.8	30.9

Table 10: Results on Eval-LogiEval (d2). The metrics BDR, DRD, DRI, PBD, REI, and REII are split into MP and MT sub-metrics. Avg. denotes the average F1 score across all reasoning patterns.

Dataset	Method	Base Model	F1 (%)												Avg.
			BDR		DRD		DRI		PBD		REI		REII		
			MP	MT	MP	MT	MP	MT	MP	MT	MP	MT	MP	MT	
Eval-LogiEval (d2)	Zero-Shot	DeepSeek-R1-32B	34.1	16.7	48.7	4.7	100	33.0	100	30.0	46.7	13.0	0.0	8.69	19.6
		DeepSeek-V3.2-671B	37.5	100	43.5	42.8	100	100	45.9	89.8	30.4	48.7	33.3	44.4	59.7
		Gemma3-27B	53.1	44.4	48.7	39.4	100	100	47.3	64.3	42.8	25.0	13.0	42.8	51.7
		o3-mini	46.7	100	34.7	100	48.7	100	48.7	64.3	17.5	48.7	48.7	100	63.2
		GPT-4o-mini	54.8	47.4	43.5	45.9	47.4	100	39.4	64.3	28.7	45.9	28.6	44.4	49.2
	Few-Shot	DeepSeek-R1-32B	64.3	47.4	47.4	45.9	100	43.0	48.7	43.7	45.8	47.4	13.0	48.7	49.6
		DeepSeek-V3.2-671B	56.0	100	40.5	100	100	100	42.8	89.9	12.5	48.7	48.7	48.7	65.6
		Gemma3-27B	41.2	3.2	48.7	0.0	100	47.0	48.7	33.3	25.6	0.0	0.0	13.3	30.1
		o3-mini	34.1	100	37.3	100	100	100	100	68.6	18.4	100	48.7	100	75.6
		GPT-4o-mini	60.0	100	46.7	45.9	47.4	100	47.3	60.1	33.3	44.4	35.5	47.4	55.7
	Fine-Tuning	DeepSeek-R1-32B	41.2	13.0	48.7	0.0	100	100	100	39.4	48.4	9.0	0.0	16.7	43.0
		Gemma3-27B	41.2	23.1	48.7	0.0	100	100	100	52.0	48.4	0.0	0.0	16.7	44.2
	LLM2ASP	DeepSeek-V3.2-671B	42.4	100	24.4	100	100	100	47.2	78.4	40.0	100	29.6	48.6	67.6
		GPT-4o-mini	40.6	42.4	20.2	100	100	45.7	38.7	72.8	40.0	100	36.7	47.2	57.0
		o3-mini	42.4	47.2	32.1	47.2	100	100	44.1	89.2	40.0	100	9.5	100	62.6
		DeepSeek-R1-32B	47.4	42.4	35.6	100	100	36.7	34.5	72.5	37.5	47.2	17.4	38.7	50.8
		Gemma3-27B	38.7	100	42.4	48.6	100	36.7	26.9	52.9	44.4	44.1	24.0	40.6	50.0
	AS-NMRer **(ours)**	DeepSeek-R1-32B	39.1	100	20.2	100	48.6	100	47.2	62.7	31.8	100	26.9	100	64.7
		DeepSeek-V3.2-671B	43.4	100	20.2	100	48.6	100	47.2	77.4	16.7	100	34.5	100	65.7
		Gemma3-27B	42.4	100	24.4	100	47.2	100	44.1	68.3	31.8	100	32.1	100	65.8
		GPT-4o-mini	28.4	100	20.2	100	100	100	42.4	52.5	11.8	100	40.6	47.8	62.0

Table 11: Results on Multi-LogiEval (d3/4/5) across different methods and models. The datasets D3 and D4 are evaluated on two subsets (d1 and d2). Avg. denotes the average F1 score.

Dataset	Method	Base Model	F1 (%)					
			D3		D4		D5	Avg.
			d1	d2	d1	d2		
Multi-LogiEval (d3/4/5)	Zero-Shot	DeepSeek-R1-32B	33.3	33.3	33.3	33.3	43.7	35.4
		DeepSeek-V3.2-671B	13.0	74.9	73.3	50.0	64.2	55.1
		Gemma3-27B	25.9	52.3	43.5	43.5	45.0	42.0
		o3-mini	43.5	56.0	84.6	79.2	45.1	61.7
		GPT-4o-mini	33.3	33.3	43.6	43.6	28.5	36.5
	Few-Shot	DeepSeek-R1-32B	16.7	79.2	79.8	89.8	70.0	67.1
		DeepSeek-V3.2-671B	19.2	70.0	79.2	49.5	64.2	56.4
		Gemma3-27B	25.9	58.3	47.8	60.1	64.2	51.3
		o3-mini	52.4	56.0	79.2	73.3	37.3	59.6
		GPT-4o-mini	23.1	33.3	43.6	33.3	25.9	31.8
	Fine-Tuning	DeepSeek-R1-32B	34.0	94.9	84.9	100	62.7	75.3
		Gemma3-27B	40.5	73.3	52.3	100	73.3	67.8
	LLM2ASP	DeepSeek-V3.2-671B	100	83.5	100	94.7	83.5	92.3
		GPT-4o-mini	52.6	73.4	62.7	50.4	52.1	58.5
		o3-mini	94.7	89.2	100	94.7	70.7	89.8
		DeepSeek-R1-32B	94.7	57.8	84.0	100	66.1	76.2
		Gemma3-27B	89.4	84.0	83.5	72.5	83.5	82.6
	AS-NMRer **(ours)**	DeepSeek-R1-32B	100	89.2	94.7	89.2	70.7	88.8
		DeepSeek-V3.2-671B	100	83.5	83.5	89.2	55.2	74.2
		Gemma3-27B	89.2	83.5	100	83.5	70.7	85.4
		GPT-4o-mini	77.4	77.4	83.5	89.2	63.5	78.2

LogicNMR assesses the model's capacity for NMR through the introduction of sequential updates to the knowledge base, denoted by U. Table 12 aggregates F1 scores across varying update frequencies, ranging from $U = 1$ to $U = 5$, thereby tracking the stability of the reasoning process as the knowledge base evolves. Complementing this, Table 13 details an ablation study using the Gemma3-27B backbone, analyzing how the synergy between varying search sizes and expert iteration steps mitigates the performance fluctuations associated with increasing knowledge base updates.

Table 12: Results on LogicNMR across different methods and models. $U = 1$ to $U = 5$ denote different uncertainty levels or settings. Avg. denotes the average F1 score.

Dataset	Method	Base Model	F1(Test)					
			U=1	U=2	U=3	U=4	U=5	Avg.
LogicNMR	Zero-Shot	DeepSeek-R1-32B	35.2	40.8	33.5	35.1	35.6	36.4
		DeepSeek-V3.2-671B	45.2	34.5	33.2	33.5	39.9	37.3
		Gemma3-27B	47.3	35.7	30.8	43.8	32.7	38.2
		o3-mini	24.8	27.6	28.7	22.9	33.5	27.8
		GPT-4o-mini	27.6	31.5	37.9	38.9	25.9	32.7
	Few-Shot	DeepSeek-R1-32B	42.5	52.3	37.3	36.6	43.4	42.9
		DeepSeek-V3.2-671B	55.5	48.2	41.7	48.5	44.5	47.6
		Gemma3-27B	58.3	38.3	36.7	40.9	37.0	42.2
		o3-mini	33.3	37.9	27.2	34.9	36.4	33.9
		GPT-4o-mini	21.3	32.4	24.0	36.7	29.6	28.8
	Fine-Tuning	DeepSeek-R1-32B	98.9	91.8	95.1	92.1	94.0	94.4
		Gemma3-27B	51.3	51.6	50.0	48.3	46.9	49.6
	LLM2ASP	DeepSeek-V3.2-671B	56.1	45.6	52.3	52.9	60.3	53.6
		GPT-4o-mini	44.8	60.8	51.1	51.6	55.1	52.8
		o3-mini	45.6	59.7	65.2	54.3	64.3	58.0
		DeepSeek-R1-32B	49.5	44.5	48.2	43.4	49.8	47.3
		Gemma3-27B	51.2	55.4	47.8	57.8	60.0	54.5
	AS-NMRer **(ours)**	DeepSeek-R1-32B	93.8	96.4	93.6	92.3	88.2	92.8
		DeepSeek-V3.2-671B	94.9	96.2	95.2	93.1	98.9	95.7
		Gemma3-27B	93.2	91.5	94.9	94.6	95.6	93.6
		GPT-4o-mini	94.9	94.6	96.9	94.8	97.2	95.8

Table 13: Results on LogicNMR using Gemma3-27B with varying iterations and search sizes. U denotes the number of updates to the knowledge base. Avg. denotes the average F1 score across different numbers of knowledge base updates.

Data set	Model	Iter	Search size	F1(%)					
				U=1	U=2	U=3	U=4	U=5	Avg.
LogicNMR	Gemma3-27B	0	1	91.9	92.3	89.2	88.1	89.2	90.2
			3	92.9	90.1	89.2	90.3	96.0	91.7
			5	93.2	91.5	94.9	94.6	95.6	93.6
		1	1	91.8	91.2	91.1	92.3	94.2	92.1
			3	95.9	88.9	85.2	87.2	91.1	89.7
			5	92.9	92.2	94.1	91.0	93.0	92.7
		2	1	92.9	91.1	94.1	91.1	93.0	92.5
			3	92.9	92.2	94.1	91.1	94.1	92.9
			5	92.9	92.2	92.2	90.1	92.1	91.8
		3	1	93.9	92.2	93.2	90.1	93.1	92.5
			3	91.8	89.9	94.1	92.0	93.1	92.6
			5	92.9	92.2	93.2	90.1	93.0	92.3

The MultiLogicNMR benchmark evaluates reasoning in complex scenarios characterized by multiple valid answer sets, where E represents the number of answer sets intrinsic to each sample. Table 14 focuses on skeptical Reasoning, reporting F1 scores where conclusions must hold true across all valid answer sets, stratified by extension counts from $E = 1$ to $E = 5$. Conversely, Table 15 evaluates credulous reasoning, where a conclusion is deemed valid if it is supported by at least one answer set, assessing the framework's ability to identify potential truths within multi-extension environments.

Table 14: Results on MultiLogic NMR (skeptical) across different methods and models. $E = 1$ to $E = 5$ denote different evaluation settings or reasoning lengths. Avg. denotes the average F1 score.

Dataset	Method	Base Model	F1(Test)					
			E=1	E=2	E=3	E=4	E=5	Avg.
MultiLogic NMR (skeptical)	Zero-Shot	DeepSeek-R1-32B	47.2	56.9	47.2	52.7	53.3	51.6
		DeepSeek-V3.2-671B	36.9	28.2	43.8	39.1	47.2	39.2
		Gemma3-27B	38.4	42.5	49.2	24.5	47.6	41.1
		o3-mini	38.1	29.4	29.4	44.4	40.5	36.8
		GPT-4o-mini	39.1	31.7	44.6	31.5	43.3	38.1
	Few-Shot	DeepSeek-R1-32B	44.0	53.2	50.3	45.8	48.4	49.2
		DeepSeek-V3.2-671B	60.2	60.9	51.5	47.7	60.0	56.3
		Gemma3-27B	56.9	52.7	54.9	41.7	50.3	51.5
		o3-mini	66.6	67.5	70.5	82.1	70.8	71.7
		GPT-4o-mini	34.4	42.8	44.5	33.3	46.5	40.4
	Fine-Tuning	DeepSeek-R1-32B	64.2	73.4	82.9	78.5	81.8	76.1
		Gemma3-27B	58.9	61.6	67.3	64.9	65.0	63.5
	LLM2ASP	DeepSeek-V3.2-671B	83.8	66.0	53.8	57.8	72.3	66.8
		GPT-4o-mini	61.7	66.5	66.6	68.0	55.3	63.8
		o3-mini	85.4	80.6	74.1	60.9	78.9	76.9
		DeepSeek-R1-32B	45.8	59.6	55.6	48.7	48.8	52.1
		Gemma3-27B	49.1	53.6	59.2	57.7	53.6	54.9
	AS-NMRer **(ours)**	DeepSeek-R1-32B	77.1	72.3	77.2	81.7	83.6	78.6
		DeepSeek-V3.2-671B	96.7	86.5	89.9	89.9	93.3	91.3
		Gemma3-27B	86.0	70.0	81.1	84.6	65.6	78.5
		GPT-4o-mini	95.1	84.7	88.2	88.3	86.3	88.6

Table 15: Results on MultiLogicNMR (credulous) across different methods and models. E=1 to E=5 denote different evaluation settings. Avg. denotes the average F1 score.

Dataset	Method	Base Model	F1(Test)					
			E=1	E=2	E=3	E=4	E=5	Avg.
MultiLogic NMR (credulous)	Zero-Shot	DeepSeek-R1-32B	49.2	50.2	47.7	40.8	48.8	47.6
		DeepSeek-V3.2-671B	50.8	42.6	43.8	30.2	33.4	40.4
		Gemma3-27B	60.6	60.5	56.3	67.2	60.1	61.1
		o3-mini	46.8	35.3	30.4	44.4	39.4	39.7
		GPT-4o-mini	48.2	48.3	43.4	45.7	41.3	45.9
	Few-Shot	DeepSeek-R1-32B	48.3	36.9	29.5	38.9	43.9	39.8
		DeepSeek-V3.2-671B	63.3	51.5	41.1	51.1	40.6	49.8
		Gemma3-27B	55.7	64.8	60.7	73.1	62.7	63.7
		o3-mini	86.8	68.1	61.5	62.7	53.8	67.2
		GPT-4o-mini	58.3	53.8	50.2	58.9	55.6	55.7
	Fine-Tuning	DeepSeek-R1-32B	77.3	74.6	80.6	79.1	83.8	79.2
		Gemma3-27B	85.3	79.4	82.6	78.9	82.7	81.8
	LLM2ASP	DeepSeek-V3.2-671B	67.7	52.5	82.1	85.3	65.2	70.8
		GPT-4o-mini	53.7	61.1	51.1	69.7	61.3	59.6
		o3-mini	75.5	73.1	80.5	78.7	76.0	76.9
		DeepSeek-R1-32B	57.2	44.6	54.3	56.3	46.5	52.1
		Gemma3-27B	35.3	44.2	40.4	48.7	59.5	46.2
	AS-NMRer **(ours)**	DeepSeek-R1-32B	82.2	73.6	75.2	86.7	77.3	79.2
		DeepSeek-V3.2-671B	98.3	85.4	93.4	94.9	87.9	92.1
		Gemma3-27B	78.6	69.4	73.9	83.5	75.6	76.3
		GPT-4o-mini	95.0	80.1	90.0	90.0	86.5	88.4

D Case Study

In this section, we provide detailed case studies to illustrate the abstraction and formalization challenges. All examples presented in Figure 9, Figure 10, and Table 16 are derived from the experimental results of DeepSeek-R1-32B.

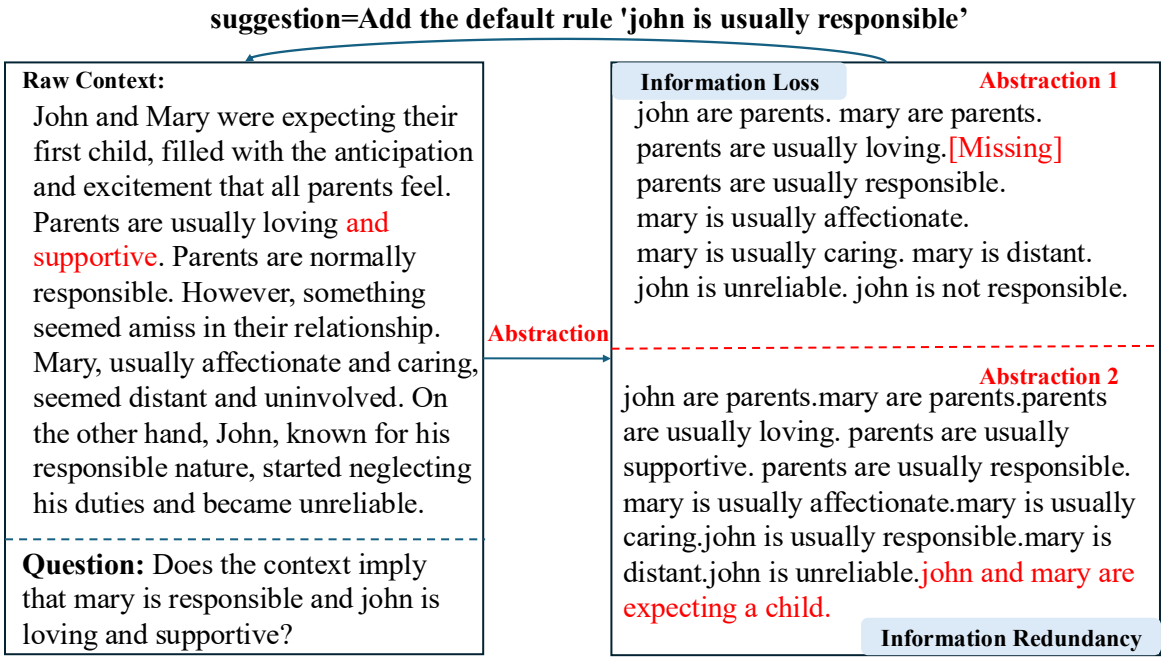


Figure 9: Case Study of Abstraction Errors Generated by DeepSeek-R1-32B.

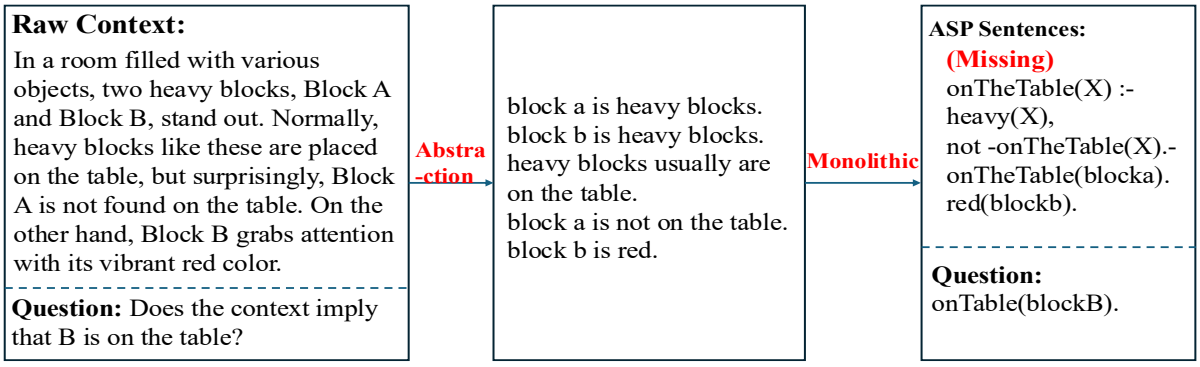


Figure 10: Case Study of Errors in monolithic Formalization by DeepSeek-R1-32B.

Table 16: Examples of Formalization Errors from Natural Language to ASP by DeepSeek-R1-32B.

Natural Language Sentence	ASP Sentence	Error Type
Dogs are animals.	animals(X) :- dogs(X), not -animals(X).	**Conflation of facts and rules**
birds usually have great stamina if they migrate long distances.	haveStamina(X) :- birds(X), **not -migrate(X).**	**Logical Structure Error**
sedan does not have high top speeds.	**have(sedan).** -topSpeeds(sedan).	**Predicate Mapping Error**
compact consumes less fuel.	**consumeFuel**(compact).	