From Hypothesis to Premises: LLM-based Backward Logical Reasoning with Selective Symbolic Translation

Qingchuan Li[1], Mingyue Cheng[1*], Zirui Liu[1], Daoyu Wang[1], Yuting Zeng[1], Tongxuan Liu[1, 2]

[1]University of Science and Technology of China,
[2]JD.com
{chouli, liuzirui, wdy030428, yuting_zeng, tongxuan.ltx}@mail.ustc.edu.cn, mycheng@ustc.edu.cn

Abstract

Logical reasoning is a core challenge in natural language understanding and a fundamental capability of artificial intelligence, underpinning scientific discovery, mathematical theorem proving, and complex decision-making. Despite the remarkable progress of large language models (LLMs), most current approaches still rely on forward reasoning paradigms, generating step-by-step rationales from premises to conclusions. However, such methods often suffer from redundant inference paths, hallucinated steps, and semantic drift, resulting in inefficient and unreliable reasoning. In this paper, we propose a novel framework, Hypothesis-driven Backward Logical Reasoning (HBLR). The core idea is to integrate confidence-aware symbolic translation with hypothesis-driven backward reasoning. In the translation phase, only high-confidence spans are converted into logical form, such as First-Order Logic (FOL), while uncertain content remains in natural language. A translation reflection module further ensures semantic fidelity by evaluating symbolic outputs and reverting lossy ones back to text when necessary. In the reasoning phase, HBLR simulates human deductive thinking by assuming the conclusion is true and recursively verifying its premises. A reasoning reflection module further identifies and corrects flawed inference steps, enhancing logical coherence. Extensive experiments on five reasoning benchmarks demonstrate that HBLR consistently outperforms strong baselines in both accuracy and efficiency.

Code — https://github.com/wufeiwuwoshihua/HBLR

1 Introduction

Logical reasoning lies at the heart of artificial intelligence (AI), playing a central role in scientific discovery, mathematical theorem proving, and complex decision-making (Bronkhorst et al. 2020). In natural language understanding, logical reasoning refers to the process of drawing valid conclusions from a set of textual premises, often requiring models to perform multi-step, structured inference (Nunes 2012; Bronkhorst et al. 2020). Despite its importance, reasoning over natural language remains a formidable challenge due to linguistic ambiguity, implicit knowledge, and the need for robust and reliable compositional generalization (Ye et al. 2023; Pan et al. 2023; Li et al. 2025; Luo et al. 2025). Effective logical reasoning requires not only the understanding of individual statements, but also the ability to model their interrelations through valid and principled logical transformations (Smith 2003).

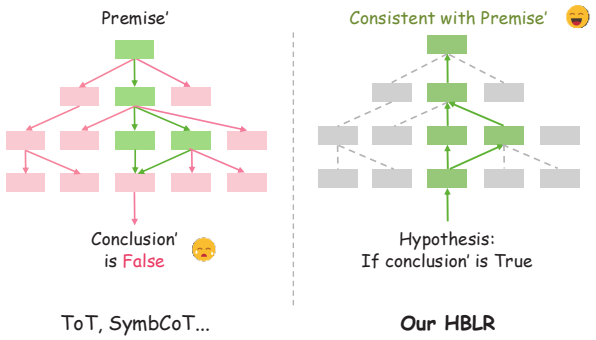

Figure 1: Conceptual comparison of reasoning paradigms. HBLR adopts a hypothesis-driven backward reasoning strategy with selective symbolic translation, enhancing precision and effectiveness.

Over the past decades, a wide range of approaches have been proposed to tackle this challenge. Early methods relied extensively on symbolic representations and rule-based inference engines, such as first-order logic (FOL) systems and logic programming frameworks. Subsequently, constraint solvers (e.g., GeCode (Schulte, Lagerkvist, and Tack 2006), PyKE (Frederiksen 2008)) and automated theorem provers (e.g., Prover9 (McCune 2009), Z3 (Microsoft Research 2015)) were introduced to strengthen formal inference capabilities. More recently, the emergence of neural-symbolic reasoning has enabled the integration of statistical learning with structured logical operations, bridging the gap between expressivity and scalability.

In particular, the emergence of large language models (LLMs) (Patel et al. 2022; Hahn et al. 2022) has greatly propelled the development of natural language reasoning. A series of studies, such as CoT (Chain-of-Thought) prompting (Wei et al. 2022), Tree of Thoughts (Yao et al. 2023a), LINC (Olausson et al. 2023), and Logic-LM (Pan et al. 2023), have demonstrated the potential of LLMs in perform-

*Corresponding Author.

ing complex deductive reasoning tasks. These methods either directly leverage LLMs to generate reasoning chains in natural language, or translate natural language into symbolic forms and invoke external solvers. Some recent works, such as SymbCoT (Xu et al. 2024), further integrate symbolic representations with LLMs to enhance reasoning faithfulness and interpretability.

Despite these promising advances, current methods still face several key limitations. First, most approaches adopt a forward reasoning paradigm, generating reasoning steps from premises toward conclusions. However, this paradigm often suffers from redundant reasoning paths, hallucinated intermediate steps, and goal deviation, resulting in unreliable or inefficient inference (Kazemi et al. 2023). Second, symbolic translation is often applied globally to the entire input, despite the fact that large language models may only be confident about specific parts of the text. Over-reliance on imperfect translation may introduce logical errors or semantic loss (Li et al. 2025). Finally, reasoning steps are seldom verified or revised once generated, making the process prone to compounding mistakes.

To address these challenges, we propose a novel framework, Hypothesis-driven Backward Logical Reasoning (HBLR), shown in Figure 1. The core idea is to reframe logical reasoning as a hybrid process that (1) selectively translates high-confidence spans of natural language into formal logic and (2) simulates human-like deductive reasoning via backward chaining. Specifically, HBLR first performs confidence-aware symbolic translation into FOL while retaining uncertain parts in natural language. A translation reflection module further ensures semantic fidelity by reverting lossy translations. Then, HBLR assumes the conclusion to be true and recursively verifies its logical support in a hypothesis-driven, backward fashion. A reasoning reflection module is also introduced to detect and correct flawed inference steps, ensuring logical coherence and robustness.

- We propose HBLR, a novel logical reasoning framework based on hypothesis-driven backward chaining, which simulates human deductive thinking by assuming the conclusion and verifying its premises in reverse, addressing the inefficiencies of forward reasoning paradigms.
- We introduce a selective symbolic translation strategy that converts high-confidence spans into logic form while retaining uncertain parts in natural language, thereby balancing formal precision and semantic flexibility.
- We conduct comprehensive experiments on five benchmark datasets and show that HBLR achieves superior performance in reasoning accuracy and efficiency.

2 Related Work

Prompt-based LLM Reasoning. Logical reasoning—deriving correct conclusions from given premises—is a core ability for large language models (LLMs) (Mondorf and Plank 2024; Sun et al. 2023; Qiao et al. 2023). Prompting provides a direct way to activate this ability. For complex tasks, step-by-step prompting has become widely used (Besta et al. 2024b; Wang et al. 2023). Chain-of-Thought (CoT) prompting (Wei et al. 2022) guides LLMs to generate intermediate steps, while Tree-of-Thought (ToT) (Yao et al. 2023a) and Graph-of-Thought (GoT) (Besta et al. 2024a) enable exploration of multiple reasoning paths. However, these methods still struggle in non-mathematical tasks and when exemplar and target question complexity differ (Sprague et al. 2024).

Another major direction is question decomposition (Zhang et al. 2023; Yao et al. 2023b; Kazemi et al. 2023). Least-to-most prompting (Zhou et al. 2023; Wang, Cheng, and Liu 2025; Jiang et al. 2025) solves problems by first handling simpler sub-questions, and divide-and-conquer strategies (Cui et al. 2023; Zhang et al. 2024; Cheng et al. 2025) further improve consistency and logical accuracy. Backward chaining (Kazemi et al. 2023) reduces the search space by breaking tasks into solvable sub-modules. Despite these advances, prompt-based reasoning still faces high computational cost and unstable performance (Yao et al. 2023a).

Symbolic-based LLM Reasoning. Symbolic reasoning uses formal logic symbols and expressions to enable consistent and precise inference, helping reduce issues in prompt-based reasoning such as inconsistency and sensitivity to premise order (Chen et al. 2024; Bao et al. 2024a).

One main line of work improves LLM reasoning by introducing explicit logical representations (Wang et al. 2022; Wan et al. 2024). Wang et al. (2022) mapped natural language to logical forms to produce more aligned and reliable answers, while Bao et al. (2024b) used structured semantic graphs for logic-driven data augmentation across diverse tasks. Xu et al. (2024) proposed a two-stage method that first generates logical forms and then uses them to guide downstream reasoning and planning more effectively. Another direction employs symbolic solvers to infer over LLM-generated logic (Olausson et al. 2023; Pan et al. 2023; Ye et al. 2023). Solver choice, such as SAT (Ye et al. 2023) or first-order logic (Pan et al. 2023), directly affects accuracy and overall generalization quality. SymBa (Lee and Hwang 2024) further integrates classical SLD resolution with LLMs, providing symbolically guided chain-of-thought reasoning that significantly improves both performance and interpretability.

Despite these strengths, symbolic methods remain limited by the translation step from natural language to logic: errors or missing information in this process can weaken downstream reasoning (Pan et al. 2023).

3 Preliminaries

3.1 Problem Definition

We study the task of natural language logical reasoning, where the input consists of a set of premises $\mathcal{P}$ and a target conclusion $\mathcal{C}$. The goal is to determine whether the conclusion can be logically inferred from the premises, denoted as $\mathcal{P} \models \mathcal{C}$, meaning that $\mathcal{C}$ is true in all possible interpretations where $\mathcal{P}$ holds. In our setting, both $\mathcal{P}$ and $\mathcal{C}$ may contain a mixture of formal logic expressions and natural language statements. After symbolic translation, the inputs are reformulated as a hybrid premise set $\mathcal{P}'$ and a hybrid conclusion $\mathcal{C}'$. The final goal is to determine whether $\mathcal{P}' \models \mathcal{C}'$.

Dataset	Solver	GPT-4	DeepSeek-V3
FOLIO	Prover9	0.7383	0.7245
ProofWriter	PyKE	0.8338	0.8211
ProntoQA	PyKE	0.9050	0.8783
Deduction	constraint	0.9599	0.8963
AR-LSAT	Z3	0.4304	0.4041

Table 1: Datasets, their associated symbolic solvers and the translation accuracy of GPT-4 and DeepSeek-V3. Note: "Deduction" denotes the LogicalDeduction dataset, and "constraint" refers to the python-constraint solver.

3.2 Empirical Exploration

Translation Module. We first evaluate the SymbCoT (Xu et al. 2024) translation module to measure how well LLMs convert natural language into formal logic. We keep the original SymbCoT translator but replace its LLM-based reasoning with symbolic solvers, using solver outputs as proxies for translation accuracy within well-defined logical domains. Experiments are run on five datasets (Saparov and He 2022; Tafjord, Mishra, and Clark 2020; Ghazal et al. 2013; Han et al. 2022; Zhong et al. 2021) and four symbolic solvers, with GPT-4 and DeepSeek-V3 as base models, following the setup of Logic-LM (Pan et al. 2023). As shown in Table 1, LLMs achieve high translation accuracy on synthetic datasets like ProntoQA and ProofWriter, which use clear logical templates. However, performance drops sharply on manually curated datasets such as FOLIO and AR-LSAT, where natural language is more varied and logical structure is often implicit. These results show that LLMs handle explicit logical inputs well but struggle when logic is expressed indirectly. This motivates a selective translation strategy that converts only high-confidence spans to logic and keeps ambiguous parts in natural language to reduce semantic drift.

To evaluate how translation errors affect SymbCoT's LLM-based reasoning module, we examine two datasets with low translation accuracy, FOLIO and AR-LSAT, and measure the share of reasoning failures linked to translation errors. As shown in Figure 2a, these errors are a major source of SymbCoT's reasoning failures on both datasets.

Reasoning Module. We further analyze the efficiency of SymbCoT's reasoning strategy. Although SymbCoT employs a plan-and-solve approach that generates reasoning chains from premises to conclusions, we observe that its LLM-based reasoning often lacks goal-directedness—frequently invoking irrelevant premises and introducing redundant steps. To quantify this inefficiency, we extract essential reasoning paths from correct GPT-4 predictions by pruning extraneous steps while retaining all necessary inferences. We then compute the ratio of tokens in the essential paths relative to the original plans (Figure 2b). The results show that forward reasoning tends to overuse available premises, resulting in unnecessarily verbose inference trajectories. These findings highlight the need for a goal-driven alternative that begins with the hypothesis and selectively identifies minimal supporting premises, thereby improving reasoning efficiency and precision.

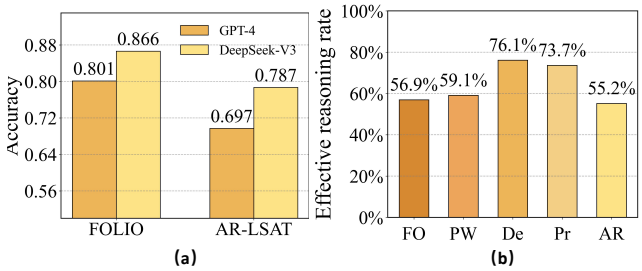

Figure 2: (a) Translation Error Rate Among SymbCoT Failure Cases in FOLIO and AR-LSAT (b) Ratio of essential tokens retained after pruning redundant steps from GPT-4-generated reasoning plans.

4 Methodology

4.1 Overview of the HBLR Framework

To overcome the limitations of existing methods in symbolic translation and reasoning, we propose Hypothesis-driven Backward Logical Reasoning (HBLR) (Figure 3). HBLR improves translation reliability and strengthens goal-directed reasoning through two components: the Confidence-aware Symbolic Translation Module (CSTM) and the Hypothesis-based Backward Reasoning Module (HBRM). CSTM selectively converts natural language into formal logic based on structural and semantic confidence, while HBRM mirrors human deductive reasoning by starting from the conclusion and working backward to find the minimal supporting premises.

4.2 Confidence-aware Symbolic Translation

To reduce errors from full symbolic translation, we introduce the Confidence-aware Symbolic Translation Module (CSTM), which selectively converts structurally sound and semantically clear statements into logical forms while keeping the rest in natural language. This hybrid strategy balances the rigor of symbolic logic with the expressiveness and robustness of natural language. CSTM uses two mechanisms—a structural rule-based pre-checker and a semantic consistency verifier—to ensure both syntactic and semantic requirements are met before translation.

Structural Filter. To determine whether a sentence s is suitable for symbolic translation, we apply a structural filter that detects logic-compatible patterns in natural language. Sentences that pass this filter are translated into a logical form ϕ, while those that do not are retained in natural language to preserve semantic fidelity.

We define logic-compatible patterns as a specific class of linguistic structures that are suitable for formal representation. A sentence is considered logic-compatible if it conforms to these predefined patterns, which include explicit predicate-argument constructions, the presence of logical connectives or quantifiers such as "if," "then," "and," or "or," and canonical formulations commonly found in propositional or first-order logic. These structural characteristics ensure that the sentence can be reliably and meaningfully translated into symbolic form.

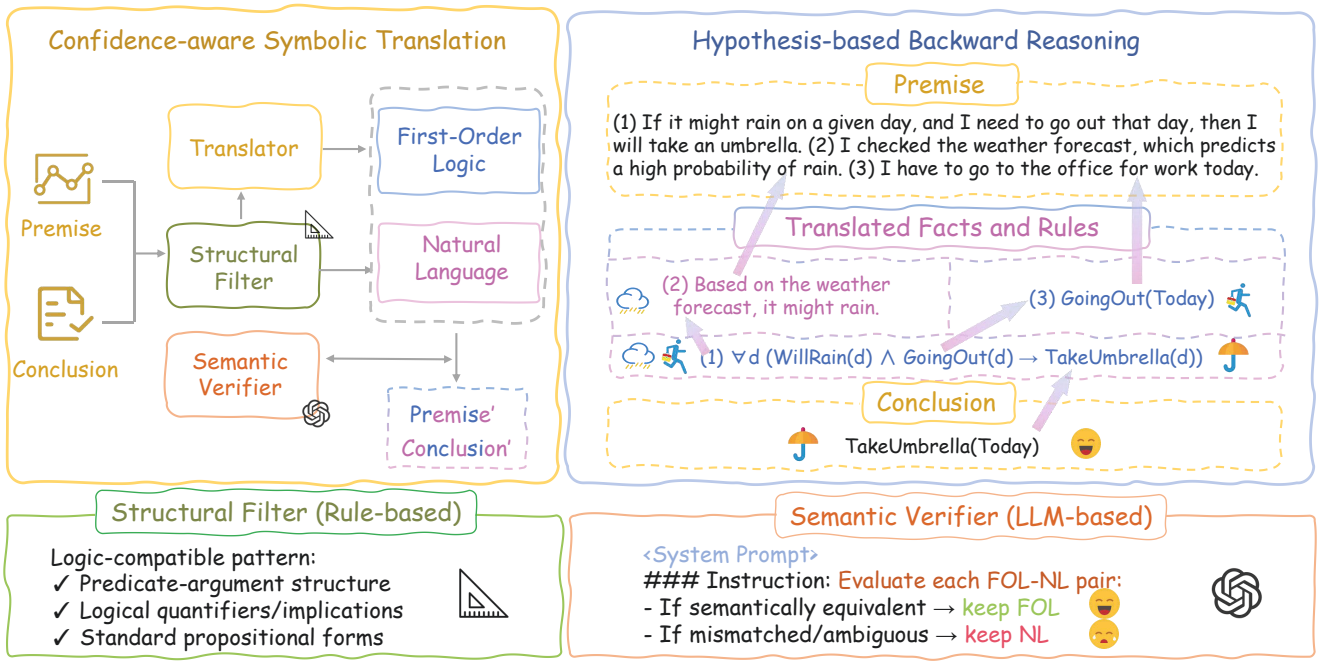

Figure 3: Overview of the Hypothesis-driven Backward Logical Reasoning (HBLR) framework. Structural Filter and Semantic Verifier below illustrate the internal mechanisms of the Confidence-aware Symbolic Translation module.

Semantic Verifier. To ensure that symbolic translations faithfully preserve the meaning of the original sentence, we introduce a semantic verifier. For each candidate logical form ϕ generated from a natural language sentence s, the verifier assesses whether ϕ semantically aligns with the intent of s. This check is performed using LLM-based prompting with carefully selected few-shot exemplars.

The verifier is designed as a conservative safeguard rather than a perfect semantic judge. Because an incorrect logical form is more harmful than retaining the original natural language, the system adopts a strict acceptance rule: ϕ is accepted only when it can be confidently verified as fully consistent with s. Any logical form with uncertainty or insufficient semantic evidence is rejected, and the corresponding s is preserved in natural language. This conservative strategy reduces semantic drift and ensures that symbolic representations remain faithful and reliable, avoiding the introduction of incorrect logic when translation is uncertain.

Hybrid Representation. The final output of CSTM consists of two complementary components: a set of high-confidence logical expressions $\{\phi_i\}_{i=1}^{m}$ and a set of natural language statements $\{s_j\}_{j=1}^{n}$, which are retained due to insufficient structural or semantic confidence for symbolic translation. These elements are integrated into a unified hybrid context that serves as the input for downstream reasoning. Formally, we define the hybrid context as:

$$\text{HybridContext} = \mathcal{P}' \cup \mathcal{C}' \tag{1}$$

where $\mathcal{P}' = \{\phi_i\} \cup \{s_j\}$ denotes the premise set, comprising both logical expressions and natural language statements, and $\mathcal{C}' = \{\phi_m\} \cup \{s_n\}$ represents the conclusion, expressed in logical or textual form depending on translation confidence. This hybrid representation enables flexible yet faithful reasoning by combining the precision of formal logic with the semantic richness of natural language.

4.3 Hypothesis-based Backward Reasoning

To address the inefficiency and lack of clear goal orientation inherent in conventional forward reasoning approaches, we introduce the Hypothesis-based Backward Reasoning Module (HBRM). Inspired by the classical hypothetical-deductive paradigm, HBRM initiates reasoning by assuming the conclusion to be true and recursively verifying the premises that would logically support it.

Backward Reasoning with Hypothesis. The input to HBRM is defined as a tuple $(\mathcal{P}', \mathcal{H})$, where $\mathcal{P}'$ denotes the hybrid premise set, containing both logical expressions and natural language statements, and $\mathcal{H}$ is the hypothesis explicitly asserting that the target conclusion $\mathcal{C}$ holds true. Reasoning unfolds by constructing a sequence of intermediate hypotheses $\{H_t\}$, each of which is iteratively evaluated through deductive inference based on the information encoded in P. This backward chaining process is formally summarized in Algorithm 1.

Stopping Criteria. The backward reasoning process terminates when any of the following conditions is met: (i) the current hypothesis H_t is directly entailed by a premise in P, indicating a successful and complete proof; (ii) H_t contradicts an existing premise in P, resulting in a logical refutation; or (iii) the maximum number of reasoning steps is reached without sufficient supporting evidence, in which case the outcome is deemed logically inconclusive.

Algorithm 1: Backward Reasoning with Hypothesis

Require: Hypothesis $\mathcal{H}$, Hybrid Premises $\mathcal{P}'$
Ensure: Validity status of $\mathcal{H}$

while steps $\leq k$ **do**
  $Z \leftarrow \text{REASONING}(\mathcal{P}', \mathcal{H})$
  if Z contradicts $\mathcal{P}'$ or $\mathcal{H}$ **then**
    return False
  else if Z supports $\mathcal{P}'$ or $\mathcal{H}$ **then**
    return True
  else
    $\mathcal{H} \leftarrow Z$
  end if
end while
return Unknown

Dimension	CoT	Logic-LM	SymbCoT	HBLR
Use of Solver	✗	✓	✗	✗
Translation	None	Full	Full	Selective
Strategy	Forward	Solver	Forward	Backward
Redundancy	High	–	High	Low
Interpretability	High	Low	High	High
Verification	No	No	Yes	Yes

Table 2: Comparison of HBLR and baseline methods. HBLR does not rely on external solvers and achieves low reasoning redundancy and high interpretability through backward reasoning with confidence-aware symbolic translation.

Verification Mechanism. To ensure logical soundness and semantic fidelity, HBRM incorporates a verification mechanism that evaluates the validity of each reasoning step. If a step contains logical errors, semantic inconsistencies, or unsupported inferences, the system reconstructs a revised reasoning path; otherwise, the original path is preserved. By guiding the inference process in a backward manner, from the conclusion toward its underlying premises, HBRM enhances goal alignment, reduces redundancy, and produces more efficient and interpretable reasoning trajectories.

4.4 Discussions

As shown in Table 2, HBLR addresses several limitations of existing reasoning frameworks. Unlike Logic-LM and SymbCoT, which fully translate all inputs into symbolic logic, HBLR performs selective symbolic translation, converting only high-confidence spans while retaining natural language when appropriate. This reduces translation errors and better preserves contextual semantics. In contrast to CoT and SymbCoT, which rely on forward reasoning, HBLR employs a hypothesis-driven backward reasoning mechanism that initiates inference from the conclusion and recursively identifies minimal supporting premises. This goal-oriented approach significantly reduces redundancy in reasoning paths. HBLR also incorporates a step-wise verification mechanism that evaluates each inference step, improving both reliability and interpretability. Overall, HBLR combines the precision of symbolic methods with the flexibility of neural reasoning, while mitigating the weaknesses of both.

5 Experiments

5.1 Experimental Settings

Evaluation Models. Experiments are conducted using four representative LLMs: the relatively less capable **GPT-3.5-Turbo** (OpenAI 2023); the more advanced **GPT-4** (Achiam et al. 2023) and **DeepSeek-V3** (Liu et al. 2024); and **DeepSeek-R1** (Guo et al. 2025), currently one of the most reasoning-capable models available.

Evaluation Datasets. Our evaluation spans five widely-used logical reasoning benchmarks: **ProntoQA** (Saparov and He 2022), **ProofWriter** (Tafjord, Mishra, and Clark 2020), **FOLIO** (Han et al. 2022), **LogicalDeduction** (Ghazal et al. 2013), and **AR-LSAT** (Zhong et al. 2021). These datasets vary in symbolic formalisms and present diverse challenges across deductive reasoning scenarios. We adopt **accuracy** as the primary evaluation metric, measuring the correctness of multiple-choice answers.

Symbolic Formalisms. For ProntoQA, ProofWriter, and FOLIO, we use first-order logic (FOL) as the primary underlying symbolic structure. To assess the broader generalizability of our approach, we additionally evaluate on constraint optimization (CO) symbolic representations in LogicalDeduction and AR-LSAT.

Baselines. We compare HBLR against several strong baselines employing distinct strategies: (1) **Direct**, which uses LLMs to directly answer questions without intermediate reasoning; (2) **CoT** (Wei et al. 2022), which applies chain-of-thought prompting to elicit step-by-step reasoning; (3) **Logic-LM** (Pan et al. 2023), which translates problems into logic and invokes symbolic solvers; and (4) **SymbCoT** (Xu et al. 2024), which combines symbolic translation with LLM-based reasoning.

5.2 Overall Evaluation Results

As shown in Table 3, HBLR consistently outperforms all baselines across five benchmark datasets. On GPT-4, it achieves gains of up to 36.74%, 22.58%, 10.39%, and 10.07% over Direct, CoT, Logic-LM, and SymbCoT, respectively. Similar trends are observed on GPT-3.5-Turbo, DeepSeek-V3, and DeepSeek-R1, confirming the robustness of HBLR across models with varying reasoning capabilities. Compared to Logic-LM and SymbCoT, the improvements highlight the effectiveness of HBLR's partial symbolic translation and backward reasoning in balancing logical formality with LLM-native reasoning. Notably, HBLR outperforms Logic-LM by an average of 14.61% on DeepSeek-R1. An exception occurs on the LogicalDeduction task with GPT-3.5-Turbo, where Logic-LM slightly outperforms HBLR. This is due to GPT-3.5-Turbo's limited reasoning ability—while it handles translation reasonably well, it struggles with complex inference. Logic-LM avoids this bottleneck by delegating reasoning to an external solver.

Another notable trend is that Logic-LM and SymbCoT often rank second-best on GPT-4, GPT-3.5-Turbo, and DeepSeek-V3, while CoT ranks second on DeepSeek-R1.

Dataset	GPT-4					GPT-3.5-Turbo				
	Direct	CoT	Logic	SymbCoT	HBLR	Direct	CoT	Logic	SymbCoT	HBLR
ProntoQA	77.40	94.79	90.50	97.16	**99.36**	46.04	67.80	67.21	71.95	**75.58**
ProofWriter	52.67	68.11	83.38	79.34	**89.41**	36.53	49.17	58.62	59.03	**63.24**
FOLIO	70.61	72.37	73.83	78.19	**84.22**	45.09	57.35	48.83	57.84	**59.12**
Deduction	71.33	75.25	95.99	93.00	**97.83**	39.15	43.67	**69.06**	45.85	49.77
AR-LSAT	34.43	35.06	43.04	37.19	**44.67**	20.34	21.31	25.15	21.59	**27.22**

Dataset	DeepSeek-V3					DeepSeek-R1				
	Direct	CoT	Logic	SymbCoT	HBLR	Direct	CoT	Logic	SymbCoT	HBLR
ProntoQA	74.93	97.67	87.83	98.43	**99.55**	97.28	99.57	84.21	98.47	**99.72**
ProofWriter	54.46	56.84	82.11	84.15	**89.48**	82.48	86.27	84.35	88.34	**92.32**
FOLIO	68.36	74.64	72.45	77.78	**85.22**	88.13	92.97	72.51	84.46	**95.60**
Deduction	67.25	72.67	90.63	89.91	**92.63**	76.17	86.45	99.50	99.03	**99.61**
AR-LSAT	36.21	42.50	40.41	44.52	**46.69**	60.40	76.91	62.10	70.87	**86.47**

Table 3: Performance Comparison between HBLR and Baselines on Logical Reasoning Datasets. The second-best score is underlined and **bold** one is the best.

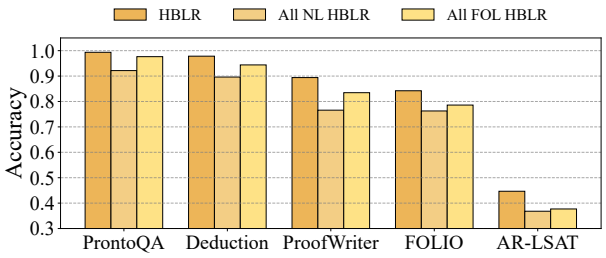

Figure 4: Comparison of selective translation (HBLR) with its two module variants: All-NL and All-FOL. HBLR consistently achieves higher accuracy across all datasets by balancing natural language and formal logic.

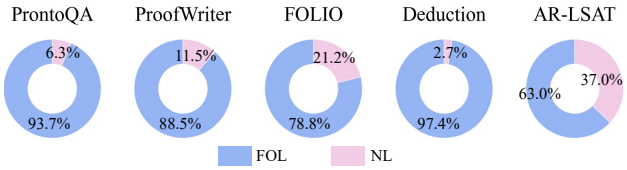

Figure 5: Proportion of natural language retained by the Translation module across datasets. Retention varies by task, reflecting differences in translation confidence.

This suggests that as model reasoning improves, fully symbolic approaches may introduce noise that offsets their benefits. HBLR remains robust by minimizing such noise while leveraging stronger reasoning capacity.

5.3 Impact of Translation Module

To assess the effectiveness of our selective translation strategy, we compare it with two ablated variants of the HBLR translation module: **All-NL**, which performs no translation and retains all inputs in natural language; and **All-FOL**, which fully translates all inputs into formal logic. As shown in Figure 4, our selective strategy consistently outperforms both variants across all five benchmarks.

On average, the selective strategy yields a 7.2% improvement over All-NL, showing that natural language alone lacks sufficient structure for accurate reasoning. It also outperforms All-FOL by 4.7%, indicating that full formalization can introduce translation errors or add unnecessary rigidity. The gains are especially large on **ProofWriter** (+12.83%) and **LogicalDeduction** (+8.21%), both requiring multi-step reasoning. In contrast, **ProntoQA** shows a small decline (-1.72%) compared to All-FOL, likely because its highly structured format benefits from full logical conversion. Overall, these results show that the selective strategy effectively combines the semantic richness of natural language with the structural precision of formal logic, allowing LLMs to adapt to task demands and input characteristics.

We analyze the proportion of natural language retained by HBLR's translation module using GPT-4 (Figure 5). On average, only 15.74% of the input remains in natural language, with the lowest retention on **LogicalDeduction** (2.65%), showing that most inputs are confidently translated into formal logic. This matches our confidence-aware design: natural language is kept only for uncertain segments, reducing ambiguity while limiting translation errors. Higher retention on **AR-LSAT** (37.03%) and **FOLIO** (21.17%) aligns with Section 3.2, where we show these datasets have greater translation uncertainty. These results confirm that the selective strategy adjusts to dataset-specific reliability and preserves robustness when symbolic translation is less certain.

5.4 Impact of Reasoning Module

To evaluate the contribution of our HBRM strategy, we compare it against a forward reasoning variant that adopts CoT prompting, while keeping the verification module unchanged. As shown in Table 4, HBLR consistently outperforms CoT across all five datasets.

On average, backward reasoning yields a 5.69% improvement. This gain can be explained as follows: forward reasoning tends to expand redundant branches during the search

Dataset	HBLR (%)	Fwd Var. (%)	Δ
ProntoQA	99.36	95.41	+3.95
ProofWriter	89.41	81.24	+8.17
FOLIO	84.22	77.58	+6.64
Deduction	97.83	93.34	+4.49
AR-LSAT	44.07	38.86	+5.21

Table 4: Comparison between HBLR and its forward reasoning variant (Fwd Var.) across five datasets. HBLR consistently achieves higher accuracy.

Dataset	SymbCoT (%)	HBLR (%)	Δ
ProntoQA	73.65	95.58	21.93
ProofWriter	59.10	81.60	22.50
FOLIO	56.94	73.29	16.35
Deduction	76.13	87.46	11.33
AR-LSAT	55.20	66.08	10.88

Table 5: Effective reasoning rates of SymbCoT and HBLR. HBLR shows consistent improvements across datasets.

and accumulate translation errors, whereas HBLR starts from the hypothesis and traces back the key premises that support it, which greatly reduces the search space and limits its error propagation. As a result, HBLR produces shorter and more accurate reasoning chains. This advantage is especially clear on ProofWriter (+8.17%), where long reasoning chains make forward reasoning prone to early errors that amplify later; HBLR's backward strategy focuses on essential premises more quickly and keeps the reasoning process stable. On AR-LSAT, which involves complex structures and diverse semantics, HBLR still achieves a +5.21% gain, demonstrating stronger robustness and generalization. Overall, hypothesis-driven backward reasoning effectively controls the direction of inference, reduces noise and redundancy, and leads to more reliable reasoning across tasks.

Following the methodology in Section 3.2, we compute the proportion of effective reasoning steps produced by HBLR using GPT-4 and compare it with the previously reported SymbCoT results. HBLR substantially increases the share of effective reasoning, with improvements up to 22.50%. It reaches over 70% effectiveness on all datasets except **AR-LSAT**, which contains complex reasoning scenarios and broad semantic variation. These results show that HBLR offers strong efficiency across diverse tasks.

5.5 Performance Across Reasoning Depths

Having established the overall superiority of our method in direct comparisons, we further analyze its performance across different levels of reasoning depth. Intuitively, greater reasoning depth corresponds to higher problem complexity. As illustrated in Figure 6, the performance gap between HBLR and CoT widens as reasoning depth increases, highlighting HBLR's advantage in handling more challenging reasoning scenarios. Notably, even at a depth of 5—the most complex setting in our evaluation—HBLR continues to achieve the highest performance among all methods.

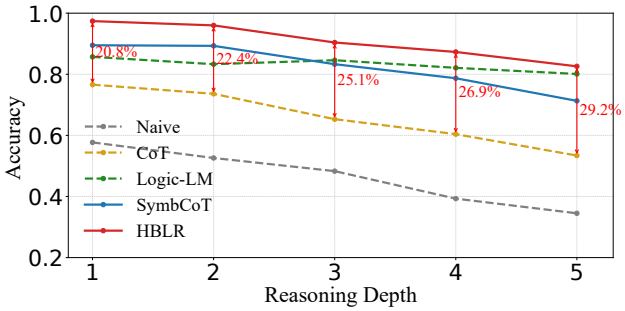

Figure 6: The effect of reasoning depth with GPT-4 on ProofWriter. The red double-headed arrow indicates our improvements over vanilla CoT.

Method	DeepSeek-V3	DeepSeek-R1	Δ
Logic-LM	74.69	80.93	+6.24
SymbCoT	78.96	88.23	+9.27
HBLR	82.31	94.34	+12.03

Table 6: Performance improvements from DeepSeek-V3 to DeepSeek-R1 on different methods.

5.6 Impact of Stronger Backbone Models

We compare the performance gains achieved when upgrading from DeepSeek-V3 to the more reasoning-capable DeepSeek-R1 model, as shown in Table 6. HBLR shows larger improvements than Logic-LM and SymbCoT, primarily because these translation-heavy methods benefit less from an upgrade that enhances reasoning ability rather than translation quality. In contrast, HBLR's confidence-aware translation mitigates this bottleneck, allowing it to better exploit the improved reasoning capacity of DeepSeek-R1. HBLR's overall gain is smaller than that of Direct and CoT, as its baseline performance on DeepSeek-V3 is already strong, especially on ProntoQA, where accuracy is near saturation. Methods with lower initial performance naturally show larger absolute improvements under the upgrade. These results highlight HBLR's adaptability and its strong potential in settings where model-side reasoning improvements are increasingly central.

6 Conclusion

This study presents HBLR, a hypothesis-driven backward reasoning framework for natural language logical reasoning. HBLR combines confidence-aware partial symbolic translation with human-inspired backward chaining to enhance precision and interpretability. It translates only high-confidence spans into formal logic based on structural and semantic cues, leaving uncertain content in natural language to balance symbolic rigor and flexibility. Reasoning begins from the hypothesis and works backward to validate supporting premises, aided by a reflection mechanism that enforces step-wise consistency. Across five benchmarks with diverse symbolic settings, HBLR consistently surpasses prior state-of-the-art methods in both accuracy and effectiveness.

Acknowledgements

This research was supported by grants from the National Natural Science Foundation of China (No. 62502486), the grants of Provincial Natural Science Foundation of Anhui Province (No. 2408085QF193), USTC ResearchFunds of the DoubleFirst-Class Initiative (No. YD2150002501), the Fundamental Research Funds for the Central Universities of China (No. WK2150110032).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bao, G.; Zhang, H.; Yang, L.; Wang, C.; and Zhang, Y. 2024a. LLMs with Chain-of-Thought Are Non-Causal Reasoners. *CoRR*, abs/2402.16048. ArXiv: 2402.16048.

Bao, Q.; Peng, A.; Deng, Z.; Zhong, W.; Gendron, G.; Pistotti, T.; Tan, N.; Young, N.; Chen, Y.; Zhu, Y.; Denny, P.; Witbrock, M.; and Liu, J. 2024b. Abstract Meaning Representation-Based Logic-Driven Data Augmentation for Logical Reasoning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 5914–5934. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; and Hoefler, T. 2024a. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16): 17682–17690. Number: 16.

Besta, M.; Memedi, F.; Zhang, Z.; Gerstenberger, R.; Blach, N.; Nyczyk, P.; Copik, M.; Kwasniewski, G.; Müller, J.; Gianinazzi, L.; Kubicek, A.; Niewiadomski, H.; Mutlu, O.; and Hoefler, T. 2024b. Topologies of Reasoning: Demystifying Chains, Trees, and Graphs of Thoughts. *CoRR*, abs/2401.14295. ArXiv: 2401.14295.

Bronkhorst, H.; Roorda, G.; Suhre, C.; and Goedhart, M. 2020. Logical reasoning in formal and everyday reasoning tasks. *International Journal of Science and Mathematics Education*, 18: 1673–1694.

Chen, X.; Chi, R. A.; Wang, X.; and Zhou, D. 2024. Premise Order Matters in Reasoning with Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Cheng, M.; Mao, Q.; Liu, Q.; Zhou, Y.; Li, Y.; Wang, J.; Lin, J.; Cao, J.; and Chen, E. 2025. A survey on table mining with large language models: Challenges, advancements and prospects. *Authorea Preprints*.

Cui, W.; Zhang, J.; Li, Z.; Lopez, D.; Das, K.; Malin, B.; and Kumar, S. 2023. A Divide-Conquer-Reasoning Approach to Consistency Evaluation and Improvement in Blackbox Large Language Models. In *Socially Responsible Language Modelling Research*.

Frederiksen, B. 2008. PyKE: Python Knowledge Engine. A knowledge-based inference engine in Python, accessed 15 May 2025.

Ghazal, A.; Rabl, T.; Hu, M.; Raab, F.; Poess, M.; Crolotte, A.; and Jacobsen, H.-A. 2013. Bigbench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, 1197–1208.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hahn, C.; Schmitt, F.; Tillman, J. J.; Metzger, N.; Siber, J.; and Finkbeiner, B. 2022. Formal specifications from natural language. *arXiv preprint arXiv:2206.01962*.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Jiang, C.; Cheng, M.; Tao, X.; Mao, Q.; Ouyang, J.; and Liu, Q. 2025. TableMind: An Autonomous Programmatic Agent for Tool-Augmented Table Reasoning. *arXiv preprint arXiv:2509.06278*.

Kazemi, M.; Kim, N.; Bhatia, D.; Xu, X.; and Ramachandran, D. 2023. LAMBADA: Backward Chaining for Automated Reasoning in Natural Language. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 6547–6568. Association for Computational Linguistics.

Lee, J.; and Hwang, W. 2024. Symba: Symbolic backward chaining for structured natural language reasoning. *arXiv preprint arXiv:2402.12806*.

Li, Q.; Li, J.; Liu, Z.; Cheng, M.; Zeng, Y.; Liu, Q.; and Liu, T. 2025. Are LLMs Reliable Translators of Logical Reasoning Across Lexically Diversified Contexts? *arXiv preprint arXiv:2506.04575*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Luo, Y.; Zhou, Y.; Cheng, M.; Wang, J.; Wang, D.; Pan, T.; and Zhang, J. 2025. Time Series Forecasting as Reasoning: A Slow-Thinking Approach with Reinforced LLMs. *arXiv preprint arXiv:2506.10630*.

McCune, W. 2009. Prover9: An Automated Theorem Prover for First-Order Logic. Developed at Argonne National Laboratory, accessed 15 May 2025.

Microsoft Research. 2015. Z3: A High-Performance SMT Solver. Developed by Leonardo de Moura and Nikolaj Bjørner, open-sourced under MIT License, accessed 15 May 2025.

Mondorf, P.; and Plank, B. 2024. Beyond Accuracy: Evaluating the Reasoning Behavior of Large Language Models - A Survey. *CoRR*, abs/2404.01869.

Nunes, T. 2012. Logical reasoning and learning. In *Encyclopedia of the sciences of learning*, 2066–2069. Springer.

Olausson, T. X.; Gu, A.; Lipkin, B.; Zhang, C. E.; Solar-Lezama, A.; Tenenbaum, J. B.; and Levy, R. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. *arXiv preprint arXiv:2310.15164*.

OpenAI. 2023. gpt-3.5-turbo. https://platform.openai.com/docs/models/gpt-3-5.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. Y. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Patel, A.; Li, B.; Rasooli, M. S.; Constant, N.; Raffel, C.; and Callison-Burch, C. 2022. Bidirectional language models are also few-shot learners. *arXiv preprint arXiv:2209.14500*.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2023. Reasoning with Language Model Prompting: A Survey. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5368–5393. Toronto, Canada: Association for Computational Linguistics.

Saparov, A.; and He, H. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Schulte, C.; Lagerkvist, M.; and Tack, G. 2006. Gecode. *Software download and online material at the website: http://www. gecode. org*, 11–13.

Smith, P. 2003. *An introduction to formal logic*. Cambridge University Press.

Sprague, Z.; Yin, F.; Rodriguez, J. D.; Jiang, D.; Wadhwa, M.; Singhal, P.; Zhao, X.; Ye, X.; Mahowald, K.; and Durrett, G. 2024. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning.

Sun, J.; Zheng, C.; Xie, E.; Liu, Z.; Chu, R.; Qiu, J.; Xu, J.; Ding, M.; Li, H.; Geng, M.; Wu, Y.; Wang, W.; Chen, J.; Yin, Z.; Ren, X.; Fu, J.; He, J.; Yuan, W.; Liu, Q.; Liu, X.; Li, Y.; Dong, H.; Cheng, Y.; Zhang, M.; Heng, P.; Dai, J.; Luo, P.; Wang, J.; Wen, J.; Qiu, X.; Guo, Y.; Xiong, H.; Liu, Q.; and Li, Z. 2023. A Survey of Reasoning with Foundation Models. *CoRR*, abs/2312.11562.

Tafjord, O.; Mishra, B. D.; and Clark, P. 2020. ProofWriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.

Wan, Y.; Wang, W.; Yang, Y.; Yuan, Y.; Huang, J.-t.; He, P.; Jiao, W.; and Lyu, M. R. 2024. A & B == B & A: Triggering Logical Reasoning Failures in Large Language Models. *CoRR*, abs/2401.00757. ArXiv: 2401.00757.

Wang, J.; Cheng, M.; and Liu, Q. 2025. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511*.

Wang, S.; Zhong, W.; Tang, D.; Wei, Z.; Fan, Z.; Jiang, D.; Zhou, M.; and Duan, N. 2022. Logic-Driven Context Extension and Data Augmentation for Logical Reasoning of Text. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, 1619–1629. Association for Computational Linguistics.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.-L.; and Hsu, W. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023b. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ye, X.; Chen, Q.; Dillig, I.; and Durrett, G. 2023. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36: 45548–45580.

Zhang, Y.; Du, L.; Cao, D.; Fu, Q.; and Liu, Y. 2024. An Examination on the Effectiveness of Divide-and-Conquer Prompting in Large Language Models.

Zhang, Y.; Yang, J.; Yuan, Y.; and Yao, A. C.-C. 2023. Cumulative Reasoning with Large Language Models. *CoRR*, abs/2308.04371. ArXiv: 2308.04371.

Zhong, W.; Wang, S.; Tang, D.; Xu, Z.; Guo, D.; Wang, J.; Yin, J.; Zhou, M.; and Duan, N. 2021. Ar-lsat: Investigating analytical reasoning of text. *arXiv preprint arXiv:2104.06598*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q. V.; and Chi, E. H. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.