Bi-Chainer: Automated Large Language Models Reasoning with Bidirectional Chaining

Shuqi Liu[1] **Bowei He**[1] **Linqi Song**[1,2*]
[1]Department of Computer Science, City University of Hong Kong
[2] Shenzhen Research Institute, City University of Hong Kong
{shuqiliu4-c, boweihe2-c}@my.cityu.edu.hk
linqi.song@cityu.edu.hk

Abstract

Large Language Models (LLMs) have shown human-like reasoning abilities but still face challenges in solving complex logical problems. Existing unidirectional chaining methods, such as forward chaining and backward chaining, suffer from issues like low prediction accuracy and efficiency. To address these, we propose a bidirectional chaining method, Bi-Chainer, which dynamically switches to depth-first reasoning in the opposite reasoning direction when it encounters multiple branching options within the current direction. Thus, the intermediate reasoning results can be utilized as guidance to facilitate the reasoning process. We show that Bi-Chainer achieves sizable accuracy boots over unidirectional chaining frameworks on four challenging logical reasoning datasets. Moreover, Bi-Chainer enhances the accuracy of intermediate proof steps and reduces the average number of inference calls, resulting in more efficient and accurate reasoning.

1 Introduction

Automated reasoning involves deriving accurate and valid conclusions from explicitly given knowledge (McCarthy and SCIENCE, 1963). Logical reasoning, particularly in the context of unstructured natural language text, is essential for automated knowledge discovery and has promising implications for advancements in diverse scientific fields. Recently, large language models (LLMs) (Touvron et al., 2023; Ouyang et al., 2022; OpenAI, 2023) have shown promising progress in emulating human-like reasoning abilities (Wei et al., 2022). However, they still face challenges when it comes to complex multi-step logical reasoning problems (Creswell et al., 2022; Kazemi et al., 2023; Valmeekam et al., 2022).

Recent studies have enhanced reasoning capabilities by employing a modular approach that breaks down complex tasks into smaller, more manageable components. Selection-Inference (SI) (Creswell et al., 2022) utilizes forward chaining that employs iterative selection and inference steps to draw conclusions. However, the absence of explicit guidance directly targeting the goal results in subpar and imprecise selection. On the other hand, LAMBADA (Kazemi et al., 2023) utilizes backward chaining to handle multi-step reasoning. It starts with the goal, recursively selects rules, and iteratively proves decomposed sub-goals. However, LAMBADA's ranking strategy, which prioritizes shorter rules assuming higher success rates, may not always be accurate. This can result in suboptimal performance and hinder the overall efficiency of the reasoning process.

This drives our exploration of the bi-directional chaining method that enables forward chaining with explicit guidance towards the goal and facilitates backward chaining using determinate facts derived from forward chaining. As illustrated in Figure 1, we present Bi-Chainer, a modular reasoning framework that incorporates bi-directional chaining. Bi-Chainer can be understood as a bi-directional depth-first search algorithm that dynamically switches to the opposite reasoning direction when it encounters multiple branching options within the current direction. Consequently, the intermediate reasoning outcomes obtained from the opposite reasoning direction can be employed as guidance to enhance the ongoing reasoning process on the current side.

We showcase the adaptability and effectiveness of Bi-Chainer on four logical reasoning datasets: ProofWriter (Tafjord et al., 2021), FOLIO (Han et al., 2022), AR-LSAT (Zhong et al., 2022), and ParaRules (Clark et al., 2021). The datasets encompass a broad range of logical reasoning problems, including deductive reasoning, first-order logic reasoning, and analytical reasoning. In addition to achieving quantitative improvements over unidi-

*Corresponding author

Findings of the Association for Computational Linguistics ACL 2024, pages 8578–8598
August 11-16, 2024 ©2024 Association for Computational Linguistics

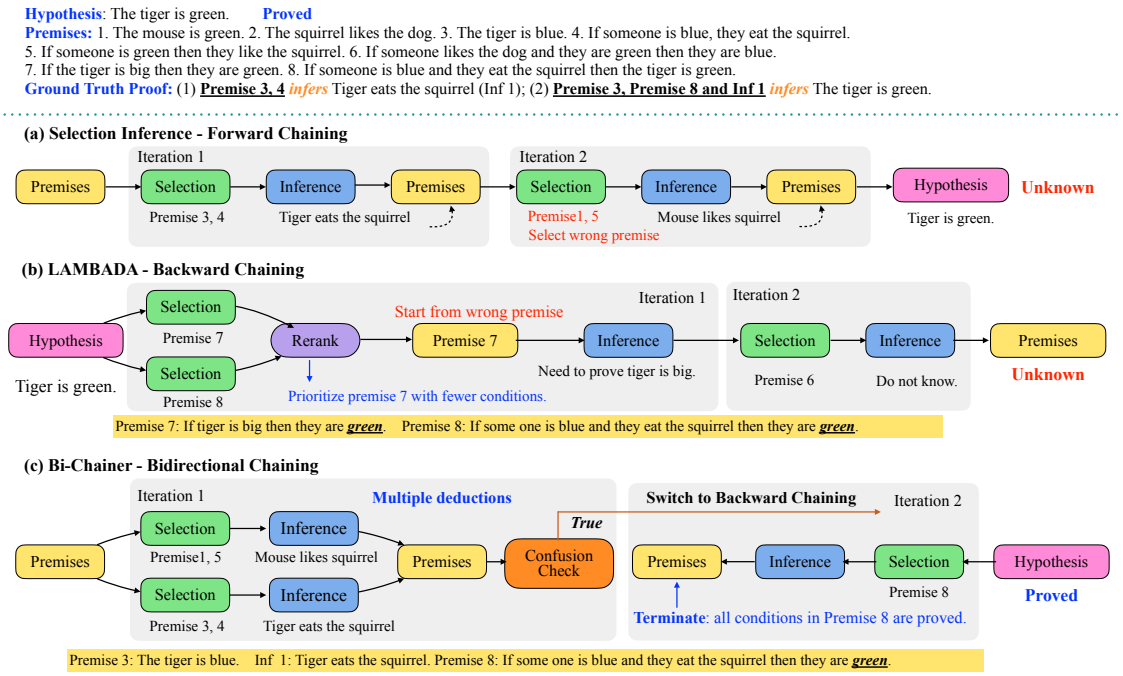


Figure 1: Bi-Chainer framework in bidirectional chaining (c) in comparison with the Selection-Inference framework in forward chaining (a) and the LAMBADA framework in backward chaining (b).

rectional chaining methods, the Bi-Chainer framework also offers qualitative advantages. Firstly, it enhances the accuracy of intermediate proof steps, resulting in more reliable and correct reasoning outcomes at each stage. Secondly, Bi-Chainer reduces the number of inference calls needed during the reasoning process. By utilizing guidance from the opposite side, Bi-Chainer eliminates unnecessary and redundant inference steps.

2 Related Works

Recent advancements in large language models, such as LLaMA (Touvron et al., 2023), PaLM (Chowdhery et al., 2023), and GPT-4 (OpenAI, 2023), have demonstrated surprising human-like intelligence in the area of multi-step logical reasoning. Due to its huge application potential for various applications, including problem-solving, decision-making, and critical thinking (Huang and Chang, 2023), numerous research efforts have been dedicated to improving or eliciting the reasoning ability of these language models. Most of them can be classified into three categories:

Fully Supervised Finetuning: Some previous methods (Rajani et al., 2019; Hendrycks et al., 2021) have employed fine-tuning techniques on pre-trained language models (LMs) using downstream datasets to generate rationales or step-by-step solutions, effectively performing reasoning until obtaining final answers. However, these methods heavily rely on meticulously constructed fine-tuning datasets that explicitly capture the reasoning process. Unfortunately, such high-quality data is often difficult to access or requires substantial resources to create. Moreover, this reliance on specific datasets can restrict the extension of the LM's reasoning abilities to other open-ended reasoning tasks beyond the domain of the fine-tuning dataset.

Prompting & In-Context Learning: The *Chain of Thought (CoT) and its variants* (Wei et al., 2022) are the most common approaches to release and utilize LLM's reasoning capabilities. CoT guides the model to generate explicit step-by-step rationale before producing the final results. The *rationale engineering* techniques like rational refinement/exploration/verification are complementary to CoT for further eliciting the reasoning capabilities more effectively via refining demonstration rationale examples (Fu et al., 2023), encouraging exploring diverse reasoning ways (Wang et al., 2023), and verifying if the generated rationales by LLMs lead to correct final answers (Weng et al., 2023). *Problem decomposition* methods (Zhou et al., 2023; Press et al., 2023) can also help facilitate the CoT reasoning when tackling complex tasks by decomposing the complex problems into relatively simpler subproblems. It should be mentioned that most

previous methods are forward chaining reasoning while only a few works (Kazemi et al., 2023) have noticed its drawbacks and tried conducting backward chaining reasoning.

Hybrid Methods: Some other methods propose to simultaneously enhance and elicit the reasoning capabilities of LLMs with training and prompting techniques, respectively, like *reasoning-enhanced training and prompting* (Chung et al., 2022) and *bootstrapping & self-improving* (Zelikman et al., 2022; Huang et al., 2023).

Our work lies in the category of prompting & in-context learning and aims to fully release the multi-step logical reasoning ability embedded in powerful LLMs like GPT-4.

3 Methodology

In this section, we introduce the Bi-Chainer framework, which automates logical reasoning over natural language premises using bidirectional chaining. The premises $\mathcal{C}$ consist of a set of facts $\mathcal{F}$ and rules $\mathcal{R}$, where rules can be deductive, first-order logic, or analytical reasoning statements. The framework aims to prove or disprove a hypothesis $\mathcal{H}$ based on the given premises. The hypothesis and the premise follow the form "If $\mathcal{P}$, then $\mathcal{Q}$", where $\mathcal{P}$ represents the condition and $\mathcal{Q}$ represents the consequent.

3.1 Bi-directional Chaining

Bidirectional chaining is a reasoning strategy that combines both forward and backward chaining to facilitate the inference process. It involves simultaneous exploration in both directions, starting from the available facts and working forward to derive new conclusions, while also starting from the goal and working backward to decompose the goal into sub-goals using applicable rules. In our research, we define the existence of multiple deductions or abductions as a confusion state since we aim to ensure a depth-first searching process, thereby reducing the number of LLM calls. In a depth-first search, when faced with multiple deductions or abductions at a single reasoning step, the challenge lies in selecting the most suitable deduction to continue the chaining process. Therefore, we describe this challenge as a confusion in the reasoning process. As the term confusion signifies the need to resolve this ambiguity and make decisions to continue the reasoning chain effectively.

Algorithm 1 Bi-Chainer

Input: Premises $\mathcal{C} = (\mathcal{F}, \mathcal{R})$, Hypothesis $\mathcal{H}$ with condition $\mathcal{P}$ and consequent $\mathcal{Q}$, Max-Depth D.

$\mathcal{F}(\mathcal{H})$ = *FactIdentify*$(\mathcal{H}, \mathcal{F})$
while not reach maximum steps D **do**
  if ForwardChaining **then**
    repeat
      $\mathcal{R}_d$ = *RuleSelection*$(\mathcal{F}(\mathcal{H}), \mathcal{R}, \mathcal{Q})$
      $\mathcal{F}_d$ = *LogicDeduction*$(\mathcal{F}(\mathcal{H}), \mathcal{R}_d)$
      Update $\mathcal{F}$ and $\mathcal{F}(\mathcal{H})$ with $\mathcal{F}_d$
      v = *FactCheck*$(\mathcal{H}, \mathcal{F})$
      c = *ConfusionCheck*$(\mathcal{F}_d)$
    until c is True
    Switch to BackwardChaining
  end if
  if BackwardChaining **then**
    repeat
      $\mathcal{R}_a$ = *RuleSelection*$(\mathcal{Q}, \mathcal{R})$
      $\mathcal{F}_a$ = *LogicAbduction*$(\mathcal{Q}, \mathcal{R}_a)$
      $\mathcal{Q} = \mathcal{F}_a$
      v = *FactCheck*$(\mathcal{Q}, \mathcal{F})$
      c = *ConfusionCheck*$(\mathcal{F}_a)$
    until c is True
    Switch to ForwardChaining
  end if
  if v is not Unknown **then**
    return v
  end if
end while
return Unknown

Figure 1 illustrates the application of bidirectional chaining in proving a hypothesis using a set of premises. Initially, forward chaining is employed to derive more definite facts and update the premises. In the forward chaining process, deductions are made based on selected premises, such as Premises 3 and 4 leading to the deduction "Tiger eats the squirrel", and Premises 1 and 5 establishing the deduction "Mouse likes squirrel". However, as multiple deductions are obtained, further forward chaining becomes confusing on which deduction to select to continue the chaining process. Therefore, the Confusion Check module triggers a switch to backward chaining. In the backward chaining phase, both Premise 7 and Premise 8 support the consequence of the hypothesis "Someone is green". However, Premise 8's conditions can all be proven using the intermediate deductions obtained from forward chaining. As a result, the hypothesis is successfully proved using bi-directional chaining.

3.2 LLM Modules in Bi-Chainer

To enable applying bidirectioanl chaining for text-based reasoning, we introduce six LLM-based modules: Fact Identification, Rule Selection, Logic Deduction, Logic Abduction, Fact Check, and Confusion Check. Each module is implemented by providing instructions with relevant in-context demonstrations to an LLM (see Appendix C.3 for details). We describe these modules and then proceed to the full algorithm.

Fact Identification Module. Given the facts $\mathcal{F}$ from the premises and the hypothesis $\mathcal{H}$, the Fact Identification module is responsible for identifying relevant facts $\mathcal{F}(\mathcal{H}) \in \mathcal{F}$ that contribute to proving the hypothesis.

Rule Selection Module. Given a set of rules $\mathcal{R}$ from the premises and a hypothesis $\mathcal{H}$, the Rule Selection module in Forward Chaining identifies a subset of rules $\mathcal{R}_d \in \mathcal{R}$ such that the condition of the rule entails with the facts $\mathcal{F}(\mathcal{H})$ and the consequent of the rule entails with the hypothesis consequent $\mathcal{Q}$. If a rule exists that satisfies these conditions, it is returned as it serves as a bridge between the known facts and the hypothesis, facilitating the concatenation of forward and backward chaining. However, if no such rule is found, only the rules that can be entailed by the known facts are returned. The Rule Selection module in Backward Chaining identifies a subset of rules $\mathcal{R}_a \in \mathcal{R}$ such that the consequent of the rule unifies with the consequent of the hypothesis $\mathcal{Q}$.

Logic Deduction & Logic Abduction Modules. The Logic Deduction module focuses on deductive reasoning, starting from known facts $\mathcal{F}(\mathcal{H})$ and the deductive rules $\mathcal{R}_d$ to derive a set of new conclusions $\mathcal{F}_d$. The Logic Abduction module, on the other hand, deals with abductive reasoning. It aims to generate plausible explanations $\mathcal{F}_a$ that best lead to the hypothesis consequent $\mathcal{Q}$ according to the abductive rules $\mathcal{R}_a$. The generated explanations are then treated as new consequences that need to be proven or validated.

Fact Check. Given the facts $\mathcal{F}$ from the premises, the Fact Check module verifies if the hypothesis $\mathcal{H}$ entails (in which case the hypothesis is proved) or contradicts (in which case the hypothesis is disproved) with the facts. If no such fact can be found, then the truth of $\mathcal{H}$ remains unknown.

Confusion Check Module. The Confusion Check module determines the moment to switch between forward and backward chaining. We define a situation where confusion happens when executing the uni-directional chaining at a single step, multiple deductions (in forward chaining) or abductions (in backward chaining) emerge. In the bidirectional chaining process, if each reasoning step produces consistent deduction $\mathcal{F}_d$ or abduction results $\mathcal{F}_a$ based on the selected rules, the reasoning continues in that direction. However, if different results emerge at each step, it indicates that the system may be confused in selecting the appropriate rule to proceed with. In such cases, the reasoning is temporarily paused, and the other direction of reasoning is allowed to continue for a few steps to gather additional information that can aid in determining the reasoning path in the current direction. Bidirectional chaining thus involves continuously switching between forward and backward chaining until a rule is found that connects the consequent of forward chaining with a plausible explanation derived from backward chaining, or until the maximum step limit is reached.

3.3 The Bi-Chainer Algorithm

Algorithm 1 provides a high-level description of how the six LLM modules described earlier can be integrated with bidirectional chaining to enable text-based logical reasoning (the function calls corresponding to LLM modules are color-coded).

Bi-Chainer can be understood as a bidirectional depth-first algorithm that focuses on reasoning with premises. It employs a depth-first search approach and switches between reasoning directions when faced with multiple branching options. Bi-Chainer takes a set of premises $\mathcal{C} = (\mathcal{F}, \mathcal{R})$, a Hypothesis $\mathcal{H}$ with condition $\mathcal{P}$ and consequent $\mathcal{Q}$, and a depth limit D as input. The algorithm starts by using the *Fact Identify* module to find facts $\mathcal{F}(\mathcal{P})$ that are essential for proving the hypothesis. It then employs forward chaining to iteratively expand the determinate facts that are associated with and supportive of the hypothesis.

During Forward Chaining, the Rule Selection module selects rules $\mathcal{R}d$ from R that are consistent with the identified facts $\mathcal{F}(\mathcal{H})$. The Logical Deduction module then applies these rules and facts to derive new conclusions $\mathcal{F}d$, which are subsequently added to the existing premises. The *Fact Check* module then verifies whether the hypothesis can be proved or disproved using the facts. If this is the case, then the algorithm stops and returns the result. If not the case, the *Confusion Check*

module examines the deduced results to identify any inconsistencies. If different deduction results emerge at each step, it suggests that further deductions based on these conclusions would lead to a significant number of branching paths, deviating from the depth-first approach. In such situations, the algorithm switches the reasoning mode from Forward Chaining to Backward Chaining. Similarly, during Backward Chaining, the *Rule Selection* module identifies rules $\mathcal{R}a$ from R that unifies with the hypothesis consequent. The *Logical Abduction* module then applies these rules to derive the plausible explanations $\mathcal{F}a$, which are then updated as the new consequent to be proved. The *Fact Check* module verifies whether the updated hypothesis can be proved or disproved using the facts enriched by Forward Chaining. On the other hand, the *Confusion Check* module examines any inconsistencies are present in $\mathcal{F}a$ to determine if a change in the reasoning mode is necessary.

4 Experimental Setup

We describe our baselines and datasets here, and provide further implementation details in Appendix B. Unless stated otherwise, all experiments are based on GPT-4 (OpenAI, 2023).

4.1 Baselines

We compare against the following four baselines.

Standard directly prompts LLM to output labels and proofs in an end-to-end manner, showcasing the lower bound of LLM's capabilities.

Chain-of-Thought (CoT) (Wei et al., 2022) adopts a step-by-step problem-solving approach, generating explanations before providing the final answer. In our work, the indeterminate explanations are the corresponding step-by-step proof.

Selection-Inference (SI) (Creswell et al., 2022) is a forward modular reasoning framework. SI starts from the facts and rules, it iteratively calls selection and inference, until the goal can be proved or disproved.

Backward Chaining Reasoning (LAMBADA) (Kazemi et al., 2023) tackles multi-step reasoning using backward chaining. LAMBADA starts from the goal, it recursively selects rules that share the same consequent as the goal and then decomposes the goal into sub-goals based on the antecedent of the selected rules. The recursive selection and decomposition process continues until the sub-goals can be proved or disproved based on the given facts.

4.2 Datasets

We experiment with four challenging logical reasoning datasets outlined below.

ProofWriter (Tafjord et al., 2021) is a commonly used synthetic dataset for testing logical reasoning. We use the ProofWriter OWA dataset of proof depth 0, 1, 2, 3 and 5. The task is to determine the provability of the hypothesis as Proved, Disproved, or Unknown based on the given premises. Our reported results include two sets: ProofWriter-PUD, containing all proven examples, and ProofWriter-PD, excluding examples labeled as Unknown. In line with the methodology outlined by Kazemi et al. (2023), we employ the first 1000 examples from the test set for our analysis.

FOLIO (Han et al., 2022) is a challenging expert-written dataset with complex first-order logic reasoning. The problems are mostly aligned with real-world knowledge and use highly natural wordings. We use the entire FOLIO test set for evaluation, consisting of 204 examples.

AR-LSAT (Zhong et al., 2022) is a challenging dataset that focuses on investigating the analytical reasoning of text. The questions are collected from the Law School Admission Test from 1991 to 2016. We use the entire test set of 230 multiple choice questions. AR-LSAT is particularly challenging, with state-of-the-art models only achieving performance slightly better than random guessing (Liang et al., 2022; Ribeiro et al., 2022).

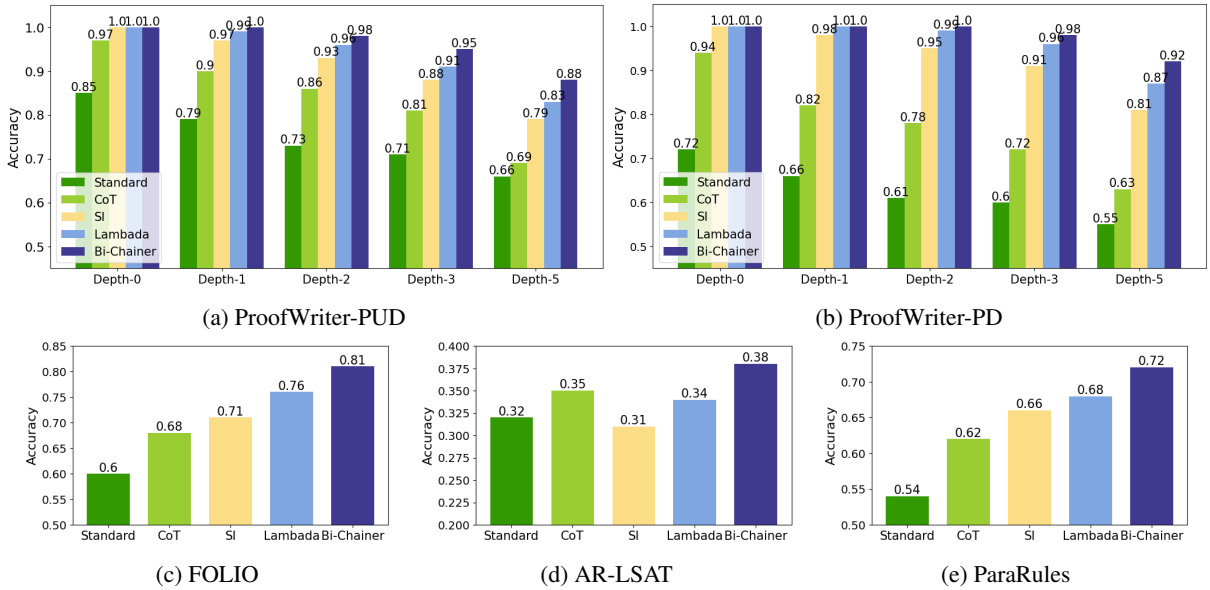
ParaRules (Clark et al., 2021) modifies from ProofWriter where the synthetically generated premises are rewritten by crowdworkers to increase diversity and naturalness. Thus, we can surpass the evaluation of reasoning limited to templatic expressions. The provided examples necessitate proof depths of up to 5, and the corresponding labels are Proved, Disproved, or Unknown. We employ the first 200 examples of the test set for evaluation.

5 Results

We now describe the results and compare Bi-Chainer with the baselines in detail.

5.1 Label Prediction Accuracy

The overall label prediction accuracy results across various reasoning frameworks are reported in Figure 2 (a)-(e). The Bi-Chainer framework, employing bi-directional chaining techniques, is observed to significantly outperform both the foundational reasoning models such as the standard and CoT



(a) ProofWriter-PUD (b) ProofWriter-PD

(c) FOLIO (d) AR-LSAT (e) ParaRules

Figure 2: Label prediction accuracies on (a)-(b) ProofWriter, (c) FOLIO, (d) AR-LSAT, and (e) ParaRules datasets.

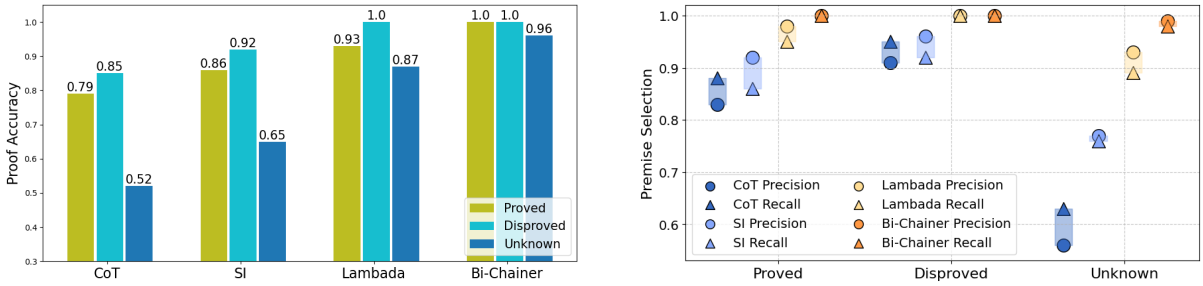
frameworks, and the more advanced, modular reasoning systems such as the SI and LAMBADA frameworks. In the evaluation of the ProofWriter-PUD dataset at a reasoning depth of 5, the comparative analysis reveals that Bi-Chainer achieves a relative improvement of 8.9% over the SI framework. Against the LAMBADA framework, Bi-Chainer maintains a strong lead with a 6.3% relative improvement. Moreover, the FOLIO dataset, which presents more difficult real-world reasoning challenges, also reflects the Bi-Chainer framework's superior performance. Here, Bi-Chainer records a relative improvement of 14.1% when compared to the SI framework. Against the backward-chaining LAMBADA framework, Bi-Chainer again prevails with a relative improvement of 6.6%.

In the context of the AR-LSAT dataset, which involves complex analytical reasoning, the modular reasoning frameworks SI and LAMBADA exhibit lower performance compared to CoT. On the other hand, Bi-Chainer demonstrates a relative increase of 8.5% in performance compared to CoT. In ParaRules, the introduction of naturalness and diversity through paraphrasing might inadvertently introduce ambiguity of the original premises, resulting in a decrease in the accuracy of label prediction compared to the ProofWriter dataset. However, Bi-Chainer demonstrates a notable relative improvement of 9.1% over the SI framework and 5.9% over the LAMBADA framework. This consistent outperformance across diverse datasets indicates the adaptability and generalization strength of the Bi-Chainer framework's reasoning mechanisms.

5.2 Proof Accuracy

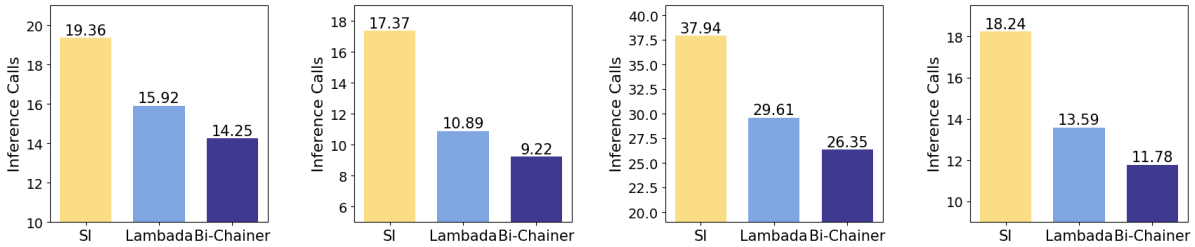
To validate whether each reasoning framework is susceptible to hallucinations, which involve correct final label predictions but incorrect intermediate steps, we conduct an assessment of the proof accuracy. We randomly selected separate sets of 50 examples from Depth-5 of the ProofWriter-PUD dataset where each reasoning framework predicted the label correctly and manually verified if the proof chain is correct or not. In each step, we compared the facts, rules, and resulting conclusions utilized in the reasoning process to corresponding steps in the reference reasoning path. A proof chain is considered to be correct if these elements are consistent with each other. The proof accuracy results are reported in Figure 3a.

We observe that different reasoning frameworks demonstrate varying levels of logical reasoning hallucinations in different cases. In general, modular reasoning frameworks, including SI, Lambada, and Bi-Chainer, are less affected by implicit patterns in language models and achieve higher proof accuracy compared to direct proof generation frameworks like CoT. CoT has an average proof accuracy of 68%, while SI achieves 78%. Lambada demonstrates an impressive proof accuracy of 94%, and Bi-Chainer surpasses all with the highest average proof accuracy of 98%. Specifically, we observe that whenever reasoning frameworks predict Proved or Disproved, the prediction is mostly cor-



(a) (b)

Figure 3: (a) Proof accuracy results on ProofWriter-PUD (Depth-5) for a set of randomly sampled examples for which the models correctly predicted the goal. (b) Precision and Recall results for Premise Selection on the selected samples from the ProofWriter-PUD (Depth-5), with shaded areas indicating the performance gap between different reasoning frameworks for the Proved, Disproved, and Unknown cases.



(a) ProofWriter (b) FOLIO (c) AR-LSAT (d) ParaRules

Figure 4: Comparing SI, LAMBADA with Bi-Chainer w.r.t. the average number of inference calls they make per example in different datasets.

rect. The accuracy is slightly more in cases where the prediction is Disproved. We believe this is because in cases where the result is Disproved, the reasoning path of the model typically involves accurately identifying contradictions or inconsistencies, thereby reducing hallucinatory reasoning.

Moreover, in the case of unknown examples, forward chaining frameworks such as CoT and SI face difficulties in accurately determining the correct reasoning path, achieving relatively low accuracies of 52% and 65%, respectively. In contrast, the Lambada framework uses backward chaining to capture goal-oriented premises, leading to a significant improvement in achieving 87% accuracy for unknown cases. On the other hand, the Bi-Chainer framework employs bidirectional chaining to assist premise selection under the guidance of other side's intermediate reasoning results, resulting in an impressive accuracy of 96%.

In practice, generated reasoning paths often exhibit partial correctness, with errors occurring during the intermediate reasoning process. These errors are mainly attributed to incorrect premise selection, as large models possess powerful single-step reasoning capabilities. To assess the extent of partially correct reasoning, we measure the precision and recall of unique premises extracted from the generated proof that are also present in the reference reasoning path. The results are presented in Figure 3b. In the case of the CoT method, it heavily relies on internal knowledge and rules within the model to generate proofs, resulting in a limited selection of premises. Consequently, the method exhibits higher recall values (around 5.3% higher) than precision values.

On the contrary, the SI method requires considering all available facts and rules that can be used for deduction at each step of the reasoning process. This leads to a larger selection of premises in the reasoning process, resulting in lower recall values (around 3.6% lower) compared to precision values. While most of the reasoning paths in Lambada are correct, in situations where there are numerous and complex facts and rules, there may be a process of error correction. Consequently, the selection of premises in the proof becomes more diverse, leading to lower recall values (around 2.3% lower) compared to precision values. In contrast, the Bi-Chainer method excels in handling scenarios with a large number of complex facts and rules. It lever-

ages the guidance provided by the forward chaining process, utilizing intermediate results to guide the backward chaining. This approach mitigates the occurrence of errors and the need for subsequent corrections, resulting in both high recall and precision values.

5.3 Number of Inference Calls

Another advantage of Bi-Chainer is its efficiency compared to other modular reasoning frameworks, such as SI and Lambada, which often require multiple LLM inference calls per example. In Figure 4, we compare the average number of LLM calls per example for our datasets. For the ProofWriter dataset, Bi-Chainer requires 14.25 LLM calls, which is 1.12 times fewer than Lambada and 1.36 times fewer than SI. In the case of the FOLIO dataset, which has a limited number of premises, Bi-Chainer requires 9.22 LLM calls, exhibiting 1.18 times fewer calls than Lambada and 1.89 times fewer calls than SI. However, the AR-LSAT dataset poses a different challenge as it contains five options per question. This requires more LLM calls for evaluating each option, resulting in 26.35 calls for Bi-Chainer. Despite this, Bi-Chainer still reduces LLM calls by 1.12 times compared to SI and 1.44 times compared to Lambada. As for the ParaRules dataset, the presence of paraphrased premises increases the difficulty of accurately selecting the relevant premises. Consequently, the number of LLM calls for ParaRules exhibits a decrease compared to the ProofWriter dataset, with Bi-Chainer requiring 11.78 calls.

6 Additional Results

Performance on Open-Source Model. We also adopt the open-sourced LLaMA2 7B model in greedy search decoding strategy to supplement the corresponding experiments on ProofWriter and ParaRules datasets.

Method	ProofWriter (d5)	ParaRules
CoT	43.4	33.5
SI	52.6	41.0
Lambada	58.9	43.5
Bi-Chainer	**62.3**	**48.5**

Table 1: Label accuracy of LLaMA2 7B model on ProofWriter and ParaRules datasets.

Individual Module Performance. To understand which components in Bi-Chainer are responsible for the failure cases, we computed the individual accuracy of the six modules described in Section 3. For this purpose, we randomly sampled 100 examples from the validation set of ProofWriter. This sampling included 20 examples for each reasoning depth. We then manually wrote the desired outputs for each module. A module prediction is considered correct if it matches our annotations. The performance of modules in Bi-Chainer is shown in Table 2.

Model	FC	FI	RS	LD	LA	CC
GPT-4	97.82	98.78	91.52	97.44	95.26	97.73
LLaMA2	83.71	86.58	65.54	93.43	89.80	95.29

Table 2: Individual module performance in Bi-Chainer.

The evaluation results indicate that the Fact Check module (FC), Fact Identify module (FI), and Confusion Check module (CC) demonstrate a better performance. On the other hand, the Rule Selection module (RS) exhibits the lowest performance among all the modules, indicating that the LLM still faces challenges in effectively selecting the appropriate rules during the reasoning process. Additionally, the Logical Abduction module (LA) performs slightly lower than the Logical Deduction module (LD), suggesting that decomposing conditions are slightly more difficult for the LLM compared to making deduction inferences.

Compare width-first search framework. Tree of Thoughts (ToT) reasoning framework (Yao et al., 2024) performs reasoning and evaluation on each intermediate result in a tree-searching manner. Thus, compared to our depth-first bi-directional searching framework, ToT is a width-first searching framework, resulting in a high number of inference calls. The result comparison between ToT and Bi-Chainer is shown in Table 3.

Method	Accuracy	Inference calls
Standard	54.0	1
CoT	61.5	1
ToT	65.0	22.79
SI	65.5	18.24
Lambada	67.5	13.59
Bi-Chainer	**72.0**	**11.78**

Table 3: ToT performance on ParaRules.

The results demonstrate that ToT surpasses SI but trails behind Lambada in terms of performance, and falls even further behind Bi-Chainer. This can be attributed to ToT's focus on solving general complex reasoning tasks, rather than being specifically tailored for goal-oriented tasks like logical reason-

ing. Besides, ToT's reasoning process, which involves tree-searching for each intermediate result, leads to a significant number of Inference calls.

Robustness Analysis. We supplement the label accuracy result (mean and standard deviation) of both CoT baseline and our Bi-Chainer under 3 GPT-4 runs on the FOLIO and AR-LSAT datasets in Table 4. We observe that the variance across multiple runs is consistently low compared to the improvement in performance, suggesting that GPT-4 is stable in performing logical reasoning tasks.

Method	FOLIO	AR-LSAT
CoT	59.64 $\pm$ 0.6112	34.93 $\pm$ 1.0974
Bi-Chainer	**81.24** $\pm$ 0.8328	**38.08** $\pm$ 0.9351

Table 4: Label accuracy of CoT and Bi-Chainer on FOLIO and AR-LSAT dataset. We report the mean and standard deviation under 3 GPT-4 runs

7 Case Study

We demonstrate a case study to understand the performance of the Bi-Chainer method compared to LAMBADA. We give a high-level overview and abbreviated examples here, leaving full detailed examples in Appendix A. Lambada experienced premise confusion, it fails to accurately determine the appropriate rule for the subsequent inference step when multiple rules unify with the consequent of the goal statement. As a result of choosing the wrong rule, the model was unable to validate the premise condition, resulting in a wrong conclusion.

Hypothesis: The cow chases the cow.
Step 1: Rule 2: If someone is rough and the tiger sees the bear then they chase the cow. Rule 3: If someone likes the tiger then they chase the cow. Multiple rules unified.
Step 2: Select the shorter rule, Rule 2. Select the wrong rule.
Further steps fail to prove the goal.
Conclusion: Unknown.
Premise confusion error: Lambada encountered premise confusion where Rule 2 and Rule 3 are both unified with the consequent of the goal statement. The model erroneously selects Rule 2 with fewer subgoals, leading to further steps that fail to prove the sub-goal.

Bi-Chainer for Lambada premise confusion

Step 1: Identify the facts about the cow, The cow is blue, and The cow chases the lion.
Step 2: In forward-chaining rule selection, we have two candidate rules: Rule 1 and Rule 6.
Step 3: Forward-chaining Logical Deduction: As the cow is blue, we can deduce that the cow chases the tiger from Rule 1. Additionally, since it is stated that the cow chases the lion, we can further deduce that the cows are rough from Rule 6.
**Detect forward chaining leads to multiple deductions, switch to backward chaining. **
Step 4: Backward-chaining Rule Selection: we have two candidate rules: Rule 6: if someone is rough and the tiger sees the bear, then they chase the cow. Rule 3 states that if someone likes the tiger, then they chase the cow.
Step 4: Backward-chaining Logical Abduction: Using Rule 6 and knowing the cow is rough and the tiger sees the bear, we can deduce that the cow chases the cow.
Conclusion: True.

8 Conclusion

We propose the bidirectional chaining method, Bi-Chainer, to overcome the limitations of existing unidirectional chaining methods for complex logical reasoning. By dynamically switching to depth-first reasoning in the opposite direction when faced with multiple branching options, Bi-Chainer leverages intermediate reasoning results to enhance the reasoning process. In the experiments, Bi-Chainer demonstrates substantial accuracy improvements over unidirectional chaining frameworks on challenging datasets. It also improves the accuracy of intermediate proof steps and reduces the average number of inference calls, resulting in more efficient and accurate reasoning.

Acknowledgements

This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grant GRF 11217823 and Collaborative Research Fund C1042-23GF, the National Natural Science Foundation of China under Grant 62371411, InnoHK initiative, the Government of the HKSAR,Laboratory for AI-Powered Financial Technologies.

Limitations

This paper presents a novel approach for enhancing reasoning capabilities in large language models through bidirectional chaining. However, it is crucial to acknowledge and address several limitations inherent in this research:

(1) **Scalability:** The proposed approach may face challenges in terms of scalability when applied to large-scale datasets or real-time applications. The computational complexity of bidirectional chaining may hinder its efficiency, potentially limiting its practicality for scenarios requiring rapid and extensive reasoning.

(2) **Dependency on Pretrained Models:** The approach heavily relies on pretrained language models, which may introduce certain limitations. Pretrained models are prone to biases and may not capture all relevant contextual information, leading to potential errors or inaccuracies in reasoning outcomes. Additionally, the reliance on pretrained models limits the flexibility and adaptability of the proposed method to new domains or specialized contexts.

(3) **Lack of Explainability:** While bidirectional chaining enhances reasoning capabilities, it may obscure the interpretability and explainability of the model's decision-making process. Understanding the reasoning steps and how conclusions are reached becomes challenging, hindering transparency and trust in the system. This limitation may impact the acceptance and adoption of the proposed approach in critical applications where interpretability is essential.

(4) **Knowledge Acquisition and Representation:** The effectiveness of bidirectional chaining heavily depends on the availability and quality of the underlying knowledge base. Incomplete or inaccurate knowledge representations may result in flawed reasoning or incorrect conclusions. Additionally, the challenge of continuously updating and maintaining the knowledge base to keep up with evolving information poses a significant obstacle.

(5) **Ethical Considerations:** The utilization of large language models raises ethical concerns, including the potential for generating biased or offensive content. Although bidirectional chaining aims to enhance reasoning, it does not inherently address these ethical challenges. Proactive measures, such as comprehensive content filtering and bias detection mechanisms, should be integrated to mitigate the risks associated with unintended outputs.

Addressing these limitations is vital for future research in automated large language models reasoning with bidirectional chaining. Overcoming scalability issues, ensuring model transparency, improving knowledge acquisition, and addressing ethical considerations will contribute to the broader adoption and practicality of the proposed approach in real-world applications.

Ethics Statement

This study utilizes publicly available datasets for our models. Prior research endeavors have generally taken ethical considerations into account. We have manually inspected a subset of samples and found no explicit ethical concerns, including violent or offensive content. Nonetheless, it is crucial to highlight that the output generated by large language models lacks the degree of control we might assume. Consequently, we are prepared to implement measures to mitigate any unforeseen outputs.

References

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1051–1068. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1049–1065. Association for Computational Linguistics.

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

John McCarthy and STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE. 1963. Programs with common sense.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4932–4942. Association for Computational Linguistics.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Henghui Zhu, Rui Dong, Deguang Kong, Juliette Burger, Anjelica Ramos, William Yang Wang, George Karypis, et al. 2022. Street: A multi-task structured reasoning and explanation benchmark. In *The Eleventh International Conference on Learning Representations*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 2550–2575. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Yining Chen, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2022. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2306–2319.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 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.

A Additional Results and Analyses

In this section, we provide some more in-depth qualitative and quantitative analysis of the results from our model and the baselines.

A.1 Biases of Reasoning Frameworks

Figure 2 (a)-(b) demonstrates a performance disparity among different reasoning frameworks when handling datasets with or without Unknown cases. To gain deeper insights into the inherent biases of each method, we provide detailed confusion matrices in Figure 5. The results reveal that Bi-Chainer consistently outperforms other reasoning frameworks across all Proved, Disproved, and Unknown cases, indicating its ability to achieve accurate and well-balanced predictions. In contrast, CoT exhibits a noticeable bias in predicting Unknown labels, with 24% of Proved cases and 39% of Disproved cases being misclassified as Unknown. Consequently, in the absence of unknown cases, the CoT method experiences a decline in model accuracy, while the other methods show an improvement.

Furthermore, we observe that forward chaining is particularly effective in handling Proved cases, while backward chaining demonstrates a more significant improvement in handling Disproved cases. Compared to CoT, the Forward chaining-based SI method shows a relative improvement of 29% for Proved cases and 28% for Disproved cases. The backward chaining-based Lambada method demonstrates a relative improvement of 31% for Proved cases and an impressive relative improvement of 47% for Disproved cases, which is 1.7 times higher than the improvement achieved by the SI method. The Bi-Chainer method, which incorporates bidirectional reasoning, combines the advantages of forward chaining that aligns with the natural flow of logical order and backward chaining that focuses on goal-oriented reasoning. It effectively addresses situations of uncertainty in one-directional reasoning by timely incorporating intermediate results from the other side as guidance. This enhances the probability of selecting accurate premises for reasoning. Consequently, the Bi-Chainer method achieves a further relative improvement of 39% in Proved cases compared to CoT, and a relative improvement of 54% in Disproved cases.

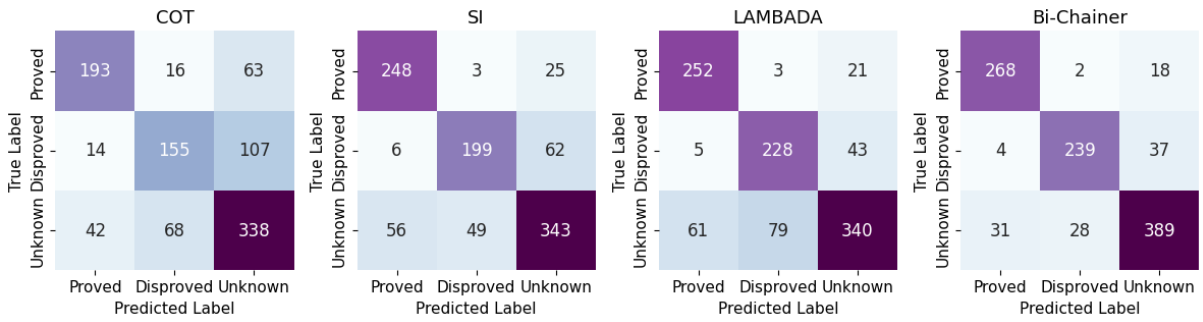


Figure 5: Confusion matrices.

A.2 Failure Modes of CoT

CoT demonstrates a self-bias. CoT tends to rely heavily on its internal knowledge and rules during the reasoning process, disregarding the externally provided facts and rules.

(GPT-4 Incorrect CoT): Let's think step by step. From Fact 9 and Rule 4, since the cat sees the bear and someone who sees the cat (which could be anyone, including the cat itself) sees the bear, they eat the dog. This rule suggests that the cat could eat the dog because the cat sees the bear and the cat sees itself, fulfilling the condition of the rule. Conclusion: Proved
Self-bias error: CoT makes an erroneous inference by concluding that the cat eats the dog based on its own speculation rather than the given facts. The model incorrectly assumes that the cat can see itself, which is not supported by the provided facts.
Hallucinatory Proof: However, CoT predict the correct label, indicating self-bias leads to the generation of hallucinatory proofs.

CoT fails to make inferences beyond the explicit rules provided. In complex scenarios where multiple steps of reasoning or indirect relationships are involved, CoT tends to focus on surface-level connections and fails to engage in deep reasoning.

Hypothesis: Karen will share "Stranger Things" to Lisa.
Premises:
1. "Stranger Things" is a popular Netflix show.
2. If a Netflix show is popular, Karen will binge-watch it.
3. If and only if Karen binge-watches a Netflix show, she will download it.
4. Karen does not download "Black Mirror".
5. "Black Mirror" is a Netflix show.
6. If Karen binge-watches a Netflix show, she will share it to Lisa.
(GPT-4 Incorrect CoT): Let's think step by step. "Stranger Things" is a popular Netflix show, so Karen will binge-watch it (Premise 1, Premise 2). However, there is no direct information provided about whether Karen has actually binge-watched "Stranger Things" or whether she has downloaded it. Therefore, we cannot definitively conclude whether Karen will share "Stranger Things" with Lisa based on the given information. Conclusion: Unknown
Multi-step reasoning error: CoT makes an erroneous inference of the unknown due to its inability to engage in deeper reasoning. CoT gives up after failing to find any direct link between watching "Stranger Things" and sharing with Lisa.

A.3 Failure modes of SI

SI encounters over-inference in the reasoning process. SI fails to effectively identify and utilize the premise information relevant to validating the hypothesis, leading to excessive processing and reasoning with all available premises, including those that do not contribute to the verification process. This redundancy and unnecessary computation reduce the efficiency and accuracy of SI's reasoning.

Hypothesis: The squirrel is blue.
Facts:
1. The dog eats the mouse.
2. The dog eats the tiger.
3. The dog visits the squirrel.
4. The mouse is green.
5. The mouse visits the tiger.
6. The squirrel is big.
7. The squirrel is round.
8. The squirrel likes the dog.
9. The tiger eats the dog.
10. The tiger visits the mouse.
Rules:
1. If someone eats the tiger and the tiger is big then they are green.
2. If someone is green then they like the squirrel.
3. If the dog is green then the dog likes the mouse.
4. If someone visits the tiger then the tiger is blue.
5. If someone visits the tiger then the tiger visits the dog.
6. If someone is blue and they eat the squirrel then the squirrel is green.
7. If someone is blue then they eat the squirrel.
8. If someone likes the dog and they are green then they are blue.
SI reasoning steps:
Step 1: Analyze Fact 5 and Rule 4.
Fact 5: the mouse visits the tiger. Rule 4: if someone visits the tiger then the tiger is blue. Deduction: The tiger is blue. Valid
Step 2: Analyze the deduced fact that the tiger is blue and Rule 7. Valid
Rule 7: if someone is blue, they eat the squirrel.
Deduction: The tiger eats the squirrel. Valid
Step 3: Analyze Fact 4 and Rule 2.
Fact 4: the mouse is green. Rule 2: if someone is green then they like the squirrel.
Deduction: Mouse likes the squirrel. Invalid
Step 4: Analyze Fact 8 and Rule 8.
Fact 8: the squirrel likes the dog. Rule 8: if someone likes the dog and they are green then they are blue.
Deduction: We do not have information about the squirrel being green, so Rule 8 does not apply. Wrong deduction
Conclusion: Unknown.
Over-inference error: Despite deducing that the squirrel is green based on the deduction results in Step 1 and 2 using Rule 6, the presence of invalid reasoning Step 3 prevents the model from progressing along the correct reasoning path within its limited multi-step reasoning capacity. As a result, the model incorrectly concludes that Rule 8 cannot be applied.

A.4 Failure Modes of LAMBADA

Lambada experienced premise confusion LAMBADA fails to accurately determine the appropriate rule for the subsequent inference step when multiple rules unify with the consequent of the goal statement. As a result of choosing the wrong rule, the model was unable to validate the premise condition, resulting in a wrong conclusion.

Hypothesis: The cow chases the bear.
Facts:
1. The bear is blue.
2. The bear is round.
3. The bear sees the cow.
4. The cow is blue.
5. The lion is rough.
6. The lion likes the tiger.
7. The lion sees the bear.
8. The tiger is cold.
9. The tiger is round.
10. The tiger sees the bear.
11. The tiger sees the cow.
Rules:
1. If someone is blue then they chase the tiger.
2. If the cow is blue and the tiger sees the bear then the cow chases the lion.
3. If someone likes the tiger then they chase the lion.
4. If someone likes the lion then the lion chases the tiger.
5. If the cow is cold and the cow chases the bear then the bear chases the tiger.
6. If someone chases the cow and they chase the lion then they chase the bear.
7. If someone is rough then they chase the cow.

8. If someone is cold then they are blue.
9. If someone is blue and they chase the lion then they are rough.

LAMBADA reasoning steps:
Step 1: Select Rule 6, If someone chases the cow and they chase the lion then they chase the bear.
Step 2: We need to prove the cow chases the cow and they chase the lion.
Step 3: To prove the cow chases the cow, select Rule 7: If someone is rough then they chase the cow.
Step 4: We need to prove the cow is rough.
Step 5: To prove the cow is rough, select Rule 9: If someone is blue and they chase the lion then they are rough.
Step 6: By checking the facts, we know that the cow is blue (Premise 4).
Step 7: We need to prove the cow chases the lion.
Step 8: To prove the cow chases the lion, we have two candidate rules. Rule 2: If the cow is blue and the tiger sees the bear then the cow chases the lion; and Rule 3: If someone likes the tiger then they chase the lion.
Step 9: As Rule 3 has fewer sub-goals, we start with proving the cow likes the tiger. Select the wrong rule based on the Rerank strategy in LAMBADA.
Step 10: Based on the given information, we were unable to find a rule or fact that directly connects or unifies with the statement "The cow likes the tiger." Therefore, the truth or validity of this statement remains unknown based on the provided context.
Conclusion: Unknown.
Premise confusion error: Lambada encountered premise confusion where Rule 2 and Rule 3 are both unified with the consequent of the goal statement. The model erroneously selects Rule 3 with fewer sub-goals, leading to further steps that fail to prove the sub-goal.

B Implementation Details

For our experiments, we used the GPT-4 (OpenAI, 2023) for all the models (both Bi-Chainer and the baselines). The decoding temperature was set to 0.1. we limit the maximum number of tokens to generate to 1024 for FOLIO and 4096 for ProofWriter, ParaRules, and AR-LSAT. We use gpt-4-0613 checkpoint of GPT-4 model and invoke the model via the OpenAI API. We prompt the model with a set of instructions and 1-8 ICL examples. The examples follow a structured text format designed to scaffold generations and facilitate post-processing. Each ICL example begins with a task description, followed by the NL hypothesis. The premises are then outlined using numbered statements. The necessary reasoning steps for each example are subsequently outlined in a separate section.

ProofWriter. We utilize a subset of the publicly available ProofWriter dataset, specifically the Open World Assumption (OWA) dataset [1]. Due to the cost of inference, we used the first 1000 examples in the test set. In the Closed World Assumption (CWA) dataset, everything is either proven True or False. However, in the OWA dataset, if a statement cannot be proven True or False, it is labeled as Unknown. For Unknown samples, where there is no explicit reasoning trace, it is essential to enumerate all possible facts. If the hypothesis has not been proven or disproven, it is classified as Unknown. For this reason, we need to manually verify if the proof chain is correct or not for proof accuracy analysis.

FOLIO. We use the publicly available FOLIO dataset [2] and use the validation split of the dataset in our evaluation as the testing split is not publicly available. The original dataset has 204 validation examples.

AR-LSAT. We use the publicly available AR-LSAT dataset [3] and use the full test set of 230 examples in our evaluation. The AR-LSAT dataset differs from other datasets in that its labels are not fixed. Each example in the AR-LSAT dataset consists of five options associated with a question. To address this, during the prompting process, we concatenate the question with each option to form a hypothesis. Consequently, each AR-LSAT example has five hypotheses that need to be validated. However, the results obtained from validating earlier hypotheses are added to the premises to reduce redundant reasoning among multiple hypotheses.

Pararules. We use a subset of the publicly avail-

[1] https://allenai.org/data/proofwriter
[2] https://github.com/Yale-LILY/FOLIO
[3] https://github.com/zhongwanjun/AR-LSAT/tree/main/data

able ParaRules dataset [4], specifically the parallel dataset that runs through the Problog reasoner that produced the same labels. The subset consists of the first 200 examples from the test set, with a reasoning depth of 5. Table 5 provides a comprehensive summary of the examples utilized in our study, which are derived from four distinct datasets representing three different types of logical reasoning problems.

C Few-Shot Prompts

We select representative samples from the training split of the dataset as our few-shot examples. These samples are chosen to ensure a balanced representation across different labels. As the training set lacks correct proofs, we manually provide the corresponding proof for each example. We utilize the FOLIO dataset for demonstrating prompts across different reasoning frameworks due to its limited number of premises, which facilitates the presentation.

C.1 Chain-of-Thought Prompting

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

Reason:
Let's think step by step. As indicated by Premise 3, Real Madrid and Barcelona are both La Liga soccer teams. From premise 4, Real Madrid received 86 points, and Barcelona received 73 points. This implies Real Madrid has more points than Barcelona. From premise 1, if a team receives more points, it ranks higher. Therefore, Real Madrid ranks higher than Barcelona based on points.

Answer:
True

C.2 Selection-Inference Prompting

SI framework iteratively calls the selection and inference module. The **selection** prompt is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to select the premises where new conclusions can be derived toward proving the goal.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid receives 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both received 3 points from the games between them.

Selected Premises:
Step 1: Premise 3: Real Madrid and Barcelona are both La Liga soccer teams.

[4] https://allenai.org/data/ruletaker

Step 2: Premise 4, Real Madrid received 86 points, and Barcelona received 73 points.
Step 3: Premise 1, If a team receives more points, it ranks higher.

The **inference** prompt is:

Task Description:
Derive the inferences based on the selected premises.

Inferences:
Step 1: From Premises 4, Real Madrid received 86 points, and Barcelona received 73 points. This implies Real Madrid has more points than Barcelona.
Step 2: From Premise 1: If a team receives more points, it ranks higher. We know that Real Madrid receives more points than Barcelona. Therefore, Real Madrid ranks higher than Barcelona.

C.3 LAMBADA Prompting

LAMBADA employs backward chaining with four modules: Fact Check, Rule Selection, Goal Decomposition, and Sign Agreement. We add instructions for LAMBADA to align with our method, the additional instructions only summarize the main idea of each module.

The prompt for Fact Chek is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to check the premises whether the hypothesis can be directly proved or disproved by one of the premises.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
...
6: In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Fact Check:
The hypothesis can be directly proved by Premise 6.

The prompt for Rule Selection is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to select the rules that share the consistent consequences as the hypothesis.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

Rule Selection:
Premise 1, A La Liga soccer team ranks higher than another if it receives more points. or
Premise 2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.

The prompt for Goal Decomposition is:

Task Description:
Analyze the plausible sub-goals for the selected rules.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Decomposed Sub-Goals:
According to Premise 1, if we want to prove a La Liga soccer team ranks higher than another, we need to prove the La Liga soccer team receives more points. or
According to Premise 2, if we want to prove a La Liga soccer team ranks higher than another, we need to prove two La Liga soccer teams receive the same points, and one of them receives more points from the games between the two teams.

The prompt for Sign-Agreement is:

Task Description:
Check whether the consequence of the rule agrees or disagrees with the hypothesis.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Rule:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Agreement Sign:
Agree.

C.4 Bi-Chainer Prompting

Bi-Chainer employs bi-directional chaining with six modules: Fact Check, Fact Identify, Rule Selection, Logical Deduction, Logical Abduction, and Confusion Check. The prompt for the Fact Check module in our approach aligns with the prompt used in LAMBADA, as presented above.

The prompt for Fact Identify is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to identify the premises where new conclusions can be derived toward proving the goal.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

Fact Identify:
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

The prompt for Rule Selection in Forward Chaining is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to select the rules whose conditions entail the identified facts and whose consequents entail the consequent of the hypothesis. If a rule satisfying these criteria is found, return it as the result. Otherwise, return only the rules that are entailed by the identified facts.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more

points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

Rule Selection:
Premise 1, A La Liga soccer team ranks higher than another if it receives more points.

The prompt for Logical Deduction:

Task Description:
Derive the inferences based on the selected premises.

Inferences:
We know that Real Madrid receives more points than Barcelona (Premise 4). Therefore, Real Madrid ranks higher than Barcelona (Premise 1).

The prompt for Rule Selection in Backward Chaining is:

Task Description:
Given a set of premises, you have to reason whether the hypothesis is true, false, or unknown. To prove the hypothesis, you need to select the rules whose consequences entail the consequence of the hypothesis.

Hypothesis:
In La Liga 2021-2022, Real Madrid ranks higher than Barcelona.

Premises:
1: A La Liga soccer team ranks higher than another if it receives more points.
2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.
3: Real Madrid and Barcelona are both La Liga soccer teams.
4: In La Liga 2021-2022, Real Madrid recieves 86 points and Barcelon recieves 73 points.
5: In La Liga 2021-2022, Real Madrid and Barcelona both recieve 3 points from the games between them.

Rule Selection:
Premise 1, A La Liga soccer team ranks higher than another if it receives more points. or
Premise 2: If two La Liga soccer teams recieve the same points, the team which recieves more points from the games between the two teams ranks higher.

The prompt for Logical Abduction:

Task Description:
Analyze the plausible explanations for the selected rules.

Plausible Reasons:
According to Premise 1, if we want to prove a La Liga soccer team ranks higher than another, we need to prove the La Liga soccer team receives more points. or
According to Premise 2, if we want to prove a La Liga soccer team ranks higher than another, we need to prove two La Liga soccer teams receive the same points, and one of them receives more points from the games between the two teams.

The prompt for Confusion Check:

Task Description:
Check whether each reasoning step produces consistent deduction or induction results after applying the selected rules.

Abduction Results:
According to Premise 1, if we want to prove a La Liga soccer team ranks higher than another, we need to prove the La Liga soccer team receives more points. or
According to Premise 2, if we want to

prove a La Liga soccer team ranks higher than another, we need to prove two La Liga soccer teams receive the same points, and one of them receives more points from the games between the two teams.

Confusion Check:
True

Problem	Example	Dataset
Deductive Reasoning	Premises: 1. The bear sees the mouse. 2. The cow visits the dog. 3. The dog visits the cow. 4. The mouse chases the bear. 5. The mouse chases the dog. 6. The mouse is young. 7. The mouse sees the bear. 8. If the mouse is rough and the mouse sees the cow then the mouse is not round. 9. If someone chases the mouse then they see the mouse. 10. If someone is big then they see the dog. 11. If someone is cold and they do not visit the mouse then the mouse sees the dog. 12. If someone sees the mouse then they are big. 13. If someone is young and they visit the cow then the cow does not visit the dog. 14. If someone sees the dog and the dog visits the cow then the cow sees the mouse. 15. If someone sees the dog then the dog sees the bear. Hypothesis: The bear is not big.	ProofWriter ParaRules
First-Order Logic	Premises: 1: "Stranger Things" is a popular Netflix show 2: If a Netflix show is popular, Karen will binge-watch it 3: If and only if Karen binge-watches a Netflix show, she will download it 4: Karen does not download "Black Mirror" 5: "Black Mirror" is a Netflix show 6: If Karen binge-watches a Netflix show, she will share it to Lisa Hypothesis: Karen will share "Stranger Things" to Lisa.	FOLIO
Analytical Reasoning	Premises: 1. The organizer of a reading club will select at least five and at most six works from a group of nine works. 2. The group consists of three French novels, three Russian novels, two French plays, and one Russian play. 3. No more than four French works are selected. 4. At least three but no more than four novels are selected. 5. At least as many French novels as Russian novels are selected. 6. If both French plays are selected, then the Russian play is not selected. Hypothesis 1: No Russian novels are selected. Hypothesis 2: Exactly one French novel is selected. Hypothesis 3: All three plays are selected. Hypothesis 4: All three Russian novels are selected. Hypothesis 5: All five French works are selected.	AR-LSAT

Table 5: A summary of the examples we use for the four datasets in our study, representing three different types of logical reasoning problems.