# Exploring the Role of Reasoning Structures for Constructing Proofs in Multi-Step Natural Language Reasoning with Large Language Models

Anonymous ACL submission

## Abstract

When performing complex multi-step reasoning tasks, the ability of Large Language Models (LLMs) to derive structured intermediate proof steps is important for ensuring that the models truly perform the desired reasoning. This paper is centred around a focused study: whether the current state-of-the-art LLMs can leverage the structures in a few examples and benefit from them to construct the proof structures when performing complex natural language reasoning. Our study specifically focuses on structure-aware demonstration and structure-aware pruning. We demonstrate that both of them help improve performance. We provide a detailed analysis to help understand the results.

## 1 Introduction

Large language models (LLMs) have played an essential role in a wide range of applications (Nori et al., 2023; Savelka et al., 2023; Wang et al., 2023; Qin et al., 2023) including as intelligent agents (Liu et al., 2023; Cheng et al., 2022). Their ability to perform complex multi-step reasoning has become critical (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023; Besta et al., 2023; Lei et al., 2023; Dalvi et al., 2021; Ribeiro et al., 2023; Saparov and He, 2023). In complex multi-hop reasoning tasks, the proof steps often form a graph but not just a chain. The capability to construct correct, structured proofs is essential for ensuring that LLMs perform the desired reasoning. The structured intermediate proof steps are also important for the explainability of the reasoning models (Dalvi et al., 2021; Ribeiro et al., 2023).

In this paper, we perform a focused study, providing evidence to help understand whether the state-of-the-art LLMs, such as GPT-4, can leverage the given proof structures of several similar examples and benefit from them to construct the proof structure for the reasoning problem under study. We investigate this in the in-context learning (Brown et al., 2020) setup because in many real-life applications, the number of available examples with proof structures is small. Specifically, we consider two key components that can utilize the known proof structures: (i) *demonstration*, and (ii) *proof path search and pruning*. Accordingly, we equip the state-of-the-art LLMs, *i.e.*, GPT-4 and GPT-3.5, with structure-aware demonstration and structure-aware pruning.

We set up our study in three benchmark datasets, `EntailmentBank` (Dalvi et al., 2021), `AR-LSAT` (Ribeiro et al., 2023) and `PrOntoQA` (Saparov and He, 2023). Our study shows that both structure-aware demonstration and structure-aware pruning improve performance. We provide a detailed analysis to help understand the results.

## 2 Related Work

In complex multi-hop reasoning tasks, the proof steps often form a graph (i.e., a tree or directed acyclic graph (DAG)). It is only recently that researchers have begun to develop evaluation datasets to measure proof structure quality in natural language (Dalvi et al., 2021; Ribeiro et al., 2023). The baseline methods proposed in these papers include smaller models such as T5 (Raffel et al., 2020) or older models such as GPT-3.

Recently, LLMs' reasoning ability has also been significantly improved. Chain-of-thought (CoT) (Wei et al., 2022; Kojima et al., 2022) is arguably the simplest but an effective way to elicit linear reasoning chains of LLMs. Tree-of-thought (Yao et al., 2023) can further provide deeper insights into the model's reasoning structures. However, ToT has been applied to tasks such as game-of-24 and creative writing, but not to natural language entailment and reasoning tasks with complex proof structures. In this paper, we will compare our models to the CoT and ToT models.

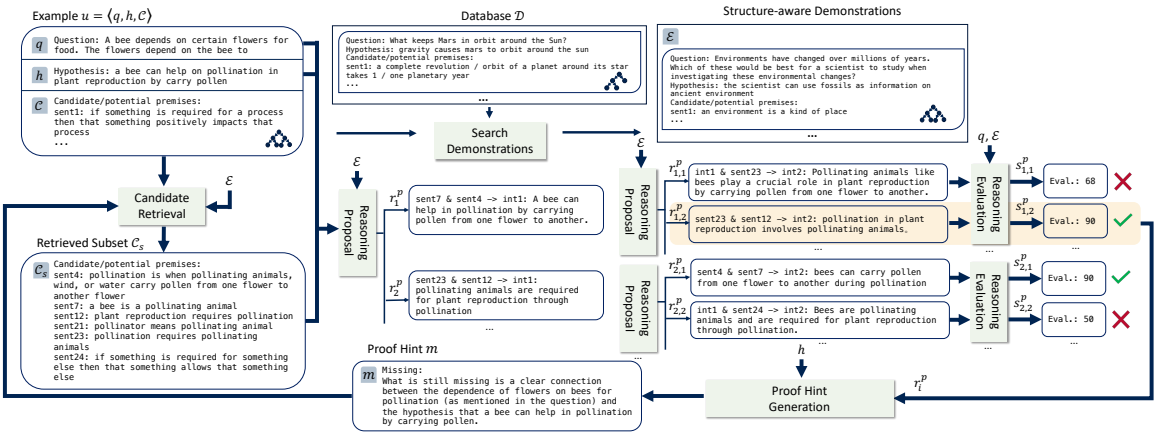


Figure 1: Overview of each module in our proposed framework.

## 3 Method

Given a question $q$, a hypothesis $h$, and a context $\mathcal{C}$ consisting of pieces of evidence or premises, the objective of the task is to provide a proof graph $\mathcal{G}$ from the premises to the hypothesis *if* the hypothesis can be proven. Formally, we denote $p_\theta$ to be a pre-trained language model with parameter $\theta$. Suppose $x = (x_1, \ldots, x_n)$ is a language sequence with $n$ tokens, the probabilistic language model can be written as $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i \mid x_{1,\ldots,i-1})$. Following previous work (Yao et al., 2023), we use the notation $p_\theta^{\text{prompt}}(y|x)$ to represent $p_\theta(y|\text{prompt}(x))$, where $\text{prompt}(x)$ is the input sentences $x$ wrapped with the prompt instructions and templates; $y$ is the output. The overall architecture of our model is depicted in Figure 1.

**Structure-aware Demonstration.** Given an example $u = \langle q, h, \mathcal{C} \rangle$ and a database $\mathcal{D}$ where instances feature structured proofs, the search for most similar demonstrations $\mathcal{E}$ can be expressed as $\mathcal{E} = \mathcal{S}(u, \mathcal{D})$. Usually, $\mathcal{S}$ is defined as manually selecting several fixed demonstrations (Wei et al., 2022; Yao et al., 2023) or choosing the top $k$ demonstrations with the example $u$ based on the similarity (Fu et al., 2022; Liu et al., 2022). In this paper, we hypothesize that the proof structure of similar examples can help LLMs construct a structured proof for the target problem. Specifically, we consider two key components that can utilize the known proof structures: demonstration, and proof-path pruning. At the initial stage, we prompt LLMs to provide a guessed proof graph $\mathcal{G}_u^a$ of the example $u$ which is used to find the most similar examples as the demonstrations. As the proof moves forward, the partially constructed proof tree will be simply merged into the guessed tree (other methods can be considered here and we will leave that as future work). Specifically, we use the graph attention network (GATv2) (Brody et al., 2022) and calculate the similarity between the proof graph $\mathbf{E}_u^a$ and each candidate demonstration $v$'s proof graph $\mathbf{E}_v$, which considers both the structure and content of the graphs. We choose the candidates with the higher similarity scores as the demonstrations.

**Candidate Retrieval.** Given $u = \langle q, h, \mathcal{C} \rangle$, a proof hint $m$ (discussed below), and a set of selected demonstrations $\mathcal{E}$, the candidate retrieval component aims to retrieve a set of most relevant evidence $\mathcal{C}_s$: $\mathcal{C}_s = \{o(z_i)\}_{i=1}^{k}$; $z_i \sim p_\theta^{\text{Retrieve}}(z|q, h, \mathcal{C}, m, \mathcal{E})$, where $z_i$ represents the generated output, which is sampled from the generative language model $p_\theta$ that takes in the retrieval prompt. Because $z_i$ contains the needed evidence sentence *id*, we need to extract the *id* from it, the $o(\cdot)$ represent that extraction process. For detailed examples of prompts, refer to Appendix I.1. As a result, $\mathcal{C}_s$ represents a set of retrieved evidence after the retrieval ran $k$ times. The proof hint $m$ measures the difference between the current proof status and the hypothesis, which will be discussed later in the *proof hint generation* section. Note that the retrieval models can be replaced by search engine, but in our study, we use a set of given candidate evidences since our focus is on reasoning itself.

**Reasoning Step Proposal.** We then prompt LLMs themselves to provide the most plausible proposal for the next reasoning steps. Formally, given $\langle q, h, \mathcal{C}_s, \mathcal{E} \rangle$, the output is reasoning candidates $r$ for the subsequent reasoning step.

$$r_i \sim p_\theta^{\text{Propose}}(r|q, h, \mathcal{C}_s, \mathcal{E}) \quad (1)$$

Then we obtained a set of reasoning steps: $\mathcal{P} = \{r_i\}_{i=1}^{k'}$. The output $r_i$ is parsed to transform the output text into a structured step $r_i^p$ such as $\text{sent}_i$ & $\text{sent}_j \rightarrow \text{int}_k$. In Figure 1, we can see one such step is $\text{sent}7$ & $\text{sent}4 \rightarrow \text{int}1$, meaning intermediate conclusion $\text{int}1$ is drawn from $\text{sent}7$ and $\text{sent}4$.

**Reasoning Step Evaluation.** Given the current structured reasoning step candidate $r_i^p$ and selected demonstrations $\mathcal{E}$, an LLM measures how likely this reasoning step can reach the final hypothesis with a score $s$.

$$s_i \sim p_\theta^{\text{Eval}}(s_i | r_i^p, \mathcal{E}) \quad (2)$$

where $s_i$ is the language model output from which the score $s_i^p$ is extracted.

**Proof Hint Generation.** This component asks LLMs to compare the intermediate conclusion $r_i^p$ with the target hypothesis $h$ to provide *proof hint*. An example is shown at the bottom of Figure 1.

$$m \sim p_\theta^{\text{Compare}}(m | h, r_i^p) \quad (3)$$

As discussed above, this will be used to guide the model to find the most relevant evidence.

**Structure-aware Pruning** During the forward proving process, we combine the typical breadth-first search (BFS) with the beam search. We maintain $b$ beams of candidates, selecting those with the highest evaluation score from the *Reasoning Evaluation* for each exploration. Furthermore, we delve into the utilization of the problem's structure in this stage. To explore the effect of structure-guiding path selection, we conducted different experiments on how the structures may be used. In our probing experiment (Appendix B) on the dev set of EntailmentBank, we found that models benefit from selecting diverse candidate proof steps; *i.e.*, the models perform better when they are encouraged to select more diverse candidates. That is, two pieces of evidence located on different subtrees are regarded as more diverse than those on the same subtree. Inspired by this, we discourage the model from using the intermediate conclusions which have been used in the previous steps, to avoid growing the tree from the evidence node that has just been generated. We call this implementation the `div` variant, which was used in our final model.

## 4 Experiment Set-Up

**Dataset.** We perform experiments on three datasets, `EntailmentBank` (Dalvi et al., 2021), `AR-LSAT` (Ribeiro et al., 2023) and `PrOntoQA` (Saparov and He, 2023). Details of the datasets can be found in Appendix C.

**Evaluation Metrics.** We evaluate the predicted proof graph $\mathcal{G}_\text{p}$ against the golden graph $\mathcal{G}_\text{g}$ using three metrics: `Evidence F1` (`Ev-F`) (Dalvi et al., 2021), `Proof-F1` (`Pr-F`) (Dalvi et al., 2021), and reasoning `Graph Similarity` (`G Sim`) (Ribeiro et al., 2023). Details can be found in Appendix E.

**Implementation Details.** To ensure replicability, we include implementation and baseline details in Appendix D.

## 5 Experiment Results

Table 1 compares our model with the off-the-shelf Chain-of-Thought (CoT) and Tree-of-Thought (ToT) models. The results show that our models outperform CoT and ToT across the three datasets under different evaluation metrics. Note that the improvement is less in PrOntoQA, which is due to the fact that a larger percentage of data in PrOntoQA has linear reasoning patterns. We refer readers to Appendix G for examples.

**Effect of Proof Structure.** To further understand the effect of proof structures of given examples, we conduct more experiments on EntailmentBank. Table 2 shows the effectiveness of different components of our model. Particularly, our focus is on the variants without structure-aware pruning ("`w/o prun.`") and without structure-aware demonstration ("`w/o demon.`"). We can see that under both GPT3.5 and GPT-4, the structure information contributes to the performance (`Ev-F` and `G Sim` scores dropped without them.). The comparison involving other variants of our model, specifically concerning the hint module and pruning strategies, is detailed in Table 7 in the Appendix.

Table 3 focuses on evaluating the impact of structure-aware demonstration. We compare the structure-aware demonstration (Ours) vs. regular sturctured-unaware simple demonstration ($\text{Ours}_\text{sim}$). We can see that our model is better under both GPT-3.5 and GPT-4. The `oracle` model means we suppose that we know in advance the proof structure of the question under study (which is not true, because the structure needs to be constructed.) and use that to select the most similar demonstrations. We can see that our model is effective as its gap from the Oracle is not large.

**Analysis on Sequential and Non-sequential Reasoning.** The EntailmentBank dataset consists of

| Dataset | Model | GPT-3.5 | | | | | | | GPT-4 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F | G Sim | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F | G Sim |
| EntBank | CoT | .283 | .160 | .204 | .092 | .043 | .059 | .037 | .326 | .270 | .295 | .152 | .110 | .128 | .105 |
| | ToT | .302 | .173 | .220 | .104 | .046 | .064 | .051 | .347 | .293 | .318 | .174 | .132 | .150 | .140 |
| | Ours | **.374** | **.236** | **.289** | **.118** | **.087** | **.100** | **.097** | **.388** | **.327** | **.355** | **.204** | **.162** | **.181** | **.162** |
| AR-LSAT | CoT | .482 | .462 | .472 | .077 | .042 | .054 | .007 | .523 | .492 | .507 | .092 | .068 | .078 | .008 |
| | ToT | .537 | .507 | .522 | .083 | .045 | .058 | .008 | .562 | .510 | .535 | .111 | .063 | .080 | .008 |
| | Ours | **.595** | **.576** | **.585** | **.086** | **.073** | **.079** | **.009** | **.602** | **.588** | **.595** | **.122** | **.075** | **.093** | **.010** |
| PrOntoQA | CoT | .802 | .782 | .792 | .782 | .740 | .760 | .447 | .843 | .811 | .827 | .812 | .800 | .806 | .528 |
| | ToT | .828 | .801 | .814 | .802 | .758 | .779 | .482 | .849 | .825 | .837 | .825 | .800 | .812 | .530 |
| | Ours | **.857** | **.817** | **.837** | **.821** | **.776** | **.798** | **.504** | **.866** | **.838** | **.852** | **.831** | **.821** | **.826** | **.533** |

Table 1: Performance of different models on test sets.

| Model | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F | G Sim |
|---|---|---|---|---|---|---|---|
| GPT-3.5 | | | | | | | |
| Ours | .374 | .236 | .289 | .118 | .087 | .100 | .097 |
| - w/o prun. | .372 | .230 | .284 | .117 | .087 | .100 | .097 |
| - w/o demon. | .332 | .182 | .235 | .107 | .053 | .071 | .067 |
| - w/o hint | .313 | .167 | .218 | .103 | .049 | .066 | .064 |
| - w/o retrieval | .311 | .166 | .216 | .092 | .047 | .062 | .058 |
| GPT-4 | | | | | | | |
| Ours | .388 | .327 | .355 | .204 | .162 | .181 | .162 |
| - w/o prun. | .382 | .311 | .343 | .192 | .159 | .174 | .158 |
| - w/o demon. | .341 | .257 | .293 | .145 | .103 | .120 | .110 |
| - w/o hint | .339 | .223 | .269 | .140 | .088 | .108 | .093 |
| - w/o retrieval | .331 | .201 | .250 | .121 | .057 | .077 | .075 |

Table 2: Cumulative ablation analysis.

| Model | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F | G Sim |
|---|---|---|---|---|---|---|---|
| GPT-3.5 | | | | | | | |
| Ours (w/o prun.) | .372 | .230 | .284 | .117 | .087 | .100 | .097 |
| $\text{Ours}_{\text{sim}}$ (w/o prun.) | .358 | .211 | .266 | .112 | .069 | .085 | .077 |
| $\text{Ours}_{\text{oracle}}$ (w/o prun.) | .392 | .259 | .312 | .153 | .132 | .142 | .138 |
| GPT-4 | | | | | | | |
| Ours (w/o prun.) | .382 | .311 | .343 | .192 | .159 | .174 | .158 |
| $\text{Ours}_{\text{sim}}$ (w/o prun.) | .367 | .258 | .303 | .149 | .121 | .134 | .100 |
| $\text{Ours}_{\text{oracle}}$ (w/o prun.) | .419 | .333 | .371 | .240 | .195 | .215 | .205 |

Table 3: Ablation of demonstration methods.

| Dep. | Sequential | | | | | | Non-sequential | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CoT | | ToT | | Ours | | CoT | | ToT | | Ours | |
| | Ev-F | Pr-F | Ev-F | Pr-F | Ev-F | Pr-F | Ev-F | Pr-F | Ev-F | Pr-F | Ev-F | Pr-F |
| GPT-3.5 | | | | | | | | | | | | |
| 3 | .328 | .138 | .330 | .143 | .330 | .144 | .238 | .108 | .257 | .129 | .282 | .135 |
| 4 | .189 | .070 | .202 | .104 | .202 | .113 | .132 | .068 | .149 | .077 | .175 | .102 |
| 5 | .082 | .003 | .123 | .007 | .125 | .007 | .049 | .002 | .069 | .005 | .093 | .006 |
| 6 | .012 | .000 | .047 | .004 | .047 | .004 | .010 | .000 | .038 | .003 | .045 | .004 |
| 7 | .002 | .000 | .005 | .001 | .006 | .001 | .002 | .000 | .004 | .001 | .005 | .001 |
| GPT-4 | | | | | | | | | | | | |
| 3 | .333 | .150 | .356 | .157 | .357 | .157 | .250 | .121 | .266 | .149 | .297 | .151 |
| 4 | .195 | .145 | .242 | .128 | .242 | .129 | .141 | .074 | .160 | .091 | .181 | .133 |
| 5 | .102 | .010 | .133 | .015 | .135 | .019 | .057 | .005 | .075 | .006 | .100 | .007 |
| 6 | .013 | .001 | .055 | .005 | .059 | .005 | .011 | .001 | .043 | .003 | .050 | .004 |
| 7 | .002 | .000 | .005 | .002 | .006 | .002 | .002 | .000 | .004 | .001 | .005 | .001 |

Table 4: Results of sequential reasoning /non-sequential reasoning.

reasoning problems that only involve sequential reasoning (the ground-truth proof paths of these problems are chains), as well as non-sequential problems. Table 4 depicts the detailed analysis of these two sub-types in the testset. We can see that our method and ToT outperform CoT in both sequential and non-sequential reasoning. Between our model and ToT, they have comparable performance on the sequential subset, while our model performs better than ToT on the non-sequential subset. Regarding different depths, our model also consistently outperforms ToT. In general, we can see that non-sequential reasoning is more challenging than sequential reasoning for all models, due to its higher demands on proof planning and development. The models not only need to explore new potential premises during reasoning but also ensure that the reasoning process remains coherent. Also, the performances of all models decrease on both sequential and non-sequential problems when the depth increases.

## 6 Conclusion

Enabling LLMs to generate their proof structure is critical for the reliability and explainability of such models. By incorporating structure-aware components into state-of-the-art LLMs, we demonstrate that LLMs can benefit from utilizing the given proof structures of similar examples. We find that measuring the gap between the intermediate steps and the final hypothesis can help narrow down the search space and enhance the performance. Further analysis of sequential and non-sequential reasoning reveals that our model offers greater advantages in the more complex task of non-sequential reasoning.

## Limitations

Our proposed method is primarily designed for the natural language reasoning task, especially the task requiring multi-step proof to obtain the final conclusion. We do not test our method on other typess of reasoning, *e.g*. mathematical reasoning and our method only tested on the English reasoning dataset.

One limitation, as mentioned in the paper, is the increased token usage with the potential reasoning branches exploration since the system uses LLM-as-a-service API. Although we apply the beam search strategy over the graph which needs less exploration compared to the naive breadth-first search, the overall cost is still high. We also leverage LLM in several modules in the system, which increases the total API calls as well. Future work will include evaluating the system with open-source LLM to conduct the comparison and save on the budget.

Another limitation is that the current system does not consider the negation proof or the conclusion that cannot be reached. The goal of the current system is to design a system that provides better proof. Proof by negation and other kinds of reasoning, *e.g*. conjunction, disjunction and conditionals, could be extended in future work.

## References


Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of Thoughts: Solving Elaborate Problems with Large Language Models.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kanzhi Cheng, Zheng Ma, Shi Zong, Jianbing Zhang, Xinyu Dai, and Jiajun Chen. 2022. Ads-cap: A framework for accurate and diverse stylized captioning with unpaired stylistic corpora. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 736–748. Springer.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Bin Lei, Chunhua Liao, Caiwen Ding, et al. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *arXiv preprint arXiv:2308.08614*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

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

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-LM: empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*, Singapore.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1339–1384, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Henghui Zhu, Rui Dong, Deguang Kong, Juliette Burger, Anjelica Ramos, zhiheng huang, William Yang Wang, George Karypis, Bing Xiang, and Dan Roth. 2023. STREET: A MULTI-TASK STRUCTURED REASONING AND EXPLANATION BENCHMARK. In *International Conference on Learning Representations*.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Jaromir Savelka, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining legal concepts with augmented large language models (gpt-4). *arXiv preprint arXiv:2306.09525*.

Yue Wang, Hung Le, Akhilesh Gotmare, Nghi Bui, Junnan Li, and Steven Hoi. 2023. CodeT5+: Open code large language models for code understanding and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1069–1088, Singapore. Association for Computational Linguistics.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate problem solving with large language models.

Hongyu Zhao, Kangrui Wang, Mo Yu, and Hongyuan Mei. 2023. Explicit planning helps language models in logical reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11155–11173, Singapore. Association for Computational Linguistics.

## A Related Work

Recently, many researchers have studied how to better leverage large language models (LLMs) to boost the performance of reasoning. Generating the intermediate reasoning steps has shown substantial improvement in many reasoning tasks.

Chain-of-thought (Wei et al., 2022; Kojima et al., 2022) can significantly improve inference accuracy and, at the same time, provide a good explanation for the reasoning process. However, when the complexity of the task increases, it is still hard for them to get a good result. Also, reasoning with LLM in this way is constrained by the LLM's architecture, with its decisions determined by the next token that has the highest probability of prediction. Tree-of-thought (ToT) (Yao et al., 2023) provides deeper insights into the model's reasoning process, offering a clearer view of how it progresses towards its conclusions. Not based on the highest probabilities of the next token, tree-of-thought makes decisions based on the evaluation of each state. However, this method has not yet been applied to the complex structured reasoning task. Following tree-of-thought, graph-of-thought (Besta et al., 2023; Lei et al., 2023) was proposed to enhance the connections between different lines of thought. (Besta et al., 2023) applied the divide-and-merge strategy and solved the simple subtasks instead of the whole complex task, while (Lei et al., 2023) started searching from the target node. However, these methods do not explore ways that are applied in natural language reasoning tasks, especially structured reasoning, and the solution search space is quite small in the tasks they perform.

Narrowed down to the natural language reasoning tasks, Selection-Inference (SI) (Creswell et al., 2022) is a strong modular reasoning approach based on forward chaining. SI contains two modules: selection and inference. The selection module selects a subset of rules and facts that can be used in the reasoning aimed at proving the goal, while the inference module performs reasoning towards the goal with the use of the chosen facts and rules.

Zhao et al. (2023) also found out that the planning stage helps improve the performance of reasoning models. Inspired by these research (Creswell et al., 2022; Zhao et al., 2023). Other works (Olausson et al., 2023; Pan et al., 2023) for forward reasoning include using a logic prover, instead of an LLM, to perform more precise reasoning. There is another line of research (Kazemi et al., 2023; Lei et al., 2023) trying to reason based on the backward chaining. Compared to forward chaining, reasoning from the conclusion to the supporting evidence is significantly more efficient at proof-finding. However, the backward reasoning method by Kazemi et al. (2023) required the dataset to identify the facts and rules which are absent in most datasets and require a lot of annotation labour in the real world.

## B Preliminary Experiments

We conduct two preliminary experiments on the dev set of EntailmentBank with GPT-3.5. For the *Preliminary Experiment I*, we provide all other proofs except for randomly deleting two pieces of evidence. We conduct three deletion strategies: two missing pieces of evidence are in the same subtree and the same reasoning step, in the same subtree but not the same reasoning step, or in a different subtree. Here, we set the depth of the subtree to 2. Specifically, "the same subtree and the same reasoning step" means the two missing pieces of evidence can together form an intermediate conclusion in the proof tree, while "the same subtree but different reasoning step" means that the intermediate conclusion from one missing piece of evidence could be combined with the other missing evidence to obtain another intermediate conclusion. "A different subtree" means the two missing pieces of evidence are not in the same 2-depth subtree. Results in Table 5 show that it is easier for the model to find evidence when they are located in a different proving subtree. We further mimic the practical searching scenario in the *Preliminary Experiment II*, where given one chosen reasoning step, *e.g.* $\text{sent}_4$ & $\text{sent}_5 \rightarrow \text{int}_1$, and missed two different reasoning step among which one is based on the given intermediate conclusion (`reuse_ic`) and the other (`div`) is not, *e.g.* $\text{sent}_3$ & $\text{int}_1 \rightarrow \text{int}_2$ and $\text{sent}_1$ & $\text{sent}_2 \rightarrow \text{int}_3$, we ask the model to provide the prediction of the reasoning step. Table 6 shows that `div` model outperforms `reuse_ic` and thus we apply `div` in the main experiment.

| Model | Ev-P | Ev-R | Ev-F |
|---|---|---|---|
| same subtree and same reasoning step | 0.62 | 0.59 | 0.61 |
| same subtree but different reasoning step | 0.62 | 0.58 | 0.60 |
| different subtree | 0.63 | 0.60 | 0.62 |

Table 5: Result of Preliminary Experiment I

| Model | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F |
|---|---|---|---|---|---|---|
| reuse_ic | 0.57 | 0.42 | 0.49 | 0.35 | 0.19 | 0.25 |
| ind | 0.59 | 0.45 | 0.51 | 0.36 | 0.19 | 0.25 |

Table 6: Result of Preliminary Experiment II

## C Dataset

**EntailmentBank** (Dalvi et al., 2021) not only lists the supporting textural evidence but also offers a hierarchical tree structure showing how the evidence organized to lead to the hypothesis. In the entailment tree, the supporting evidence is the leaf node, the hypothesis is the root node, and the intermediate conclusions are the internal nods. EntailmentBank is also included in the STREET benchmark (Ribeiro et al., 2023). We exclude the cases which only need one reasoning step, *i.e.*, proof depth and length equal to 1.

**AR-LSAT** is the Analytical Reasoning -Law School Admission Test task from the STREET benchmark (Ribeiro et al., 2023). STREET benchmark is a unified multi-task and multi-domain natural language reasoning and explanation benchmark. Unlike other existing question-answering (QA) datasets, models are expected to not only answer questions but also produce step-by-step structured explanations describing how premises in the question are used to produce intermediate conclusions that can prove the correctness of a certain answer. We only include AR-LSAT in addition to EntailmentBank because the other datasets in STREET focus on math problems or the sequence process prediction which needs different prompts, especially for the comparison module, with those regarding to logic reasoning in this paper. For QA datasets, we keep the question as the input $q$ and append the question and correct answer as the input hypothesis $h$.

**PrOntoQA** (Saparov and He, 2023) is a synthetic question-answering dataset, where each example is generated from a synthetic world model represented in first-order logic. The rules applied during the synthetic generation endow it with extractable structural information. We applied a simi-

lar process on this QA dataset as AR-LSAT except that some examples reasoned by negative deduction are removed in this version.

## D Implementation Details

We retrieve 5 times independently and take the union set as the result of the retrieval component. For each step, we propose 3 potential reasoning steps at each node and we keep the beam size as 3 in the breadth-first search. The number of demonstrations is set to 3 for all few-shot models. The max iteration number is set to 5 times of the max reasoning depth for each dataset. We conduct the experiments on gpt-3.5-turbo-0613 version of GPT-3.5 and gpt-4-0125-preview version of GPT-4. For GATv2, we train the model with the training set of EntailmentBank.

**Baseline.** We implement two baselines, CoT and ToT, with three demonstrations. We adapt the ToT to the natural language reasoning task. Specifically, the thought generator outputs the potential reasoning step and the depth-first-search strategy is applied.

## E Evaluation Metrics

We evaluate the predicted proof graph $\mathcal{G}_{\text{pred}}$ against the golden graph $\mathcal{G}_{\text{gold}}$ with three metrics, describing evidence, proof and graph similarity. Unlike previous work, we target the model's ability to provide correct proofs more than the true or false result.

**Evidence.** Following (Dalvi et al., 2021), we perform an evaluation over the chosen evidence to check whether the predicted proof graph uses the correct evidence. Suppose $E_{\text{pred}}$ and $E_{\text{gold}}$ are the selected evidence set for the predicted proof graph $\mathcal{G}_{\text{pred}}$ and the golden graph $\mathcal{G}_{\text{gold}}$ respectively. We compute precision (`Ev-P`), recall (`Ev-R`) and F1 (`Ev-F`) score by comparing $E_{\text{pred}}$ and $E_{\text{gold}}$.

**Proof** Following (Dalvi et al., 2021), we evaluate over individual reasoning steps to check whether the predicted proof graph is structurally correct. Suppose $P_{\text{pred}}$ and $P_{\text{gold}}$ are the reasoning step set for the predicted proof graph $\mathcal{G}_{\text{pred}}$ and the golden graph $\mathcal{G}_{\text{gold}}$ respectively. We compute precision (`Pr-P`), recall (`Pr-R`) and F1 (`Pr-F`) score by comparing $P_{\text{pred}}$ and $P_{\text{gold}}$.

**Graph Similarity.** Following (Ribeiro et al., 2023), we compute the reasoning graph similarity (`G Sim`) $\text{sim}\left(\mathcal{G}_p, \mathcal{G}_g\right)$ by comparing the predicted and the golden reasoning graphs through $\delta\left(\mathcal{G}_p, \mathcal{G}_g\right)$ where $\delta$ is a graph edit distance function using insertion, deletion and substitution as elementary edit operator over nodes and edges. This can be computed as

$$\text{sim}\left(\mathcal{G}_p, \mathcal{G}_g\right) = 1 - \left[\frac{\delta\left(\mathcal{G}_p, \mathcal{G}_g\right)}{\max\left(|N_p| + |E_p|, |N_g| + |E_g|\right)}\right] \quad (4)$$

## F Other Variants

Table 7 shows the analysis with other variants of our model. The `reuse_ic` variant requires the model to reuse the intermediate conclusion generated in the previous iteration in the 2nd iteration's reasoning, while `div` variant forces the model to explore the reasoning step from the untouched premises. The `w/o hint` includes all modules except the *proof hint generation* module. We modify the prompt in this module into asking the model what is the next step of reasoning in `what's next`. Our findings indicate that the `div` variant has higher performance than the `reuse_ic` and `w/o pruning` variant, showcasing the effectiveness of the structure-aware pruning.

| Model | Ev-P | Ev-R | Ev-F | Pr-P | Pr-R | Pr-F | G Sim |
|---|---|---|---|---|---|---|---|
| GPT-3.5 | | | | | | | |
| Ours (w/o hint) | .359 | .220 | .273 | .100 | .057 | .073 | .072 |
| Ours (what's next) | .363 | .221 | .275 | .108 | .077 | .090 | .089 |
| Ours (w/o pruning) | .372 | .230 | .284 | .117 | .087 | .100 | .097 |
| Ours (reuse_ic) | .363 | .231 | .282 | .117 | .082 | .096 | .095 |
| Ours (div) | .374 | .236 | .289 | .118 | .087 | .100 | .097 |
| GPT-4 | | | | | | | |
| Ours (w/o hint) | .371 | .247 | .297 | .136 | .102 | .117 | .101 |
| Ours (what's next) | .379 | .253 | .303 | .158 | .121 | .137 | .121 |
| Ours (w/o hint) | .382 | .311 | .343 | .192 | .159 | .174 | .158 |
| Ours (reuse_ic) | .380 | .309 | .341 | .192 | .157 | .173 | .158 |
| Ours (div) | .388 | .327 | .355 | .204 | .162 | .181 | .162 |

Table 7: Ablation analysis on EntailmentBank.

## G Case Study

**Proof Hint Generation.** Table 8 shows two examples and we conduct a comparison between the model with or without the *proof hint generation* module. In the first example, both models could make the correct reasoning in the first iteration and the intermediate conclusion finds out that carbon dioxide is required photosynthesis process. Without the *proof hint generation* module, the model could not retrieve the wanted sentences, while with the *proof hint generation* module, the model succeeds in focusing on the missing relationship with

'step'. Similarly, in the second example, both models could correctly retrieve sent6. However, with the *proof hint generation* module, the model cares more about the information of Earth itself, not the moon. The examples show that the *proof hint generation* module explicitly asks the model to think about the missing part between the current intermediate conclusion and the final goal and the model could retrieve relevant information based on this action.

**Structure-aware Demonstration.** Table 9 shows the example with structure-aware demonstrations. For the page limit, we only show the proof structure of one demonstration in the table. We observe that the model is prone to providing the proof that is structurally similar to the proofs given in the demonstration and we attribute the performance improvement brought by structure-aware demonstrations to this observation.

## H Computation Cost

We observe that the cost of experimenting is higher than the baselines. We leverage the language model in several different modules and apply the beam search strategy in the breadth-first search. We keep $a$ most promising states per step and $b$ beams of candidates with the highest evaluation score for each exploration in the beam search strategy. Although we cut down the total number of explored cases of $n$ reasoning iterations to $a+(n-1)\times b\times a$ from $a + a^2 + a^3 + \cdots + a^n$ because of the beam search over the tree, it is still higher than CoT (1) and ToT ($n \times a$). Table 4 shows our benefits on non-sequential reasoning but similar performance with ToT on sequential reasoning. Considering the computation cost, our model might not be a good choice if most data belongs to sequential reasoning.

## I Example Prompts

We provide three demonstrations in all few-shot models, but we only show one in the example in this section.

### I.1 Candidate Retrieval

System: Below, you are given a question, a hypothesis and a set of candidate premises. You are required to select a small set of candidates (at least provide 3 sentences) to deduce the hypothesis. Please only filter out the sentences that you are sure of.

[example]
Question: What keeps Mars in orbit around the Sun?
Hypothesis: gravity causes mars to orbit around the sun
Candidate/potential premises:
sent1: a complete revolution / orbit of a planet around its star takes 1 / one planetary year
sent2: our sun is located at the center of our solar system
sent3: celestial objects are located in outer space
sent4: gravity causes orbits
sent5: orbit is a kind of characteristic
sent6: a star usually is larger than a planet
sent7: revolving around something means orbiting that something
sent8: a satellite orbits a planet
sent9: uranus is a kind of planet
sent10: planets are found in space
sent11: gravity means gravitational pull / gravitational energy / gravitational force / gravitational attraction
sent12: as mass of a planet / of a celestial body increases, the force of gravity on that planet will increase
sent13: the sun is the strongest source of gravity in the solar system
sent14: a galaxy is made of stars
sent15: orbit means orbital path
sent16: can be means able to be
sent17: celestial bodies / celestial objects are found in space
sent18: satellites are found in space
sent19: proxima centauri is a kind of star
sent20: planets in the solar system orbit the sun
sent21: mars is a kind of planet
sent22: venus is a kind of planet
sent23: mars is located in the solar system
sent24: isaac newton discovered the theory of gravity
sent25: a comet is a kind of celestial body

Retrieval sentences (at least 3): sent4, sent20, sent21, sent23
Proof: sent20 & sent4 -> int1: gravity causes the planets in the solar system to orbit the sun; sent21 & sent23 -> int2: mars is a planet in the solar system; int1 & int2 -> hypothesis;
—-
[Question]
Question: A bee depends on certain flowers for food. The flowers depend on the bee to

| Model | Previous Intermediate Conclusion | Retrieved Premises |
|---|---|---|
| | Case 1:<br>Hypothesis: taking in carbon dioxide is a step in the photosynthesis process<br>sent2: when carbon dioxide in the atmosphere is absorbed by plants , the amount of carbon dioxide in the atmosphere is reduced in the atmosphere<br>sent4: if something is required in a process then obtaining that something is a step in that process<br>sent8: taking something in is a kind of method for obtaining that something<br>sent17: a step in a process means a stage in a process<br>sent25: taking in and releasing means exchanging<br>Expected: sent4 & (sent9 & sent19) | |
| w/o hint | sent9 & sent19 -> int1: Carbon dioxide is a required raw material in the photosynthesis process. | sent2, sent12, sent17, sent25 |
| w/ hint | sent9 & sent19 -> int1: Carbon dioxide is required as a raw material in the photosynthesis process. | sent2, **sent4**, sent8, sent17 |
| | Case 2:<br>Hypothesis: the difference between the earth and the moon is that the moon revolves around a planet<br>sent1: celestial bodies / celestial objects are found in space<br>sent3: earth is a kind of planet<br>sent4: moons / comets / planets are part of the solar system<br>sent6: the earth revolves around the sun<br>sent9: the sun is a kind of star<br>sent10: a moon is a kind of satellite<br>sent11: revolving around means orbiting<br>sent23: the moon is earth 's moon<br>sent24: a celestial body travelling around another celestial body means that celestial body completes a cycle around that other celestial body<br>Expected: (sent6 & sent9) & ((sent25 & sent3)& sent11) | |
| w/o hint | sent3 & sent25 -> int1: The Earth and the Moon are both planets, but the Moon orbits the Earth. | sent1, sent4, **sent6**, sent10, sent23, sent24 |
| w/ hint | sent3 & sent25 -> int1: Earth is a planet and the Moon orbits it. | sent3, **sent6**, **sent9**, sent10, **sent11** |

Table 8: 2nd iteration of reasoning examples for w/ and w/o proof hint generation module

| Model | Demonstration Proof | Final Proof |
|---|---|---|
| | Hypothesis: wood boards are a kind of building material that is made of a renewable natural resource<br>sent3: wood boards are made of wood<br>sent7: wood is a renewable resource<br>sent8: a renewable resource is a kind of natural resource<br>sent17: wood boards can be used to build houses<br>sent19: a house is a kind of building<br>sent23: building materials are used to build buildings<br>Expected: ((sent19 & sent23) & sent17) & ((sent7 & sent8) & sent3) | |
| Text-aware Demonstration | (sent25 & sent3) & sent2 | ((sent7 & sent8) & sent17) |
| Structure-aware Demonstration | ((sent26 & sent3) & sent1) & ((sent7 & sent9) & sent10) | ((sent19 & sent23) & sent17) & ((sent7 & sent8) & sent3) |

Table 9: Final proof for structure-aware demonstration and demonstration with the most similar context

Hypothesis: a bee can help on pollination in plant reproduction by carry pollen
Candidate/potential premises:
sent1: if something is required for a process then that something positively impacts that process
sent2: pollinated means after pollination
sent3: pollinating is a kind of function
sent4: pollination is when pollinating animals, wind, or water carry pollen from one flower to another flower
sent5: if something causes a process then that something is required for that process
sent6: seed dispersal has a positive impact on a plant / a plant's reproduction sent7: a bee is a pollinating animal
sent8: flowers sometimes become fruits after pollination
sent9: if a living thing requires something then that something has a positive impact on that living thing
sent10: flowers are a source of fruit
sent11: if something is required then that something must be provided
sent12: plant reproduction requires pollination
sent13: needing something means depending on that something
sent14: to be used for something means to be required by that something
sent15: flowers often have a sweet smell to attract pollinators
sent16: to carry means to transport
sent17: a bird is a pollinating animal
sent18: a flower's purpose is to produce seeds
sent19: when pollen sticks to a hummingbird, that pollen will move to where the hummingbird moves
sent20: plant requires seed dispersal for reproduction
sent21: pollinator means pollinating animal
sent22: seed dispersal is a kind of method of sexual reproduction
sent23: pollination requires pollinating animals
sent24: if something is required for something else then that something allows that something else
sent25: requiring something means needing that something

Retrieval sentences (at least 3):

### I.2 Reasoning Step Proposal

System: Provide me several sentences with the sentence number and one intermediate conclusion that are possible to be used in the next step in this small set. If the deduction reaches the hypothesis, tell me 'Finish'; otherwise please provide the

(intermediate) conclusion.

[example]
Question: What keeps Mars in orbit around the Sun?
Hypothesis: gravity causes mars to orbit around the sun
Candidate/potential premises:
sent4: gravity causes orbits
sent5: orbit is a kind of characteristic
sent12: as mass of a planet / of a celestial body increases, the force of gravity on that planet will increase
sent20: planets in the solar system orbit the sun
sent21: mars is a kind of planet
sent22: venus is a kind of planet
sent23: mars is located in the solar system
sent24: isaac newton discovered the theory of gravity

Possible Next Reasoning: sent20 & sent4 -> int1: gravity causes the planets in the solar system to orbit the sun
—
[Question]
Question: A bee depends on certain flowers for food. The flowers depend on the bee to
Hypothesis: a bee can help on pollination in plant reproduction by carry pollen
Candidate/potential premises:
sent4: pollination is when pollinating animals, wind, or water carry pollen from one flower to another flower
sent7: a bee is a pollinating animal
sent12: plant reproduction requires pollination
sent21: pollinator means pollinating animal
sent23: pollination requires pollinating animals
sent24: if something is required for something else then that something allows that something else

Possible Next Reasoning:

### I.3 Reasoning Step Evaluation

System: Evaluate whether these intermediate conclusions could reach the hypothesis with candidates. Provide me the number of possibilities (0-99) of these intermediate conclusions: Surely: 85-99, Likely: 50-84, Impossible: 0-49

[example]
Question: What keeps Mars in orbit around the Sun?
Hypothesis: gravity causes mars to orbit around the sun
Candidate/potential premises:
sent4: gravity causes orbits
sent5: orbit is a kind of characteristic
sent12: as mass of a planet / of a celestial body increases , the force of gravity on that planet will increase
sent20: planets in the solar system orbit the sun
sent21: mars is a kind of planet
sent22: venus is a kind of planet
sent23: mars is located in the solar system
sent24: isaac newton discovered the theory of gravity

Possible Next Reasoning: sent20 & sent4 -> int1: gravity causes the planets in the solar system to orbit the sun
Evaluate: 99
—
[Question]
Question: The body of a fish is covered by scales for
Hypothesis: scales are used for protection by fish
Candidate/potential premises:
sent1: a fish is a kind of scaled animal
sent8: scales are a covering around the body of a scaled animal
sent12: scales are used for protection by scaled animals
sent15: protecting is a kind of function

### I.4 Proof Hint Generation

System: Compare the intermediate conclusion with the hypothesis and the question, and provide me one sentence of what is still missing.

Question: The body of a fish is covered by scales for
Hypothesis: scales are used for protection by fish
Intermediate Conclusion: int1: scales cover the body of a fish
Missing: