FRVA: Fact-Retrieval and Verification Augmented Entailment Tree Generation for Explainable Question Answering

Yue Fan[1], **Hu Zhang**[1,2*], **Ru Li**[1,2*], **Yujie Wang**[1], **Hongye Tan**[1,2], **Jiye Liang**[1,2]
[1]School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]Key Laboratory of Computational Intelligence and Chinese Information
Processing of Ministry of Education, Shanxi University, Taiyuan, China
yuefan24@163.com, {zhanghu,liru}@sxu.edu.cn, init_wang@foxmail.com,
{tanhongye,ljy}@sxu.edu.cn

Abstract

Structured entailment tree can exhibit the reasoning chains from knowledge facts to predicted answers, which is important for constructing an explainable question answering system. Existing works mainly include directly generating the entire tree and stepwise generating the proof steps. The stepwise methods can exploit combinatoriality and generalize to longer steps, but they have large fact search spaces and error accumulation problems resulting in the generation of invalid steps. In this paper, inspired by the Dual Process Theory in cognitive science, we propose FRVA, a Fact-Retrieval and Verification Augmented bidirectional entailment tree generation method that contains two systems. Specifically, System 1 makes intuitive judgments through the fact retrieval module and filters irrelevant facts to reduce the search space. System 2 designs a deductive-abductive bidirectional reasoning module, and we construct cross-verification and multi-view contrastive learning to make the generated proof steps closer to the target hypothesis. We enhance the reliability of the stepwise proofs to mitigate error propagation. Experiment results on EntailmentBank show that FRVA outperforms previous models and achieves state-of-the-art performance in fact selection and structural correctness.

1 Introduction

Automated reasoning, the process of reasoning from given *explicit* knowledge to generate valid conclusions, has always been a goal pursued in Artificial Intelligence (Wos, 1985; Mercier and Sperber, 2011; Xu et al., 2023). Interpreting the reasoning process from question to answer can provide a human-understandable inspection for the QA system, which can help improve the debuggability and trustworthiness of the model.

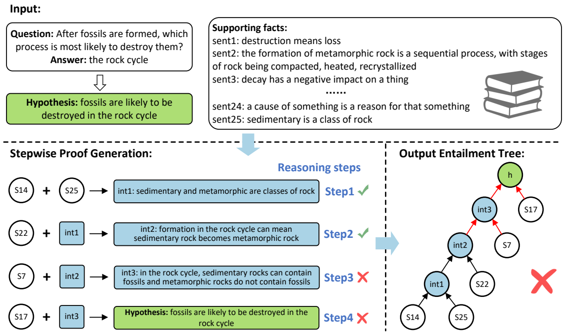

Input:
Question: After fossils are formed, which process is most likely to destroy them?
Answer: the rock cycle
Hypothesis: fossils are likely to be destroyed in the rock cycle
Supporting facts:
sent1: destruction means loss
sent2: the formation of metamorphic rock is a sequential process, with stages of rock being compacted, heated, recrystallized
sent3: decay has a negative impact on a thing
......
sent24: a cause of something is a reason for that something
sent25: sedimentary is a class of rock
Stepwise Proof Generation:
Reasoning steps
S14 + S25 → int1: sedimentary and metamorphic are classes of rock
Step1 ✓
S22 + int1 → int2: formation in the rock cycle can mean sedimentary rock becomes metamorphic rock
Step2 ✓
S7 + int2 → int3: in the rock cycle, sedimentary rocks can contain fossils and metamorphic rocks do not contain fossils
Step3 ✗
S17 + int3 → Hypothesis: fossils are likely to be destroyed in the rock cycle
Step4 ✗
Output Entailment Tree:
h
int3
S17
int2
S7
int1
S22
S14
S25
✗

Figure 1: The task of entailment tree generation. Given question-answer pairs, hypotheses (green boxes), and supporting fact sentences, the model needs to generate tree-structured reasoning chains and natural language intermediate conclusions (blue boxes).

Explainable Question Answering (XQA) aims to answer a question and give a corresponding explanation (Schuff et al., 2020), and current related works focus on three aspects: extracting keywords or sentences that contain the answer (Yang et al., 2018; Serrano and Smith, 2019), constructing the multi-hop explanation chain (Xu et al., 2021), and generating free-form explanations (Wei et al., 2022; Yoran et al., 2023). Among the various explanation methods, Dalvi et al. (2021) proposes to construct multi-step reasoning chains in the form of entailment tree, which shows *how* hypotheses (question + answer) can be explained from simple textual evidence. As shown in Figure 1, given a hypothesis and a set of supporting facts, the goal is to generate an entailment tree consisting of multiple premise proof steps, which has better expressive compared to other methods. For example, for the question "*After fossils are formed, which process is most likely to destroy them?*", the reasoning process can explain *why* the answer "*the rock cycle*" is predicted.

One line of previous work converts a multi-step entailment tree into a linearized sequence (Dalvi

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 9111–9128
August 11-16, 2024 ©2024 Association for Computational Linguistics

et al., 2021; Tafjord et al., 2021; Bostrom et al., 2021), which directly outputs the entire reasoning chain using the sequence-to-sequence model (seq2seq). Although this method simplifies the generation task, it is challenging to guarantee structural correctness when the steps are longer. Therefore, subsequent works propose to use stepwise generation by splitting the tree into multiple steps and training the model to perform single-step reasoning (Liu et al., 2022; Yang et al., 2022; Hong et al., 2022; Morishita et al., 2023; Su et al., 2023; Yuan et al., 2024). Compared to generating the entire reasoning chain directly, the stepwise generation can more fully exploit the combinability of proofs, which makes the model easier to learn and generalize to longer proof steps.

However, existing stepwise methods still face two challenges: (1) combination failure due to the large fact search space and (2) generation invalid steps due to error accumulation. The current works perform single-step reasoning mainly using deductive reasoning, which is a bottom-up process that requires iterative search and generation of intermediate conclusions from known facts until the hypothesis is proved. The search space grows with new conclusions to join the known facts, easily leading to combinatorial failure. Moreover, the stepwise methods have the problem of error accumulation, e.g., the third step "`sent7 & int2 -> int3`" in Figure 1 is not correct, which affects the next step generation. The model may quickly generate invalid proofs with the increase of proof steps, which leads to incorrectness of the tree structure.

To tackle the above problems, we propose FRVA, a bidirectional stepwise proofs generation method. We simulate the cognitive process of human reasoning and divide the entailment tree generation into two systems. In System 1, we design a fact retrieval module, which scores all candidate facts from a global perspective and initially eliminates knowledge facts with low relevance to the hypothesis. In System 2, we propose a deductive-abductive bidirectional proof generator. In this process, forward deductive and backward abductive reasoning generate candidate steps and verify each other to satisfy bidirectional consistency. Meanwhile, we design a multi-view contrastive loss integrating the local level and global level information to pull the semantic distance between the golden facts and hypotheses, which can help the model generate proof steps closer to the hypotheses.

We improve the reliability of the stepwise proof step in terms of both supportiveness (cross-verification) and similarity (contrastive learning). Finally, we align and fuse the bidirectionally generated structure trees and select the best proof tree. Experiments on EntailmentBank show that FRVA can obtain more reliable and valid reasoning paths and outperform existing baselines regarding fact selection and structural correctness. We also demonstrate that the backward reasoning remains valid in proof discovery. In summary, our contributions are threefold:

- We design the fact retrieval module which dynamically filters irrelevant knowledge facts to the target hypothesis to reduce the search space.
- We propose a bidirectional proof generation method based on deductive-abductive reasoning, which mitigates error propagation from both supportability and similarity perspectives through cross-verification and multi-view contrastive learning.
- Through extensive experiments and analysis, we shed light that FRVA can generate more correct and reliable reasoning paths and improve performance.

2 Related Work

Explainability in Question Answering. Recent works explore different forms of QA explanation, including retrieval to obtain multiple supporting facts related to the question or answer (Yang et al., 2018; Serrano and Smith, 2019; Niu et al., 2020; Schuff et al., 2020; DeYoung et al., 2020; Valentino et al., 2021). However, this method does not show how the evidence reasoning leads to the question hypothesis. In addition, some researchers use multi-hop reasoning chains to combine model outputs with human-understandable explanations (Yang et al., 2018; Jhamtani and Clark, 2020; Xu et al., 2021), where the explanation consists of a series of inferences rather than a simple textual explanation. Some works explain QA systems in a generative manner, which connects question to answer as explanations in the free-form text (Wei et al., 2022; Yoran et al., 2023; Wang et al., 2023), but this explanation is not always reasonable and faithful. Our work generates explanation in the form of multi-step entailment trees proposed by Dalvi et al. (2021). This form can show *which* facts and *how* to combine them to produce new intermediate conclusions and ultimately prove the target hypothesis, which makes it easier to check the reasoning behind the model.

Structured entailment tree generation. Existing methods can be categorized into two types: single-step direct generation (Tafjord et al., 2021; Dalvi et al., 2021) and stepwise generation (Liu et al., 2022; Hong et al., 2022; Qu et al., 2022; Fei et al., 2022; Bostrom et al., 2022; Morishita et al., 2023; Kazemi et al., 2023; Creswell et al., 2023; Zhao et al., 2023; Hong et al., 2023; Yuan et al., 2024; Chen et al., 2024). In direct generation methods, the entailment tree is modeled as a linearized sequence, and the entire reasoning chain is directly generated in one shot using a seq2seq model. Such methods simplify the task but make generating entailment trees with longer steps difficult.

A series of recent works exploring stepwise generation methods include two main core components: fact sentence selection and intermediate conclusion generation. RLET (Liu et al., 2022) introduces reinforcement learning to entailment tree generation for the first time. NLProofs (Yang et al., 2022) introduces an independent verification mechanism to check the validity of the proof step and prevent the generation of hallucinatory invalid steps. IRGR (Ribeiro et al., 2022) proposes an iterative retrieval-generation architecture, which combines the generated intermediate conclusions and premises for retrieval to generate entailment trees better. FLD (Morishita et al., 2023) proposes a deductive rule generation method based on formal logic theory, which further enhances the deductive reasoning ability of the model. FAME (Hong et al., 2023) uses Monte Carlo planning to implement faithful question answering, but they focus on QA while we focus on explainability.

Prior works focus on forward reasoning, and MetGen (Hong et al., 2022) further reveals the effectiveness of backward reasoning. However, they discretely perform single-step proof generation (i.e., it needs to enumerate different combinations of facts to select the best single step). Our method is also a bidirectional proof generation that enhances the reliability of stepwise proof steps by cross-verification and multi-view contrastive learning during the generation process.

3 Task Formulation

As shown in Figure 1, the input of the entailment tree generation task consists of the hypothesis H and a set of supporting facts $S = \{\mathrm{sent}_1, \mathrm{sent}_2, \ldots, \mathrm{sent}_n\}$. H is a declarative sentence consisting of question-answer pairs (QA), which can be proved by constructing one or more reasoning steps with the knowledge from S. The output is an entailment tree T where root nodes are the target hypotheses H, leaf nodes are the facts chosen from S, and intermediate nodes are the conclusions int_i generated by the model. The tree T is considered valid if all non-leaf nodes can be effectively entailed by their children. The entailment tree can be formalized as $T = \{H, \mathcal{J}, \mathcal{M}, \mathcal{V}\}$, where the leaf node $l_i \in \mathcal{J} \in S$, the intermediate conclusion $\mathrm{int}_i \in \mathcal{M} \notin S$, and the proof step $\mathrm{step}_i \in \mathcal{V}$ (a step consists of an intermediate conclusion and its premise, e.g., "`sent1 & sent2 -> int1`"). We denote the gold entailment tree as T_{gold} and all the leaf nodes it contains as the gold fact S_{gold}. According to the different composition of the input supporting facts S, it is divided into three increasingly difficult tasks:

Task1 (no-distractor): Inputs = H + QA + leaf sentences S_{gold}

Task2 (distractor): Inputs = H + QA + leaf sentences S_{gold} + 15-20 distractor sentences

Task3 (full-corpus): Inputs = H + QA + a corpus C

4 Method

The reasoning process of FRVA is shown in Figure 2, which consists of two systems. System 1 retrieves relevant facts, and System 2 contains the bidirectional proof generator and the proof alignment and fusion module. We introduce the above modules in section 4.1, 4.2 and 4.3.

4.1 Fact Retrieval

In the tree generation process, the search space of reasoning grows with the increase of input facts, and it is easy to generate complex tree branches. Therefore, we first filter irrelevant facts from S to reduce the search space and help efficiently and accurately generate the proof steps.

Specifically, we aim to score all candidate facts based on their relevance to hypotheses. Firstly, we splice hypothesis H and fact sent_i and input them into the pre-training model for encoding. The sentence embedding $\boldsymbol{h}$ of hypothesis H and $\boldsymbol{f_i}$ of each fact are obtained by average pooling. Then, we use a multi-layer perceptron to obtain the correlation

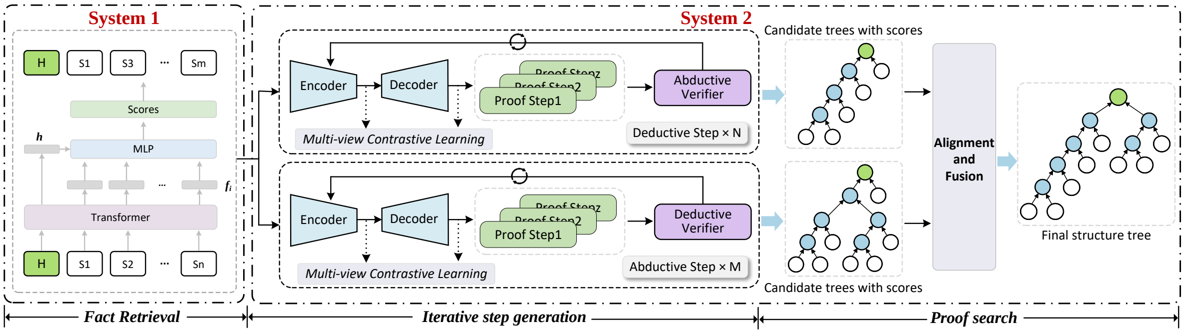


Figure 2: The reasoning process of FRVA. The goal is to obtain one or more proofs steps based on given facts to prove the hypothesis and generate a valid structured tree.

score of each fact:

$$\mathcal{R}_{fact}(\mathrm{sent}_i) = \sigma(\mathrm{MLP}_{fact}([\boldsymbol{h}, \boldsymbol{f_i}])) \tag{1}$$

where σ is the sigmoid activation function, $[\cdot]$ represents the splicing operation, and MLP_{fact} is a multi-layer perceptron composed of two-layer feed-forward neural networks. In the tree structure, a fact with a smaller depth should be closer to the target hypothesis than a larger one (Hong et al., 2022). Therefore, we use a marginal ranking as the loss of the retriever during training:

$$\begin{aligned}\mathcal{L}_{fact} =& \frac{1}{N_1} \sum_{\mathrm{s}_1^+, \mathrm{s}_2^+ \in \mathcal{S}_{gold}} \psi(\mathcal{R}_{fact}(\mathrm{s}_1^+), \mathcal{R}_{fact}(\mathrm{s}_2^+), \mathcal{W}_{fact}) \\ &+ \frac{1}{N_2} \sum_{\mathrm{s}^- \in \mathcal{S}_{gold}} -\log\left(1 - \mathcal{R}_{fact}(\mathrm{s}^-)\right)\end{aligned} \tag{2}$$

where s_i denotes sent_i, s_1^+ is the fact with depth less than s_2^+ in the golden tree, s^- is the rest of irrelevant facts in S, N_1 is the number of $(\mathrm{s}_1^+, \mathrm{s}_2^+)$ pairs, N_2 is the number of disturbing facts, $\mathcal{W}_{fact}$ is the margin of fact, $\psi(x_1, x_2, \mathcal{W}) = \max(0, x_2 - x_1 + \mathcal{W})$ is the marginal loss. Based on the above fact scores, we design a threshold γ to dynamically screen each candidate fact and obtain a new set S' containing $m (m < n)$ facts.

4.2 Proof Generation

We build a bidirectional proof generator and design cross-verification and multi-view contrastive loss to enhance the reliability of proof steps from the perspective of supportability and similarity.

Deductive-Abductive Generation (supportive): The backward reasoning is a top-down process, which starts with a question and iteratively decomposes the sub-questions until all of them can be solved with the existing knowledge. We use a pre-trained T5 model (Raffel et al., 2020) as the stepwise deductive and abductive proofs generator. We construct deductive and abductive training data from the proof step of the golden tree. **For deductive**, we construct it in a bottom-up manner, where the inputs of each step include the outputs of the previous steps in addition to the hypothesis H and the fact set S'. As shown in Figure 1, the output of the first step model is: "`$proof$ = sent14 & sent25 -> int1`". We then use that step as part of the model input for the next step and iteratively generate proof steps until the output "`hypothesis`" terminates.

For abductive, we construct training data from the golden tree in a top-down manner. Taking the deductive step "`sent7 & int2 -> int3`" in the golden tree in Figure 1 as an example, the abduction model should output: "`$proof$ = int3 & sent7 -> int2`". In this case, `int3` is the intermediate conclusion generated by the previous abductive model. Then, we use this step as input to the next step of the abductive model and iteratively generate the step, terminating when no more intermediate conclusions are generated.

In the above way, we construct training data and train the deductive and abductive generators:

$$\begin{aligned}G_{\mathrm{ded}}^t &= \mathrm{Encoder} - \mathrm{Decoder}(H, S', G_{\mathrm{ded}}^{1:t-1}) \\ G_{\mathrm{abd}}^t &= \mathrm{Encoder} - \mathrm{Decoder}(H, S', G_{\mathrm{abd}}^{1:t-1})\end{aligned} \tag{3}$$

where t represents the number of steps in iterative generation, and G^t represents the proof step generated at time t. We maximize the conditional probability likelihood loss during training. It is worth noting that joint training of deductive and abductive generators may get better gains, and limited by resources, we train the two individually.

In reasoning, we use the trained proofs generators G_{ded}^t and G_{abd}^t to generate u candidate

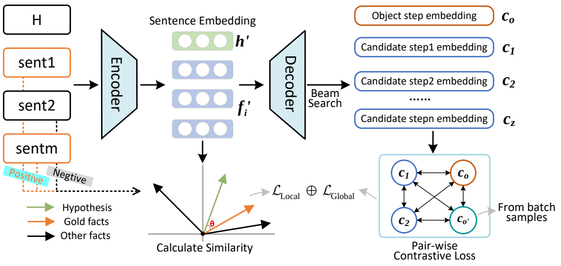


Figure 3: Multi-view contrastive learning method.

steps by beam search respectively, and then design a cross-verification mechanism to select the most appropriate candidate step. Specifically, the candidate deductive step $\text{step}^t_{\text{ded}}$ is checked with the abductive verifier $\mathcal{V}_{\text{abd}}$, and vice versa. Because we think deductive and abductive reasoning should work together, the proof steps with bidirectional consistency are more reliable. For the training of the verifier, we convert the step (`sent1 & sent2 -> inti`) in the golden tree into abductive steps (`inti & sent2 -> sent1, inti & sent1 -> sent2`), and construct deductive pairs (`sent1, sent2, inti`) and abductive pairs (`inti, sent2, sent1`), (`inti, sent1, sent2`), respectively. Then, we jointly learn the deductive and abductive verifier $\mathcal{V}_{\text{ded}-\text{abd}}$ by fine-tuning the pre-trained albert-xxlarge-v2 (Lan et al., 2020) model. We denote both the above deductive and abductive pairs as $(x1, x2, y)$ and use a multilayer perceptron to score the candidate step:

$$\begin{aligned}\mathcal{V}_{\text{ded}}(step_{\text{ded}}) &= \sigma(\text{MLP}_{\text{ded}}([x1; x2])) \\ \mathcal{V}_{\text{abd}}(step_{\text{abd}}) &= \sigma(\text{MLP}_{\text{abd}}([y; x2]))\end{aligned} \tag{4}$$

The golden proof step is used as the positive example, and the negative example is constructed by randomly replacing one of the premises with a non-golden fact in S, and the positive-to-negative proportion is set to 1:1. The loss function for training the verifier is the binary cross-entropy loss.

Multi-view Contrastive Learning (similarity): In the stepwise generator of deductive and abductive, we design a multi-view contrastive loss integrating local and global information to pull the distance between the golden facts and target hypothesis, which makes the generation of proof steps closer to the hypothesis, as shown in Figure 3.

Specifically, we first design the local contrastive information at the Encoder stage. We encode the hypothesis H and fact S' to obtain the sentence representation $\boldsymbol{h}'$ for the hypothesis and $\boldsymbol{f}'$ for each fact. Then, we take the leaf node S_{gold} of the golden tree as positive examples and calculate the cosine similarity between non-golden facts in S' and the hypothesis. The facts with scores greater than μ are also taken as positive examples, and other facts in S' are taken as negative examples:

$$\text{sent}_i = \begin{cases} positive, & \text{if sent}_i \in \mathcal{J}\text{or score}_{cos} > \mu \\ negative, & \text{otherwise} \end{cases} \tag{5}$$

where score_{cos} is the cosine similarity calculation. We construct the local contrastive loss based on the above positive and negative samples:

$$\mathcal{L}_{\text{Local}} = -\log \sum_{\boldsymbol{f_i}'^+ \in S'_{gold}} \frac{\exp\left(\text{sim}(\boldsymbol{h}', \boldsymbol{f}'^+_i)/\tau\right)}{\sum_{\boldsymbol{f_i}' \in S'} \exp\left(\text{sim}(\boldsymbol{h}', \boldsymbol{f}'_i)/\tau\right)} \tag{6}$$

where $\boldsymbol{f_i}'^+$ denotes the embedding of the gold fact sentence, S'_{gold} is the set of gold facts from S', τ is a configurable temperature coefficient hyperparameter, and $\text{sim}(\cdot)$ is the similarity measure function:

$$\text{sim}(x, y) = \frac{x^T y}{\|x\| \|y\|} \tag{7}$$

In the Decoder stage, we design the contrastive loss of global information. For the output of the generation model, the difference in semantic information between sentences cannot be completely separated by positive and negative sample labels. In other words, even negative samples are not necessarily irrelevant to the target output.

Therefore, we obtain z candidate steps and their embedding representations $\{c_1, c_2, \ldots, c_z\}$ through beam search, and then we construct the global contrastive loss by combining $\{c_1, c_2, \ldots, c_z\}$ and the embedding representation c_o of the golden step and the embedding representation $c_{o'}$ of the golden step for the rest of the samples within the batch. In this way, we can construct the sample set $W = \{k_1, k_2, \ldots, k_r\}$ and sample pair $(k^+, k^-) \in W$, where $+$ and $-$ are determined by the ranking of the samples, which is obtained by calculating sequence-level scores (e.g., BLEU) with the target output, and this can reflect the relative differences between the comparison samples. The global contrastive learning loss is:

$$\begin{aligned}\mathcal{L}_{\text{Global}} = \sum_{(k^+, k^-) \in W} & \max(0, \cos(E_x, E_{k^+}) \\ & - \cos(E_x, E_{k^-}) + \mathcal{W}_{pair})\end{aligned} \tag{8}$$

where E_x is the embedding of the target output c_o, E_{k^+}, E_{k^-} is the embedding of the sample, and $\mathcal{W}_{pair}$ is the margin.

Loss Function. In the training stage, we jointly learn the local and global contrastive loss with the stepwise proof generator. For example, the loss of the deductive generator is computed as:

$$\mathcal{L}_{\text{ded}} = \mathcal{L}_{\text{Local}} + \mathcal{L}_{\text{Global}} + \mathcal{L}^{\text{ded}}_{\text{generator}} \tag{9}$$

where $\mathcal{L}^{\text{ded}}_{\text{generator}}$ is the conditional probability likelihood loss of the seq2seq generative model:

$$\mathcal{L}^{\text{ded}}_{\text{generator}} = \sum_{i=1}^{Z} \log p_\theta(G_i^t | H, S', G_i^{1:t-1}) \tag{10}$$

$$\begin{aligned} p_\theta(G_i^t | H, S', G_i^{1:t-1}) &= p_\theta(s_1^t, s_2^t, \dots, s_L^t | X) \\ &= \prod_{l=1}^{L} p_\theta(s_l^t | s_{<l}^t, X) \end{aligned} \tag{11}$$

where $X = (H, S', G_i^{1:t-1})$, each token in step G_i^t is denoted as $\{s_1^t, s_2^t, \dots, s_L^t\}$, Z denotes the total number of training data and L denotes the total number of tokens. The loss of the abductive generator $\mathcal{L}_{\text{abd}}$ is calculated in the same way as above.

4.3 Alignment and Fusion

We further align the trees T_{ded} and T_{abd} obtained by G^t_{ded} and G^t_{abd} into a proof graph and search for the best structure tree according to the score of each node, where the node of the graph is $\text{node} \in \{\mathcal{J} \cup \mathcal{M}\}$, and the edge is the premise node of the proof step $\text{step}_i \in \mathcal{V}$ points to the conclusion node.

Specifically, we first construct the initial graph $\mathcal{G}_{init}$ according to T_{ded}, and then integrate the abduction tree T_{abd} whose confidence is greater than the threshold β into $\mathcal{G}_{init}$, expanding the nodes and edges of the proof graph while assigning scores to each node:

$$node_i = \begin{cases} 1.0, & \text{if node}_i \in \mathcal{J} \\ \log_s + \text{ver}_s, & \text{otherwise} \end{cases} \tag{12}$$

where $\log_s$ is the likelihood score when the autoregressive model generates an intermediate conclusion, and ver_s is the step score of the verifier for the intermediate conclusion. As described in section 4.2, we use the abductive score for the deductive step, and vice versa. For searching, we follow the work of Yang et al. (2022) explore the proof of different paths on the graph and extract the best proof tree based on node and step scores. The overall flow of FRVA is shown in Algorithm 1 in *Appendix* A.

	Train	Dev	Test	All
Question / Tree	1,131	187	340	1,840
Reasoning steps	4,175	597	1,109	5,881

Table 1: Summary statistics for EntailmentBank dataset splits.

5 Experiments

5.1 Dataset

We evaluate our method on EntailmentBank (Dalvi et al., 2021), an expert-annotated benchmark for QA explanation in the form of entailment trees. It contains a total of 1,840 entailment trees and 5,881 reasoning steps, each showing how QA pairs are entailed by a small number of related sentences. On average, each entailment tree has 7.6 nodes (including leaf nodes, intermediate nodes, and root nodes) and about 3.2 entailment steps. The detailed statistics of the data are shown in Table 1.

5.2 Baselines and Evaluation Metrics

We use the direct generation method EntailmentWriter (Dalvi et al., 2021) and the recent stepwise generation methods IRGR (Ribeiro et al., 2022), RLET (Liu et al., 2022), MetGen (Hong et al., 2022), NLProofs (Yang et al., 2022) and FLD (Morishita et al., 2023) as comparison baselines. We evaluate the structured tree using the metrics proposed by Dalvi et al. (2021), which includes the F1 and accuracy of the Leaves, Steps and Intermediate conclusions, in addition to the overall accuracy of the tree structure. The baselines, evaluation metrics, and implementation details can be found in *Appendix* B.

5.3 Main Results

The results of the experiments on EntailmentBank are shown in Table 2. We can find the following conclusions from the experimental results.

First, FRVA generates more correct proofs. In task1, we improve the overall accuracy of the tree by 1.4% with the same settings (the FLD method uses a lot of extra data for training, nevertheless, our method is also 1.1% better than theirs). Meanwhile, FRVA achieves SOTA performance on the proof step and intermediate metrics, and the accuracy improves by 2.1% and 1.8%, respectively. In Task2, FRVA improves the overall accuracy of the tree from 33.3% to 34.4%. It also achieves the best performance on the proof step and interme-

Task	Method	Leaves		Steps		Intermediates		Overall
		F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
Task1	EntailmentWriter (T5-11B)	99.0	89.4	51.5	39.2	71.2	38.5	35.3
	EntailmentWriter	98.7	86.2	50.5	37.7	67.6	36.2	33.5
	IRGR*	97.6	89.4	50.2	36.8	62.1	31.8	32.4
	RLET	**100.0**	**100.0**	54.6	40.7	66.9	36.3	34.8
	MetGen	**100.0**	**100.0**	57.7	41.9	70.8	39.2	36.5
	FLD†	99.0	92.7	55.5	42.2	73.4	41.3	39.2
	NLProofs	97.8 ± 0.2	90.1 ± 1.2	55.6 ± 0.6	42.3 ± 0.4	72.4 ± 0.5	40.6 ± 0.7	38.9 ± 0.7
	FRVA (ours)	98.2 ± 0.3	94.0 ± 1.0	**57.8 ± 0.4**	**44.4 ± 0.6**	**73.5 ± 0.4**	**42.4 ± 0.3**	**40.3 ± 0.7**
Task2	EntailmentWriter (T5-11B)	89.1	48.8	41.4	27.7	66.2	31.5	25.6
	EntailmentWriter	84.3	35.6	35.5	22.9	61.8	28.5	20.9
	IRGR*	69.9	23.8	30.5	22.4	47.7	26.5	21.8
	RLET	81.0	39.0	38.5	28.4	56.3	28.6	25.7
	MetGen	82.7	46.1	41.3	29.6	61.4	32.4	27.7
	FLD†	88.4	53.6	45.6	33.8	67.9	36.1	32.6
	NLProofs	90.3 ± 0.4	58.8 ± 1.8	47.2 ± 1.7	34.4 ± 1.7	70.2 ± 0.5	37.8 ± 1.6	33.3 ± 1.5
	FRVA (ours)	**91.3 ± 0.3**	**60.5 ± 0.8**	**48.0 ± 0.6**	**35.8 ± 0.5**	**71.1 ± 0.4**	**39.1 ± 0.9**	**34.4 ± 0.8**
Task3	EntailmentWriter (T5-11B)	39.9	3.8	7.4	2.9	35.9	7.1	2.9
	EntailmentWriter	35.7	2.9	6.1	2.4	33.4	7.7	2.4
	IRGR*	**45.6**	**11.8**	**16.1**	**11.5**	38.8	**20.9**	**11.5**
	RLET	38.3	9.1	11.5	7.1	34.2	12.1	6.9
	MetGen	34.8	8.7	9.8	8.6	36.6	20.4	8.6
	FLD†	43.6	9.7	12.1	8.3	43.0	20.1	8.3
	NLProofs	43.2 ± 0.6	8.2 ± 0.7	11.2 ± 0.6	6.9 ± 0.7	42.9 ± 1.0	17.3 ± 0.5	6.9 ± 0.7
	FRVA (ours)	44.0 ± 0.2	8.9 ± 0.3	11.6 ± 0.2	7.53 ± 0.2	**43.2 ± 0.4**	17.9 ± 0.3	7.5 ± 0.2

Table 2: Results of proof generation on EntailmenBank. All baselines using the T5-large model, except for those marked in parentheses. *: IRGR in task3 retrieves facts from the entire corpus, while the rest of the methods use the retrieved 25 fact sentences provided in the original dataset. †: FLD uses additional data to generate deductive steps for training. Bold and underlined texts highlight the best method and the runner-up.

diate metrics. This shows that our bidirectional generation method can improve the reliability of the proofs step, thus mitigating the error propagation during stepwise generation and enabling the model to generate better intermediate conclusions.

Second, FRVA can identify relevant supporting facts more effectively. In task1, FRVA improves the F1 and accuracy of leaf nodes by 0.4% and 3.9% compared to NLProofs and achieves a large improvement compared to all baselines except RLET and MetGen. Among them, MetGen manually annotates templates for different reasoning types and augments proof generation with additional Wikipedia training data. In task2, FRVA improves the F1 and accuracy of leaf nodes by 1.0% and 1.7%, respectively, compared to the best baseline, and it achieves a more significant improvement compared to the other baseline methods, which can be improved by 8.7% and 19.3% on average. This indicates that fact retrieval can filter irrelevant facts and reduce the search space in the subsequent reasoning process. Moreover, we design cross-verification and contrastive learning that can help the model more accurately select facts needed to prove the hypotheses. For task3, we use the same retrieval results generated by Dalvi et al. (2021). We can see that our methods achieve competitive performance with the baseline except for the IRGR and MetGen.

We also use a larger pre-trained model (e.g., Flan-T5-Large (Chung et al., 2022)) as the proof step generator, and the experiment results are shown in *Appendix* C. We also prompt large language models (LLMs) with in-context samples to test their ability of generating proof steps. *Appendix* G shows that our method is significantly beyond the LLMs.

6 Analysis

6.1 Ablation Study

To evaluate the effectiveness of each module, we conduct an ablation study on Task2 as shown in Table 3. We can see that all the metrics exhibit an increasing trend when we introduce each component sequentially. For example, we introduce contrastive learning in deductive generation, which can significantly improve the structural correctness of the tree, which benefits from the contrastive loss pulling the distance between proof and target hy-

Method	**Leaves**		**Steps**		**Intermediates**		**Overall**
	F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
Only Deductive generate	89.9	56.5	45.7	32.9	69.1	36.5	31.5
+ Multi-view CL	90.1	56.8	46.4	33.8	69.0	37.4	33.3
+ Abductive generate	89.8	57.1	47.3	34.4	69.6	36.5	33.5
+ Fact Retrieval	90.9	**61.2**	47.4	35.3	71.1	38.2	33.5
+ Cross Verifier (FRVA)	**91.2**	60.3	**48.8**	**36.2**	**71.6**	**38.5**	**34.7**

Table 3: Results of the ablation experiments on the Task2 test set, where we sequentially add each of the components proposed on the stepwise deductive generation baseline.

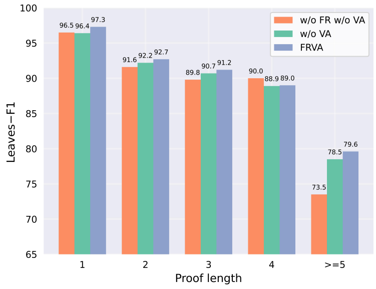


Figure 4: Results of test sets for Leaves F1 with different proof step lengths in Task2.

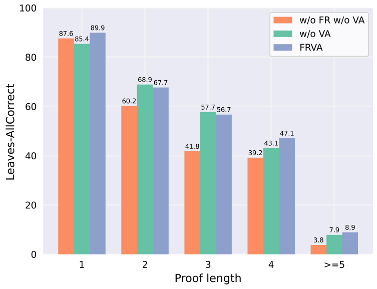


Figure 5: Results of test sets for Leaves AllCorrect with different proof step lengths in Task2.

Method	**Task1**	**Task2**
SI+Halter	72.4	55.9
SI+Halter+Search	83.2	72.9
FAME	91.5	78.2
FRVA (Ours)	**92.2**	**85.8**

Table 4: Experiment results on the EntailmentBankQA. The results for SI are from Creswell et al. (2023), and the results for FAME are from Hong et al. (2023).

Method	**eQASC**		**eOBQA**	
	P@1	NDCG	P@1	NDCG
EntailmentWriter	52.48	73.14	69.07	89.05
EntailmentWriter-Iter	52.56	73.28	72.15	90.19
MetGen	55.81	74.19	74.89	90.50
FRVA (Ours)	**60.12**	**89.81**	**76.68**	**93.46**

Table 5: Experiment results on the eQASC and eOBQA.

pothesis. Moreover, when we introduce the fact retrieval module, Leaves F1 and AllCorrect metrics improve by 1.1 and 4.1, respectively, which shows that irrelevant facts can be effectively reduced. When we introduce the cross-verification, it can be seen that all metrics have made significant improvements, which indicates that more reliable steps can be selected through bidirectional consistency verification to better generate a proof tree.

6.2 Effect of different Proof Length

Multi-step proof trees are difficult to generate accurately due to their complex structure. To explore the effectiveness of FRVA on multi-step generation, we break down the test performance of Task2 by the step length of the golden tree. The performance of the leaf node is shown in Figure 4, and the rest of the metrics are shown in *Appendix* D. We observe that the performance decreases significantly when proof length increases, which indicates that the generation of multi-step proof trees remains challenging. Nevertheless, we can see that improving leaf node accuracy is more significant for FRVA than without cross-verification and fact retrieval. In the multi-step case, the above two modules can eliminate the distraction of some irrelevant facts and provide more reliable knowledge facts for proving the target hypothesis.

6.3 Cross-dataset Experiments

To further validate the generalization of our method, we perform cross-dataset experiments on other datasets. Specifically, we conduct experiments on the EntailmentBankQA, eQASC, and eOBQA datasets. Among these, EntailmentBankQA was constructed by Hong et al. (2023) following Creswell and Shanahan (2022) conversion of EntailmentBank to a more challenging version of QA by adding 4-way multiple options from the ARC

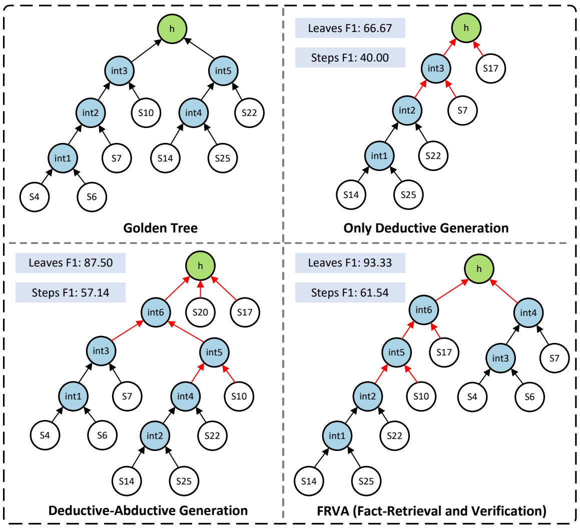


Figure 6: Examples of multi-step proofs generated by different models in Task2.

dataset for each hypothesis. eQASC and eOBQA were constructed by Jhamtani and Clark (2020) that collect one-step entailment trees from QASC (Khot et al., 2020) and OpenBookQA (Mihaylov et al., 2018) questions, which require the selection of valid single-step trees from the candidate set and evaluate the results with P@1 and NDCG metrics (Jhamtani and Clark, 2020).

Following the previous work, we directly apply Task2's model for question answering on the above three datasets, and the experimental results are shown in Table 4 and 5. We can see that FRVA exhibits a great advantage in generalization across datasets, which validates the effectiveness of our method.

6.4 Case Study

We further perform a case study on the proof tree generated by FRVA, as shown in Figure 6. The golden tree contains six proof steps, whereas the previous deductive generation method generated only four proof steps. We can see that the step "`sent6 & sent4 -> int1`" is further supplemented to make the prediction tree closer to the standard tree in terms of the overall structure by introducing abductive reasoning. However, we find that it still has unreliable steps, such as `sent20` and `sent17`. Therefore, we introduce fact retrieval and bidirectional verification, and we find that they can remove irrelevant facts and select more reliable steps. Finally, FRVA can obtain results closer to those of the golden tree.

6.5 Error Analysis

To understand the shortcomings of our model, we further analyze the output of FRVA. We select 50 proof trees of generated errors on the test set of Task2, and we classify the error types into the following two categories:

Reasoning process mistake. The missing or redundant leaves lead to incorrect proof steps, making the model generate incorrect reasoning processes (42% of errors). For example, when explaining the hypothesis "*forming fossil fuels requires deposition and burial of decaying plants*", the absence of the premises "*forming fossil fuels requires deposition and burial of decaying vegetation*" and "*plants are a kind of vegetation*" makes it impossible to explain the hypothesis.

Tree structure diversity. For proof trees with steps greater than 1, there are usually multiple reasoning paths from facts to hypotheses, so it is difficult to capture the validity of different structure trees by evaluating them directly against the golden tree (36% of errors). As shown in Figure 12 of *Appendix* E, the prediction tree contains three premise steps, and the Steps and Overall score are 0 under the automatic evaluation, because the prediction tree does not match the golden tree. However, the prediction tree should be valid because each step can be entailed by the premises. In the future, other evaluation methods should be introduced to better evaluate the different structural trees and thereby present the true logical reasoning ability of the model. In addition, some of the errors are repeating premises as conclusions, or re-generating steps that are already in place (22% of errors).

7 Conclusion

In this paper, we propose FRVA, a bidirectional entailment tree generation method based on fact retrieval and verification augmented, which divides the generation process into two systems. System 1 makes the initial judgment, and System 2 makes the refined reasoning. We design a deductive-abductive bidirectional reasoning method and enhance the effectiveness of the proof step from supportive and similarity perspectives through cross-verification and multi-view contrastive learning. Experiments show that FRVA outperforms the existing baselines regarding fact selection and structural correctness. It is worth exploring to design more efficient bidirectional reasoning methods and comprehensive evaluation systems in the future.

Limitations

Despite our method to achieving the performance increase, there is still substantial room for future improvements. First, we set the termination condition for the backward abductive reasoning that no new intermediate conclusions will be generated in the iterative step generation process. However, this will ignore the proof steps that contain multiple intermediate conclusions as premises. If there are better judgment conditions, it will be more helpful to generate the structure tree. Second, like prior work (Tafjord et al., 2021; Dalvi et al., 2021), we concatenate the filtered facts into a long text sequence and encode it with the language model, which can be limited by the input length constraints of the language model and affect the practical application of proof generation. Finally, the current automatic evaluation makes it difficult to accurately capture different tree structures (as we discussed in Section 6.5), which can underestimate the reasoning ability of the language model. Therefore, we leave the exploration of better backward reasoning and accurate evaluation of structural trees for future work.

Ethics Statement

Explainable QA is an important branch of the question answering domain, where the explanations given by the model must be faithful and reliable. Therefore, giving the reasoning process from the known facts to the answer is crucial for the transparency of the model. The data used in our work come from public datasets. Our proposed bidirectional proof generation method can improve the reliability of the reasoning process.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key Research and Development Program of China (2020AAA0106100) and National Natural Science Foundation of China (62376144, 62176145, 62076155).

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, and Ben He. 2023. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. *CoRR*, abs/2303.16421.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. 2022. Natural language deduction through search over statement compositions. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 4871–4883.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 6266–6278.

Guoxin Chen, Kexin Tang, Chao Yang, Fuying Ye, Yu Qiao, and Yiming Qian. 2024. SEER: facilitating structured reasoning and explanation via reinforcement learning. *CoRR*, abs/2401.13246.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *CoRR*, abs/2208.14271.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations, ICLR*. OpenReview.net.

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, EMNLP*, pages 7358–7370.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 4443–4458.

Zichu Fei, Qi Zhang, Xin Zhou, Tao Gui, and Xuanjing Huang. 2022. Proofinfer: Generating proof via iterative hierarchical inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 10883–10892.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. METGEN: A module-based entailment tree generation framework for answer explanation. In *Findings of the Association for Computational Linguistics: NAACL*, pages 1887–1905.

Ruixin Hong, Hongming Zhang, Hong Zhao, Dong Yu, and Changshui Zhang. 2023. Faithful question answering with monte-carlo planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3944–3965. Association for Computational Linguistics.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 37–42. Association for Computational Linguistics.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 137–150.

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), ACL*, pages 6547–6568.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8082–8090. AAAI Press.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. *CoRR*, abs/2306.03872.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2022. RLET: A reinforcement learning based approach for explainable QA with entailment trees. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 7177–7189.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*.

Hugo Mercier and Dan Sperber. 2011. Why do humans reason? arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, EMNLP*, pages 2381–2391.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2023. Learning deductive reasoning from synthetic corpus based on formal logic. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 25254–25274.

Yilin Niu, Fangkai Jiao, Mantong Zhou, Ting Yao, Jingfang Xu, and Minlie Huang. 2020. A self-training method for machine reading comprehension with soft evidence extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 3916–3927.

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

Hanhao Qu, Yu Cao, Jun Gao, Liang Ding, and Ruifeng Xu. 2022. Interpretable proof generation via iterative backward reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL*, pages 2968–2981.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henghui Zhu, Xinchi Chen, Peng Xu, Zhiheng Huang, Andrew O. Arnold, and Dan Roth. 2022. Entailment tree explanations via iterative retrieval-generation reasoner. In *Findings of the Association for Computational Linguistics: NAACL*, pages 465–475.

Hendrik Schuff, Heike Adel, and Ngoc Thang Vu. 2020. F1 is not enough! models and evaluation towards user-centered explainable question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 7076–7095.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 2931–2951.

Ying Su, Xiaojin Fu, Mingwen Liu, and Zhijiang Guo. 2023. Are llms rigorous logical reasoner? empowering natural language proof generation with contrastive stepwise decoding. *CoRR*, abs/2311.06736.

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*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based reconstruction of multi-hop explanations for science questions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL*, pages 200–211.

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*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*.

Larry Wos. 1985. What is automated reasoning? *J. Autom. Reason.*, 1(1):6–9.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views. *arXiv preprint arXiv:2306.09841*.

Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Dynamic semantic graph construction and reasoning for explainable multi-hop science question answering. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1044–1056.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. Generating natural language proofs with verifier-guided search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 89–105.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 2369–2380.

Yao Yao, Zuchao Li, and Hai Zhao. 2023. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. *CoRR*, abs/2305.16582.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 5942–5966.

Li Yuan, Yi Cai, Haopeng Ren, and Jiexin Wang. 2024. A logical pattern memory pre-trained model for entailment tree generation. *arXiv preprint arXiv:2403.06410*.

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, EMNLP*, pages 11155–11173.

A The overall flow of FRVA

The overall flow of FRVA is shown in Algorithm 1.

B Experiment details

We describe the baselines, evaluation metrics, and implementation details used in the experiment.

B.1 Baselines

EntailmentWriter (Dalvi et al., 2021) provides a powerful baseline by linearizing the tree structure and generates the entire tree as well as intermediate conclusions in one shot using a sequence-to-sequence model. It is available in two versions, implemented on T5-11B (11 billion parameters) and T5-Large (770 million parameters) (Raffel et al., 2020).

IRGR (Ribeiro et al., 2022) designs an iterative retrieval generation framework that improves the retrieval results of Task 3.

RLET (Liu et al., 2022) introduces reinforcement learning for the first time to entailment tree generation tasks, where single-step reasoning is performed iteratively through sentence selection and deductive generation modules.

MetGen (Hong et al., 2022) iteratively generates entailment trees through multiple modules and reasoning controllers.

NLProofs (Yang et al., 2022) guides the generation of proof steps through an independent verifier.

FLD (Morishita et al., 2023) designs new deductive data generation methods based on the synthetic corpora, and uses the new data to train and then fine-tune the EntailmentBank data.

B.2 Evaluation Metrics

Leaves (F1, AllCorrect): To evaluate the performance of the model in recognizing facts related to

Algorithm 1: FRVA Proof Tree Generation

```
Input: Hypothesis $H$, Supporting facts
       $S = \{sent_1, sent_2, ..., sent_n\}$, Fact
       Retriever $\mathcal{R}_{fact}$, Deductive and
       Abductive generator $G_{ded}$, $G_{abd}$,
       Cross-verifier $\mathcal{V}_{ded-abd}$
Output: Proof Tree $T_{pred}$
Fact Retrieval (System 1):
$S' = \{\}$;
for $sent_i$ in $S$ do
    fact_score = $\mathcal{R}_{fact}(sent_i)$;
    if fact_score $> \gamma$ then
        Add $sent_i$ to $S'$;
Detailed stepwise reasoning (System 2):
for $t = 1$ to $max\_step$ do
    $step^t_{ded}, s^t_{ded} \leftarrow G_{ded}(H, S', step^{t-1}_{ded})$;
    $v^t_{ded} \leftarrow \mathcal{V}_{abd}(step^t_{ded}) + s^t_{ded}$;
    if $step^t_{ded}$ is final then
        $proof^{pred}_{ded} = step^{1:t}_{ded}$, Aggre($v^{1:t}_{ded}$);
        break;
for $t = 1$ to $max\_step$ do
    $step^t_{abd}, s^t_{abd} \leftarrow G_{abd}(H, S', step^{t-1}_{abd})$;
    $v^t_{abd} \leftarrow \mathcal{V}_{ded}(step^t_{abd}) + s^t_{abd}$;
    if $step^t_{abd}$ is final then
        $proof^{pred}_{abd} = step^{1:t}_{abd}$, Aggre($v^{1:t}_{abd}$);
        break;
$T^{pred}_{ded}$, $s_{ded} = proof^{pred}_{ded}$;
$T^{pred}_{abd}$, $s_{abd} = proof^{pred}_{abd}$;
$\mathcal{G} \leftarrow$ initialize($T^{pred}_{ded}$, $s_{ded}$);
if $s_{abd} > \beta$ then
    $\mathcal{G} \leftarrow$ update($\mathcal{G}$, $T^{pred}_{abd}$, $s_{abd}$);
$T_{pred} \leftarrow$ extract_proof($\mathcal{G}$);
return $T_{pred}$;
```

γ	Coverage	Number	Leaves
3.9e-7	96.2%	14	60.3
2.5e-7	98.8%	17	61.5
2.1e-7	99.5%	19	58.9

Table 6: Statistics for different thresholds γ on the task2 test set. "Coverage" is the golden fact coverage, "Number" is the average number of facts, and "Leaves" is the accuracy of the leaf nodes on task2.

questions and answers. We compute F1 by comparing the leaf node S_{pred} of T_{pred} with the leaf node S_{gold} of T_{gold}, and AllCorrect is 1 if they match exactly.

Steps (F1, AllCorrect): To evaluate whether the predicted steps are structurally correct. We compute F1 by comparing the set of premises (child nodes) of the intermediate conclusion node int$_{pred}$ of T_{pred} and the aligned conclusion node int$_{gold}$ in T_{gold}, and AllCorrect is 1 if they match exactly.

Intermediates (F1, Allcorrect): To evaluate whether the intermediate conclusion of the prediction is correct or not. We compute F1 by comparing the intermediate conclusion node int$_{pred}$ of the prediction tree T_{pred} with the aligned conclusion node int$_{gold}$ in the gold tree T_{gold}. Then we compute the BLEURT[*] score between them, if it is greater than 0.281 (we following Dalvi et al. (2021)), then the intermediate conclusion of the prediction int$_{pred}$ is considered to be correct, and if the intermediate conclusions of the prediction tree T_{pred} are all correct, then AllCorrect is 1.

Overall (AllCorrect): Test the above three metrics together. If the AllCorrect scores for Leaves, Steps, and Intermediates of the prediction tree T_{pred} are all 1, the overall correctness of the tree is 1. Note that this is a strict metric, as any error in T_{pred} results in a score of 0. For all metrics we report the results generated by their official evaluation code[†].

B.3 Implementation Details

Following previous work, our proofs generators (deductive and abductive generators) also use a pre-trained T5-Large (Raffel et al., 2020) model. The fact retriever and deductive-abductive cross-validator are implemented with fine-tuned albert-xxlarge-v2 (Lan et al., 2020). We use AdamW (Loshchilov and Hutter, 2019) as the optimizer with a learning rate of 1e-5 and batch size of 4. We set the maximum length of the input sequence to 1024. The threshold γ is set to 2.5e-7, and the marginal $\mathcal{W}_{fact}$ is set to 0.1 in section 4.1. We filter the facts after the first stage to get a different number of knowledge facts for each hypothesis (the average number of facts is reduced from 25 to 17 as shown in Table 6). The temperature coefficient τ is set to 0.07, the marginal $\mathcal{W}_{pair}$ is set to 0.1, the beam size u is set to 10, μ is set to 1.0, and z is set to 10 in section 4.2. β is set to 0.85 in section 4.3. Hyperparameters are tuned to the validation data separately for each task/method. For the deductive and abductive proof generators, we set a maximum

[*] We use the `bleurt-large-512` model following Dalvi et al. (2021)

[†] `https://github.com/allenai/entailment_bank`

Method	Leaves		Steps		Intermediates		Overall
	F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
ConDec (Flan-T5-Large 0.8B)†	90.73	58.82	49.17	36.18	69.56	36.76	33.53
ConDec (Flan-T5-XL 3B)†	91.10	60.59	**50.70**	**37.35**	70.74	38.24	**34.71**
FRVA (T5-Large 0.8B)	91.23	60.29	48.77	36.18	**71.55**	**38.53**	**34.71**
FRVA (Flan-T5-Large 0.8B)	**92.04**	**62.94**	49.68	35.88	71.43	37.65	34.41

Table 7: Comparison results with larger models on Task 2 test set of EntailmenrBank. † are results from Su et al. (2023). Bold and underlined texts highlight the best method and the runner-up.

Task	Method	Leaves		Steps		Intermediates		Overall
		F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
Task1	LMPM (Yuan et al., 2024)	99.8	99.4	57.8	43.8	72.8	**42.8**	38.5
	SEER (Chen et al., 2024)	**100.0**	**100.0**	**67.6**	**52.6**	70.3	42.6	40.6
	FRVA-one (ours)	98.4	95.3	58.0	45.0	**74.1**	**42.8**	**41.2**
Task2	LMPM (Yuan et al., 2024)	81.1	47.1	42.6	31.4	61.7	34.3	29.4
	SEER (Chen et al., 2024)	86.4	53.5	**56.8**	**39.7**	66.3	38.3	34.7
	FRVA-one (ours)	**91.5**	**61.4**	48.8	36.4	**71.5**	**40.6**	**35.3**

Table 8: Comparison results with the latest related work on Task1 and Task2. We report here the best results of FRVA on the EntailmentBank test set.

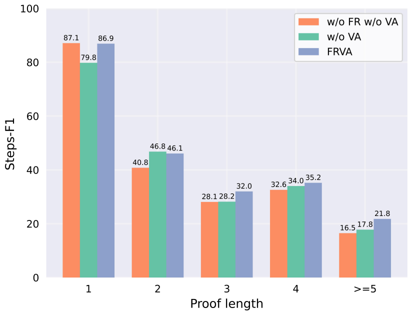


Figure 7: Results of test sets for Steps F1 with different proof step lengths in Task2.

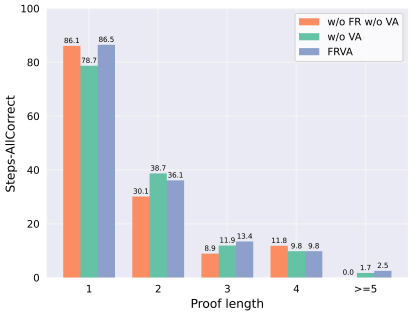


Figure 8: Results of test sets for Steps AllCorrect with different proof step lengths in Task2.

number of steps of 20 as an additional condition for termination to prevent the infinite generation of steps.

We train the model on Task1 and Task2, respectively. For Task3, Dalvi et al. (2021) retrieves 25 supporting facts for each hypothesis, and we use the same retrieval results. Following their work, we utilize the model trained on Task2 to test its zero-shot performance on Task3. All comparative baseline results report the results in the original paper. For our method on test set, we report the average performance and standard deviation over 5 independent runs. All experiments are run on a machine with one NVIDIA Tesla A100 (40GB) GPU.

C Results of other experiments

Su et al. (2023) propose a contrast stepwise decoding method, ConDec, that employs an additional checker (using the T5-11B model) to construct the contrast samples. It is worth noting that their experiment uses a larger model Flan-T5-XL (3B) (Chung et al., 2022) as the backbone and points out that the model with larger parameters has better logical reasoning ability. Our work belongs to the contemporaneous work with ConDec. Nevertheless, we also compare with it as shown in Table 7.

It can be seen that FRVA outperforms the ConDec method when using the Flan-T5-Large model, and at the same time achieves a new state-of-the-art performance in the Leaves F1 and AllCorrect metrics, which are even higher than the larger model

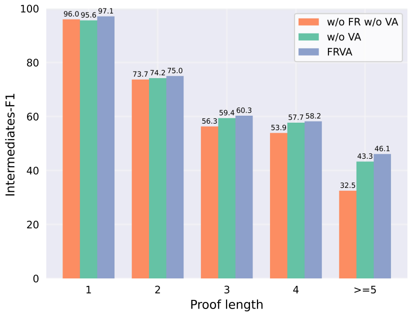


Figure 9: Results of test sets for Intermediates F1 with different proof step lengths in Task2.

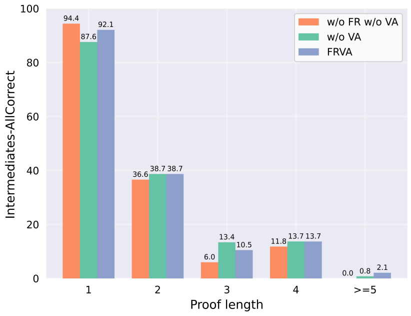


Figure 10: Results of test sets for Intermediates AllCorrect with different proof step lengths in Task2.

ConDec (Flan-T5-XL (3B)) by 0.94 and 2.35. Due to resource constraints, we have not yet experimented on Flan-T5-XL (3B), but we still achieve comparable or better results than the larger model.

Furthermore, we compare FRVA with the most recent related work (Yuan et al., 2024; Chen et al., 2024), as shown in Table 8. We can see that FRVA outperforms all baselines except the Steps metric. It is worth noting that Overall AllCorrect is a strict metric that is 1 when the leaf nodes and intermediate conclusions match the golden tree exactly. We can see that FRVA is achieved the optimization on this metric, which demonstrates that our method is able to obtain a more accurate entailment tree to support the construction of a explainable QA system.

D Experiment results for different proof length

The results of the metrics for different proof lengths on the test set of task2 as shown in Figures 7, 8, 9, 10, 11. It is worth noting that the automatic evaluation metrics underestimate the logical reasoning power of language models because the entailment tree containing multiple steps may have multiple reasoning paths, as we described in Section 6.5. Therefore, the performance for different lengths only roughly reflects the multi-step reasoning capabilities of different models.

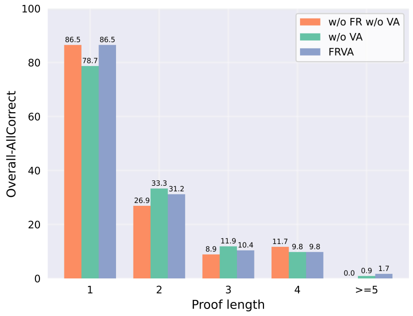


Figure 11: Results of test sets for Overall AllCorrect with different proof step lengths in Task2.

	NLProofs	FRVA
Facts Correctness	4.52	4.58
Single-step Validity	3.80	4.14
Conclusion Consistency	3.86	4.22
Overall Correctness	25	30

Table 9: Human evaluation results for 50 randomly selected samples in the Task 1 test set.

E Error Analysis Sample

An entailment tree containing multiple proof steps may have multiple reasoning paths pointing from premises to hypotheses, as shown in Figure 12. The current automatic evaluation often struggles to capture the diversity of tree structures effectively. Therefore, it remains worth exploring how to evaluate the validity of structured trees more effectively in the future.

F Human Evaluation

As mentioned above, due to the diversity of tree structures, automatic evaluation metrics have limitations and do not accurately evaluate the structured reasoning ability of the model. Therefore, we further perform a human evaluation. We randomly selected 50 instances from EntailmentBank's Task1 test set and evaluated the model results against four criteria: **(1) Facts Correctness:** evaluates whether the leaf nodes of the prediction tree are correct and necessary to prove the hypothesis. **(2) Single-step Validity:** evaluates whether the generated intermediate conclusion can be derived from two or more premises. **(3) Conclusion Consistency:** assesses

Method	Leaves		Steps		Intermediates		Overall
	F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
EntailmentWriter (T5- Large)	86.2	43.9	40.6	28.3	67.1	34.8	27.3
EntailmentWriter (T5-11B)	89.4	52.9	46.6	35.3	69.1	36.9	32.1
GPT-3 (7-shot ICL)†	64.2 ± 2.3	15.3 ± 1.9	17.6 ± 0.6	12.3 ± 1.4	53.6 ± 1.4	22.3 ± 1.1	12.3 ± 1.4
Codex (7-shot ICL)†	68.9 ± 3.7	19.8 ± 3.2	21.4 ± 3.0	14.6 ± 1.7	55.6 ± 2.2	23.2 ± 1.9	14.4 ± 1.4
GPT-3.5-turbo (7-shot ICL)	60.9	3.5	16.5	2.8	41.7	9.6	3.2
GPT-3.5-turbo (SCOT)	62.3	9.6	17.7	6.4	43.4	14.0	5.9
GPT-4 (7-shot ICL)	75.0	24.1	23.0	4.3	60.5	20.9	10.5
GPT-4 (SCOT)	76.1	25.4	23.3	16.1	62.3	27.8	15.5
FRVA (ours T5-Large)	90.1	56.2	50.8	38.0	72.6	40.7	35.8
FRVA (ours Flan-T5-Large)	**91.8**	**60.0**	**54.6**	**42.3**	**74.0**	**43.3**	**40.1**

Table 10: Results of different models on EntailmentBank's Task2 validation set. † are results from Yang et al. (2022). We prompt the large language model using 7-shot ICL and 1-shot COT, respectively.

whether the intermediate conclusions generated are consistent with facts and common sense and whether they are simple repetitions of the premises. **(4) Overall Correctness:** evaluates whether the final hypothesis can be deduced from all generated intermediate conclusions in the prediction tree. For the overall correctness metric, we count the number of valid trees. For the remaining metrics, we rate the generated reasoning steps from 1 (poor) to 5 (very good) and report the average score. We compared the results of FRVA with the baseline method, NLProofs (Yang et al., 2022), and the results are shown in Table 9. We can see that our method exhibits excellent performance on all metrics, especially on the single-step validity and conclusion consistency metrics. This demonstrates that our method is better able to generate intermediate reasoning steps.

G Prompting with LLMs

With the help of large-scale pre-training, instruction fine-tuning, and human feedback reinforcement learning strategies, Large Language Models (LLMs) have made significant progress in natural language processing (Ling et al., 2023; Yao et al., 2023). However, researchers have begun to doubt the effectiveness of LLMs in complex logical reasoning tasks and to evaluate the capability of LLMs from different reasoning perspectives (Xu et al., 2023), such as commonsense reasoning (Bian et al., 2023), mathematical reasoning (Imani et al., 2023), and multilingual reasoning (Bang et al., 2023) and so on. The entailment tree comprises multiple reasoning steps annotated by experts, which involves complex reasoning in real-world scenarios and presents a great challenge to the reasoning ability of the model. Therefore, we test the performance of LLMs on this task and detect their logical reasoning ability.

We explore the logical reasoning ability of LLMs using both in-context-learning (ICL) and thought of chaining (COT) for LLMs, as shown in Figures 13 and 14. We use the above instructions to prompt for ChatGPT[‡] and GPT-4 (OpenAI, 2023). Specifically, we randomly sample 7 context examples from the training set of EntailmentBank (Task 2) to prompt the LLMs directly. In addition, we also design a stepwise thought of chaining (SCOT) method to further stimulate the step-by-step reasoning ability of the large model. The experiment results on the validation set of task 2 are shown in Table 10.

We set only 1-shot instructions for the input of COT, limited by the input length and resources of LLMs. Nevertheless, we can see that GPT-4 generally outperforms GPT-3.5-turbo in all metrics, which demonstrates that GPT-4 has stronger logical reasoning and better instruction following. We also find that COT outperforms ICL, which indicates that the stepwise guide to the larger model can obtain a more accurate proof. GPT-4 can choose more accurate facts and generate more plausible intermediate conclusions, but our fine-tuned smaller models (e.g., T5, Flan-T5) significantly outperform LLMs on all metrics, which indicates that well-trained language models can capture the correlations between the knowledge facts and target hypotheses. Although large language models have some logical reasoning capabilities, it remains challenging to generate complex multi-step proofs accurately.

[‡] https://openai.com/blog/chatgpt

Question: Students are learning about the natural resources in Maryland. One group of students researches information about renewable natural resources in the state. The other group researches information about nonrenewable natural resources in the state. The resources the students investigate include plants, animals, soil, minerals, water, coal, and oil. Which of the following human activities negatively affects a natural resource?

Answer: directing runoff from cropland into a lake

Support Facts:

sent1: absorbing something harmful has a negative impact on a thing
sent2: damming a river can cause a lake to form
sent3: erosion sometimes decreases the amount of nutrients in soil
sent4: decreasing something positive has a negative impact on a thing
sent5: nature is the source of natural resources
sent6: nature means a natural environment
sent7: if something has a negative impact on something else then increasing the amount of that something has a negative impact on that something else
sent8: a body of water contains water
sent9: acid rain causes water pollution
sent10: a lake is a kind of body of water
sent11: runoff is when cropland water enters bodies of water
sent12: bodies of water are located on the surface of the earth
sent13: pollution has a negative impact on the environment / air quality / water quality / society
sent14: as the level of water rises , the amount of available land will decrease
sent15: a body of water is a source of water
sent16: loss of resources has a negative impact on the organisms in an area
sent17: runoff is a stage in the water cycle process
sent18: if something causes a process then that something is required for that process
sent19: soil erosion means soil loss through wind / water / animals
sent20: runoff from a cropland causes water pollution
sent21: a natural resource is a kind of environmental factor
sent22: high runoff causes flooding
sent23: as the amount of a source of something decreases , the amount of that something will decrease
sent24: water is a kind of natural resource
sent25: soil erosion is when wind / moving water / gravity moves soil from fields / environments

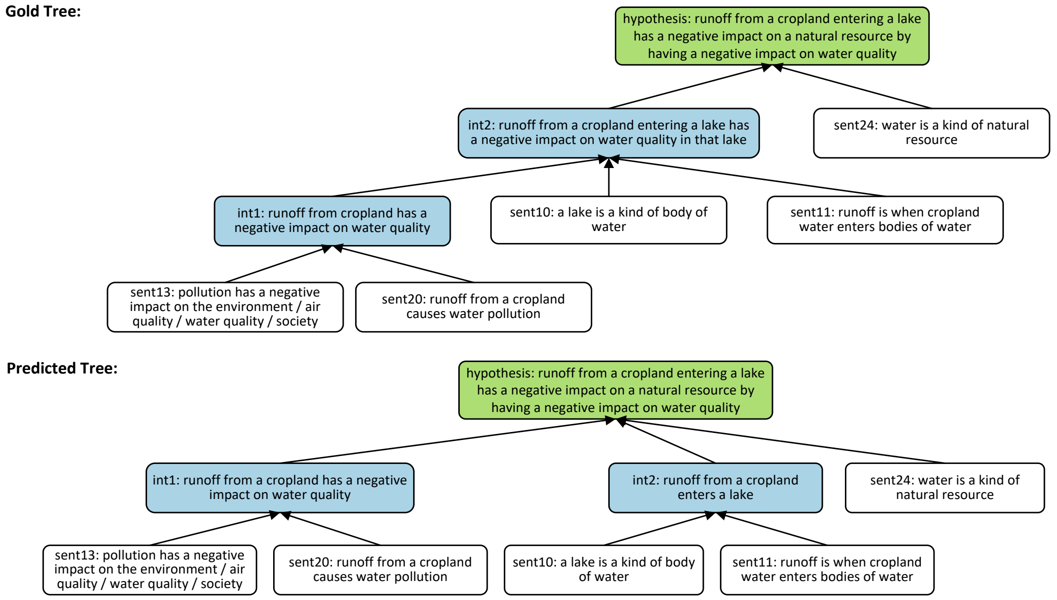
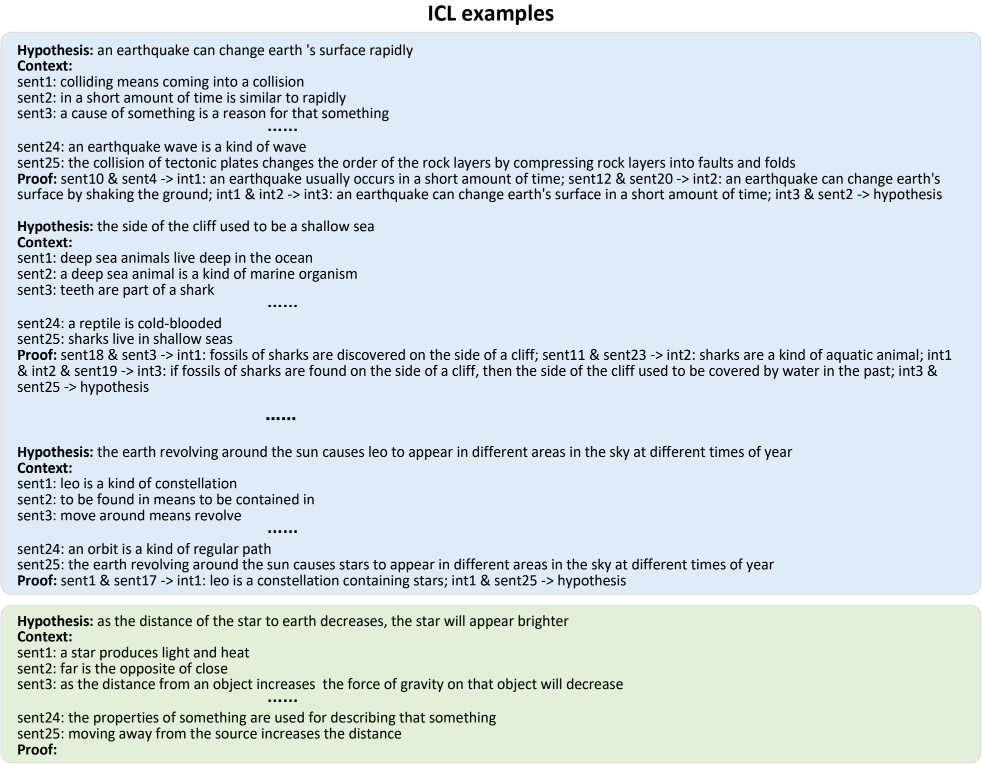

Gold Tree:
hypothesis: runoff from a cropland entering a lake has a negative impact on a natural resource by having a negative impact on water quality
int2: runoff from a cropland entering a lake has a negative impact on water quality in that lake
sent24: water is a kind of natural resource
int1: runoff from cropland has a negative impact on water quality
sent10: a lake is a kind of body of water
sent11: runoff is when cropland water enters bodies of water
sent13: pollution has a negative impact on the environment / air quality / water quality / society
sent20: runoff from a cropland causes water pollution
Predicted Tree:
hypothesis: runoff from a cropland entering a lake has a negative impact on a natural resource by having a negative impact on water quality
int1: runoff from a cropland has a negative impact on water quality
int2: runoff from a cropland enters a lake
sent24: water is a kind of natural resource
sent13: pollution has a negative impact on the environment / air quality / water quality / society
sent20: runoff from a cropland causes water pollution
sent10: a lake is a kind of body of water
sent11: runoff is when cropland water enters bodies of water

Figure 12: The case for diversity in tree structures. Although the prediction tree does not exactly match the golden tree, it is possible to complete the reasoning process with two premise steps.



ICL examples

Hypothesis: an earthquake can change earth 's surface rapidly
Context:
sent1: colliding means coming into a collision
sent2: in a short amount of time is similar to rapidly
sent3: a cause of something is a reason for that something
……
sent24: an earthquake wave is a kind of wave
sent25: the collision of tectonic plates changes the order of the rock layers by compressing rock layers into faults and folds
Proof: sent10 & sent4 -> int1: an earthquake usually occurs in a short amount of time; sent12 & sent20 -> int2: an earthquake can change earth's surface by shaking the ground; int1 & int2 -> int3: an earthquake can change earth's surface in a short amount of time; int3 & sent2 -> hypothesis

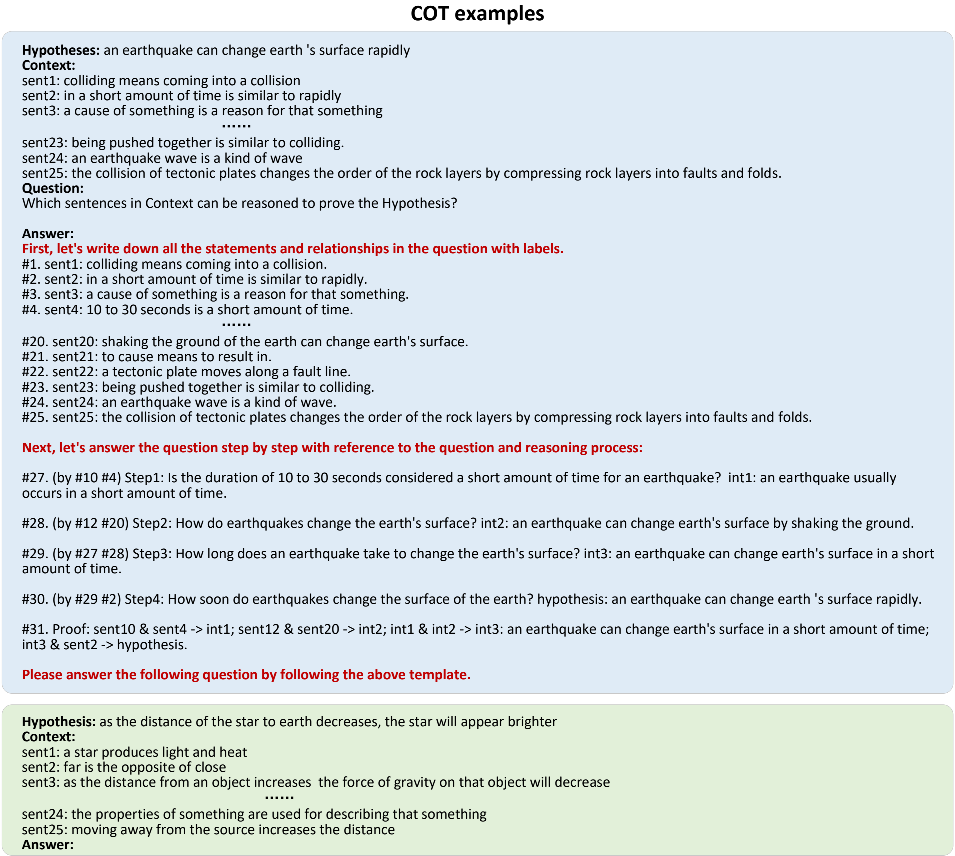
Hypothesis: the side of the cliff used to be a shallow sea
Context:
sent1: deep sea animals live deep in the ocean
sent2: a deep sea animal is a kind of marine organism
sent3: teeth are part of a shark
……
sent24: a reptile is cold-blooded
sent25: sharks live in shallow seas
Proof: sent18 & sent3 -> int1: fossils of sharks are discovered on the side of a cliff; sent11 & sent23 -> int2: sharks are a kind of aquatic animal; int1 & int2 & sent19 -> int3: if fossils of sharks are found on the side of a cliff, then the side of the cliff used to be covered by water in the past; int3 & sent25 -> hypothesis

……

Hypothesis: the earth revolving around the sun causes leo to appear in different areas in the sky at different times of year
Context:
sent1: leo is a kind of constellation
sent2: to be found in means to be contained in
sent3: move around means revolve
……
sent24: an orbit is a kind of regular path
sent25: the earth revolving around the sun causes stars to appear in different areas in the sky at different times of year
Proof: sent1 & sent17 -> int1: leo is a constellation containing stars; int1 & sent25 -> hypothesis

Hypothesis: as the distance of the star to earth decreases, the star will appear brighter
Context:
sent1: a star produces light and heat
sent2: far is the opposite of close
sent3: as the distance from an object increases the force of gravity on that object will decrease
……
sent24: the properties of something are used for describing that something
sent25: moving away from the source increases the distance
Proof:

Figure 13: In-context-learning prompts (ICL). The blue box shows the sample prompts of the input and the green box shows the test data for the LLMs.



COT examples

Hypotheses: an earthquake can change earth 's surface rapidly
Context:
sent1: colliding means coming into a collision
sent2: in a short amount of time is similar to rapidly
sent3: a cause of something is a reason for that something
……
sent23: being pushed together is similar to colliding.
sent24: an earthquake wave is a kind of wave
sent25: the collision of tectonic plates changes the order of the rock layers by compressing rock layers into faults and folds.
Question:
Which sentences in Context can be reasoned to prove the Hypothesis?

Answer:
First, let's write down all the statements and relationships in the question with labels.
#1. sent1: colliding means coming into a collision.
#2. sent2: in a short amount of time is similar to rapidly.
#3. sent3: a cause of something is a reason for that something.
#4. sent4: 10 to 30 seconds is a short amount of time.
……
#20. sent20: shaking the ground of the earth can change earth's surface.
#21. sent21: to cause means to result in.
#22. sent22: a tectonic plate moves along a fault line.
#23. sent23: being pushed together is similar to colliding.
#24. sent24: an earthquake wave is a kind of wave.
#25. sent25: the collision of tectonic plates changes the order of the rock layers by compressing rock layers into faults and folds.

Next, let's answer the question step by step with reference to the question and reasoning process:

#27. (by #10 #4) Step1: Is the duration of 10 to 30 seconds considered a short amount of time for an earthquake? int1: an earthquake usually occurs in a short amount of time.

#28. (by #12 #20) Step2: How do earthquakes change the earth's surface? int2: an earthquake can change earth's surface by shaking the ground.

#29. (by #27 #28) Step3: How long does an earthquake take to change the earth's surface? int3: an earthquake can change earth's surface in a short amount of time.

#30. (by #29 #2) Step4: How soon do earthquakes change the surface of the earth? hypothesis: an earthquake can change earth 's surface rapidly.

#31. Proof: sent10 & sent4 -> int1; sent12 & sent20 -> int2; int1 & int2 -> int3: an earthquake can change earth's surface in a short amount of time; int3 & sent2 -> hypothesis.

Please answer the following question by following the above template.

Hypothesis: as the distance of the star to earth decreases, the star will appear brighter
Context:
sent1: a star produces light and heat
sent2: far is the opposite of close
sent3: as the distance from an object increases the force of gravity on that object will decrease
……
sent24: the properties of something are used for describing that something
sent25: moving away from the source increases the distance
Answer:

Figure 14: Stepwise Chain-of-thoughts prompts (COT). The blue box shows the sample prompts of the input and the green box shows the test data for the LLMs.