

# From Sentences to Proof Trees: Leveraging Language Models for Structured Reasoning

Aayushee Gupta

International Institute of Information  
Technology Bangalore, India  
aayushee.gupta1@iiitb.ac.in

## Abstract

The ability of AI systems to not only answer complex natural language questions, but also transparently justify their reasoning, is crucial for building trust and enabling effective human-AI collaboration. In domains requiring multi-hop reasoning, answers must often be constructed by combining multiple relevant sentences from a knowledge base to build an inferential path from the question toward the answer. We tackle this challenge by exploring a neuro-symbolic approach to reasoning through the generation of entailment trees – structured, step-by-step proof trees – using Large Language Models (LLMs). These trees provide interpretable justifications for the inference process. Using the EntailmentBank (Dalvi et al., 2021) data set, we evaluated a diverse set of prompting strategies across multiple models, along with a proposal of an inference-guided prompting approach that performs well. We also fine-tuned LLMs trained specifically for proof generation by applying several data augmentation, curriculum learning, and reinforcement-guided optimization strategies. Our results show that the fine-tuned model outperforms all prompting strategies, achieving superior performance across multiple structural and semantic metrics. We also provide a detailed evaluation of which training strategies are helpful towards proof generation. Our findings highlight the importance of proof tree generation as a benchmark for evaluating structured reasoning in LLMs.

## 1 Introduction

Multi-hop inferencing over Knowledge Bases is widely used in answering complex questions that usually require a chain of facts to be presented, reflecting the reasoning behind the answer. But to a layman, simply looking at the facts relevant to a query is not enough; the path of reasoning created with the facts and connected to both the query and the answer ascertains how and why it is the correct answer. This path of reasoning can also be

structured as a multi-step proof tree that proves the hypothesis [query concatenated with the correct answer] via multiple pieces of relevant facts paired with intermediate natural language proof-step conclusions. Multi-step entailment trees (Dalvi et al., 2021) from a science text KB can be used to explain the line of reasoning behind the answers to grade-school science questions. Students can examine the tree structure to grasp the step-by-step logic behind inferences. Proof tree generation involves picking facts from the corpus and composing them recursively, thereby building a proof/explanation tree for a query.

Language models capable of producing explicit proof trees offer improved interpretability, debuggability, and transparency compared to unstructured text outputs, addressing concerns around black-box reasoning and supporting user trust. In this work, we focus on the foundational problem of generating high-quality deductive proof trees in a static, one-shot setting, which we view as a necessary step toward more interactive human-AI reasoning systems. More broadly, our results contribute to the neuro-symbolic AI literature by demonstrating that LLMs can be guided to produce structured symbolic representations for multi-hop inference. We evaluate the boundaries of structured reasoning by answering two central research questions: (a) How do prompting techniques impact the structural and semantic accuracy of entailment trees across different model scales? and (b) How does task-specific fine-tuning compare to prompting, and what impact does Group Relative Policy Optimization (GRPO) (Shao et al., 2024) have on logical soundness versus structural correctness?

The main contributions of our work are as follows:

### 1. Comprehensive Evaluation of Prompting Strategies for Entailment Tree Generation

We present systematic comparisons of multi-

ple prompting strategies—zero-shot, few-shot, chain-of-thought, for generating multi-step entailment trees using LLMs.

## 2. Inference-Guided Prompting Strategy for Entailment Tasks

We introduce an *Inference-Guided Prompting* strategy, where the prompts incorporate abstract reasoning templates that reflect common inference types (for example: substitution, rule-based inference, inheritance). These templates guide the model’s internal reasoning process without appearing in the final output, enabling more coherent and logically grounded generation.

## 3. Demonstration of Fine-tuning Superiority on EntailmentBank Task 1 for Structured Reasoning

We show that fine-tuning a general-purpose LLM like Meta-Llama (Grattafiori et al., 2024) model on Task 1 of the EntailmentBank dataset significantly outperforms all prompting strategies, establishing a new performance benchmark for entailment tree generation in the no-distractor setting.

## 4. Positioning Proof Tree Generation as a Benchmark for LLM Reasoning

We propose entailment tree generation as a rigorous and interpretable benchmark to evaluate the multi-hop reasoning skills of LLMs.

This paper is organized as follows: Section 2 reviews prior research on entailment tree generation, task description and data description in Section 3 followed by details of our prompting and finetuning approaches in Sections 4 and 5, evaluation in Section 6 along with results & discussion in Section 7, finishing with conclusion and future work in Section 8.

## 2 Related Work

Several attempts at explanation tree generation use generative models like T5 to fine-tune and generate the complete proof tree given the relevant input data (Dalvi et al., 2021; Ribeiro et al., 2022; Tafjord et al., 2020).

A few recent works, like NLProofS (Yang et al., 2022), also explore step-wise proof generation while still using the fine-tuned T5 model for generation, but conditioned on the hypothesis and use

a fine-tuned RoBERTa-based model for preventing hallucination and proof step verification. A search algorithm then uses the validation scores to decide which path to explore next. Ribeiro et al. (2022) iteratively generate entailment trees by retrieving relevant premises and producing one step at a time, showing better accuracy than single-pass generation with gold premises, though their model struggles with trees having more than four steps. The MetGen system (Hong et al., 2022) iteratively generates the entailment tree using both deductive and abductive approaches, using the intermediate generated conclusions in the next round of tree generation. They explicitly model different types of logical reasoning as separate modules, using a controller to orchestrate their execution at every tree generation step. A probing study of multi-step reasoning capabilities of LLMs was done, showcasing their attention patterns encoding the reasoning tree (Hou et al., 2023), which shows promise, but their study is limited to depth-1 proof trees. Contrastive decoding with a hard negative strategy is suggested by Su et al. (2023) to improve the accuracy of finding the correct leaf and proof steps in the tree, but without improvement in generating correct intermediate conclusions. Shi et al. (2024) generate proof trees by finding and reusing similar logical examples through a prototypical network and information entropy-based reranking, demonstrating an improved performance on the EntailmentBank dataset, but only involving three types of logical patterns found in the dataset. SEER (Chen et al., 2024) employs reinforcement learning to generate logically coherent entailment trees by capturing the hierarchical and branching structures inherent in complex reasoning tasks through a structure-based return. A two-system approach (FRVA) is suggested by Fan et al. (2024) that intuitively filters irrelevant facts via System 1 and employs bidirectional reasoning with cross-verification and contrastive learning via System 2, depicting state-of-the-art performance on the EntailmentBank dataset. Zheng et al. (2024) propose prompting and decoding refinements for generating proof trees by LLMs; however, their benchmark results are reported in a different format than those in the original EntailmentBank dataset paper that we use as baseline. Similarly, Zhang et al. (2024) leverage rhetorical perception to identify relations between sentences, enhancing the interpretability of generated trees.

Prompting strategies have been instrumental in

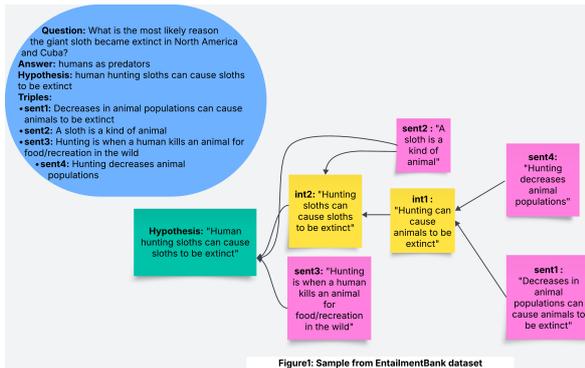


Figure 1: Example from the EntailmentBank Task 1 dataset, illustrating the construction of a proof tree using provided KB facts for a given Question, Answer, and Hypothesis. The ‘int’ labels denote intermediate conclusions derived from selected facts.

eliciting complex reasoning from LLMs. Several recent works (Wei et al., 2022; Lu et al., 2022; Yao et al., 2023b; Fu et al., 2022) demonstrated the effectiveness of Chain-of-Thought (CoT) prompting, while Yao et al. (2023a) extended this idea to Tree-of-Thought (ToT) prompting for deliberate multi-path reasoning. These methods suggest that strategic prompt design can significantly impact reasoning faithfulness and structure.

In this work, we diverge from existing literature by exploring two distinct, complementary directions: (1) Inference-Guided prompting, which employs abstracted logical templates (e.g., substitution, property inheritance) to provide models with a structural blueprint, and (2) model-level optimization through Supervised Fine-Tuning and Group Relative Policy Optimization (GRPO) to improve proof tree generation without requiring complex external controllers.

### 3 Task and Data Description

#### 3.1 Task Description

Explanation tree generation is framed as a multi-step, multi-premise entailment task. Given a hypothesis  $h$  – typically formed by concatenating the question and answer – and a set of relevant knowledge base sentences  $S$ , the objective is to construct a valid entailment tree. The tree must have its leaf nodes drawn from  $S$ , and intermediate nodes  $D$  (i.e., derived facts) generated from leaf nodes to logically support back the conclusion at the root node (hypothesis  $h$ ). A valid entailment tree must preserve the property of entailment between each parent node and its child nodes.

We operate under the **no-distractor setting** (Task 1), where all leaf sentences  $l$  (given facts) are already known to be relevant to the query and are drawn from the KB (i.e.,  $l \in S$ ). Alongside the question, answer, and hypothesis  $h$ , each example in this task can be represented as a tuple  $\langle q, a, h, l \rangle$ , for which the goal is to generate a valid proof tree  $P$ .

**Motivation for Task 1.** We focus on the no-distractor setting (Task 1), where all leaf sentences  $l$  are known to be relevant. This choice is motivated by the need to isolate the LLM’s structural reasoning and generation abilities from its retrieval performance. Task 1 provides a controlled environment to benchmark any model’s capacity for multi-step entailment tree generation so we can more accurately measure the impact of our proposed approaches without the confounding variable of retrieval noise.

#### 3.2 Dataset Description

The EntailmentBank (Dalvi et al., 2021) dataset consists of 1840 hypothesis [question and answer combined] samples along with their corresponding multistep entailment trees as well as the hypothesis-relevant facts/triples from a textual KB. The dataset is varied with 50% small [3–5 nodes] and 50% large entailment trees belonging to the grade-school level science domain. Each step of the entailment tree is an entailment and encodes a single inference. The complete dataset consists of 5,881 discrete entailment steps, wherein each entailment tree includes 7.6 nodes and 3.2 entailment steps on average. The given dataset has examples in the format: Question, Answer, Hypothesis, Given KB Facts, Core concepts, Proof Tree. We perform an evaluation on all 340 test examples from the EntailmentBank dataset<sup>1</sup>.

Figure 1 shows a typical multi-step entailment tree from the dataset. To prove the hypothesis that human hunting causes sloths to go extinct, the model must bridge the gap between specific species (sloths) and general biological principles. First, **sent4** and **sent1** are combined to form **int1** via Inference from Rule, establishing a general rule that hunting leads to extinction in animals via population decrease. Simultaneously, the model must recognize the taxonomic relationship in **sent2** (“A sloth is a kind of animal”). By

<sup>1</sup>[https://github.com/allenai/entailment\\_bank](https://github.com/allenai/entailment_bank)

applying the general rule (**int1**) to the specific category (**sent2**) through Substitution (or Property Inheritance), the model derives **int2** (“Hunting sloths can cause sloths to be extinct”). Finally, this is combined with the context of human action in **sent3** to reach the hypothesis. While this proof tree structure assumes the inheritance of properties (what applies to animals applies to sloths), it provides a verifiable trace of the logical flow in the multi-step entailment trees.

**Why EntailmentBank matters.** Given the reasoning effort required to construct natural language proof trees, we consider EntailmentBank a valuable stepping stone for evaluating and developing future LLMs with multi-hop reasoning capabilities. It tests logical skills such as taxonomic inference, rule application, conjunction, and compositional reasoning in the context of scientific understanding. In contrast to other popular LLM evaluation question answering datasets – such as Codeforces (Penedo et al., 2025) (programming), MGSM (Shi et al., 2022), GSM8K (Cobbe et al., 2021), AIME (MAA, 2024), and FrontierMath (Glazer et al., 2024) (math reasoning), MMLU (Hendrycks et al., 2020) (multi-domain QA), or GPQA Diamond (Rein et al., 2024) (PhD-level science) and several other reasoning tasks (Yu et al., 2022) – EntailmentBank uniquely focuses on explicit multi-step explanation generation and includes human-authored ground truth entailment trees. This makes it particularly suitable for benchmarking interpretability, reasoning faithfulness, and explanation quality in next-generation language models. Moreover, He et al. (2023) demonstrate that the incorporation of natural language explanations increases the robustness of using LLMs with adversarial datasets such as natural language inference and paraphrase identification. A model trained on EntailmentBank-style proof trees could further enhance this robustness by promoting more systematic and logically grounded reasoning.

## 4 Prompting Approach for Tree Generation

### 4.1 Prompting Strategies

We explore the following prompting strategies in this work:

- **Zero-shot prompting:** The LLM is presented with the expected proof format, the test question with its correct answer, hypothesis, and

the set of facts required to construct the proof tree.

- **Few-shot prompting:** The zero-shot prompt is enhanced with  $k = 2$  examples of constructed proof trees sampled from the training dataset.
- **Chain-of-Thought (CoT) prompting:** The few-shot prompt is augmented with natural language explanations for each reasoning step, provided in a sequential fashion to guide the proof generation process.
- **Inference-Guided prompting:** The few-shot prompt is augmented with abstracted reasoning templates that reflect common logical patterns (e.g., substitution, inheritance) as identified by Dalvi et al. (2021). These templates provide the model with a structural blueprint for composing logical steps without appearing in the final output. Figure 2 illustrates the specific rules used.

All prompts explicitly specify the desired entailment tree format to promote consistent and well-structured outputs and are presented in Appendix A.

## 4.2 Models Evaluated

We evaluate the following language models with the prompting approaches, chosen to represent a range of model scales, design philosophies, and availability: **o1-mini** (OpenAI, 2025), **Phi-4** (Abdin et al., 2024), **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024), **Gemini Flash 2.0** (Comanici et al., 2025).

## 5 Finetuned Model for Proof Tree Generation

While chain-of-thought and inference-guided prompting improve reasoning quality, our experiments show that they remain insufficient for generating structured entailment trees. This motivates the use of supervised fine-tuning (SFT) to reliably produce explicit and verifiable proof structures. In SFT, a pre-trained language model is optimized using negative log-likelihood to generate the exact token sequences of gold proofs, allowing it to adapt to both the required output format and domain-specific terminology.

We fine-tune the Meta LLaMA-3.1 8B model using 1,313 training examples from EntailmentBank

**Abstracted Inference Templates** (for model’s internal reasoning only): These templates guide how to abstract and combine sentences logically:

- **SUBSTITUTION:** [General Entity] has [Property] & [Specific Entity] is a kind of [General Entity] → int: [Specific Entity] has [Property]
- **INFERENCE FROM RULE:** If [Condition], then [Effect] & [Case] exhibits [Condition] → int: [Case] exhibits [Effect]
- **FURTHER SPECIFICATION:** [Entity] requires [Need] in [Context] & [Mechanism] provides [Need] → int: [Mechanism] provides [Need] in [Context]
- **INFER CLASS:** A [Class] is made of [Components] & [Instance] is made of [Same Components] → int: [Instance] is a kind of [Class]
- **PROPERTY INHERITANCE:** An [Entity]’s [Part] is [Feature] & Something [Feature] is used for [Purpose] → int: [Entity]’s [Part] is used for [Purpose]
- **SEQUENTIAL INFERENCE:** [Event-B] follows [Event-A] & descriptions of both → int: [Overall Process] involves [A] then [B]

Figure 2: Abstracted Inference Templates guiding the model’s internal logical reasoning.

Task-1, where each input consists of a question, answer, hypothesis, and relevant KB facts. Training is performed using a 4-bit quantized model with parameter-efficient fine-tuning (PEFT) for 10 epochs. Despite hyperparameter tuning, evaluation loss consistently increases and test accuracy quickly plateaus, indicating overfitting. This observation motivates the exploration of alternative fine-tuning strategies beyond standard SFT.

### 5.1 Dataset Augmentation

We use the following dataset augmentation approaches to increase the dataset size and variety:

1. Triples Permutation: Modify the order of given sentence facts in the training sample, leading to corresponding modification in the output proof. We generated 2230 permuted training samples with this approach.
2. Paraphrasing: Mistral-7B-Instruct (Jiang et al., 2024) model is used to generate a paraphrase of the input sentences. The sentences to paraphrase from each training sample are chosen at random, which include any of the question, hypothesis, or relevant KB facts. This does not alter the proof to be generated. We generated 1313 paraphrased training samples.

3. Commutation Permutation: Modify the order of commutative operands in the proof structure randomly without modifying the example text. We generated 1313 training samples with this approach.

### 5.2 Curriculum Training

Bengio et al. (2009) suggest significant improvement in generalization while using ‘Curriculum Learning’ - an approach that sorts training examples into a sequence that illustrates the simpler concepts first. This acts as a form of continuation method, speeding up convergence and finding a better local minimum. We use this approach to sort all training examples by proof length and then train the model through SFT. This helps the model to prevent overfitting and generalize better to new and more complex multi-step examples.

### 5.3 Reinforcement-Guided Learning

Since we noticed the limit of what SFT could achieve on the entailment tree generation task, we also explored reinforcement-guided learning through the Group Relative Policy Optimization (Shao et al., 2024) (GRPO) technique proven to be effective at enhancing math reasoning capabilities, and aligned it to our task of proof generation.

This method generates a group of multiple proof completions for a single example from the SFT

model. A reward model, composed of several reward functions, then computes rewards by comparing these proofs to the ground truth. Finally, the SFT model’s policy is updated based on a group-normalized advantage calculation. We design the following reward functions based on the step proof structure and intermediate node text generation:

1. Overall Tree BLEURT Score: Rewarding the exact overall tree generated as the ground truth proof tree
2. Intermediate Node Text BLEURT Score: Rewarding correctly generated intermediate node text when compared to the ground truth intermediate node text.
3. Intermediate Node Reuse Frequency: Rewarding correct reuse of intN nodes across steps and penalizing “dead-end” intermediate nodes that aren’t reused
4. Steps Match BLEURT Score: Rewarding correct order of proof step resolution (i.e., leaves first  $\rightarrow$  root)
5. Proof Tree Length Match Score: Rewarding completions with correct proof length compared to the ground truth proof

We denote the model policy by

$$\pi_{\theta}(y | x) = \prod_{t=1}^T \pi_{\theta}(y_t | x, y_{1:t-1}).$$

For each prompt  $x_i$ , we sample a group of  $K$  completions  $\{y_{i,1}, \dots, y_{i,K}\} \sim \pi_{\theta_{\text{old}}}(\cdot | x_i)$  and compute scalar rewards  $R_{i,j} \equiv R(y_{i,j}, x_i)$  constructed as a weighted sum of  $M$  components:

$$R(y, x) = \sum_{m=1}^M w_m r_m(y, x),$$

where  $r_m$  are the BLEURT-based and heuristic rewards described above.

For group-normalized advantages, we compute

$$\mu_i = \frac{1}{K} \sum_{j=1}^K R_{i,j}, \quad \sigma_i = \sqrt{\frac{1}{K} \sum_{j=1}^K (R_{i,j} - \mu_i)^2},$$

and define

$$\hat{A}_{i,j} = \frac{R_{i,j} - \mu_i}{\sigma_i + \varepsilon}.$$

The GRPO surrogate loss (critic-free) minimized w.r.t.  $\theta$  is:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{NK} \sum_{i=1}^N \sum_{j=1}^K \hat{A}_{i,j} \log \pi_{\theta}(y_{i,j} | x_i).$$

Unlike PPO-based RLHF methods, GRPO omits an explicit KL regularization term and achieves stability through group-relative normalization of rewards. Thus, no KL penalty term is used in our implementation. Model parameters are updated via stochastic gradient descent on  $\mathcal{L}_{\text{GRPO}}(\theta)$ , and the sampling policy is periodically refreshed as  $\theta_{\text{old}} \leftarrow \theta$ . We used  $K = 4$  in our experiments.

## 6 Evaluation

### 6.1 Evaluation Metrics

We adopt multiple evaluation metrics to assess proof tree generation. These are derived from the official EntailmentBank<sup>2</sup> evaluation script, which compares each generated proof tree  $\hat{T}$  against a gold-standard tree  $T$  using a structured alignment process based on the “sent\*” labels. Once aligned, performance is evaluated across four categories:

- **Leaves (F1, AllCorrect)**: These metrics assess whether the leaf nodes in  $\hat{T}$  match those in  $T$ . The F1 score captures the overlap, while the AllCorrect score is 1 only if there is a perfect match.
- **Steps (F1, AllCorrect)**: These metrics evaluate the structural correctness of proof steps, i.e., whether internal nodes in  $\hat{T}$  correctly reproduce the logical steps in  $T$  based on aligned children.
- **Intermediates (F1, AllCorrect)**: These assess semantic similarity for intermediate conclusions (non-leaf, non-root nodes). An intermediate node is considered correct if the BLEURT score (int-BLEURT) with the aligned node in  $T$  exceeds a threshold (0.28).
- **Overall Proof (AllCorrect)**: This metric evaluates full-tree correctness. A score of 1 is assigned only if all leaves, steps, and intermediate nodes are perfectly aligned with the ground truth proof tree.

We further evaluate how the distribution of generated proof trees varies across entailment step counts

<sup>2</sup>[https://github.com/allenai/entailment\\_bank](https://github.com/allenai/entailment_bank)

(i.e., proof tree depths) for our fine-tuned models, as compared to the ground truth in Figure 3. This analysis helps assess how closely the generated proof tree lengths align with the reference trees. Figure 4 additionally presents the accuracy comparison across different proof tree depths for the model-generated proof trees and ground truth trees.

## 7 Results and Discussion

### 7.1 Performance of Prompting Strategies

Table 1 summarizes the performance of models under different prompting strategies. Our experiments reveal four main insights:

- **Few-shot prompting consistently outperforms zero-shot prompting.** Most models benefit from example-driven generalization, while zero-shot remains insufficient for structured generation (with Flash 2.0 as an outlier).
- **Inference-guided prompting improves over few-shot.** Our domain-specific reasoning templates help constrain generation and yield better proof trees, though effectiveness drops without at least one illustrative example. All models show a huge jump in Leaves-F1 and Leaves-AllCorrect metrics with Inference-Guided prompting compared to other strategies.
- **High premise recall, weak multi-hop reasoning.** Llama and Phi models identify relevant leaves accurately specifically but struggle on step-level inference.
- **Chain-of-Thought prompting is model-dependent.** It works well for Flash 2.0 but structured proof generation failed with Llama and Phi in our experiments, suggesting that some models cannot reliably translate reasoning chains into structured proofs.

These findings reinforce the need for more effective prompt engineering, targeted fine-tuning, and enhanced supervision signals to bridge the gap between factual retrieval and reliable multi-hop reasoning.

### 7.2 Performance of Supervised Fine-tuned LLMs

Table 2 shows the results of the fine-tuned Llama 3.1 8B models under different training strategies. We observe that:

- **Fine-tuning outperforms prompting.** Task-specific training yields the highest scores across all metrics, highlighting its effectiveness for structured reasoning. Compared to the prompting strategies in Table 1, our best supervised fine-tuned model (37.35% Overall AllCorrect) demonstrates a substantial improvement over the strongest prompting baseline, Gemini Flash 2.0 with Chain-of-Thought (29.71%) across all evaluation metrics.
- **Dataset augmentation helps but saturates.** Structural and lexical variations improve performance, though additional augmentation offers diminishing returns.
- **Curriculum learning aids stability.** It reduces overfitting and aligns proof length with gold trees for shallower depths (Figures 3 and 4), but accuracy declines beyond depth 5.
- **Best results from combined augmentation.** The strongest model uses the original dataset plus triples permutation and paraphrasing, with curriculum training providing similar gains.
- **Other LLMs underperform.** Fine-tuning Flash, Deepseek, and Qwen variants did not surpass fine-tuned Llama models.

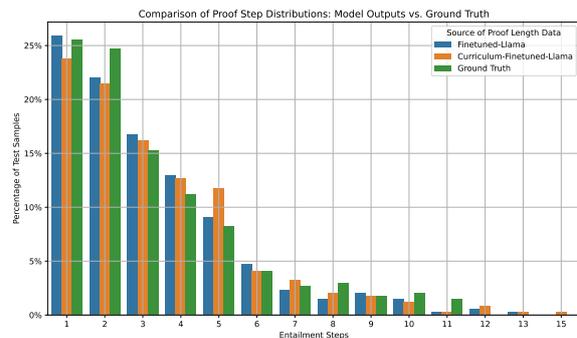


Figure 3: Comparison of the number of steps, i.e., proof tree depth in entailment trees generated by different approaches.

### 7.3 Impact of Reinforcement-Guided Optimization

Table 3 presents overall accuracy results from supervised fine-tuned Llama 3.1 8B LLM vs GRPO Reinforcement-guided model with the designed reward functions. Following are the findings from the GRPO experiments:

Model	Prompting Strategy	Leaves-F1	Leaves-AllCorrect	Steps-F1	Steps-AllCorrect	int-BLEURT-F1	int-BLEURT-AllCorrect	Overall-AllCorrect
Flash 2.0	Few-shot	91.39	61.47	40.33	26.47	65.78	31.76	23.82
Flash 2.0	Zero-shot	90.99	59.12	42.19	30.29	63.64	31.47	27.06
Flash 2.0	Inference-Guided	94.20	68.50	43.20	26.40	67.03	31.47	25.29
Flash 2.0	<b>Chain of Thought</b>	92.88	63.24	<b>44.95</b>	<b>31.47</b>	<b>68.13</b>	<b>35.59</b>	<b>29.71</b>
o4mini	Zero-shot	92.99	64.71	37.48	20.59	61.98	26.18	16.76
o4mini	Few-shot	94.48	70.29	39.97	25.29	65.00	28.53	20.88
o4mini	Inference-Guided	94.44	70.00	43.32	25.59	67.71	29.71	22.35
Llama 3.1 Instruct	Few-shot	96.50	77.90	32.00	10.80	59.60	23.50	9.70
Llama 3.1 Instruct	Inference-Guided	<b>97.33</b>	80.88	30.88	9.12	60.38	23.24	8.53
Phi-4	Few-shot	94.24	67.94	40.26	26.47	62.85	30.88	24.71
Phi-4	Inference-Guided	97.26	<b>81.18</b>	43.94	30.29	64.85	32.35	27.35

Table 1: Performance of LLMs on Entailment Tree Generation with different prompting strategies

Data Augmentation	Leaves-F1	Leaves-AllCorrect	Steps-F1	Steps-AllCorrect	int-BLEURT-F1	int-BLEURT-AllCorrect	Overall-AllCorrect	Epochs	Curriculum
No Augmentation	99.71	96.76	48.31	37.35	69.91	36.76	34.11	10	No
No Augmentation	99.57	94.41	48.76	37.94	69.15	37.06	35.00	5	Yes
Paraphrased	99.70	95.88	51.76	38.24	70.45	37.35	36.18	10	No
Paraphrased	99.71	95.88	50.10	38.53	71.05	<b>40.29</b>	37.06	5	Yes
Commutation Permutation	99.72	95.88	<b>52.60</b>	39.71	<b>72.12</b>	38.53	35.59	10	No
Commutation Permutation	99.61	95.29	47.77	34.71	69.42	35.29	31.76	5	Yes
Triples Permutation	99.54	94.71	51.20	39.71	71.20	39.12	35.29	10	No
Triples Permutation	99.69	95.59	50.85	38.24	70.37	39.71	35.59	5	Yes
Comm. Perm. + Paraphrased	99.74	96.18	50.04	36.76	70.34	36.18	34.12	5	Yes
Triples Perm. + Paraphrased	<b>99.80</b>	<b>97.06</b>	52.36	38.82	71.89	39.71	36.47	5	Yes
Triples Perm. + Paraphrased	99.59	95.00	54.52	<b>41.47</b>	72.03	40.00	<b>37.35</b>	20	No

Table 2: Performance of Llama 3.1 8B finetuned models on EntailmentBank Task-1 test dataset across data augmentation strategies, with and without curriculum training.

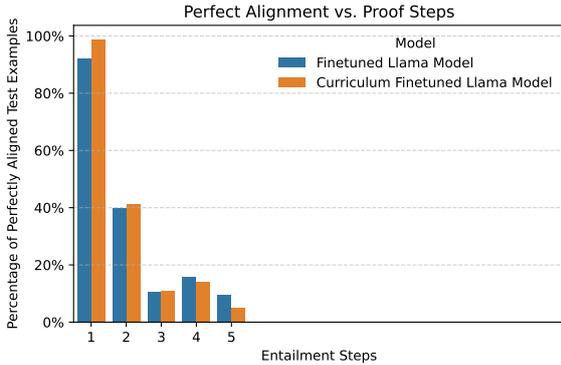


Figure 4: Comparison of overall accuracy (i.e., perfect tree alignment score) across different proof tree depths for Fine-tuned Llama and Curriculum-Finetuned Llama models

- GRPO yields limited improvements when the baseline model (SFT or curriculum-SFT) already performs well, with more consistent gains observed only when the initial error margin is larger, particularly for curriculum-trained models.
- GRPO encourages better-structured proof trees that reuse relevant sentence facts from the context, but does not explicitly enforce logical validity of intermediate inferences, suggesting the need for logic-aware reward functions.
- Reward trajectories exhibit high variance rather than stable convergence. While the re-

ward\_chain component quickly saturates, indicating consistent reuse of intermediate nodes, this saturation may restrict further structural improvements.

- GRPO achieves optimal results after a single epoch. Unlike the 10-epoch SFT phase required for knowledge acquisition, GRPO acts as a policy refinement stage. With a group size of  $G = 4$ , each training step is significantly more computationally dense and sample-efficient than standard SFT. Continued training leads to performance degradation, likely due to reward exploitation where structural optimization begins to compromise logical validity.
- Qualitative analysis (Appendix B) reveals recurring reasoning errors not corrected by GRPO, including hallucinated intermediates, misuse of facts, and structurally divergent yet semantically plausible proofs, highlighting limitations of purely structural evaluation metrics.

## 7.4 Overall Results and Comparative Analysis

Table 4 presents the results from our best finetuned model (Llama 3.1 8B optimized with GRPO). We compare our approach against prior state-of-the-art systems including EntailmentWriter (Dalvi et al., 2021), MetGen (Hong et al., 2022), NL-ProofS (Yang et al., 2022), and FRVA (Fan et al.,

Data Augmentation Technique	Overall-AllCorrect-SFT	Overall-AllCorrect-GRPO	Epochs-SFT	Epochs-GRPO	SFT Curriculum Training	Metric Improvements ( $\Delta$ )
Commutation Permutation	35.59	35.88	10	1	No	Int-BLEURT, Overall Acc
Triples Perm. + Paraphrased	37.35	37.94	20	1	No	Leaves, Steps, Int-BLEURT, Overall Acc
Triples Perm. + Paraphrased	36.47	37.35	5	1	Yes	Steps, Int-BLEURT, Overall Acc

Table 3: Comparison of SFT LLaMA model vs GRPO finetuned model performance on Task-1 test set from EntailmentBank under different data augmentation settings. Metric improvements ( $\Delta$ ) indicate where GRPO outperformed SFT.

Evaluation Level	Leaves		Steps		Intermediates		Overall
	F1	AllCorrect	F1	AllCorrect	F1	AllCorrect	AllCorrect
<b>Ours (Best Finetuned Model)</b>	99.67	95.0	54.93	42.06	72.43	40.59	37.94
EntailmentWriter (T5-11B) (Dalvi et al., 2021)	99.0	89.4	51.5	38.2	71.2	38.5	35.3
MetGen (Hong et al., 2022)	<b>100.0</b>	<b>100.0</b>	57.7	41.9	70.8	39.2	36.5
NLProofS (Yang et al., 2022)	97.8 $\pm$ 0.2	90.1 $\pm$ 1.2	55.6 $\pm$ 0.6	42.3 $\pm$ 0.4	72.4 $\pm$ 0.5	40.6 $\pm$ 0.7	38.9 $\pm$ 0.7
FRVA (Fan et al., 2024)	98.2 $\pm$ 0.3	94.0 $\pm$ 1.0	<b>57.8 <math>\pm</math> 0.4</b>	<b>44.4 <math>\pm</math> 0.6</b>	<b>73.5 <math>\pm</math> 0.4</b>	<b>42.4 <math>\pm</math> 0.3</b>	<b>40.3 <math>\pm</math> 0.7</b>

Table 4: Evaluation results from our best fine-tuned model and comparison with recent work on the Task-1 test dataset from EntailmentBank

2024). Our model shows strong performance, particularly in identifying the correct leaf facts. While our model is not the top performer across all metrics, it is highly competitive. Its Overall Acc score of 37.94 places it in the top three, just behind FRVA (40.3), and is closely aligned with NLProofS and FRVA for correct intermediate conclusion and steps generation. A clear and expected trend across all models is the significant drop in performance as the task becomes more complex, i.e., moving from leaves to generating the multi-step proof tree.

A direct comparison with the prompting results in Table 1 reveals that even the most sophisticated prompting on larger models, such as Gemini Flash 2.0, fails to match the structural precision of a smaller, task-specifically fine-tuned model. Specifically, our fine-tuned model achieves an absolute improvement of 8.23% in Overall AllCorrect metrics over the strongest CoT baseline and 10.59% over our proposed Inference-Guided prompting. This suggests that the complex constraints of multi-step entailment – such as maintaining intermediate state consistency – are more effectively captured through direct weight updates and reinforcement learning than through structured prompts alone.

## 8 Conclusion and Future Work

In this work, we study multi-step entailment tree generation for explaining complex queries over a textual knowledge base. Our experiments show a clear progression from prompting to task-specific fine-tuning: while few-shot and inference-guided prompting improve over zero-shot baselines, they remain inadequate for complex reasoning. Supervised fine-tuning yields the strongest performance,

yet models continue to struggle with deeper reasoning chains. Reinforcement-based optimization (GRPO) provides only limited benefits, highlighting the need for reward functions that assess logical soundness rather than structural correctness alone.

Future work will focus on designing reward functions that penalize unsupported or logically inconsistent intermediate steps, encouraging grounded and verifiable reasoning during reinforcement learning. We also plan to develop evaluation metrics that better capture semantically valid but structurally divergent entailment trees, enabling more faithful assessment of reasoning quality. In addition, integrating search-based approaches such as Tree-of-Thought (Yao et al., 2023a) with fine-tuning may allow models to explore and self-correct multiple reasoning paths, improving reasoning depth and reliability. We also aim to extend our evaluation to the Task 2 and Task 3 settings of EntailmentBank to assess how reinforcement-guided models handle the presence of irrelevant distractors. Finally, extending proof generation to dynamic settings where evidence evolves over time would broaden the applicability of these models to adaptive explanation generation.

## Limitations

Despite the improvements obtained through supervised finetuning and GRPO-based reinforcement learning, our approach inherits several limitations of current LLM-based reasoning systems:

- **Hallucinations and logical inconsistencies.** LLMs may still produce unsupported or logically invalid intermediate steps. This occurs most prominently in prompting-based settings,

although finetuning and GRPO reduce but do not eliminate such errors.

- **Structural sensitivity in evaluation metrics.** Existing metrics for entailment tree generation emphasize structural matching and may penalize semantically correct but structurally different reasoning paths. This limits our ability to fully assess the quality of reasoning.
- **Generalization to longer reasoning chains.** All evaluated models struggle with deep multi-step reasoning, often breaking down as the required proof depth increases. Improving robustness for long-chain reasoning remains an open challenge.
- **Benchmark availability.** EntailmentBank remains the only benchmark with gold multi-step natural language proof trees. The lack of additional datasets with comparable annotations limits our ability to assess cross-domain generalization. While our study focuses on Task 1 to isolate reasoning performance, future work is needed to evaluate these methods in "distractor" settings (Tasks 2 and 3) to assess robustness against irrelevant information.
- **Scope limited to Deductive and Monotonic Reasoning.** Our approach, like EntailmentBank itself, focuses on deductive inference over a fixed set of premises and does not address defeasible or non-monotonic reasoning, where conclusions may change when new or conflicting evidence arises.

## References

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, and 1 others. 2024. [Phi-4 technical report](#). *Preprint*, arXiv:2412.08905.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Guoxin Chen, Kexin Tang, Chao Yang, Fuying Ye, Yu Qiao, and Yiming Qian. 2024. [SEER: Facilitating structured reasoning and explanation via reinforcement learning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5901–5921, Bangkok, Thailand. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, and 1 others. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. Google DeepMind. Accessed: 2025-07-08.
- 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*, pages 7358–7370.
- Yue Fan, Hu Zhang, Ru Li, Yujie Wang, Hongye Tan, and Jiye Liang. 2024. Frva: Fact-retrieval and verification augmented entailment tree generation for explainable question answering. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9111–9128.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.
- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and 1 others. 2024. [The Llama 3 Herd of Models](#). *Preprint*, arXiv:2407.21783.
- Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2023. Using natural language explanations to improve robustness of in-context learning. *arXiv preprint arXiv:2311.07556*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. Metgen: A module-based entailment tree generation framework for answer explanation. *arXiv preprint arXiv:2205.02593*.
- Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. *arXiv preprint arXiv:2310.14491*.

- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, and 1 others. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.
- Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.
- MAA. 2024. American invitational mathematics examination (aime). Accessed: July 2025.
- OpenAI. 2025. Introducing o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>. Accessed: 2025-05-01.
- Guilherme Penedo, Anton Lozhkov, Hynek Krdlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. 2025. Codeforces. <https://huggingface.co/datasets/open-r1/codeforces>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Danilo Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henry Zhu, Xinchu Chen, Zhiheng Huang, Peng Xu, Andrew Arnold, and 1 others. 2022. Entailment tree explanations via iterative retrieval-generation reasoner. *arXiv preprint arXiv:2205.09224*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.
- Jihao Shi, Xiao Ding, and Ting Liu. 2024. Case-based deduction for entailment tree generation. *Mathematics*, 12(18):2893.
- Ying Su, Xiaojin Fu, Mingwen Liu, and Zhijiang Guo. 2023. Are llms rigorous logical reasoner? empowering natural language proof generation with contrastive stepwise decoding. *arXiv preprint arXiv:2311.06736*.
- Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, 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*.
- Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. Generating natural language proofs with verifier-guided search. *arXiv preprint arXiv:2205.12443*.
- Shunyu Yao, Jeffrey Zhao, Izhak Gurion, Dian Yu, Yuchen Zhang, Yoav Artzi, and Yejin Choi. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10683*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
- Ping Yu, Tianlu Wang, Olga Golovneva, Badr Alkhamissi, Siddharth Verma, Zhijing Jin, Gargi Ghosh, Mona Diab, and Asli Celikyilmaz. 2022. Alert: Adapting language models to reasoning tasks. *arXiv preprint arXiv:2212.08286*.
- Longyin Zhang, Bowei Zou, and Aiti Aw. 2024. Empowering tree-structured entailment reasoning: rhetorical perception and llm-driven interpretability. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5783–5793.
- Zi'ou Zheng, Christopher Malon, Martin Renqiang Min, and Xiaodan Zhu. 2024. Exploring the role of reasoning structures for constructing proofs in multi-step natural language reasoning with large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15299–15312.

## A Prompting Templates

### A.1 Prompt A: Zero-shot Prompting

#### Prompt A – Zero-shot Template

You are a helpful assistant that generates step-by-step proof trees for grade 3-8 science problems. Each proof tree must use the given question, answer, hypothesis, and relevant triples. Rules for generating proof trees:

1. Output the proof tree exactly in the following format: sent1 & sent2 -> int1: [Intermediate node text]; int1 & sent3 -> hypothesis;
2. Each proof step must combine only necessary triples and/or previously derived intermediate nodes which must entail the hypothesis/intermediate conclusions.
3. Introduce an intermediate node only when necessary to bridge facts. If a direct step to the hypothesis is possible, avoid extra nodes.
4. Limit proof steps to a maximum of 15. Do not repeat the same intermediate node text again.
5. Do not generate any more text once the 'hypothesis' is reached.
6. Do not copy full sentence text from triples into intermediate node text.

### A.2 Prompt B: Few-shot Prompting

#### Prompt B – Few-shot Template

### Proof Tree Rules:

- **Format:** 'sent1 & sent2 -> int1: [Intermediate node text]; int1 & sent3 -> hypothesis;'
- **Concise:** Do not copy sentence triples or hypothesis from the input.
- **Stop Condition:** The proof must **end immediately** once the 'hypothesis' is reached, finishing with ';

### Example Proof:

Question: During the Apollo 14 moon landing, astronauts played golf on the moon. Which of the following would be less on the moon than on Earth?

Answer: The weight of the golf ball

Hypothesis: the weight of a golf ball will be less on the moon than on Earth

Triples:

sent1: As the mass of a celestial object decreases, the surface gravity of that celestial object weakens sent2: A golf ball is a kind of object sent3: A planet is a kind of celestial object

t/body sent4: As the force of gravity decreases, the weight of the object will decrease sent5: The Moon is a kind of moon sent6: The Earth has more mass than the Moon sent7: Earth is a kind of planet sent8: A moon is a kind of celestial object/body

Proof: sent3 & sent7 -> int1: Earth is a celestial object; sent5 & sent8 -> int2: The Moon is a celestial object; int1 & int2 -> int3: Earth and the Moon are celestial objects; int3 & sent1 & sent6 -> int4: The surface gravity on the Moon will be less than the surface gravity on Earth; int4 & sent4 -> int5: The weight of an object on the Moon will be less than the weight of an object on Earth; int5 & sent2 -> hypothesis;  
### Generate the proof tree for the following grade 3-8 science problem following the above rules using the given question, answer, hypothesis, and **only the necessary** relevant triples and stop at the final proof step:

### A.3 Prompt C: Inference-Guided Prompting

#### Prompt C – Inference-Guided + Few-shot Prompt Template

### Proof Tree Output Rules:

- **Format:** 'sent1 & sent2 -> int1: [Intermediate node text]; int1 & sent3 -> hypothesis;'
- **Concise:** Do not copy sentence triples or hypothesis from the input.
- **Stop Condition:** The proof tree must **end immediately** once the 'hypothesis' is reached, finishing with ';

### Abstracted Inference Templates (for internal reasoning only — do not output these directly): These templates guide how to abstract and combine sentences logically:

- **SUBSTITUTION:** '[General Entity] has [Property]' & '[Specific Entity] is a kind of [General Entity]' → int: '[Specific Entity] has [Property]'
- **INFERENCE FROM RULE:** 'If [Condition], then [Effect]' & '[Case] exhibits [Condition]' → int: '[Case] exhibits [Effect]'
- **FURTHER SPECIFICATION:** '[Entity] requires [Need] in [Context]' & '[Mechanism] provides [Need]' → int: '[Mechanism] provides [Need] in [Context]'
- **INFER CLASS:** 'A [Class] is made of [Components]' & '[Instance] is made of [Same Components]' → int: '[Instance] is a kind of

[Class]‘

- **PROPERTY INHERITANCE:** ‘An [Entity]’s [Part] is [Feature]‘ & ‘Something [Feature] is used for [Purpose]‘ → int: ‘[Entity]’s [Part] is used for [Purpose]‘

- **SEQUENTIAL INFERENCE:** ‘[Event-B] follows [Event-A]‘ & descriptions of both → int: ‘[Overall Process] involves [A] then [B]‘  
### Example Proof:

Question: During the Apollo 14 moon landing, astronauts played golf on the moon. Which of the following would be less on the moon than on Earth?

Answer: The weight of the golf ball

Hypothesis: the weight of a golf ball will be less on the moon than on Earth

Triples: sent1: As the mass of a celestial object decreases, the surface gravity of that celestial object weakens sent2: A golf ball is a kind of object sent3: A planet is a kind of celestial object/body sent4: As the force of gravity decreases, the weight of the object will decrease sent5: The Moon is a kind of moon sent6: The Earth has more mass than the Moon sent7: Earth is a kind of planet sent8: A moon is a kind of celestial object/body

Proof: sent3 & sent7 → int1: Earth is a celestial object; sent5 & sent8 → int2: The Moon is a celestial object; int1 & int2 → int3: Earth and the Moon are celestial objects; int3 & sent1 & sent6 → int4: The surface gravity on the Moon will be less than the surface gravity on Earth; int4 & sent4 → int5: The weight of an object on the Moon will be less than the weight of an object on Earth; int5 & sent2 → hypothesis;

### Generate the proof tree for the following grade 3-8 science problem following the above format rules using the given question, answer, hypothesis, and relevant triples and stop at the final proof step:

#### A.4 Prompt D: Chain of Thought Prompting

##### Prompt D – Chain-of-Thought + Few-shot Prompt Template

### Example Proof:

Question: Drew knows that Earth is tilted on its axis. He also knows this tilt is responsible for the season that a region on Earth will experience. When the South Pole is tilted toward the Sun, what season will it be in Florida?

Answer: winter

Hypothesis: it is winter in Florida

Triples: sent1: florida is a state located in the united states of america sent2: united states is located in the northern hemisphere sent3: the winter in the northern hemisphere is during the summer in the southern hemisphere sent4: the south pole is tilted toward the sun sent5: summer is when a hemisphere is tilted towards the sun sent6: the south pole is located in the southern hemisphere

Let’s think step by step. First, we know the South Pole is tilted toward the sun (sent4). And summer is when a hemisphere is tilted towards the sun (sent5). Therefore, it is summer in the South Pole. Next, the South Pole is located in the southern hemisphere (sent6). So, it is summer in the southern hemisphere. The winter in the northern hemisphere is during the summer in the southern hemisphere (sent3). Therefore, it is winter in the northern hemisphere. Florida is a state located in the United States of America (sent1), and the United States is located in the northern hemisphere (sent2). So, Florida is located in the northern hemisphere. Since it is winter in the northern hemisphere and Florida is in the northern hemisphere, it is winter in Florida. Therefore, it is winter in Florida. Now, let’s format this into a proof tree:

### Generated Proof: sent4 & sent5 → int1: it is summer in south pole; int1 & sent6 → int2: it is summer in southern hemisphere; int2 & sent3 → int3: it is winter in the northern hemisphere; sent1 & sent2 → int4: florida is located in the northern hemisphere; int3 & int4 → hypothesis;  
### Proof Tree Rules:

- **Format:** ‘sent1 & sent2 → int1: [Intermediate node text]; int1 & sent3 → hypothesis;‘

- **Concise:** Do not copy sentence triples or hypothesis from the input.

- **Stop Condition:** The proof must **end immediately** once the ‘hypothesis‘ is reached, finishing with ‘;‘

### Generate the proof tree for the following grade 3-8 science problem following the above rules thinking step by step using the given question, answer, hypothesis, and *only the necessary* relevant triples and stop at the final proof step:

## B Qualitative Error Analysis

To better understand model behavior beyond aggregate metrics, we manually analyzed several examples from the EntailmentBank Task-1 test set. Each case compares the model-predicted proof to the gold proof and annotates the type of reasoning error (if any). These examples illustrate diverse error patterns, including hallucinated intermediate steps, irrelevant or misused facts, invalid logical compositions, and multiple forms of structural divergence (extra steps, missing steps, or reordered but semantically valid proofs). Such patterns illustrate both the strengths and limitations of current evaluation metrics, which often penalize structurally different yet semantically valid reasoning.

Error Type	Description / Example Case
<b>Hallucination</b>	Invented or unsupported intermediate steps (Case 1)
<b>Structural Divergence: Extra Step</b>	Over-elaboration through unnecessary intermediate nodes (Case 2)
<b>Structural Divergence: Proof Compression</b>	Collapsed multi-step reasoning into a single inference (Case 3)
<b>Structural Divergence: Reordered Proof</b>	Semantically valid reasoning expressed with a different ordering (Case 5)
<b>Fact Misuse / Irrelevant Step</b>	Incorporates irrelevant triples or unnecessary inference paths (Case 4)
<b>Logical Inference Error</b>	Incorrect logical linkage or misuse of hierarchical relationships (Case 6)

Table 5: Summary of qualitative error types observed in model predictions (examples follow).

---

### Case 1: Logical Inconsistency / Hallucination

---

**Question:** Why is the gravitational attraction of the Moon a greater factor in determining tides than the gravitational attraction of the much larger Sun?

**Hypothesis:** The gravitational pull of the moon on Earth affects the tides more than the gravitational pull of the sun.

**Relevant Facts:** sent1: the cause of something affects that something. sent2: the gravitational pull of the sun on Earth’s oceans causes the tides. sent3: the gravitational pull of the moon on Earth’s oceans causes the tides. sent4: as distance from an object decreases, gravitational pull increases. sent5: the moon is closer to Earth than the sun.

**Gold Proof:** sent4 & sent5 → int1: the gravity pull from the moon on earth is greater than the sun on earth; int1 & sent1 & sent2 & sent3 → hypothesis.

**Predicted Proof:** sent3 & sent4 & sent5 → int1: the gravitational pull of the moon on earth’s oceans causes the tides increases; sent2 & sent4 & sent5 → int2: the gravitational pull of the sun on earth’s oceans causes the tides decreases; int1 & int2 → int3: the gravitational pull of the moon on earth affects the tides more than the gravitational pull of the sun; int3 & sent1 → hypothesis.

**Issue:** Predicted steps contain unsupported comparative claims (“tides increase/decrease”), showing hallucinated intermediate reasoning.

---

Table 6: Example showing hallucinated intermediate steps.

---

### Case 2: Structural Divergence: Extra Step

---

**Question:** Tides on Earth are influenced more by the Moon than the Sun because the Moon is...

**Hypothesis:** The gravitational pull of the moon on Earth affects the tides more than the gravitational pull of the sun.

**Relevant Facts:** sent1: the moon is closer to Earth than the sun. sent2: gravitational pull of the moon on Earth’s oceans causes tides. sent3: gravitational pull of the sun on Earth’s oceans causes tides. sent4: gravitational pull increases as distance decreases.

**Gold Proof:** sent1 & sent4 → int1: the gravity pull from the moon on earth is greater than the sun on earth; int1 & sent2 & sent3 → hypothesis.

**Predicted Proof:** sent2 & sent4 → int1: as the distance from the moon decreases, the gravitational pull of the moon on earth will increase; int1 & sent1 → int2: the gravitational pull of the moon on earth is greater than the gravitational pull of the sun on earth; int2 & sent3 → hypothesis;

**Issue:** No logical error. The model introduces an extra intermediate step, producing a different but valid structure. Illustrates the need for structure-insensitive metrics.

---

Table 7: Structurally divergent through over-elaboration but semantically valid reasoning.

---

**Case 3: Structural Divergence: Proof Compression**

---

**Question:** An object composed mainly of ice is orbiting the Sun in an elliptical path. This object is most likely...?

**Hypothesis:** A comet is an object made of ice that orbits the Sun elliptically.

**Relevant Facts:** sent1: comets orbit the sun. sent2: a comet is often made of ice. sent3: comet orbits are elliptical.

**Gold Proof:** sent1 & sent3  $\rightarrow$  int1: comets orbit the sun in an elliptical path; int1 & sent2  $\rightarrow$  hypothesis.

**Predicted Proof:** sent2 & sent3 & sent1  $\rightarrow$  hypothesis.

**Issue:** Model collapses steps into one shallower inference. Logically correct. Demonstrates over-penalization by structural metrics.

---

Table 8: Model predicts a shorter but correct proof.

---

**Case 4: Irrelevant Step / Fact Misuse**

---

**Question:** One difference between the Moon and Earth is that the Moon...

**Hypothesis:** The moon revolves around a planet.

**Relevant Facts:** sent1: the sun is a kind of star. sent2: the moon orbits the earth. sent3: revolving around means orbiting. sent4: the earth revolves around the sun. sent5: earth is a planet.

**Gold Proof:** sent1 & sent4  $\rightarrow$  int1: the earth revolves around a star; sent2 & sent5  $\rightarrow$  int2: the moon orbits a planet; int2 & sent3  $\rightarrow$  int3: the moon revolves around a planet; int1 & int3  $\rightarrow$  hypothesis.

**Predicted Proof:** sent5 & sent4  $\rightarrow$  int1: the earth is a planet that revolves around the sun; sent1 & sent2  $\rightarrow$  int2: the moon orbits the earth; int2 & sent3  $\rightarrow$  int3: the moon revolves around the earth; int1 & int3  $\rightarrow$  hypothesis.

**Issue:** Model creates intermediate steps using loosely related facts (sent1 & sent2), introducing irrelevant logic. Shows difficulty filtering essential vs. peripheral information.

---

Table 9: Example of irrelevant or unnecessary inference.

---

**Case 5: Structural Divergence: Reordered Proof**

---

**Question:** Which best describes the Sun?

**Hypothesis:** The Sun is a yellow dwarf with medium size.

**Relevant Facts:** sent1: the sun is a kind of yellow dwarf. sent2: medium means average. sent3: the sun is average in size for a star in our galaxy.

**Gold Proof:** sent1 & sent3  $\rightarrow$  int1: the sun is a yellow dwarf with average size; int1 & sent2  $\rightarrow$  hypothesis.

**Predicted Proof:** sent2 & sent3  $\rightarrow$  int1: the sun is a yellow dwarf with average size; int1 & sent1  $\rightarrow$  hypothesis.

**Issue:** Both proofs are correct. Model simply uses a different but valid ordering. Penalized under strict structural matching.

---

Table 10: Example of semantically equivalent alternative reasoning path.

---

**Case 6: Logical Inference Error**

---

**Question:** Which category best describes the Sun?

**Hypothesis:** The sun is a yellow main-sequence star.

**Relevant Facts:** sent1: the sun is a kind of yellow dwarf. sent2: a yellow dwarf is a kind of main-sequence star. sent3: main-sequence stars are generally yellow.

**Gold Proof:** sent1 & sent2  $\rightarrow$  int1: the sun is a kind of main-sequence star; int1 & sent3  $\rightarrow$  hypothesis.

**Predicted Proof:** sent3 & sent2  $\rightarrow$  int1: main-sequence stars are yellow in color; int1 & sent1  $\rightarrow$  hypothesis.

**Issue:** Intermediate inference (sent3 & sent2  $\rightarrow$  int1) is not logically valid; correct chain requires specific category hierarchy (Yellow Dwarf  $\rightarrow$  Main-Sequence).

---

Table 11: Example of logically incorrect intermediate inference.