TEACHING TRANSFORMERS CAUSAL REASONING THROUGH AXIOMATIC TRAINING

Aniket Vashishtha[1]‡, Abhinav Kumar[3], Atharva Pandey[5], Abbavaram Gowtham Reddy[2]§, Kabir Ahuja[6], Vineeth N Balasubramanian[5]¶, Amit Sharma[5]
[1]UIUC, [2]CISPA Helmholtz Center for Information Security, Germany,
[3]MIT, [4]IIT Hyderabad, India, [5]Microsoft Research, India, [6]University of Washington
Corresponding author: amshar@microsoft.com

ABSTRACT

For text-based AI systems to interact in the real world, causal reasoning is an essential skill. Since active interventions are costly, we study to what extent a system can learn causal reasoning from symbolic demonstrations of causal axioms. Specifically, we present an axiomatic training method where the system learns from multiple demonstrations of a causal axiom (or rule), rather than incorporating the axiom as an inductive bias or inferring it from data values. A key question is whether the system would learn to generalize from the axiom demonstrations to more complex scenarios. Our results, based on applying axiomatic training to learn the transitivity axiom and d-separation rule, indicate that such generalization is possible. To avoid data contamination issues, we start with a 67 million parameter transformer model and train it from scratch. On both tasks, we find that a model trained on linear causal chains (along with some noisy variations) can generalize well to complex graphs, including longer causal chains, causal chains with reversed order, and graphs with branching. To handle diverse text inputs, the same method is extended to finetune language models. Finetuning Llama-3.1 8B model on our axiomatic data leads to significant gains on causal benchmarks such as Corr2Cause and CLEAR, in some cases providing state-of-the-art performance surpassing GPT-4.

1 INTRODUCTION

Causal reasoning can be defined as a set of reasoning procedures consistent with pre-defined axioms or rules that are specific to causality (Galles & Pearl, 1997). For instance, under stable causal models, the transitivity axiom ("if A causes B and B causes C, then A causes C") helps answer questions of cause and effect between pairs of variables in a system. Similarly, the d-separation rule connects independence of variables and their causal graph structure, and forms the basis of many graph discovery and effect identification algorithms. Given a causal task and data observations from a system, axioms or rules are typically incorporated as inductive biases in a machine learning (ML) algorithm, through regularization, model architecture, or the choice of variables for a particular analysis. Depending on the kind of available data—observational, interventional, or counterfactual—Pearl's ladder of causation (Bareinboim et al., 2022) defines the kinds of causal reasoning that is possible.

As axioms are the building blocks of causality, we study whether it is possible to directly learn the axioms or rules using ML models. That is, rather than learning from data that is the result of axioms obeyed by a data-generating process, what if a model can learn an axiom (and thus causal reasoning) directly from symbolic demonstrations of the axiom? This question gains relevance as language models make it possible to learn over symbolic data expressed in natural language. In fact, recent studies have evaluated causal reasoning capabilities of large language models (LLMs) by encoding

‡Work primarily done as a Research Fellow at MSR India, with additional contributions made at UIUC.
§Work primarily done at IIT-Hyderabad, with additional contributions made at CISPA.
¶Work primarily done at IIT Hyderabad, with additional contributions made at MSR India.

causal reasoning problems in natural language (Kıcıman et al., 2023; Jin et al., 2024a;b). Our goal is to study whether directly teaching the axioms can be a viable way to improve causal reasoning of language models.

Specifically, we propose a new way to learn causal reasoning through axiomatic training. We posit that causal axioms or rules can be expressed as the following symbolic tuple, $\langle$*premise*, *hypothesis*, *result*$\rangle$ where *hypothesis* refers to a causal claim and *premise* refers to any relevant information to decide whether the claim is true or not (*conclusion*). The conclusion could simply be "Yes" or "No". For example, consider the task of inferring causal relationships from correlational statements in the Corr2Cause dataset (Jin et al., 2024b), which we empirically study in this paper. The *premise* can be statements about statistical (in)dependence: *"Premise: T causes Eg. e causes T. e causes ID. e causes 2EN. ID causes T. ID causes Eg. ID causes 2EN"*; the hypothesis can be a question about cause-and-effect, *"Are 2EN and T d-separated given* ID, e*?"*; and the *conclusion* would be *"Yes"*. This tuple is a demonstration of the *d-separation* rule (Pearl, 2009b) (see Section 3 for definition). Based on this template, our key insight is that a large number of synthetic tuples can be generated, e.g., by changing the variable names, changing the number of variables, changing the order, and so on. The question is: if a model is trained on such data, would it learn to apply the axiom to new, more complex scenarios?

To answer this question, we consider a setup where a model is trained on axiomatic demonstrations over simple chain-like graphs of size 3-6 nodes and evaluated on more complex graphs, including longer chain-like graphs of size 7-15, graphs with branching, longer variable names, and edge direction perturbations. To avoid any contamination concerns with the pre-training data of an existing language model, we first train a transformer model from scratch. For both transitivity and d-separation, we find that a model trained on axiomatic demonstrations learns to apply the axiom multiple times to answer questions over more complex graphs. Diversity in the training data matters. For transitivity, a model trained only on simple directed chains generalizes to longer length chains, but is unable to generalize to graphs with branching or edge direction perturbations. In comparison, a model trained on a combined dataset of simple chains and chains with some edges randomly reversed, generalizes well across all kinds of evaluation scenarios. In particular, for d-separation, our 67 million parameter model outperforms billion-scale models such as GPT-4 under both zero-shot and multi-shot settings. Extending the findings on positional embedding for length generalization in NLP tasks (Kazemnejad et al., 2023; Bhattamishra et al., 2020; Haviv et al., 2022), we find that rotary position embedding works the best for causal generalization.

Next, we study whether the same axiomatic training dataset can also help to improve causal reasoning of pre-trained large language models. We fine-tune Llama-3.1-8B-Instruct model over axiomatic datasets for transitivity and d-separation. We evaluate on two benchmarks: CLEAR Chen et al. (2024b), that includes a test set for measuring d-separation capabilities; and Corr2Cause Jin et al. (2024b) on inferring causal structure from correlational statements. Note that our model is not trained on any of these datasets. We find significant gains due to axiomatic fine-tuning: in CLEAR, the accuracy on d-separation increases from 30 to 70 % on *Yes/No task*, and goes from 33 to 50% on *Multi-Choice Questions*. On Corr2Cause, we see a jump in performance of almost 20 % (after fine-tuning on transitivity axiomatic instances).

On Corr2Cause, the F1 score improved significantly, increasing by up to 20% compared to the baseline after finetuning on axiomatic instances and outperforms GPT-4 by 10%, thusb highlighting the impact of axiomatic training on more informal and complex real world datasets for inferring causal relationships.

Our work provides a new paradigm of teaching models causal reasoning through symbolic demonstrations of axioms or rules, which we call *axiomatic training*. Such symbolic data can be cheaply generated for multiple axioms and added to the finetuning data of language models. More generally, our results contribute to the literature on causal learning from *passive data* (Lampinen et al., 2023), showing a general way to learn causal reasoning with passive demonstrations.

2 Related Work

LLMs for Knowledge-Driven Causal Reasoning: Recent developments in Large Language Models (LLMs) have highlighted their potential for knowledge-driven causal discovery. Unlike traditional

methods which focus on statistical patterns or correlations, LLMs utilize knowledge acquired through their pretraining to reason about and identify causal structures based on metadata of variables (Kıcıman et al., 2023; Ban et al., 2023; Long et al., 2023; Willig et al., 2022; Vashishtha et al., 2023). However, possibility of memorization of existing benchmarks in the pretraining of these LLMs has been a major criticism. As a result, recent work (Zečević et al., 2023) argues that LLMs are not actually performing causal reasoning, but simply learning correlations about causal facts. In addition, there are critical failure modes of using LLMs for causal discovery due to hallucinations or not obeying the acyclic constraint when generating graph edges (Vashishtha et al., 2023). To evaluate causal reasoning capabilities of LLMs, (Jin et al., 2024b) and (Jin et al., 2024a) propose formal causal inference evaluation benchmarks to infer direct and indirect causal relationships, and highlight the failure of LLMs in performing accurate causal reasoning.

Impact of Positional Encoding on Generalization: Length generalization capabilities of transformers has been studied in the past to better understand their different failure modes across various settings (Hupkes et al., 2020; Zhang et al., 2023; Furrer et al., 2021). Previous work (Kazemnejad et al., 2023; Bhattamishra et al., 2020; Haviv et al., 2022; Shen et al., 2023) emphasizes the impact of positional encoding in length generalization capability of transformers. To understand how transformers can be optimized for learning through axiomatic training and generalizing to unseen larger causal structures, we also examine different types of positional encoding such as no positional encoding (PE), Learnable PEs (Radford et al., 2018) and Sinusoidal PEs (Vaswani et al., 2023).

Synthetic data generation for teaching transformers reasoning: Synthetic data generation has been explored for optimising model training for reasoning. For example, (Li et al., 2023; Gunasekar et al., 2023) use LLM-generated synthetic text for training Phi-1 and Phi-1.5 models and show impressive performance for reasoning-based tasks. (Trinh et al., 2024) introduce a novel neuro-symbolic framework to pre-train a transformer model for Olympiad-level math problems. Morishita et al. (2024) construct synthetic training data to improve language models' performance on logical reasoning tasks. Building on this stream of work, we apply synthetic data generation for teaching causal reasoning.

3 Preliminaries: Causal Axioms and Rules

Instead of performing causal reasoning using observational or interventional data, we study whether it is possible to learn general rules of causality directly from symbolic axioms. There has been fundamental work from Galles & Pearl (1997) where they axiomatize causal relevance (or equivalently irrelevance). They show that for a given *stable probabilistic* causal model (defined below), there exists a finite set of axioms that are completely characterized by axioms of path interception in corresponding directed graphs. Additionally, causal inference in practice depends on a few key rules, such as d-separation and do-calculus rules. Learning these rules can have a tangible impact on practical causal tasks such as graph discovery and effect inference. While we call the method axiomatic training, we consider learning both causal axioms and rules. Throughout this work, we assume no unobserved confounders.

Notation. We denote a random variable with an uppercase letter (e.g., X, Y, Z) and use lowercase letters (e.g., x, y, z) to denote the values taken by the corresponding random variable, written as $X = x, Y = y, Z = z$. We represent the probability of a random variable X_i as $\mathbb{P}(X_i)$. Let $\mathcal{G}(\mathbf{X}, \mathbf{E})$ be a directed acyclic graph (DAG) consisting of a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and a set of directed edges $\mathbf{E}$ among variables in $\mathbf{X}$. Let $pa(X_i) = \{X_k | X_k \to X_i\}$, $de(X_i) = \{X_k | X_k \leftarrow \cdots \leftarrow X_i\}$, $ch(X_i) = \{X_k | X_i \to X_k\}$ denote the set of *parents*, *descendants* and *children* of X_i respectively. Given two nodes X_i, X_j we call a third node X_k as a *collider* if both X_i and X_j are parents of X_k.

3.1 Axioms of Causality: Transitivity

Definition 3.1 (**Causal Irrelevance**, adapted from Defn. 7 in (Galles & Pearl, 1997))**.** X is probabilistically causally irrelevant to Y given Z, written ($X \nrightarrow Y | Z$) iff: $\mathbb{P}(y|z, do(X) = x) = \mathbb{P}(y|z, do(X) = x')\,, \forall x, x', y, z$ i.e., once we hold Z fixed at z, intervening on X will not change the probability of Y.

Next, we restate the stability assumption for a causal model from Galles & Pearl (1997) that gives a richer set of finite axiomatization for probabilistic causal irrelevance.

Assumption 3.2 (Stability, Definition 9 in Galles & Pearl (1997))**.** Let $\mathcal{M}$ be a causal model. Then an irrelevance $(X \nrightarrow Y|Z)$ in $\mathcal{M}$ is stable if it is shared by all possible probability distribution over $\mathcal{M}$. The causal model $\mathcal{M}$ is stable if all of the irrelevances in $\mathcal{M}$ are stable.

Under the stability assumption (see Assumption 3.2), Galles & Pearl (1997) states six axioms that completely characterize causal irrelevance (Definition 3.1) via axioms of path interception in the directed graphs. An axiom of causal irrelevance is of the form (given in conjunctive normal form):

$$\bigwedge_s \bigvee_t (\boldsymbol{X}_i^{s,t} \nrightarrow \boldsymbol{X}_j^{s,t} | \boldsymbol{X}_k^{s,t}) \implies \bigwedge_l \bigvee_n (\boldsymbol{X}_i^{l,n} \nrightarrow \boldsymbol{X}_j^{l,n} | \boldsymbol{X}_k^{l,n})$$

where $\wedge$ is "logical and", $\vee$ is "logical or" and for a given (s,t) or (l,n) pair, $\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{X}_k$ are disjoint subsets of observed variables $\boldsymbol{X}$. In the above causal irrelevance statement, if the antecedent is true, the consequent is also true.

Transitivity Axiom. We illustrate our axiomatic training procedure through the transitivity axiom. Following the stability assumption above, we consider the class of interventional distributions in which the transitivity causal irrelevance axiom holds (Sadeghi & Soo, 2024). Formally, for a stable probabilistic causal model (§3), given variables X, Y, Z in the system, the transitivity axiom is:

$$(X \nrightarrow Y|Z) \implies (A \nrightarrow Y|Z) \vee (X \nrightarrow A|Z) \forall A \notin X \cup Z \cup Y$$

which can be simplified using the contrapositive.

$$\exists A \notin X \cup Y \cup Z \;\; s.t. \underbrace{(X \to A|Z) \wedge (A \to Y|Z)}_{P:\text{premise}} \implies \underbrace{(X \to Y|Z)}_{H:hypothesis} \tag{1}$$

We call the LHS as *Premise* and the RHS as *Hypothesis*.

3.2 D-SEPARATION RULE

The d-separation rule connects causal graph structure with conditional independence in $\mathbb{P}(\boldsymbol{X})$.

Definition 3.3 (Definition 1.2.3 in Pearl (2009a))**.** Given a DAG $\mathcal{G}(\boldsymbol{X}, \boldsymbol{E})$, two sets of random variables $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are said to be d-separated by a third set $\boldsymbol{X}_z$ if all the *paths* between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in $\mathcal{G}$ are blocked by $\boldsymbol{X}_z$. A *path* p between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is said to be blocked by a set of nodes $\boldsymbol{X}_z$ iff 1) p contains a fork (i.e., $\cdot \leftarrow A \rightarrow \cdot$) or a chain (i.e., $\cdot \rightarrow A \rightarrow \cdot$) such that the middle node A is in $\boldsymbol{X}_z$, or 2) p contains a collider ($\cdot \rightarrow A \leftarrow \cdot$) such that the middle node A is not in $\boldsymbol{X}_z$ and no descendant of A is in $\boldsymbol{X}_z$.

Given $\mathbb{P}(\boldsymbol{X})$ is markov with respect to $\mathcal{G}$, if two sets of random variable $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are d-separated by $\boldsymbol{X}_z$, then they are conditionally independent of each other given $\boldsymbol{X}_z$.

4 AXIOMATIC TRAINING FOR TRANSFORMERS

Given an axiom, our key idea is to generate thousands of synthetic symbolic expressions that can be used to train a transformer on how to use the axiom. The trained model is then evaluated on whether it can apply these axioms to new causal structures that were not available in the training set. Below we describe how we generate the training data and the model architecture details.

4.1 TRAINING DATA: DIVERSITY IS KEY

As mentioned above, an axiom consists of a tuple, $\langle premise, hypothesis, conclusion \rangle$. Based on the specific axiom, we can map a hypothesis given the premise to its correct label ('*Yes*' or '*No*'). To create a training dataset, we randomly sample a causal DAG $\mathcal{G}$ and enumerate N random tuples of $\{(P, H, L)\}_N$ where P is the premise, H is the hypothesis and L is the label *(Yes/No)*. The premise describes the edges of the graph and is expressed in natural language, e.g., *"X causes Y. Z causes Y."*. Given a premise P based on the causal graph's edges, if the hypothesis can be derived by applying the specified axiom (once or multiple times), then label L is *Yes*; otherwise, *No*. For example, for the transitivity axiom, suppose the underlying true causal graph of a system is a chain, $X_1 \rightarrow X_2 \rightarrow X_3$. Then, the premise will be $X_1 \rightarrow X_2 \wedge X_2 \rightarrow X_3$. A corresponding hypothesis for the transitivity axiom could be $X_1 \rightarrow X_3$ will have label *Yes* whereas another hypothesis $X_3 \rightarrow X_1$ will have label *No*. The former would create a training data instance with the following text, *"$X1$ causes $X2$. $X2$*

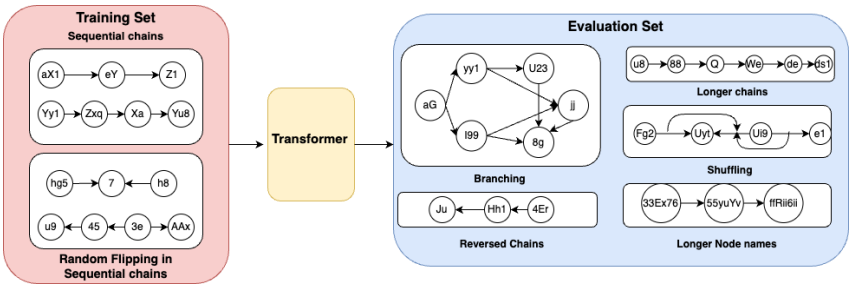

Figure 1: **Evaluating structural generalization of transformers through axiomatic training.** We train a transformer on two simple causal structures: chains and chains with random flipping of some edges. All training instances consist of 3-6 nodes. The trained model is evaluated on significantly more complex structures: bigger causal chains with >6 nodes, general branched networks with higher average in-degree and out-degree, complete reversals, longer sequences, shuffled natural language statements of sequences and longer node names.

causes $X3$*. Does* $X1$ *cause* $X3$*? Yes.*". Note that the axiom can be inductively applied multiple times to generate more complex training tuples. Another possible hypothesis for the *d-separation* rule could be "Are X_1 and X_2 d-separated given $\{X_3\}$?" and the label will be *No*.

We train the model on data from simple causal graphs such as sequential chains with 3-6 nodes and evaluate its performance on more complex graphs 1. To enhance generalization, we introduce structured perturbations in the training data across three axes: node names, causal structure types, and the number of nodes in the causal graph.

1. **Node names**: Each node in the graph is represented by an alphanumeric name comprising 1-3 characters. The length of a name and the specific characters are randomly selected during data generation.
2. **Causal Graph Topology**: We consider two main types of causal graphs in the training set.
   (a) **Sequential**: All causal edges are directed forward, thus forming a typical chain DAG, e.g. $X \rightarrow Y \rightarrow Z$.
   (b) **Random Flipping**: Given a chain of sequential nodes, we randomly reverse some edges eg. $X \rightarrow Y \leftarrow Z$. This can be expressed simply through natural language like: *"X causes Y. Z causes Y."*. This introduces forks and colliders that help add complexity to model training, thus aiding generalization across a larger space of graphs.
3. **Number of nodes in graph**: To facilitate the generalization of transformers over graphs of different sizes we incorporate chains of varying lengths, ranging from 3 to 6 nodes in our training set.

4.2 Tokenization, Training Loss & Architecture

We train the transformer model from scratch to ensure that the model has not seen such axioms in the pertaining step and thus requires a true correct understanding of axioms to perform well. Later we also tested on a pre-trained model fine-tuned on our dataset.

Tokenization. Since the training dataset follows a specific structure, we develop a custom tokenizer. Alphanumeric node names are tokenized at a character level, while special terms such as *'causes'*, *'Does'*, *'cause'*, *'Yes'*, and *'No'* are tokenized at the word level. Such an approach avoids out-of-vocabulary (OOV) tokens at test time since the alphanumeric node names in the test set can be different than those in the training set. Following this approach, the vocabulary size of our transformer model is 69.

Loss function. Given a dataset, the loss function is defined based on the ground truth label for each tuple, represented as $\mathop{\mathbb{E}}_{(P,H,L)\sim\mathbb{P}_{\text{train}}} -\log \mathbb{P}(L|P,H)$. A preliminary analysis indicated promising results with this loss formulation compared to next token prediction loss.

Positional Encoding. In addition to the training data and loss function, recent work (Kazemnejad et al., 2023) has shown that the choice of positional encoding is important for generalizing a transformer to longer or complex inputs. Therefore, we experiment with different positional encoding to understand their impact on generalization in causal tasks: learnable (LPE) (Radford et al., 2018), sinusoidal (SPE) (Vaswani et al., 2023), rotary (RoPE) position encodings (Su et al., 2023), and

no positional encoding (NoPE) (Kazemnejad et al., 2023; Haviv et al., 2022). See Appendix E for details.

Finetuning. Apart from training a transformer from scratch, we also fine-tune a pre-trained language model (Llama-3.1-8b-Instruct (gra, 2024)) on our axiomatic training data.

4.3 Evaluation setup: Assessing Axiomatic Learning in Transformers

We consider two types of evaluation: 1) on synthetic datasets where we directly test the models on axioms and, 2) on existing benchmarks corresponding to different high-level causal tasks where we expect the axioms to be helpful.

Synthetic evaluation. To evaluate if a trained model has learned the correct understanding of an axiom instead of shortcuts or correlation-based features, designing an out-of-distribution (OOD) evaluation set is important. We evaluate our model on multiple types of complex graphs that are unseen during training.

1. **Length**: Evaluating whether our model accurately infers causal relationships for chain graphs (both sequential and ones with random flipping) longer than those in the training set. Specifically, we train the model on chains with size 3-6 and evaluate on chains of size 7-15.
2. **Node Name Shift**: Testing the model's performance on longer node names, from 1-3 characters used in the training set to 8-10 characters. This is motivated by Jin et al. (2024b) who show how change in node names results in generalization failure on causal tasks for language models.
3. **Order of Chains**: a) **Completely reversed chains**: This evaluation is inspired by the reversal curse (Berglund et al., 2024) that revealed generalization failure of LLMs in answering questions in reversed sequences despite knowing the answers in the original order. We evaluate our model on completely reversed chains, a structure that was not in the training data. A completely reversed chain will be of the form $X \leftarrow Y \leftarrow Z$, written in natural language as: *"Y causes X. Z causes Y."*, where X, Y, Z are replaced by random alphanumeric names. b) **Shuffling of Sequences:** Here we shuffle the order of causal edges presented in each training row to add complexity and break sequential order.
4. **Branching Factor**: One of the most complex evaluation setups, with DAGs containing multiple branches, colliders, and forks. Let the branching factor be defined as the ratio between a number of edges and a number of nodes. Thus, the branching factor for the training set is ≤ 1. Then, we create a different evaluation set with multiple densely branched networks constructed using the Erdös-Rényi model, with different branching factors.

Benchmark evaluation. To test whether such simple axiomatic training is helpful in more complex scenarios, we evaluate our models on existing causal reasoning benchmarks, Corr2Cause (Jin et al., 2024b) and CLEAR (Chen et al., 2024b). Models trained from scratch on axiomatic instances with limited vocabulary and capability cannot be tested on diverse datasets (due to out-of-vocabulary issues). Therefore we use finetuned language models on these datasets.

5 Axiomatic Training For Transitivity Axiom

5.1 Training and Evaluation Datasets

In all our experiments, we consider an empty conditioning set Z for simplicity.

Training Datasets. The training data consists of sequential chains of lengths from [3,6]. In addition to sequential chains, random flipping of edges is done with 0.5 probability. See Appendix F for details on these hyperparameters. Our training data consists of 175k axiom demonstrations. We use three versions of training data to evaluate the impact of different data perturbations.

1. **Only Causal Chains (OCC)**: This set comprises of graphs with only sequential chains (see causal graph topology in Sec 4.1 for details).
2. **Training Setup 1 (TS1)**: This setup comprises of 73k examples where the underlying graphs has random flipping and 101k causal graphs where the underlying graphs has sequential chains. Since flipping is done randomly across all consecutive pairs of nodes in the given chain, some complete reversals are also formed. In this training set, around 12k completely reversed chains are present.

3. **Training Setup 2 (TS2)**: This setup comprises more simple sequential chains (132k), while we decrease chains with random flipping (42k) to keep the overall size around 175k to see the effect of adding examples with complicated graphs on model's performance.

Evaluation Datasets. In addition to the evaluation sets described earlier (length generalization, node name shift, order of chains, and branching), we add another evaluation set that is a combination of three shifts.

$\mathbf{MultiEval_{SLR}}$ **(Shuffling + Random Flipping + Length Sequence)**: This setup involves evaluation on 3 levels of complexities together: shuffling of sentence for representing the sequences, each sequence having random flipping, and some sequences having longer length than sequences in training set (upto 9).

5.2 Implementation Details: Architecture and Training Procedure

We train a decoder-based 67 million parameter model based on GPT-2's architecture. The model has 12 attention layers, 8 attention heads and 512 embedding dimensions. The model is trained from scratch on each of our training datasets. All models are trained for 100 epochs using the AdamW optimizer with 1e-4 learning rate. We use three variant of positional encoding when training the transformer: SPE: Sinusoidal Positional Encoding (PE), LPE: Learnable PE, RoPE: Rotary PE, NoPE: without any PE.

5.3 Baselines Using Existing LLMs

Given recent work on how LLMs can be leveraged for causal reasoning (Kıcıman et al., 2023; Vashishtha et al., 2023; Ban et al., 2023), we include language models such as GPT-4 *(gpt-4-32k)* ope (2024), Gemini *(gemini-pro)* gem (2024) and Phi-3 *(Phi-3-mini-128k-instruct)* abd (2024) as baselines. Note that each of these models is significantly larger than our model and known to perform well on reasoning tasks, with the smallest baseline model Phi-3 having 3.8 billion parameters (Li et al., 2023).

Zero Shot Setting To evaluate the baseline models, we follow a simple zero-shot prompting strategy. For each tuple, we provide the natural language expression of the causal graph *(Premise)* followed by the question *(Hypothesis)* and prompt the LM to answer it in either 'Yes' or 'No' *(Label)*. Here is an example prompt: *"EX causes T. T causes 9. 9 causes W. W causes 7. 7 causes M. M causes a. Does EX cause T? Answer in 'Yes' or 'No' only"*. See Table A1 contains examples of prompts used.

Multi Shot Setting (In-context-learning). Since these LLMs might not have seen such tasks before, we include some examples along with their true label in the prompt and then we add the evaluation example in the end. This ensures LLMs can do in-context learning and thus a fair comparison against our model that is trained explicitly on the axiomatic dataset. We present few-shot instances from our training set that includes sequential causal chains, along with a few examples with random flipping of edges. Appendix B.1 contains the multishot prompt used for querying baseline LLMs.

5.4 Results: Generalization to Complex Causal Scenarios

We train our model using axiomatic training on different kinds of datasets, TS1, TS2, and OCC, with different positional encoding (NoPE, LPE, and SPE) as described in Sec 5.1. Results on all evaluation settings are in Appendix Tables A3, A4 and A5.

$\mathbf{MultiEval_{SLR}}$ **Dataset.** We evaluate our models and other baselines on the challenging $\mathbf{MultiEval_{SLR}}$ dataset since it includes example that are different from training dataset in terms of size and complexity of causal graph. Table 1 summarizes the results for this dataset. While GPT-4 performs best, models trained with `RoPE` position encodings still achieve strong results, surpassing Gemini Pro and Phi-3 in both zero-shot and multi-shot settings for majority of node lengths.

A similar trend is seen for completely reversed sequences (Table A2). This task presents extreme out-of-distribution data, as the training data contains left-to-right edges, while the test data has only right-to-left edges. `TS2` (`NoPE`) consistently outperforms Gemini-Pro and Phi-3, and remains competitive with GPT-4 (zero shot). In particular, its accuracy (0.94 for chains of length 6) is substantially higher than Gemini Pro and Phi-3 (0.71 and 0.75 respectively).

Model/Nodes	3	4	5	6	7	8	9
Baselines (Zero Shot)							
GPT-4	0.99	0.97	0.89	0.85	**0.95**	**0.90**	0.90
Gemini Pro	0.75	0.73	0.72	0.76	0.71	0.68	0.74
Phi-3	0.88	0.86	0.82	0.79	0.76	0.73	0.79
Baselines (Multi Shot)							
GPT-4	**1.00**	**0.99**	**0.97**	**0.95**	0.94	**0.90**	**0.92**
Gemini Pro	0.95	0.85	0.83	0.79	0.79	0.73	0.75
Phi-3	0.88	0.83	0.82	0.80	0.83	0.76	0.78
Axiomatic Training							
TS1 w NoPE	**1.00**	0.93	0.85	0.83	0.78	0.73	0.73
TS1 w LPE	**1.00**	0.93	0.87	0.83	0.79	0.74	0.73
TS1 w SPE	0.99	0.92	0.85	0.81	0.76	0.74	0.61
TS1 w RoPE	**1.0**	0.93	0.87	0.85	0.81	0.78	0.76
TS2 w NoPE	0.99	0.93	0.86	0.82	0.79	0.74	0.73
TS2 w LPE	**1.00**	0.92	0.85	0.83	0.77	0.74	0.71
TS2 w SPE	0.99	0.94	0.86	0.81	0.76	0.72	0.64
TS2 w RoPE	**1.0**	0.95	0.89	0.86	0.82	0.79	0.76
OCC w RoPE	0.78	0.71	0.64	0.65	0.63	0.61	0.61

Table 1: **Evaluation on $\mathbf{MultiEval_{SLR}}$ dataset.** Accuracy of axiomatically trained models with another baseline on the most complicated setups. For OCC we only report performance with RoPE encodings, which is the best performing setup for this dataset. See Sec 5.4 for details. **Bold** numbers denote the highest value on a test set, while the underlined ones denote the second best.

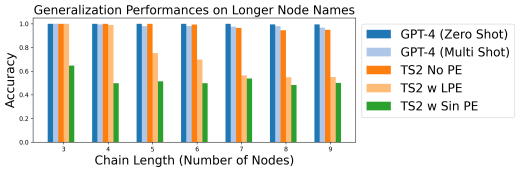

Figure 2: Evaluating generalization on causal sequences (without random flipping) with longer node names (than the ones used in sequences in train set). TS-2 training set with no positional encoding leads to the best performance. Refer table A4 for complete results.

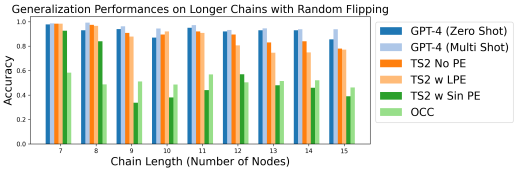

Figure 3: Generalizing to longer unseen causal sequences (>6 nodes) with random flipping on TS2 and OCC (with NoPE) train sets. OCC-trained models struggle due to limited edge-level variability, while TS2 NoPE consistently performs well. Refer table A3 for complete results

Branched Causal Graphs. Even though our models are trained on simpler graphs like sequential and random-flipping, we want to test our models on structurally harder graphs not considered in $\mathbf{MultiEval_{SLR}}$. To do so, we introduce general Erdos-Renyi graphs as the causal sequences while the training data contains only linear chains. We vary the branching factor of the graph as defined in Sec 4.3 between 1.4 and 2 for all our experiments on graphs with different numbers of nodes. Table A6 summarizes the results of this experiment. While GPT-4 achieves the highest accuracy as graph sizes increase, our `TS1 (RoPE)` and `TS2 (RoPE)` model outperforms Gemini Pro (branching factor 1.4) in for graphs with size 5 and 8 under zero-shot settings. On graphs with 12 nodes and a 1.4 branching factor, `TS2 (RoPE)` achieves 68% accuracy, far better than random (50%), despite training only on graphs with branching factors ≤ 1. Although LLMs excel in multi-shot settings, our model's performance is comparable even on more complicated causal graphs than the ones they were trained on.

Further Ablations. We run further ablations to understand the generalization behavior of our models. In particular, we experiment with generalization to unseen node names (Appendix Table A4), generalization to unseen lengths for graphs with linear chains and random flipping of edges (Appendix Table A3), and generalization to graphs with branching factor of 2 (A5).

6 AXIOMATIC TRAINING FOR D-SEPARATION RULE

Similar to the transitivity axiom, we now train our model on instances on of **d-separation** rule from multiple causal graphs and different premise, hypothesis pairs.

6.1 TRAINING DATASET AND SETUP

We follow a similar strategy to generate the training dataset as we did for the transitivity axioms. The training dataset consists of graphs with lengths from $[3, 6]$ with branching factor in range $[0.6, 0.8]$ and uses the same premises from the `TS2` training set as we did for transitivity axiom (see Sec 5.1). Given a premise, we create the hypotheses as follows: First, select all pairs of nodes in the graphs (x_1, x_2), then select the conditioning set $\mathcal{C}$ from the remaining sets of nodes of size up to 5 nodes at a time. The ground truth labels denote whether x_1 and x_2 are d-separated given conditioning set $\mathcal{C}$ for the given causal graph in the premise. From this exhaustive set of hypotheses, we randomly subsample 175k examples.

6.2 EVALUATION

Building on the success of axiomatic training for the transitivity axiom, we extend our approach to d-separation by introducing structurally more complex scenarios, such as branching and longer chains with random flipping. Unlike transitivity, which primarily involves reasoning over linear chains, d-separation is inherently more challenging due to its dependence on various structural patterns, including colliders, chains, and forks.

We evaluate the trained 67M model on two evaluation settings: longer sequences with random flipping A8 and branching A7, to cover a range of structural variations. Overall, model trained with RoPE emerges as the best performer, with NoPE based model following as close second. While GPT-4 struggles to perform better than random baselines in both settings, our models trained from scratch perform much better than random baseline, and the performance goes down as size increases.

7 EVALUATING ON COMPLEX TASKS WITH AXIOMATIC FINETUNING

We trained Llama-3.1-8B-Instruct using supervised fine-tuning on the same axiomatic training data, where the model takes causal graph premises and hypotheses (for transitivity or d-separation) as inputs and predicts 'Yes'/'No' as output labels. Refer I for more details.

7.1 EVALUATION ON CLEAR BENCHMARK

Chen et al. (2024a) introduced CLEAR, a comprehensive benchmark assessing LLMs' causal reasoning across 20 tasks, including backdoor adjustment, d-separation and others. It features diverse question types beyond Yes/No, allowing for evaluation on more complex tasks. We test our model, fine-tuned on axiomatic d-separation instances, in a zero-shot setting on CLEAR's Yes/No (YN) and Multi-Choice (MC) questions. Despite training on YN labels, our model outperformed GPT-4 on MC tasks. Fine-tuning on axiomatic instances of d-separation yielded a 20% improvement over the base model, surpassing GPT-4, while model showed improvement for YN task as well. Despite differences in semantic structure and wordings of finetuning and evaluation instances, our model, exhibited a significant performance gain over the base model. The substantial zero-shot gains across diverse tasks, semanmtic structure and question types, outperforming larger models like GPT-4, suggest that the model effectively applies d-separation rather than relying on spurious associations. Refer Table 2 for results.

7.2 EVALUATION ON CORR2CAUSE DATASET

Jin et al. (2024c) proposed a more complex dataset to evaluate models on different causal tasks. Each data instance in the benchmark includes correlational relationships described in natural language for graphs with 3 to 6 nodes; the goal is to infer the truth value of a hypothesis. The hypothesis consists of six different kinds of graphical relationships between pairs of variables: Parent, Ancestor, Child, Descendant, Collider, and Confounder. This task is significantly harder than applying a single axiom.

CLEAR D-Separation task (YN)	
Models	**Accuracy**
Llama-3-8b-Instruct	60.0
Llama-3-8b-Instruct Finetuned	70.0
GPT-4	63.33
CLEAR D-Separation task (MC)	
Models	**Accuracy**
Llama-3-8b-Instruct	33.0
Llama-3-8b-Instruct Finetuned	50.0
GPT-4	36.67

Table 2: **Evaluation on CLEAR dataset Chen et al. (2024b)** We finetune the LMs on our d-separation dataset and evaluate on the CLEAR dataset in a zero-shot setting. We observe a significant increase in performance compared to the baseline, which highlights the efficacy of axiomatic training. For discussion refer Sec 7.1.

Model	**F1**	**Prec**	**Rec**	**Acc**
Llama-3-8b-Instruct	0.18	0.14	0.23	0.67
Llama-3-8b-Instruct Finetuned Dsep (175k)	0.20	0.17	0.24	0.70
Llama-3-8b-Instruct finetuned transitivity (300k)	0.37	0.28	0.57	0.70
GPT-4 (from paper)	0.29	0.21	0.48	0.64

Table 3: **Evaluation on the Corr2Cause Task from Jin et al. (2024b).** We finetune the LMs on our transitivity and d-separation dataset and evaluate the Corr2Cause dataset in a zero-shot setting. We observe a significant increase in performance compared to the baseline, which highlights the efficacy of axiomatic training. For discussion refer Sec 7.2.

First, one needs to infer the causal graph from the correlation statements. This requires knowledge of d-separation statements and Markov condition Appendix D.1. Then, one needs to use the transitivity axiom to identify the direct effect and indirect effect to identify the children, ancestors, colliders, and confounders in the causal graph.

Comparison with Baselines: Our results highlight the Llama-3-8b-Instruct Base model's poor performance on this task, while axiomatic fine-tuning enables it to handle the complexity effectively. Fine-tuning on transitivity and d-separation improved performance, despite their simplicity. Notably, transitivity fine-tuning led to the largest gains, even surpassing GPT-4, likely due to its direct relevance in inferring graph relationships. This demonstrates the potential of our minimalist training setup to tackle complex, language-based causal reasoning tasks.

8 Discussion and Conclusion

In this paper, we provide a general framework, *axiomatic training*, to add axioms and simple rules of causality as inductive prior in the ML models, which can then further help in downstream causal discovery and causal inference tasks. We demonstrate their usefulness by training/fine-tuning models using our axiomatic training framework on two simple rules – transitivity and d-separation rules. We also discuss various modelling choices and how they effect the learning and generalization of causal axioms. We observe that a transformer model trained from scratch on a large axiomatic dataset can learn to apply axioms effectively. On causal tasks like graph traversal via transitivity and inferring causal relationships from correlation, small 67M transformers generalize well to unseen complex graphs, often outperforming models like GPT-4, Phi-3, and Gemini Pro. We then demonstrate the usefulness of adding these rules as inductive prior to downstream tasks on two downstream datasets – CLEAR and Corr2Cause. We observe that fine-tuning LLMs using axiomatic training help perform better on these dataset in the zero-shot setting, i.e., having never seen examples from these datasets. Fine-tuning on d-separation and transitivity-based axiomatic instances led to performance improvements. The model fine-tuned on transitivity exhibited the highest performance gain on templates like Parent, where distinguishing direct and indirect relationships is crucial. Similarly, fine-tuning on d-separation instances resulted in performance improvements on templates like idenrifying collider, where identifying colliders is key. Since the Child template contains no "No" labels, its reported F1 score is 0.

References

Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguis-

tics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Phi-3 technical report: A highly capable language model locally on your phone, 2024.

Gemini: A family of highly capable multimodal models, 2024.

The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Gpt-4 technical report, 2024.

Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data. *arXiv preprint arXiv:2306.16902*, 2023.

Elias Bareinboim, Juan Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference (1st edition). In Hector Geffner, Rita Dechter, and Joseph Halpern (eds.), *Probabilistic and Causal Inference: the Works of Judea Pearl*, pp. 507–556. ACM Books, 2022.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on "a is b" fail to learn "b is a", 2024.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages, 2020.

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. CLEAR: Can language models really understand causal graphs? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6247–6265, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.363. URL https://aclanthology.org/2024.findings-emnlp.363/.

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. Clear: Can language models really understand causal graphs?, 2024b. URL https://arxiv.org/abs/2406.16605.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 619–634, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.49. URL https://aclanthology.org/2021.emnlp-main.49.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures, 2021.

David Galles and Judea Pearl. Axioms of causal relevance. *Artificial Intelligence*, 97(1):9–43, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(97)00047-7. URL https://www.sciencedirect.com/science/article/pii/S0004370297000477. Relevance.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1382–1390, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.99. URL https://aclanthology.org/2022.findings-emnlp.99.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise?, 2020.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. Cladder: Assessing causal reasoning in language models, 2024a.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation?, 2024b.

Zhijing Jin, Jiarui Liu, Zhiheng LYU, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*, 2024c. URL https://openreview.net/forum?id=vqIH0ObdqL.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24892–24928. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4e85362c02172c0c6567ce593122d31c-Paper-Conference.pdf.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

Andrew Lampinen, Stephanie Chan, Ishita Dasgupta, Andrew Nam, and Jane Wang. Passive learning of active causal strategies in agents and language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1283–1297. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/045c87def0c02e3ad0d3d849766d7f1e-Paper-Conference.pdf.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023.

Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal discovery with language models as imperfect experts. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *arXiv preprint arXiv:2411.12498*, 2024.

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. Making transformers solve compositional tasks, 2022.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009a. ISBN 052189560X.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009b. ISBN 052189560X.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Kayvan Sadeghi and Terry Soo. Axiomatization of interventional probability distributions. *Biometrika*, pp. asae043, 08 2024. ISSN 1464-3510. doi: 10.1093/biomet/asae043. URL https://doi.org/10.1093/biomet/asae043.

Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. Positional description matters for transformers arithmetic, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024. doi: 10.1038/s41586-023-06747-5. URL https://doi.org/10.1038/s41586-023-06747-5.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. Causal inference using llm-guided discovery, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Moritz Willig, Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. Probing for correlations of causal facts: Large language models and causality. 2022.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. Causal parrots: Large language models may talk causality but are not causal, 2023.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task, 2023.

APPENDIX

Query Type (Train/ Eval)	Data Instance Example (Premise-Hypothesis-Label)	Structure Type	Network Size (number of nodes)
Train	Mhb causes iqB. iqB causes G. Does G cause iqB?: No	Short Linear Sequence	3-6
Train	N5w causes s. 6D causes s. Does N5w cause s?: Yes	Short Sequence with Random Flipping	3-6
Eval	w3 causes ROv. w3 causes tQC. H causes ROv. H causes tQC. b causes ROv. b causes w3. b causes H. Does tQC cause ROv?: No	Branching	5,8,10,12
Eval	LKk causes 5Ov. Kk causes L0. L0 causes KWO. 5Ov causes c. Does KWO cause L0?: No	Shuffled Sequences	3-9
Eval	FDAH26mV7 causes 7tzaIHjlY. 7tzaIHjlY causes 0kspcX95Im. 0kspcX95Im causes 7rhFSlx2o9. 7rhFSlx2o9 causes 1PlG5LHVqp. Does FDAH26mV7 cause 7tzaIHjlY?: Yes	Sequences with Longer Node Names	3-9
Eval	r causes rZ. rZ causes L. L causes bUx. bUx causes Pbr. Pbr causes 1w. 1w causes c3. c3 causes yBQ. yBQ causes yK. yK causes w. w causes P. P causes kH. kH causes 1u. 1u causes jV7. jV7 causes i. Does r cause rZ?: Yes	Long Linear Sequences	7-15
Eval	rU6 causes eF. eF causes ivC. 3R causes ivC. 3R causes A8. 2 causes A8. 2 causes i. i causes a03. y causes a03. b causes y. b causes h. h causes yN. ic0 causes yN. ic0 causes Hd. Hd causes U. Does rU6 cause eF?: Yes	Long Sequences with Random Flipping	7-15

Table A1: Table with examples of data instances of different causal structural networks used for training and evaluating models. Each instance is broken down into premise, hyopthesis, and label. During evaluation, only the premise followed by the corresponding hypothesis is provided, whereas during training of transformer, the model is trained on the loss of prediction of the label token.

A TRANSITIVITY AXIOMS

Length Generalization: Table A3 shows the accuracy of different models when evaluated on larger causal chains that were not seen during training. Among the baseline pre-trained LMs, GPT-4 obtains the highest accuracy on both sequential and randomly flipped chains for the multi-shot setting. It is remarkable that our `TS2 (NoPE)` model obtains competitive performance to the trillion-scale GPT-4 model. In particular, for chains of size 7-12, `TS2 (NoPE)` obtains higher or comparable accuracy

Model	3	4	5	6
Baselines				
Zero Shot				
GPT-4	0.97	<u>0.99</u>	<u>0.98</u>	0.92
Gemini Pro	0.61	0.59	0.66	0.62
Phi-3	0.80	0.69	0.73	0.69
Multi Shot				
GPT-4	**1.00**	**1.00**	**1.00**	**0.99**
Gemini Pro	0.95	0.87	0.77	0.71
Phi-3	0.93	0.89	0.75	0.75
Axiomatic Training				
TS1 w NoPE	0.99	0.97	0.90	0.91
TS1 w LPE	0.99	<u>0.98</u>	0.95	0.93
TS1 w SPE	**1.00**	<u>0.98</u>	0.95	0.96
TS1 w RoPE	0.97	0.97	0.96	<u>0.98</u>
TS2 w NoPE	0.98	0.96	0.90	0.91
TS2 w LPE	0.99	0.97	0.92	0.96
TS2 w SPE	0.99	0.97	0.93	0.94
TS2 w RoPE	<u>0.99</u>	<u>0.98</u>	0.97	<u>0.98</u>
OCC w NoPE	0.41	0.24	0.18	0.13
OCC w RoPE	0.59	0.26	0.22	0.20

Table A2: Following (Berglund et al., 2024), we evaluate models on inferring cause-and-effect from fully reversed sequences absent in training data. Models trained on OCC perform worse, highlighting the importance of edge-level perturbations for generalization. Accuracy metric is reported, with random baseline = 0.5. Best performance is bolded, while second best is underlined.

than GPT-4 on both sequential and randomly flipped chains. Similar trends are observed for chains of size 7-13 when compared to GPT-4 in the zero-shot setting. Our model's accuracy decreases for chains of length 14-15 (0.85 for sequential chains and 0.78 for randomly flipped chains) but is still significantly higher than that of LMs like Gemini-Pro and Phi-3. Although in-context examples in *multi-shot* setting improve the performance of baseline LLMs, `TS2` (`NoPE`) still outperforms both Gemini Pro and Phi-3 in the multi-shot setting. Note that a random prediction would yield a 50% accuracy, indicating that the axiomatically-trained `TS2` (`NoPE`) model can generalize its reasoning to causal chains much longer than 6 even though it was trained only on chains up to length 6.

Node Name Shift: For models trained on `TS2` dataset, we also evaluate generalization to changes in variable names (Figure 2). We find that `TS2` (`NoPE`) is robust to node name changes and retains its high accuracy as new, longer names are introduced. It also retains its generalizability to longer sequences with new node names, performing similarly to GPT-4.

Summary: Across all evaluation setups, our axiomatically trained model `TS2` (`NoPE`) performs significantly better than random baselines even as chain lengths are increased beyond its training data. In particular, even though our model was not trained on fully reversed chains, it performs at par with the significantly larger GPT-4 model (Fig. **??**), while easily outperforming other billion scale models even under multi-shot settings. For other tasks, it often outperforms or matches the accuracy of billion-scale models like Gemini Pro and Phi-3. These results indicate that a model trained axiomatically can learn to reason about more complex causal structures from demonstrations of simple causal sequences.

A.1 Additional Results: Role of Data Diversity and Positional Encoding

Importance of Data Perturbations. We find that diversity of the sequences in train data plays an important role. Model trained on only causal chains (OCC) generalize to longer chains (Table A3) but not to other DAG structures (see Figure 3 for edge flip, Table A6 for branching). Models trained on TS1 or TS2 generalize across all scenarios, including random flip, order permutations, and branching; thus highlighting the impact of incorporating variability at the edge level through random flipping. However, across tasks, we find that `TS2` yields higher accuracy than `TS1`, even as `TS1` has more variations due to random flipping. This suggests that while perturbations aid structural generalization, excessive perturbations can hinder it (in particular, random flipping may decrease the length of available causal paths during training).

Role of Positional Encodings. When comparing models based on positional encoding, we find that models without positional encoding generalize well to both longer chains (up to length 15) and unseen complex graph structures, despite being trained only on linear chains with 3-6 nodes. Models with SPE and LPE perform well on longer chains but struggle with longer node names, even in smaller graphs (Figure 2), highlighting their sensitivity to minor perturbations. SPE also underperforms in branching and order-based settings like shuffling and reversal. Learnable PE works up to 9-length chains but drops afterward. Overall, our results extend earlier work on the utility of `NoPE` (Kazemnejad et al., 2023; Haviv et al., 2022) to the task of understanding causal sequences and generalizing to both longer length and complex structure at test time. Interestingly, all PEs perform well in randomly flipped sequences, likely due to the short effective path lengths caused by the 0.5 probability of forward-directed edges.

B Multi-Shot Prompt

B.1 Cause-Effect Inference Task

Chain lengths of the in context examples ranged from 3 to 6 to maintains consistency with the training and testing paradigm used for our 67-million-parameter model.

The following multi-shot prompt was used to evaluate the baselines and models across different test sets, assessing their generalization based on length, order, and branching.

Following the given examples answer the question regarding causal relationship between two variables: '5e0 causes vAf. vAf causes VO. Does vAf cause VO?: Yes'
'5e0 causes vAf. vAf causes VO. Does vAf cause 5e0?: No'

Model	7		8		9		10		11		12		13		14		15	
	FS	RF	FS	RF	FS	RF	FS	RF	FS	RF	FS	RF	FS	RF	FS	RF	FS	RF
Baselines																		
Single Shot																		
GPT-4	0.95	0.98	0.97	0.93	0.87	0.94	0.91	0.87	**0.90**	0.95	0.92	0.92	0.85	0.93	**0.93**	0.93	**0.89**	0.86
Gem-Pro	0.63	0.73	0.69	0.74	0.64	0.75	0.65	0.81	0.72	0.78	0.60	0.80	0.59	0.68	0.67	0.64	0.61	0.66
Phi-3	0.81	0.85	0.96	0.85	0.85	0.85	0.87	0.89	**0.90**	0.86	0.84	0.85	0.91	0.84	0.90	0.80	0.78	0.85
Multi Shot																		
GPT-4	0.97	**0.99**	0.93	**0.99**	**0.92**	**0.96**	0.88	**0.94**	0.89	**0.97**	0.89	**0.93**	0.88	**0.95**	**0.93**	**0.94**	0.86	**0.94**
Gem-Pro	0.80	0.82	0.81	0.79	0.78	0.81	0.67	0.79	0.73	0.82	0.74	0.83	0.67	0.78	0.72	0.78	0.68	0.78
Phi-3	0.83	0.92	0.89	0.88	0.75	0.86	0.66	0.87	0.80	0.90	0.80	0.85	0.79	0.82	0.71	0.81	0.72	0.82
Axiomatic Training																		
TS1 w NoPE	0.99	0.96	0.97	0.95	0.86	0.92	0.78	0.87	0.77	0.90	0.76	0.82	0.77	0.82	0.75	0.83	0.70	0.76
TS1 w LPE	0.98	0.96	0.89	0.94	0.81	0.91	0.61	0.86	0.64	0.87	0.64	0.79	0.60	0.80	0.59	0.81	0.57	0.73
TS1 w SPE	0.99	0.91	0.88	0.92	0.73	0.77	0.62	0.69	0.63	0.65	0.69	0.60	0.62	0.62	0.59	0.58	0.63	0.58
TS1 w RoPE	0.99	0.96	0.97	0.95	0.89	0.90	0.82	0.84	0.81	0.84	0.86	0.76	0.76	0.81	0.82	0.70	0.78	0.75
TS2 w NoPE	0.98	0.93	0.93	0.92	0.82	0.88	0.74	0.84	0.70	0.85	0.70	0.80	0.71	0.76	0.71	0.77	0.66	0.73
TS2 w LPE	0.99	0.95	0.96	0.94	0.86	0.90	0.72	0.86	0.69	0.85	0.80	0.78	0.73	0.78	0.75	0.80	0.68	0.77
TS2 w SPE	0.97	0.92	0.91	0.92	0.76	0.85	0.58	0.72	0.60	0.66	0.61	0.56	0.60	0.56	0.58	0.56	0.56	0.59
TS2 w RoPE	0.99	0.97	0.98	0.96	0.90	0.89	0.85	0.87	0.84	0.82	0.87	0.74	0.78	0.80	0.86	0.69	0.78	0.71
OCC w NoPE	0.99	0.61	0.98	0.62	0.89	0.62	0.90	0.57	0.90	0.57	0.93	0.52	0.87	0.55	0.93	0.50	0.87	0.53
OCC w RoPE	0.96	0.65	0.98	0.71	0.84	0.68	0.84	0.64	0.80	0.65	0.88	0.56	0.76	0.60	0.84	0.60	0.79	0.55

Table A3: **Accuracy of different models on Transitivity axioms.** In this table, we show the accuracy of different models on the transitivity axioms. The rows shows different models considered for comparison. The top rows denote the performance of baseline models with different prompting strategies i.e. single shot and multi-shot prompt (see Sec 5.3 for details). The models listed after axiomatic training shows the performance of transformer models trained from scratch on our axiomatic dataset. TS1 and TS2 denote pretraining data setups 1 and 2 as described in Sec 5.1 and different modifiers are: SPE: Sinusoidal Positional Encoding (PE), LPE: Learnable PE, w/o PE: No PE, RoPE: Rotary Position Embedding. For axiomatic training, the model remains the same across all setups (67 Million parameters based). The training dataset contain graphs of size 3-6 however the models are tested on graphs of size 7-15 (as mentioned in different columns). FS denotes the graphs that only contain chains that are oriented in forward direction and RF contains the graphs that also includes random flipping (see Sec 4.1 for details) same as training set.

'e0F causes Z. Z causes 0U. 0U causes mR. mR causes 1L. Does mR cause 1L?: Yes'
'e0F causes Z. Z causes 0U. 0U causes mR. mR causes 1L. Does Z cause e0F?: No'
'b causes K. K causes qPv. 5 causes qPv. Does b cause qPv?: Yes'
'b causes K. K causes qPv. 5 causes qPv. Does b cause 5?: No'
'Mhb causes t0a. 6Eh causes Mhb. NS causes 6Eh. n causes NS. n causes xu. Does xu cause 6Eh?: No'
'Mhb causes t0a. 6Eh causes Mhb. NS causes 6Eh. n causes NS. n causes xu. Does n cause NS?: Yes'

C Results and Analysis

D Preliminaries and Notations

Causal Models Let $\mathcal{M} = (\boldsymbol{X}, \boldsymbol{U}, \mathcal{F})$ be a causal model defined over a set of endogenous variables $\boldsymbol{X}$, exogenous variables $\boldsymbol{U}$ and the causal relationship between then defined by set of structural equations $\mathcal{F}$ (Galles & Pearl, 1997). Let $\mathcal{G}$ be the causal graph associated with the causal model $\mathcal{M}$ where the nodes $\boldsymbol{V}$ in $\mathcal{G}$ correspond to the variables in $\mathcal{M}$ and an edge $V_i \rightarrow V_j$ between any two nodes V_i, V_j denote the causal relationship between them. The causal relationship of node X_i is characterized by the functional relationship $f_i \in \mathcal{F}$ s.t., $x_i = f_i(\boldsymbol{pa}_i, \boldsymbol{u}_i)$. Here $\boldsymbol{pa}_i$ are the parent of the node X_i is the corresponding causal graph $\mathcal{G}$ and $\boldsymbol{u_i} \subseteq \boldsymbol{U}$ are set of exogenous variables influencing the exogenous variable X_i. In our work, we assume that there are no hidden confounders so we have one exogenous variable corresponding to every endogenous variable i.e. $\boldsymbol{u}_i = u_i$. Each exogenous variable has an associated probability distribution which quantifies the uncertainty in the system i.e. $u_i \sim \mathbb{P}(u_i)$. Thus the joint distribution of the exogenous variable is given by $\mathbb{P}(\boldsymbol{U})$. Since any endogenous variable is a deterministic function of other endogenous and exogenous variables

Model	3	4	5	6	7	8	9
Baselines							
Single Shot							
GPT-4	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Gemini Pro	0.96	0.94	0.86	0.81	0.76	0.73	0.71
Phi-3	0.99	0.98	0.95	0.94	0.96	0.95	0.93
Multi Shot							
GPT-4	**1.00**	**1.00**	0.98	0.98	0.98	0.98	0.97
Gemini Pro	**1.00**	**1.00**	0.91	0.90	0.86	0.88	0.84
Phi-3	0.93	0.89	0.89	0.84	0.82	0.77	0.79
Axiomatic Training							
TS1 w NoPE	**1.00**	**1.00**	**1.00**	0.99	0.98	0.92	0.88
TS1 w LPE	**1.00**	**1.00**	0.99	0.97	0.92	0.83	0.74
TS1 w SPE	0.76	0.61	0.58	0.57	0.54	0.50	0.54
TS1 w RoPE	0.65	0.56	0.56	0.49	0.45	0.49	0.50
TS2 w NoPE	**1.00**	0.99	0.92	0.84	0.76	0.71	0.69
TS2 w LPE	**1.00**	0.99	0.96	0.90	0.86	0.76	0.74
TS2 w SPE	0.82	0.66	0.60	0.58	0.57	0.55	0.53
TS2 w RoPE	0.51	0.48	0.48	0.50	0.46	0.46	0.48
OCC w NoPE	**1.00**	0.99	0.98	0.96	0.96	0.91	0.93
OCC w RoPE	0.90	0.77	0.67	0.64	0.65	0.59	0.62

Table A4: Results on node name length generalization. TS1 and TS2 denote Training Data setup 1 and 2 from Section 4. OCC is the third data setup comprising of sequential causal chains. SPE: Sinusoidal PE, LPE: Learnable PE, w/o PE: No PE, RoPE: Rotary Position Embedding. Model remains the same across all setups (67 Million parameter based). For longer node names, NoPE performs best on sequential linear setup. Accuracy metric is used.

Model	5		8		10		12	
	BF=2	BF=1.4	BF=2	BF=1.4	BF=2	BF=1.4	BF=2	BF=1.4
Baselines								
Zero shot								
GPT-4	0.98	0.95	0.91	0.90	0.84	0.88	0.82	0.86
Gemini Pro	0.77	0.74	0.72	0.76	0.71	0.73	0.73	0.71
Phi-3	0.87	0.83	0.82	0.79	0.77	0.77	0.75	0.80
Multi shot								
GPT-4	**0.99**	**0.97**	**0.94**	**0.93**	**0.90**	**0.94**	**0.89**	**0.93**
Gemini Pro	0.81	0.76	0.77	0.79	0.75	0.77	0.78	0.79
Phi-3	0.77	0.78	0.79	0.82	0.78	0.79	0.80	0.79
Axiomatic Training								
OCC w RoPE	0.74	0.72	0.70	0.68	0.66	0.66	0.65	0.62
TS1 w LPE	0.76	0.82	0.70	0.72	0.67	0.69	0.63	0.68
TS1 w SPE	0.65	0.78	0.57	0.61	0.55	0.59	0.53	0.57
TS1 w NoPE	0.78	0.82	0.70	0.74	0.67	0.69	0.62	0.66
TS1 w RoPE	0.81	0.86	0.75	0.79	0.73	0.74	0.69	0.71
TS2 w LPE	0.73	0.80	0.68	0.72	0.65	0.67	0.61	0.64
TS2 w SPE	0.65	0.79	0.53	0.59	0.52	0.54	0.52	0.52
TS2 w NoPE	0.75	0.82	0.68	0.73	0.67	0.68	0.62	0.64
TS2 w RoPE	0.81	0.88	0.74	0.79	0.70	0.72	0.68	0.68

Table A5: **Evaluation with variation in branching factor.** Accuracy of axiomatically trained models with LM baselines on the causal graphs with higher branching factor than that in training. See Sec 5.4 for details.

the probability distribution corresponding to the endogenous variable is the push-forward of the exogenous variable i.e $\mathbb{P}(\boldsymbol{X}) \triangleq \mathbb{P}^{\#}(\boldsymbol{U})$.

Model/Nodes	5	8	10	12
Baselines (Zero shot)				
GPT-4	0.95	0.90	0.88	0.86
Gemini Pro	0.74	0.76	0.73	0.71
Phi-3	0.83	0.79	0.77	0.80
Baselines (Multi shot)				
GPT-4	**0.97**	**0.93**	**0.94**	**0.93**
Gemini Pro	0.76	0.79	0.77	0.79
Phi-3	0.78	0.82	0.79	0.79
Axiomatic Training				
OCC w RoPE	0.72	0.68	0.66	0.62
TS1 w LPE	0.82	0.72	0.69	0.68
TS1 w SPE	0.78	0.61	0.59	0.57
TS1 w NoPE	0.82	0.74	0.69	0.66
TS1 w RoPE	0.86	0.79	0.74	0.71
TS2 w LPE	0.80	0.72	0.67	0.64
TS2 w SPE	0.79	0.59	0.54	0.52
TS2 w NoPE	0.82	0.73	0.68	0.64
TS2 w RoPE	0.88	0.79	0.72	0.68

Table A6: **Evaluation with branching factor 1.4.** Accuracy of axiomatically trained models with LM baselines on the causal graphs with branching factor 1.4. See Sec 5.4 for details.

D.1 Definitions

Following the formal definitions provided by (Jin et al., 2024b), we explain the following terminologies:

Markov Property In a directed acyclic graph (DAG) G, the Markov property asserts that each node X_i is conditionally independent of its non-descendants given its parents. This can be written as $X_i \perp\!\!\!\perp \text{NonDe}(X_i) \,|\, \text{Pa}(X_i)$, where $\text{NonDe}(X_i)$ represents the set of non-descendants of X_i, excluding the node itself, and $\text{Pa}(X_i)$ denotes its parents. Leveraging the Markov property, the joint distribution over all the nodes can be factorized as:

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i \,|\, \text{Pa}(X_i)).$$

Markov Equivalence Class Two directed acyclic graphs (DAGs) are considered Markov equivalent if they induce the same joint distribution $P(X)$. The collection of DAGs that are Markov equivalent is referred to as a Markov equivalence class (MEC). Causal graphs within the same MEC can be easily recognized as they share the same skeleton (i.e., undirected edges) and V-structures (i.e., configurations of the form $A \rightarrow B \leftarrow C$, where A and C are not directly connected).

E Positional Encodings and their Role in Generalization

Positional Encoding (PE) play a crucial role of providing information about the absolute and relative position of tokens in a sequence (Vaswani et al., 2023). (Vaswani et al., 2023) propose an absolute positional encoding strategy using periodic functions (e.g., sinusoidal or cosine) to initialize these encodings. Absolute positional encoding provides definite values for all positions across any sequence length. However, studies (Ontañón et al., 2022; Csordás et al., 2021) show absolute positional encoding fails in length generalization tasks for transformers. In the learnable APE variant (Radford et al., 2018), each positional embedding is randomly initialized and trained with the model. This approach falters with sequences longer than those seen in training, as the new positional embeddings remain untrained and randomized. Interestingly, recent findings (Kazemnejad et al., 2023; Haviv

et al., 2022) indicate that removal of PEs in auto-regressive models can improve model's length generalization capabilities, wherein the attention mechanism during auto-regressive decoding is sufficient to encode positional information. We also experiment with Rotary Position Encodings Su et al. (2023), which have shown superior length generalization. We use $\theta = 10000.0$ for the base period of RoPE embeddings.

F Formalising training and Evaluation Setup

Let f_{dim} represent the maximum value for a given perturbation dimension dim, along which we construct train and evaluation sets for our axiomatic framework. For each dimension, we choose a threshold $\tau_{dim} \in L$, such that $f_{dim} < \tau_{dim}$ forms our training set and $f_{dim} \geq \tau_{dim}$ forms the evaluation set. So, $f_{dim} \in \{f_{len},\ f_{branch},\ f_{nodelen},\ f_{revfactor},\ f_{shuffle}\}$ where:

- $f_{len} = \max_{\forall i}(len(V_i))$, gives the maximum number of nodes across all causal sequences. τ_{len} for length is set at 6, with $f_{len} \in [3, 6]$.
- $f_{branch} = \max_{\forall i}(|X_i|/|V_i|)$ gives the maximum branching factor in a dataset, with $\tau_{branch} = 0.8$ (for 6 node linear sequences). For sequences in the train set, the branching factor ranges from 0.6 to 0.8 for 3 to 6 length sequences.
- Let $l_{i,j}$ be the length of the name of the node $X_{i,j}$, then $l_{i,j} = (len(X_{i,j}))$.Therefore, the maximum length of node names across all nodes in all causal sequences can be represented as: $f_{nodenamelen} = \max_{1\leq i\leq n,\, 1\leq j\leq m} l_{i,j}$. We set $\tau_{nodelen}$ for train set as 3, with $f_{nodelen} \in [1, 3]$.
- Given any causal sequence X_i and a function N, where $N(X_{i,j}, X_{i,j+1})$ returns natural language representation of a directed edge between j and $j + 1$ node in the causal chain X_i. $f_{shuffle} = \cap_{\forall i,j}\text{Perm}(N(X_{i,j}, X_{i,j+1}))$, where $N(X_{i,j}, X_{i,j+1})$ represents deviation from original sequential order of natural language sentences to represent X_i.
- Given a causal sequence X_i and let $R(X_i, f_{revfactor})$ be an operation on the causal chain that flips the direction of every edge in the sequence with probability $f_{revfactor}$. In the training set, there is a directed edge between every sequential pair of nodes $X_{i,j}, X_{i,j+1}$ with $f_{revfactor} = 0$ (for linear sequence, $X_{i,j} \rightarrow X_{i,j+1}$) or 0.5 (for sequence with random flipping, $X_{i,j} \rightarrow X_{i,j+1}$ or $X_{i,j} \leftarrow X_{i,j+1}$) In the evaluation set $f_{revfactor} = 1$ i.e., all sequences for reversal evaluation setup are completely reversed unlike in train set where no sequence is present where all edges are completely reversed.

G Results of Dsep on CLEAR dataset

Premise: *Given a DAG (directed acyclic graph) with nodes C, Z, P, V and directed edges C->V, P->V, C->Z, Z->P, Z->V.*
Hypothesis: "Which of the following nodesets can d-separate node C and node P?
A. `{'Z', 'V'}`
B. `{'V'}`
C. `{'Z'}`
D. `set()`
Answer: *C*

Figure A1: Example instance of Multiple Choice (MC) question type from Chen et al. (2024b) dataset describing d-separation rule problem defined with a different hypothesis type and semantic structure then the one our models are finetuned on.

H Results on Corr2Cause dataset

I Implementation Details

We used a learning rate of 1e-4 with linear scheduling and 3% warmup ratio, training for 4102 max steps on axiomatic instance samples with sequences of maximum length 4096 tokens. We employed mixed precision (bfloat16) training with flash attention for efficiency. After training, the LoRA weights were merged with the base model for inference. We used Huggingface wol (2020) for implementation. The fine-tuning used LoRA with rank 64, alpha 16, and dropout 0.1. Training was

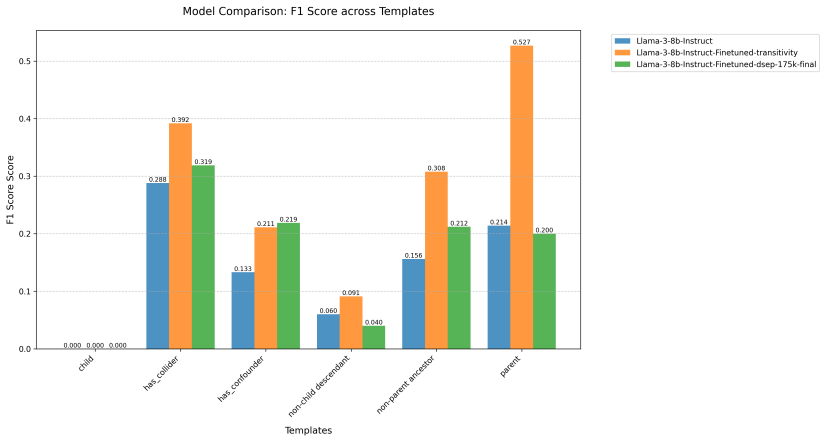

Figure A2: Model Comparison: F1 Score across Templates, finetuning on dsep and transitivity based axiomatic instances lead to performance improvement. Model finetuned on transitivity sees the highest jump for templates like **Parent**, where identifying direct-indirect relations is important. Finetuning on D-separation instances see a jump in performances for templates like `has_collider`, where identifying a collider is important. Since child template has 0 'No' labels, the F1 score reported is 0.

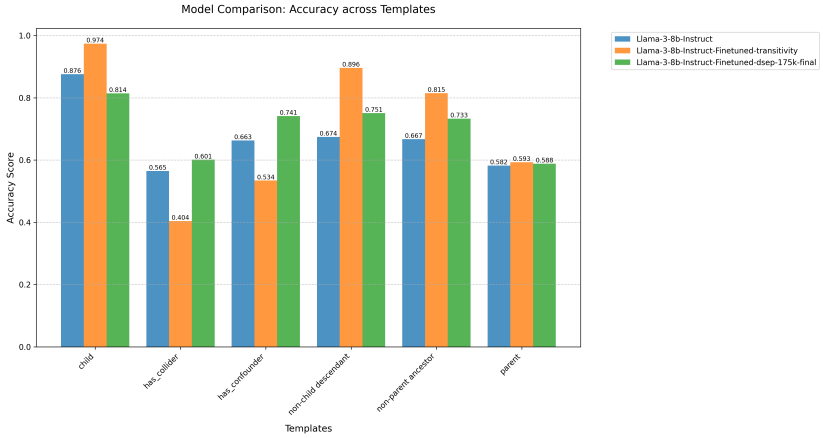

Figure A3: Model Comparison: Accuracy Score across Templates

Figure A4: Comparison of Model Performance across Different Relationship Templates. Accuracy plots show consistent trends, with model finetuned on transitivity consistently outperforming base models, on templates where direct-indirect relationship identification is required. Finetuning on D-separation instances see a performance jump over the base instruct model, for templates identifying colliders and confounders.

Branching (Bfactor = 1.4)				
Models	**5**	**8**	**10**	**12**
Baselines				
GPT-4	0.53	0.544	0.62	0.52
Finetuned Results				
Llama-3-8b-Instruct	0.474	0.490	0.470	0.482
Llama-3-8b-Instruct-Finetuned	**0.796**	**0.738**	**0.718**	**0.670**
Models with different PEs trained from scratch				
SPE	0.67	0.59	0.56	0.55
LPE	0.67	0.61	0.57	0.56
NoPE	0.63	0.58	0.53	0.53
RoPE	0.70	0.58	0.54	0.52

Table A7: We evaluate the effectiveness of axiomatic training for d-separation under two training paradigms: training a model from scratch and fine-tuning a pretrained Llama model. The training setup consists of linear sequential causal chains, along with some variations where edges are randomly flipped. However, we assess model performance on significantly more complex causal graphs featuring branching structures and additional nodes, similar to our transitivity analysis. The base Llama Instruct model (prior to fine-tuning) performs on par with random baselines. In contrast, the fine-tuned model demonstrates a substantial improvement, particularly on branched networks. Unlike our transitivity analysis—where GPT-4 significantly outperformed all other models—GPT-4 struggles in this setting, performing no better than the random baseline. While multi-shot prompting led to consistent and significant improvements in transitivity experiments, it fails to enhance GPT-4's performance on d-separation, even when using the multi-shot prompt described in Section **??**. Additionally, our experiments with a decoder-based model trained from scratch show superior performance compared to baselines.

Models	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
GPT-4	0.57	0.64	0.52	0.50	0.53	0.52	0.51	0.50
Llama-3-8b-Instruct	0.51	0.50	0.54	0.51	0.48	0.51	0.53	0.51
Llama-3-8b-Instruct-Finetuned	0.952	0.948	0.954	0.850	0.87	0.88	0.73	0.66
Models with different PEs trained from scratch								
SPE	0.93	0.95	0.97	0.95	0.71	0.61	0.80	0.44
LPE	0.96	0.93	0.99	0.92	0.68	0.71	0.62	0.47
NoPE	0.89	0.93	0.85	0.94	0.68	0.65	0.60	0.5
RoPE	0.96	0.91	0.96	0.92	0.70	0.69	0.54	0.58

Table A8: Performance on DSEP of longer chains with random flipping

performed on 3 GPUs using DeepSpeed Stage 3 with a total batch size of 128 (16 samples per GPU with gradient accumulation).