CausalGraph2LLM: Evaluating LLMs for Causal Queries

Anonymous ACL submission

Abstract

Causality is essential in scientific research, enabling researchers to interpret true relationships between variables. These causal relationships are often represented by causal graphs, which are directed acyclic graphs. With the recent advancements in Large Language Models (LLMs), there is an increasing interest in exploring their capabilities in causal reasoning and their potential use to hypothesize causal graphs. These tasks necessitate the LLMs to encode the causal graph effectively for subsequent downstream tasks. In this paper, we propose a comprehensive benchmark, *CausalGraph2LLM*, encompassing a variety of causal graph settings to assess the causal graph understanding capability of LLMs. We categorize the causal queries into two types: graph-level and node-level queries. We benchmark both open-sourced and closed models for our study. Our findings reveal that while LLMs show promise in this domain, they are highly sensitive to the encoding used. Even capable models like GPT-4 and Gemini-1.5 exhibit sensitivity to encoding, with deviations of about 60%. We further demonstrate this sensitivity for downstream causal intervention tasks. Moreover, we observe that LLMs can often display biases when presented with contextual information about a causal graph, potentially stemming from their parametric memory.

1 Introduction

The recent success of Large Language Models (LLMs) [Brown et al., 2020, Achiam et al., 2023, Reid et al., 2024] across various applications has opened up new avenues for their use beyond standard Natural Language Processing (NLP) tasks [Srivastava et al., 2022, Wei et al., 2022]. Trained on massive corpora of structured and unstructured data [Achiam et al., 2023], these models have demonstrated the ability to extract insights and exhibit emergent behaviors that can be harnessed for a wide range of applications [Bubeck et al., 2023, Qi et al., 2023, Wang et al., 2023, Zhao et al., 2024].

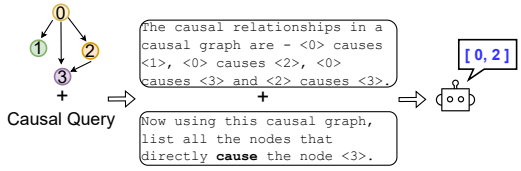


Figure 1: Causal Graphs are ingested into LLMs via prompt encoding strategies which are evaluated for Causal Queries.

Causal reasoning plays a pivotal role in guiding scientific research to establish causal relationships between different variables of an environment [Pearl, 2009]. These relationships are often represented and modeled using causal graphs, which are directed and acyclic graphs. Traditionally, causal inference and discovery have been largely driven by observational data obtained through experiments [Spirtes and Zhang, 2016, Nogueira et al., 2022, Huang et al., 2020, Cooper and Yoo, 2013]. However, inferring causal graphs from observational data alone is a challenging problem [Spirtes and Zhang, 2016, Brouillard et al., 2020]. This bottleneck of causal discovery has led to an increasing interest in the potential of LLMs to assist in this process [Vashishtha et al., 2023, Abdulaal et al., 2024, Liu et al., 2024, Ban et al., 2023b,a]. Therefore, the current paradigm for employing LLMs in causal discovery typically involves the use of metadata, particularly in the form of variable names to guide the models in identifying and interpreting causal relationships. Existing works utilize LLMs in various roles such as priors, critics, and post-processors for causality-related tasks.

While LLMs have demonstrated competitive performance [Abdulaal et al., 2024] against traditional data-driven methods, their effectiveness is constrained by their sequential text-based training paradigm. Current models typically necessitate

a user to decompose their causal reasoning task into first textualizing a causal graph and then the task prompt. In essence, for downstream tasks, the LLMs need to be able to handle and manipulate textual representations of causal graphs effectively. This capability of LLMs being able to process causal graphs as text efficiently with any encoding is often assumed in current works. Fatemi et al. [2024] have demonstrated sensitivity to prompts and encoding strategy for graphs, however, the focus of these works is on graph theory-based tasks, different from causal queries.

We challenge the assumption that LLMs can seamlessly encode causal graphs and evaluate their true capabilities in this area. By introducing the benchmark, we aim to shed light on the strengths and limitations of these models in encoding causal graphs. To fully harness the potential of LLMs for causal reasoning, it is crucial to understand not only the opportunities they present but also the risks and precautions necessary for their effective use. While LLMs can enhance our ability to discover and understand causal relationships, they may also propagate biases from their training data and their performance can vary based on the prompting strategy and task. Therefore, careful evaluation and consideration of potential biases and limitations are essential when using LLMs for causal reasoning. Considering the application of using LLMs as causal hypothesis generators [Liu et al., 2024, Kıcıman et al., 2023, Ban et al., 2023a, Sheth et al., 2024], it is imperative to evaluate their basic causal graph understanding capabilities before advancing to more complex tasks. Addressing any potential challenges early on can help in refining the models, making them more robust and effective as tools for causal reasoning and hypothesis generation.

In this work, we aim to investigate the ability of LLMs to encode causal graphs and their effectiveness in assisting with causal reasoning tasks. We propose the benchmark, *CausalGraph2LLM*, designed to evaluate LLMs on tasks related to understanding causal graphs. Our work is the first work to focus on the encoding strategies and sensitivities of LLMs in the context of causal reasoning. We assess the performance of a variety of LLMs across a broad spectrum of tasks, each inspired by potential subtasks that LLMs might need to solve for a downstream task. This benchmark serves as a foundational reference for future works that use LLMs for causal reasoning-based tasks. Our contributions can be summarised as follows:

- We conduct a comprehensive study on various techniques to encode causal graphs into text for an LLM.
- We break down the task into several subtasks and graph-level and node-level queries to better understand the capabilities and limitations of LLMs in causal reasoning.
- Our work revealed biases in model performances when the contextual information was part of the pretraining data.
- We conduct extensive experiments on both open-source and closed models to uncover the limitations of LLMs in perfectly understanding causal graphs.

2 Related Works

Causal discovery and inference have predominantly been dominated by data-driven methods [Spirtes and Zhang, 2016]. However, due to the complexity of inferring causal structures, previous works have introduced priors on causal graphs in terms of interventions, domain expertise, edge existence, or ancestral constraints [Constantinou et al., 2023, Ban et al., 2023b, Brouillard et al., 2020]. These priors help to reduce the search spaces of potential causal graphs. Recent advancements in LLMs have motivated the use of LLM-based priors and causal discovery [Long et al., 2023, Cai et al., 2023, Abdulaal et al., 2024, Jin et al., 2023a, Kıcıman et al., 2023]. Unlike data-driven methods, LLMs leverage causal variable names to evaluate the existence of edges between them, thereby constructing causal graphs. The rich parametric knowledge of LLMs has proven to be almost as effective in discovering causal structures as traditional data-driven methods [Vashishtha et al., 2023, Kıcıman et al., 2023]. These initial results have motivated the integration of LLMs as priors combined with different statistical causal discovery methods. For instance, Vashishtha et al. [2023] used pairwise queries to discover the existence of edges between different causal variables and then applied methods such as PC [Spirtes et al., 2001] to reorient the edges, whereas Ban et al. [2023b] utilized LLM-based priors for scoring-based discovery methods. Vashishtha et al. [2023] suggest triplet-based prompting strategies, and Jiralerspong et al. [2024] proposed reducing the prompting complexity by prompting in a depth-first search (DFS) manner. More recently, Abdulaal et al. [2024] proposed an iterative collaboration between LLMs and structural causal models, where the LLM refines

the output of SCMs. Chen et al. [2024] benchmarked the performance of different LLMs against various causal queries. Combined with external tools, Jin et al. [2023b] demonstrated the use of LLMs for causal inference tasks, albeit on 3-4 node tasks. Another line of previous works [Girju et al., 2002, Hassanzadeh et al., 2020, Tan et al., 2023] explored the use of LLMs to discover potential causal structures from unstructured data.

Most of these works assume a specific prompting strategy. However, it remains unclear which strategy would be most effective. In this paper, we aim to contribute to this line of research by benchmarking a variety of LLMs on a range of tasks related to causal graphs and exploring the effectiveness of different causal graph encoders.

3 CausalGraph2LLM

Understanding causal graphs is a crucial step in harnessing LLMs for tasks based on causal graphs. Our benchmark, CausalGraph2LLM, is designed to evaluate the proficiency of LLMs in interpreting and utilizing causal graphs, a skill that is vital for applications in causal inference and discovery. An overview of the benchmark is depicted in Figure 1. By evaluating the ability of these models to process and comprehend the structure and implications of causal graphs, we aim to gain a deeper understanding of their potential and limitations in complex reasoning tasks.

3.1 Preliminaries

Causal graphs serve as an effective medium for conveying the perceived interactions among variables in a research context. These assumptions are visually depicted in a causal Directed Acyclic Graph (DAG), enabling researchers to infer which variables need to be controlled to reduce bias and identify those that could potentially introduce bias if controlled in the analysis. A causal graph is mathematically denoted as $G = (V, E)$, where V is a finite set of vertices or nodes, represented as $\{v_1, v_2, \ldots, v_n\}$, with each v_i signifying a distinct variable in the system. E is a set of ordered pairs of vertices, denoted as $\{(v_i, v_j) \mid v_i, v_j \in V \text{ and } i \neq j\}$, with each pair (v_i, v_j) representing a directed edge from node v_i to node v_j. The presence of an edge from v_i to v_j signifies a causal effect of the variable represented by v_i on the variable represented by v_j. Importantly, G is a DAG, meaning that for any node v_i, there does not exist a directed sequence of edges that starts and ends at v_i.

3.2 Prompting

Instruction-tuned LLMs are gaining popularity for solving tasks using LLMs, driven by their ability to leverage pre-trained knowledge to infer causal structures by simply prompting these models. Therefore, we employ prompting in our approach. We benchmark the ability of LLMs to understand causal graphs by transforming the causal graph into a verbalized prompt. Given a causal graph G, we define a prompting function $p : G \rightarrow P$, where P is the space of all possible prompts. This function transforms the graph into a verbalized format that the LLM can process. In this benchmark, we use various prompting transformations as used in current works or graph learning literature. We perform extensive experiments on 7 different encoding strategies. A brief overview of the encoding functions is illustrated in Figure 2. In Appendix B, we go into the details for each encoding strategy. We hypothesize that popular graph representations can influence performance in downstream tasks. While the specific graph structures encountered during LLM pretraining are unknown, our benchmark aims to benchmark which graph representation enhances performance for LLMs. The different encoders we use:

- **JSON** - This encoding represents the causal graph in a JSON format, capturing nodes and their causal relationships hierarchically [Abdulaal et al., 2024].
- **Adjacency** - This encoding lists all edges in the graph, showing direct causal relationships between nodes [Fatemi et al., 2024].
- **Adjacency matrix** - This encoding uses a matrix to represent the graph, with each cell indicating the presence (1) or absence (0) of a direct causal relationship between nodes.
- **GraphML** - This encoding uses the GraphML format, a comprehensive XML-based format for graph representation.
- **Graphviz** - This encoding uses the DOT language, which can be visualized using Graphviz tools.
- **Single node** - This encoding lists each node with its direct effects in a straightforward textual description.
- **Multi node** - This encoding provides a multi-node description where each cause is followed by all its effects.

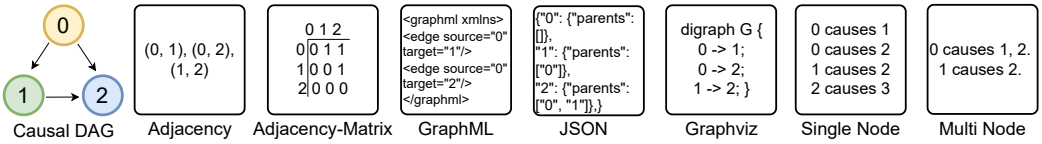


Figure 2: Different prompting transformation functions for the same causal graph.

3.3 Tasks

For the benchmark, we consider various causality-based tasks that could serve as potential subtasks, each of which may be crucial for a language model's downstream performance in understanding causal graphs. Following the graph encoding prompt, a task question prompt is added. Each task is designed to evaluate an LLM's ability to interpret causal graphs from different aspects of causality. Given a causal graph $G = (V, E)$, we explore various types of relationships between nodes, each with its relevance to causality, causal inference, and causal discovery.

Child and Parent: If there is a directed edge from node v_i to node v_j in E, then v_i is the parent of v_j, and v_j is the child of v_i. This relationship signifies a direct causal effect from the parent variable to the child variable.

Source and Sink: A source node is a node that has no incoming edges. It represents a variable that is not caused by any other variable in the system, often serving as the starting point of causal chains. A sink node is a node that has no outgoing edges. It represents a variable that does not cause any other variable in the system, often serving as the endpoint of causal chains.

Mediator: A mediator node in V is a node that lies on the path between two other nodes in E. It represents a variable that mediates the causal effect from one variable to another, playing a key role in the propagation of causal effects.

Confounder: A confounder node in V is a node that has outgoing edges to two or more other nodes in E. It represents a variable that can induce spurious associations between its child variables if not properly controlled.

By evaluating the LLM's ability to identify and interpret these different types of relationships, we aim to gain a comprehensive understanding of its capabilities and limitations in the context of causal reasoning.

3.3.1 Downstream Task

We build upon the graph interventional effect task as proposed by Kasetty et al. [2024] to evaluate the intervention reasoning abilities of an LLM as a downstream task.

We expand the dataset to include larger graphs beyond 3 variables. In causal inference, interventions alter variable values within a causal graph, breaking their causal dependencies. This is represented using Pearl's do-calculus notation as $do(X = x)$, where X is the intervened variable and x is the set value. This intervention isolates X from its original causes, allowing for the analysis of the intervention's downstream effects.

In a causal graph $G = (V, E)$, where V is a set of variables and E is a set of directed edges representing causal relationships, applying $do(X = x)$ results in a modified graph where X is fixed at x and any incoming edges to X are removed.

The task is to determine the intervention's impact on other variables. The LLM must infer whether the intervened variable X causes changes in the other variables, based on the causal graph's structure. The LLM's task is to identify and quantify the intervention $do(X = x)$'s downstream effects on the variables $V \backslash X$. This requires the LLM to interpret the causal graph and perform interventional reasoning. We evaluate this task by measuring the accuracy of LLM predictions.

3.4 Experimental setup

We evaluate our benchmark setup on a diverse array of datasets, including synthetic, semi-synthetic, and realistic scenarios. To control the complexity and structure of the graphs, we construct synthetic DAGs. Our benchmark includes commonly used causal graphs from recent literature [Ban et al., 2023b, Vashishtha et al., 2023, Ban et al., 2023a], which incorporate contextual information relevant to the causal relationships. We consider the causal graphs part of BNLearn repository - Insurance:$G(27, 52)$ [Binder et al., 1997], and Alarm:$G(37, 46)$ [Beinlich et al., 1989].

We evaluate the benchmark for various open-source and closed models. The models we use are GritLM [Muennighoff et al., 2024], Mistral-7B-Instruct-v0.2 [Jiang et al., 2023], Mixtral-8x7BInstruct-v0.1 [Jiang et al., 2024], Gem-

ini [Reid et al., 2024] , GPT-3.5 [Brown et al., 2020], and GPT-4 [Achiam et al., 2023].

4 Results

This section presents our benchmark results on causal graph understanding through causal queries. We investigate how effectively LLMs can interpret and reason about causal graphs encoded in different formats, addressing both graph-level and node-level queries. Additionally, we explore biases introduced by graph contextual information. The variances are reported in Appendix C.1 for brevity.

4.1 Graph-level queries

To evaluate the baseline graph level causal queries, we prompt the LLMs with causal query tasks resembling those encountered in larger causal reasoning tasks with different encodings. We measure the performance of these queries using the F1 score.

LLMs may struggle with simple causal query tasks. From Table 1, we observe a range of performances across different models and encoding types, highlighting the variability in how well each LLM handles causal graph encoding and interpretation. Out of Source and Sink based queries, interestingly the LLM has stronger performance on performing source tasks. We ablate in Appendix C.2 and observe that the *order* of causal graph description also has an impact on the performance of source and sink queries. This implies that the model's understanding of causal relationships may be influenced by the sequence in which information is presented. Tasks of greater complexity, such as identifying mediators, appear to be more challenging. This could be because the task of identifying a mediator intuitively involves breaking down the task into identifying 'child' and 'parent' elements, adding a layer of complexity to the task. In Figure 9 and Figure 10 we conclude a correlation with graph complexity and performance.

Average Performance. Observing the average performance for each model across different encodings suggests that the LLMs are highly sensitive to graph encoding. Adjacency-matrix encoding generally results in the lowest average performance across all models, despite being a popular format to represent causal graphs in code.

High sensitivity to the causal graph representation. We observe that different encodings for the same causal graphs have different performances across each causal query. For instance, for the Mistral model, JSON encoding has the F1 score of 0.21, however for GraphML or GraphViz encoding the performance increases to 0.46 for the Mediator task. GPT-4 and Gemini 1.5 Pro perform exceptionally well with certain encodings like GraphML and JSON, respectively, indicating that these formats might align better with the potential pretraining of the model. GritLM and Mistral show greater variability in their average performance, highlighting their sensitivity to the encoding methods used.

Correlation between Query and Encodings. Some queries may seem easier due to the definition of the encoding and its potential alignment with the encoding. For instance, for JSON encoding, identifying parent nodes might be relatively easier for all LLMs. This could be because the JSON-based prompt used by [Abdulaal et al., 2024] defines the dictionary by specifying the parents of each node. This alignment between the query and encoding likely facilitates the model's understanding of the causal relationships, resulting in improved performance on tasks involving parent nodes. This shows the importance of considering the encoding method coupled with the query when concerned with a causal graph-level reasoning task.

4.2 Effect of pretraining knowledge on causal graph understanding

In our previous experiment, we used synthetically generated causal graphs to assess the ability of LLMs to interpret and reason about causal relationships in a controlled setting. However, in this experiment, we aim to evaluate the impact of pretraining knowledge on the understanding of causal graphs. Current research often employs LLMs to extract causal priors by leveraging the semantic information embedded in variable names [Abdulaal et al., 2024, Vashishtha et al., 2023, Ban et al., 2023a, Sheth et al., 2024]. Their goal is to harness the knowledge and emergent reasoning abilities of LLMs to generate causal hypotheses. The primary motivation there is to harness the knowledge and emergent reasoning abilities of LLMs to generate causal hypotheses. Consequently, in this experiment, we specifically test causal graphs that incorporate contextual knowledge, allowing us to directly assess how pretraining influences LLMs' performance. We specifically consider two popular causal DAGs - Insurance [Binder et al., 1997], and Alarm [Beinlich et al., 1989]. These graphs were presented in two formats: one set featured contextual causal knowledge with semantically meaningful labels, and the other set consisted of the same

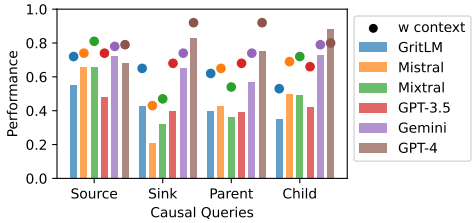
Model	Encoding	Source	Sink	Parent	Child	Mediator	Confounder	*Avg*
GritLM	JSON	0.25	0.30	0.15	0.20	0.10	0.15	0.19$\pm$0.10
	Adjacency	0.20	0.26	0.12	0.06	0.35	0.26	0.20$\pm$0.12
	Adjacency-M	0.00	0.05	0.08	0.11	0.06	0.06	0.06$\pm$0.03
	GraphML	0.38	0.24	0.14	0.21	0.18	0.29	0.24$\pm$0.08
	GraphViz	0.15	0.25	0.19	0.23	0.17	0.22	0.20$\pm$0.03
	Multi node	0.11	0.32	0.10	0.43	0.19	0.24	0.23$\pm$0.12
	Single node	0.12	0.34	0.18	0.36	0.25	0.17	0.23$\pm$0.10
	$\bar{x}$ / σ	0.18 / 0.38	0.27 / 0.29	0.14 / 0.11	0.20 / 0.37	0.19 / 0.29	0.20 / 0.23	
Mistral	JSON	0.30	0.04	0.58	0.20	0.21	0.19	0.25$\pm$0.18
	Adjacency	0.36	0.15	0.26	0.56	0.28	0.31	0.32$\pm$0.13
	Adjacency-M	0.07	0.16	0.11	0.10	0.09	0.10	0.10$\pm$0.03
	GraphML	0.18	0.21	0.31	0.59	0.46	0.61	0.39$\pm$0.18
	GraphViz	0.35	0.27	0.36	0.43	0.46	0.39	0.37$\pm$0.06
	Multi node	0.37	0.25	0.24	0.45	0.31	0.42	0.34$\pm$0.08
	Single node	0.50	0.22	0.44	0.43	0.33	0.20	0.35$\pm$0.12
	$\bar{x}$ / σ	0.32 / 0.43	0.21 / 0.23	0.30 / 0.47	0.38 / 0.49	0.33 / 0.41	0.30 / 0.27	
Mixtral	JSON	0.61	0.04	0.54	0.18	0.22	0.43	0.33$\pm$0.22
	Adjacency	0.32	0.56	0.45	0.49	0.44	0.32	0.43$\pm$0.09
	Adjacency-M	0.11	0.08	0.09	0.12	0.10	0.09	0.10$\pm$0.01
	GraphML	0.38	0.14	0.30	0.39	0.45	0.37	0.34$\pm$0.10
	GraphViz	0.76	0.50	0.46	0.39	0.55	0.37	0.50$\pm$0.14
	Multi node	0.39	0.49	0.27	0.29	0.49	0.19	0.35$\pm$0.12
	Single node	0.71	0.33	0.48	0.42	0.54	0.39	0.48$\pm$0.13
	$\bar{x}$ / σ	0.48 / 0.65	0.31 / 0.52	0.38 / 0.37	0.33 / 0.45	0.44 / 0.34	0.33 / 0.40	
GPT-3.5	JSON	0.75	0.25	0.47	0.08	0.37	0.26	0.36$\pm$0.23
	Adjacency	0.47	0.29	0.44	0.77	0.65	0.84	0.57$\pm$0.21
	Adjacency-M	0.05	0.19	0.10	0.11	0.15	0.10	0.12$\pm$0.11
	GraphML	0.72	0.51	0.50	0.61	0.36	0.37	0.51$\pm$0.13
	GraphViz	0.70	0.18	0.58	0.77	0.55	0.43	0.53$\pm$0.12
	Multi node	0.39	0.24	0.50	0.70	0.64	0.59	0.51$\pm$0.17
	Single node	0.70	0.30	0.56	0.67	0.55	0.45	0.54$\pm$0.14
	$\bar{x}$ / σ	0.57 / 0.70	0.31 / 0.33	0.48 / 0.48	0.50 / 0.69	0.50 / 0.50	0.47 / 0.74	
Gemini	JSON	0.80	0.77	0.97	0.56	0.68	0.72	0.76$\pm$0.13
	Adjacency	0.53	0.62	0.66	0.74	0.64	0.73	0.66$\pm$0.07
	Adjacency-M	0.12	0.49	0.07	0.12	0.11	0.07	0.22$\pm$0.16
	GraphML	0.84	0.54	0.76	0.56	0.67	0.60	0.67$\pm$0.11
	GraphViz	0.48	0.56	0.57	0.64	0.59	0.69	0.58$\pm$0.07
	Multi node	0.50	0.73	0.70	0.70	0.63	0.59	0.64$\pm$0.08
	Single node	0.88	0.62	0.69	0.73	0.71	0.57	0.71$\pm$0.10
	$\bar{x}$ / σ	0.65 / 0.76	0.62 / 0.28	0.69 / 0.90	0.64/0.62	0.64 / 0.66	0.62 / 0.68	
GPT-4	JSON	0.68	0.69	0.52	0.43	0.75	0.74	0.80$\pm$0.13
	Adjacency	0.77	0.58	0.69	0.69	0.84	0.75	0.73$\pm$0.09
	Adjacency-M	0.10	0.18	0.21	0.11	0.10	0.13	0.14$\pm$0.04
	GraphML	0.80	0.80	0.85	0.90	0.76	0.75	0.81$\pm$0.05
	GraphViz	0.67	0.67	0.80	0.85	0.70	0.69	0.71$\pm$0.07
	Multi node	0.66	0.65	0.73	0.88	0.84	0.79	0.75$\pm$0.09
	Single node	0.80	0.42	0.89	0.90	0.69	0.87	0.77$\pm$0.18
	$\bar{x}$ / σ	0.68 / 0.70	0.61 / 0.62	0.71 / 0.68	0.71 / 0.79	0.73 / 0.80	0.72 / 0.74	

Table 1: Performance comparison across methods and encodings. $\bar{x}$ denotes the average performance for each task and σ denotes the difference between the best and the worst encoding.

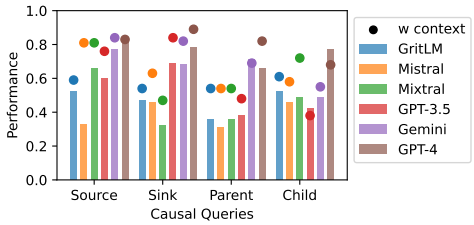
graphs labeled with random identifiers.

From Figure 3 we observe that giving contextual knowledge in terms of semantically meaningful causal variable names for the causal understanding tasks improves the performance across the models. This boost suggests that LLMs are effectively utilizing their pretraining on vast text corpora, where they have been exposed to a wide range of contexts and scenarios involving related variables. The semantics of the variable names likely help the models to enable a more accurate interpretation of the causal relationships by activating their parametric memory.

Risks associated with reliance on contextual knowledge. While the improvement in performance with contextual knowledge is promising, it also raises some concerns. Primarily, the potential (over-) reliance on semantically meaningful



(a) Insurance.



(b) Alarm.

Figure 3: Performance of different models across (a) Insurance and (b) Alarm graphs. Bars represent performance without context, while dots indicate performance with context. The results are averaged over different encodings for each model.

variable names can introduce biases based on the language and cultural context inherent in the training data of the LLMs. We additionally observe that due to contextual knowledge, false positives were increased as in the case of GPT-4 for the Child query for the Insurance graph. This occurs because the LLM assumes there are more causal relationships than are specified by the causal graphs. This observation aligns with findings from [Vashishtha et al., 2023], where it was also observed that LLMs suggest more causal relationships as a causal prior in comparison to the ground truth causal graph. Moreover, when flipping the directions of the DAG to generate an anti-commonsense DAG, there is a drop in performance (see Table 9). This suggests that if a causal DAG with contextual information does not follow the pretraining of the LLM, due to inherited biases of the LLM, it may lead to more incorrect causal query inferences.

4.3 Node-level queries are simpler for LLM

In the above experiments, we considered *graph-level* tasks only, that require the LLM to identify and list all instances of the mentioned node type present in a given causal graph. This requires the model to understand the entire graph structure and identify all nodes that follow the given task criteria (such as source, sink). In this experiment, we break down the graph-level tasks into binary queries for *node-inspection* tasks to better understand the model's performance on simpler, more

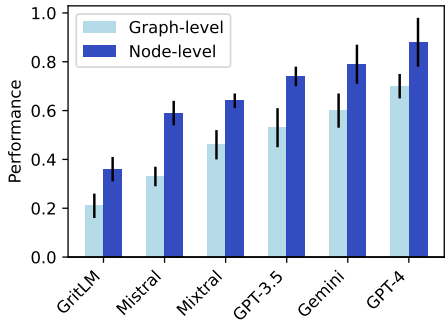


Figure 4: Node-level vs. graph-level query performances.

focused queries. The graph-level tasks are converted into binary queries reducing the processing load on the model. Each binary query asks whether a specific node in the graph is of a given type allowing LLM to focus on individual nodes.

Our results, summarized in Figure 4, indicate that LLMs generally perform better on node-level query tasks. Lower performance for graph-level tasks can be attributed to the difficulty of the tasks. This could be explained as the graph-level queries require the LLM to maintain a holistic understanding of the graph's structure. Secondly, an error in identifying one node incorrectly can lead to a cascade of errors, hence reducing LLM's ability to understand the entire graph structure. In contrast, *node-inspection* tasks isolate each node-based query, reducing the impact of individual errors and leading to more accurate overall performance.

4.3.1 Overestimation and Underestimation biases

For the binary *node-level* queries, we can further break down the results for each LLM and evaluate their false positives (FP) and false negatives (FN). This allows for an insight into the LLM to evaluate where the error truly comes from. False positives occur when the LLM incorrectly identifies a node as a specific type when it is not. False negatives occur when the LLM fails to identify a specific node type. We average the ratio of FP to FN across every task and embedding to report this ratio (τ) for each model. A high τ ratio indicates that the model is more prone to false positives, meaning it frequently identifies nodes as a specific type even when they are not. Conversely, a low τ ratio (i.e., $\tau < 1$) would indicate a higher rate of false negatives, where the model fails to identify nodes that are of a specific type. This underestimation suggests that the model might be overly conservative

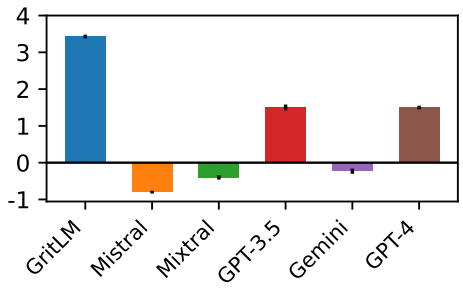


Figure 5: Evaluation of over- and underestimation biases.

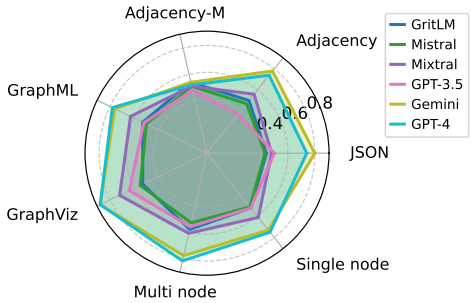


Figure 6: Downstream task performance.

in identifying causal relationships.

From Figure 5, we observe that GritLM, GPT-3.5, and GPT-4 have $\tau > 1$ ratios. It suggests that the model tends to causal overestimation bias even without the influence of contextual information. Some of the recent works have also explored the overestimation phenomenon of LLMs [Herrera-Berg et al., 2023, Li et al., 2024]. However, Gemini, Mistral, and Mixtral portray opposite behaviors, and their False Negative predictions are higher. Such biases could arise from their RLHF fine-tuning stage as well and require further investigation to explore such biases for causal queries.

4.4 Downstream task performance

In this work, we aimed to benchmark LLM performance for various causal graph queries. From Table 1, we observed that different encodings have varied effects across different causal tasks. In this experiment, however, we aim to observe this effect on a downstream task.

From Figure 6, we observed variability in model performances due to different encoding functions for downstream intervention tasks. At first glance, it may seem that adjacency-matrix encoding has higher performance in the intervention tasks as compared to graph-level queries. However, the random baseline is 0.50, and the results here for the adjacency matrix are close to the random baseline even for the best-performing GPT-4 model.

4.5 Discussion

Our study, centered on zero-shot prompting, assesses how the comprehensive training of current models impacts their responses to causal queries. As shown in Table 1, models can handle causal queries reasonably well with appropriate encoding. However, performance varies based on the encoding used, likely due to differences in understanding the rich distribution within each encoding. We hypothesize that fine-tuning could improve performance across different encodings by allowing models to adapt their pre-existing knowledge to specific causal queries. We tested this by fine-tuning the Mistral-7b model in Table 8. We observe the biggest increase in the performance of adjacency matrix encoding. In addition to textual encoding, we also explored visual graph encoding with more advanced models. In this approach, the LLM is prompted with an image of the causal graph. Interestingly, we found that the performance using visual encoding outperformed some of the textual encodings, although it did not emerge as the best approach (see Table 7).

Despite the improvements brought about by fine-tuning, the performance indicates that there is still room for enhancement. Additionally, while fine-tuning proved effective, the influence of contextual knowledge and biases on performance remains an open question. We anticipate that this benchmark will serve as a valuable tool in addressing these questions and developing defence method against the biases.

5 Conclusion

With the increasing of LLMs to assist with causal inference and causal discovery tasks, it is important to explore the opportunities and the limitations due to the nature of LLMs. In this paper, we proposed the first benchmark, *CausalGraph2LLM* to evaluate the encoding capabilities of LLMs for causal DAGs, encompassing both graph-level and node-level queries. Our findings also shed light on the potential risks associated with employing LLMs for causal reasoning tasks, particularly emphasizing the potential biases stemming from their pre-trained knowledge. These insights serve as a valuable reference for future research leveraging LLMs in causal DAG manipulation.

6 Liminations and Future Work

The scope of the evaluation is primarily limited to synthetic and well-known causal graphs, which may not fully capture the complexity of real-world causal graphs. We presented 6 diverse tasks, which can be built upon for future work. Future work can expand the diversity of causal graphs and models evaluated, develop more robust encoding techniques, and explore methods to mitigate contextual biases. Enhancing the models' ability to handle complex tasks and improving downstream task performance will also be crucial. Additionally, a deeper investigation into biased sources can provide a more nuanced understanding of LLM capabilities in causal inference. Given the modular nature of the benchmark, we aim to continue to build up this benchmark to assess newer models as they come.

7 Ethics and Risks

All of the datasets used are publicly available. Our implementation utilizes the PyTorch 1.12 framework, an open-source library. Our research is conducted per the licensing agreements of the Mistral-7B, GPT-3.5, and GPT-4 models. We ran our experiments on A100 Nvidia GPU and via OpenAI API.

Further, we recognize the risks of using LLMs for causal discovery. Our work attempts to shed light on the risks and biases that may affect the performance for causal reasoning.

References

Ahmed Abdulaal, adamos hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C. Castro, and Daniel C. Alexander. 2024. Causal modelling agents: Causal graph discovery through synergising metadata- and data-driven reasoning. In *ICLR*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv*.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. 2023a. Causal structure learning supervised by large language model. *arXiv*.

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

Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. 1989. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89: Second European Conference on Artificial Intelligence in Medicine, London, August 29th–31st 1989. Proceedings*, pages 247–256. Springer.

John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. 1997. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. 2020. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems*, 33:21865–21877.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv*.

Hengrui Cai, Shengjie Liu, and Rui Song. 2023. Is knowledge all large language models needed for causal reasoning? *arXiv*.

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. 2024. Clear: Can language models really understand causal graphs? *arXiv preprint arXiv:2406.16605*.

Anthony C Constantinou, Zhigao Guo, and Neville K Kitson. 2023. The impact of prior knowledge on causal structure learning. *Knowledge and Information Systems*, 65(8):3385–3434.

Gregory F Cooper and Changwon Yoo. 2013. Causal discovery from a mixture of experimental and observational data. *arXiv*.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *ICLR*.

Roxana Girju, Dan I Moldovan, et al. 2002. Text mining for causal relations. In *FLAIRS conference*, pages 360–364.

Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2020. Causal knowledge extraction through large-scale text mining. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 13610–13611.

Eugenio Herrera-Berg, Tomás Vergara Browne, Pablo León-Villagrá, Marc-Lluís Vives, and Cristian Buc Calderon. 2023. Large language models are biased to overestimate profoundness. In *EMNLP*.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. 2020. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv*.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023a. CLadder: Assessing causal reasoning in language models. In *NeurIPS*.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. 2023b. Can large language models infer causation from correlation? *arXiv*.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient causal graph discovery using large language models. *arXiv preprint arXiv:2402.01207*.

Tejas Kasetty, Divyat Mahajan, Gintare Karolina Dziugaite, Alexandre Drouin, and Dhanya Sridhar. 2024. Evaluating interventional reasoning capabilities of large language models. *arXiv preprint arXiv:2404.05545*.

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

Tianlin Li, Xiaoyu Zhang, Chao Du, Tianyu Pang, Qian Liu, Qing Guo, Chao Shen, and Yang Liu. 2024. Your large language model is secretly a fairness proponent and you should prompt it like one. *arXiv*.

Chenxi Liu, Yongqiang Chen, Tongliang Liu, Mingming Gong, James Cheng, Bo Han, and Kun Zhang. 2024. Discovery of the hidden world with large language models. *arXiv*.

Stephanie Long, Tibor Schuster, Alexandre Piché, ServiceNow Research, et al. 2023. Can large language models build causal graphs? *arXiv*.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv*.

Ana Rita Nogueira, Andrea Pugnana, Salvatore Ruggieri, Dino Pedreschi, and João Gama. 2022. Methods and tools for causal discovery and causal inference. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 12(2):e1449.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. 2023. Large language models are zero shot hypothesis proposers. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*.

Ivaxi Sheth, Sahar Abdelnabi, and Mario Fritz. 2024. Hypothesizing missing causal variables with llms. *arXiv preprint arXiv:2409.02604*.

Peter Spirtes, Clark Glymour, and Richard Scheines. 2001. *Causation, prediction, and search*. MIT press.

Peter Spirtes and Kun Zhang. 2016. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, pages 1–28. Springer.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv*.

Fiona Anting Tan, Xinyu Zuo, and See-Kiong Ng. 2023. Unicausal: Unified benchmark and repository for causal text mining. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 248–262. Springer.

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

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D Goodman. 2023. Hypothesis search: Inductive reasoning with language models. *arXiv*.

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

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

A Reproduciblility

We will release our code, prompts, evaluation setup, and all models' outputs of our experiments. For reproducibility, we used temperature 0 and top-p value as 1 across all of the models. We also mentioned the snapshot of the model used. We have also included the prompts and examples below. Our code will be made public post the anonymity period.

The Alarm and Insurance datasets are under CC BY-SA 3.0 which allows us to freely modify the datasets for benchmarking. Our benchmark will be released under the CC BY-SA License.

Mistral and GritLM were run on 1 A100 GPU whereas Mixtral was run on 8 A100 GPUs. Since we used off-the shelf LLM, each graph-level experiment took no more than 30 minutes to run (longer for mediator, child, parent, confounder whereas source and sink took $\approx$ 3 mins to run). All of the experiments for each model took $\approx$ 38 hours. GPT-3.5 GPT-4 were accessed via API.

A.1 Dataset descriptions

The datasets used can be divided into two: 1. realistic datasets and 2. synthetic datasets.

We use the two real-world-based datasets. These are semi-synthetic datasets available from the BNLearn library. The first graph, named **Alarm**, is a well-known benchmark in the field of causal inference. The Alarm dataset (see Figure 11) is designed to model the relationships and dependencies in an intensive care unit (ICU) monitoring system. It includes variables such as heart rate, blood pressure, and other vital signs, making it a complex and realistic representation of medical data. This dataset is particularly useful for evaluating the ability of LLMs to handle intricate causal relationships in a medical high-stakes environment.

The second dataset, **Insurance**, is another widely used benchmark that models the risk factors and dependencies in the insurance domain. This graph (see Figure 12) includes variables related to policyholders, such as age, driving history, and vehicle type, and their relationships to insurance claims and premiums. The Insurance dataset provides a different context from the medical domain, allowing us to assess the versatility of LLMs in understanding and reasoning about causal relationships in a financial setting.

A.2 Synthetic dataset

In addition to real-world-based datasets, we created synthetic datasets with varying levels of difficulty to rigorously evaluate the performance of LLMs. These synthetic datasets were designed to systematically vary in complexity by adjusting the number of nodes and edges in the causal graphs. This variation allows us to assess how well the models handle different levels of graph complexity and density. The synthetic datasets serve as a controlled environment to test the models' ability to interpret and reason about causal relationships under varying conditions. By incrementally increasing the number of edges while keeping the number of nodes constant, we can observe how the models' performance scales with the complexity of the causal structure. This approach provides valuable insights into the strengths and limitations of LLMs in handling more intricate causal graphs, which is crucial for understanding their potential applications in real-world scenarios. For the experiments, we synthesized graphs with 20 and 30 nodes. For each of these node variables, we experimented with different densities of nodes. Hence we had density = 1 x nodes, 1.5 x nodes and 2 x nodes.

B Prompting stratergies

```
( 0 , 1 )
( 0 , 2 )
( 1 , 3 )
( 2 , 3 )
( 2 , 4 )
( 3 , 4 )
( 0 , 3 )
( 1 , 4 )
( 0 , 4 )
( 1 , 2 )
```

	0	1	2	3	4
0	0	1	1	1	1
1	0	0	1	1	1
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawi
  <graph edgedefault="directed">
    <node id="0"/>
    <node id="1"/>
    <node id="2"/>
    <node id="3"/>
    <node id="4"/>
    <edge source="0" target="1"/>
    <edge source="0" target="2"/>
    <edge source="1" target="3"/>
    <edge source="2" target="3"/>
    <edge source="2" target="4"/>
    <edge source="3" target="4"/>
    <edge source="0" target="3"/>
    <edge source="1" target="4"/>
    <edge source="0" target="4"/>
    <edge source="1" target="2"/>
  </graph>
</graphml>
```

```
digraph G {
  0 -> 1;
  0 -> 2;
  1 -> 3;
  2 -> 3;
  2 -> 4;
  3 -> 4;
  0 -> 3;
  1 -> 4;
  0 -> 4;
  1 -> 2;
}
```

```
{
    "0": {
        "parents": []
    },
    "1": {
        "parents": [
            "0"
        ]
    },
    "2": {
        "parents": [
            "0",
            "1"
        ]
    },
    "3": {
        "parents": [
            "0",
            "1",
            "2"
        ]
    },
    "4": {
        "parents": [
            "0",
            "1",
            "2",
            "3"
        ]
    }
}
```

0 causes 1, 2, 3, 4. 1 causes 3, 4, 2. 2 causes 3, 4. 3 causes 4.

0 causes 1. 0 causes 2. 0 causes 3. 0 causes 4. 1 causes 3. 1 causes 4. 1 causes 2. 2 causes 3. 2 causes 4. 3 causes 4.

C Experiments

C.1 Variance

C.2 Ordering of prompt matter for causal queries

In BFS, the traversal starts from the source nodes, while in BFS-R, the traversal begins from the sink nodes. The values in the table represent the per-

Model	Enc	Source	Sink	Parent	Child	Mediator	Confounder	*Avg*
GritLM	JSON	0.25 ±0.07	0.30 ±0.05	0.15 ±0.02	0.20 ±0.07	0.10 ±0.08	0.15 ±0.07	0.19±0.10
	Adjacency	0.20 ±0.03	0.26 ±0.04	0.12 ±0.01	0.06 ±0.02	0.35 ±0.05	0.26 ±0.04	0.20±0.12
	Adjacency-M	0.00 ±0.00	0.05 ±0.01	0.08 ±0.01	0.11 ±0.02	0.06 ±0.01	0.06 ±0.01	0.06±0.03
	GraphML	0.38 ±0.06	0.24 ±0.04	0.14 ±0.03	0.21 ±0.05	0.18 ±0.04	0.29 ±0.05	0.24±0.08
	GraphViz	0.15 ±0.03	0.25 ±0.05	0.19 ±0.04	0.23 ±0.04	0.17 ±0.03	0.22 ±0.04	0.20±0.03
	Multi node	0.11 ±0.02	0.32 ±0.06	0.10 ±0.02	0.43 ±0.08	0.19 ±0.04	0.24 ±0.05	0.23±0.12
	Single node	0.12 ±0.03	0.34 ±0.06	0.18 ±0.04	0.36 ±0.07	0.25 ±0.05	0.17 ±0.04	0.23±0.10
Mistral	JSON	0.30 ±0.03	0.04 ±0.01	0.58 ±0.06	0.20 ±0.02	0.21 ±0.02	0.19 ±0.02	0.25±0.18
	Adjacency	0.36 ±0.04	0.15 ±0.02	0.26 ±0.03	0.56 ±0.06	0.28 ±0.03	0.31 ±0.03	0.32±0.13
	Adjacency-M	0.07 ±0.01	0.16 ±0.02	0.11 ±0.01	0.10 ±0.01	0.09 ±0.01	0.10 ±0.01	0.10±0.03
	GraphML	0.18 ±0.02	0.21 ±0.02	0.31 ±0.03	0.59 ±0.06	0.46 ±0.05	0.61 ±0.06	0.39±0.18
	GraphViz	0.35 ±0.04	0.27 ±0.03	0.36 ±0.04	0.43 ±0.04	0.46 ±0.05	0.39 ±0.04	0.37±0.06
	Multi node	0.37 ±0.04	0.25 ±0.03	0.24 ±0.02	0.45 ±0.05	0.31 ±0.03	0.42 ±0.04	0.34±0.08
	Single node	0.50 ±0.05	0.22 ±0.02	0.44 ±0.04	0.43 ±0.04	0.33 ±0.03	0.20 ±0.02	0.35±0.12
Mixtral	JSON	0.61 ±0.06	0.04 ±0.01	0.54 ±0.05	0.18 ±0.02	0.22 ±0.02	0.43 ±0.04	0.33±0.22
	Adjacency	0.32 ±0.03	0.56 ±0.05	0.45 ±0.04	0.49 ±0.05	0.44 ±0.04	0.32 ±0.03	0.43±0.09
	Adjacency-M	0.11 ±0.01	0.08 ±0.01	0.09 ±0.01	0.12 ±0.01	0.10 ±0.01	0.09 ±0.01	0.10±0.01
	GraphML	0.38 ±0.04	0.14 ±0.01	0.30 ±0.03	0.39 ±0.04	0.45 ±0.04	0.37 ±0.04	0.34±0.10
	GraphViz	0.76 ±0.07	0.50 ±0.05	0.46 ±0.04	0.39 ±0.04	0.55 ±0.05	0.37 ±0.04	0.50±0.14
	Multi node	0.39 ±0.04	0.49 ±0.05	0.27 ±0.03	0.29 ±0.03	0.49 ±0.05	0.19 ±0.02	0.35±0.12
	Single node	0.71 ±0.07	0.33 ±0.03	0.48 ±0.05	0.42 ±0.04	0.54 ±0.05	0.39 ±0.04	0.48±0.13
GPT-3.5	JSON	0.75 ±0.07	0.25 ±0.03	0.47 ±0.05	0.08 ±0.01	0.37 ±0.04	0.26 ±0.03	0.36±0.23
	Adjacency	0.47 ±0.05	0.29 ±0.03	0.44 ±0.04	0.77 ±0.08	0.65 ±0.07	0.84 ±0.09	0.57±0.21
	Adjacency-M	0.05 ±0.01	0.19 ±0.02	0.10 ±0.01	0.11 ±0.01	0.15 ±0.02	0.10 ±0.01	0.12±0.11
	GraphML	0.72 ±0.07	0.51 ±0.05	0.50 ±0.05	0.61 ±0.06	0.36 ±0.04	0.37 ±0.04	0.51±0.13
	GraphViz	0.70 ±0.07	0.18 ±0.02	0.58 ±0.06	0.77 ±0.08	0.55 ±0.06	0.43 ±0.04	0.53±0.12
	Multi node	0.39 ±0.04	0.24 ±0.02	0.50 ±0.05	0.70 ±0.07	0.64 ±0.06	0.59 ±0.06	0.51±0.17
	Single node	0.70 ±0.07	0.30 ±0.03	0.56 ±0.06	0.67 ±0.07	0.55 ±0.06	0.45 ±0.05	0.54±0.14
Gemini	JSON	0.80 ±0.08	0.77 ±0.08	0.97 ±0.10	0.56 ±0.06	0.68 ±0.07	0.72 ±0.07	0.76±0.13
	Adjacency	0.53 ±0.05	0.62 ±0.06	0.66 ±0.07	0.74 ±0.07	0.64 ±0.06	0.73 ±0.07	0.66±0.07
	Adjacency-M	0.12 ±0.01	0.49 ±0.05	0.07 ±0.01	0.12 ±0.01	0.11 ±0.01	0.07 ±0.01	0.22±0.16
	GraphML	0.84 ±0.08	0.54 ±0.05	0.76 ±0.08	0.56 ±0.06	0.67 ±0.07	0.60 ±0.06	0.67±0.11
	GraphViz	0.48 ±0.05	0.56 ±0.06	0.57 ±0.06	0.64 ±0.06	0.59 ±0.06	0.69 ±0.07	0.58±0.07
	Multi node	0.50 ±0.05	0.73 ±0.07	0.70 ±0.07	0.70 ±0.07	0.63 ±0.06	0.59 ±0.06	0.64±0.08
	Single node	0.88 ±0.09	0.62 ±0.06	0.69 ±0.07	0.73 ±0.07	0.71 ±0.07	0.57 ±0.06	0.71±0.10
GPT-4	JSON	0.68 ±0.07	0.69 ±0.07	0.52 ±0.05	0.43 ±0.04	0.75 ±0.08	0.74 ±0.07	0.80±0.13
	Adjacency	0.77 ±0.08	0.58 ±0.06	0.69 ±0.07	0.69 ±0.07	0.84 ±0.08	0.75 ±0.08	0.73±0.09
	Adjacency-M	0.10 ±0.01	0.18 ±0.02	0.21 ±0.02	0.11 ±0.01	0.10 ±0.01	0.13 ±0.01	0.14±0.04
	GraphML	0.80 ±0.08	0.80 ±0.08	0.85 ±0.09	0.90 ±0.09	0.76 ±0.08	0.75 ±0.08	0.81±0.05
	GraphViz	0.67 ±0.07	0.67 ±0.07	0.80 ±0.08	0.85 ±0.09	0.70 ±0.07	0.69 ±0.07	0.71±0.07
	Multi node	0.66 ±0.07	0.65 ±0.07	0.73 ±0.07	0.88 ±0.09	0.84 ±0.08	0.79 ±0.08	0.75±0.09
	Single node	0.80 ±0.08	0.42 ±0.04	0.89 ±0.09	0.90 ±0.09	0.69 ±0.07	0.87 ±0.09	0.77±0.18

Table 2: Performance comparison across methods and encodings.

formance of the models on the tasks, with higher values indicating better performance.

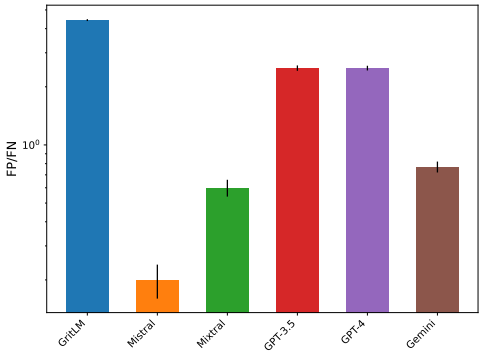


Figure 7: Evaluation of over- and underestimation biases.

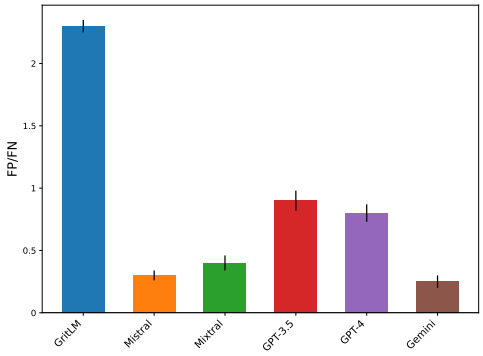


Figure 8: Evaluation of over- and underestimation biases for downstream task.

The results show that the traversal order significantly impacts the performance of the models. For instance, GritLM performs better on source tasks when the traversal is in BFS order, while it performs better on sink tasks when the traversal is in BFS-R order. This pattern is consistent across all models, suggesting that BFS is more suitable for identifying source nodes, while BFS-R is more suitable for identifying sink nodes.

D	Model	Source		Sink	
		BFS	BFS-R	BFS	BFS-R
Synthetic	GritLM	0.18	0.24	0.27	0.0.47
	Mistral	0.32	0.26	0.21	0.39
	Mixtral	0.48	0.40	0.31	0.44
	GPT-3.5	0.57	0.48	0.31	0.64
	Gemini	0.65	0.54	0.62	0.82
	GPT-4	0.68	0.57	0.61	0.89

Table 4: Comparing the order for prompts, BFS means it starts from source and BFS-R means it starts from sinks.

C.3 Downstream performance under/over bias

In the main paper, we analyzed over and underestimation bias for the binary node inspection task. We can conduct a similar analysis on the downstream task. Here, we observe a similar trend to the estimation biases in Section 4.3.1. Notably, GPT-3.5 and GPT-4 usually have FP/FN ratios closer to 1. See Figure 8.

C.4 Effect of node explanations

In our experimental setup, we took an approach to defining each task for every metric. This was primarily due to the varying terminologies used in causal inference across different academic circles. For instance, what some researchers might refer to as a 'source', others might call a 'root'. To avoid any potential confusion, we provided clear definitions for each term used in our causal queries.

Since pretraining for each model was not known, this adds an element of uncertainty to the task. To counteract this, we explicitly mentioned the query in our experiments. We conducted a set of preliminary experiments without an explanation of the query to demonstrate its effectiveness. The results showed a decrease in model performance, suggesting that providing explicit direction in the form of a mentioned query can be beneficial.

C.5 Node Complexity

C.6 Further models

C.6.1 LLama3.1 models

We additionally tested models from the LLama-3.1 family. We observe that LLama3.1 models also observe sensitivity to the graph encoding.

C.6.2 Multimodal models

In this work we focus on textual encodings into LLMs, however with the developments of multimodal models, we can test LLM's ability to answer causal queries when presented with image inputs. We performed our experiment on GPT-4 model with T=0. Future works can be built upon to test better image inputs for multimodal models.

Model	Enc	Source	Sink	Parent	Child	Mediator	Confounder
GPT-3.5	JSON	0.52	0.25	0.47	0.08	0.30	0.31
	Adjacency	0.32	0.26	0.44	0.65	0.72	0.51
	Adjacency-M	0.06	0.15	0.10	0.11	0.08	0.12
	GraphML	0.34	0.38	0.50	0.61	0.37	0.39
	GraphViz	0.42	0.19	0.58	0.77	0.52	0.28
	Multi node	0.39	0.24	0.50	0.70	0.64	0.27
	Single node	0.45	0.27	0.56	0.67	0.39	0.50

Table 5: Performance comparison across methods and encodings for GPT-3.5 without causal query explanations.

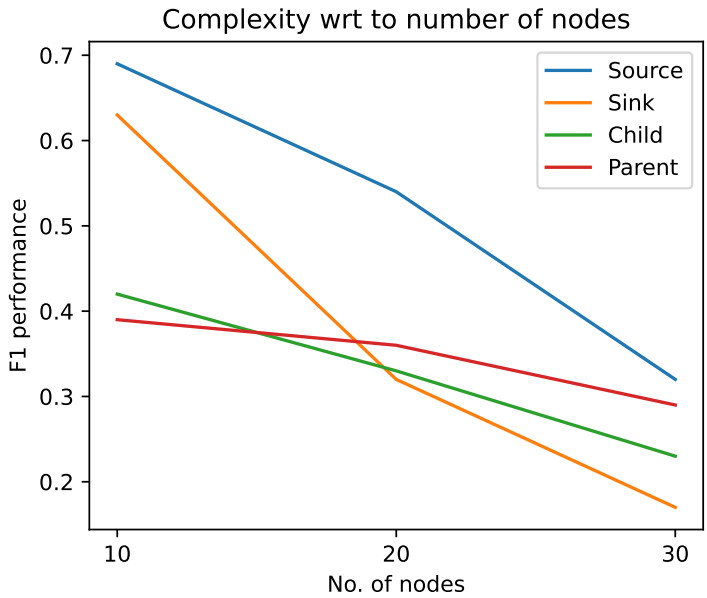


Figure 9: With an increase in graph complexity by increasing the number of nodes, we observe poorer performance of the LLM - Mistral model.

Model	Enc	Source	Sink	Parent	Child	Mediator	Confounder
8b	JSON	0.30	0.35	0.22	0.25	0.20	0.22
	Adjacency	0.28	0.30	0.18	0.20	0.22	0.23
	GraphML	0.35	0.31	0.28	0.29	0.18	0.31
	Single node	0.36	0.27	0.33	0.40	0.36	0.38
70b	JSON	0.62	0.65	0.52	0.55	0.48	0.50
	Adjacency	0.55	0.58	0.48	0.50	0.50	0.52
	GraphML	0.63	0.62	0.60	0.62	0.58	0.62
	Single node	0.71	0.75	0.72	0.74	0.68	0.69
405b	JSON	0.80	0.82	0.74	0.76	0.70	0.72
	Adjacency	0.75	0.78	0.70	0.72	0.68	0.70
	GraphML	0.85	0.83	0.80	0.82	0.77	0.79
	Single node	0.88	0.90	0.85	0.87	0.82	0.84

Table 6: Performance comparison across methods and encodings for llama models.

Source	Sink	Child	Parent	Mediator	Confounder
0.58	0.62	0.71	0.65	0.58	0.63

Table 7: Performance metrics for vision GPT-4.

C.7 Effect of finetuning

In this paper, we focused on zero-shot prompting as the current models have billions of trainable parameters and have been trained on a plethora of training data, potentially including causal graphs.

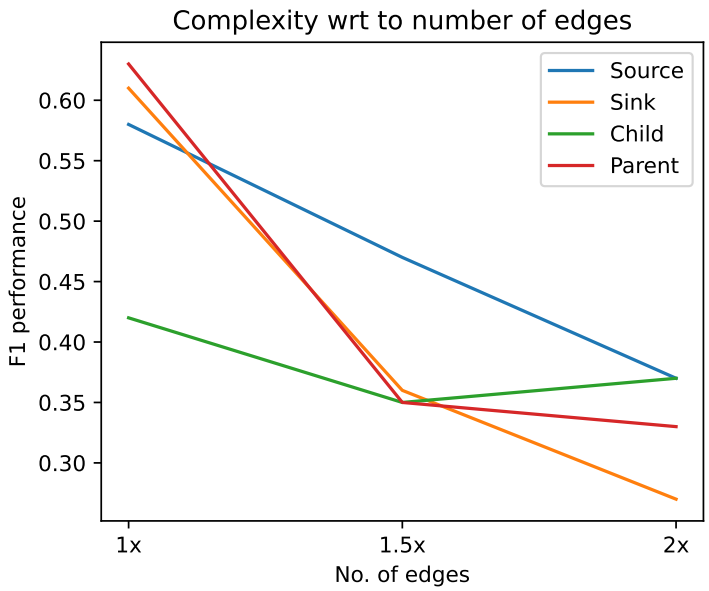


Figure 10: With an increase in graph complexity by increasing the number of edges, we observe poorer performance of the LLM - Mistral model.

We hence aimed to evaluate how this reflects in the causal queries. Additionally, most current methods utilize LLMs without fine-tuning for causal discovery queries, and our study aimed to replicate this environment to provide a realistic benchmark. We performed QLORA on Mistral 7b specifically on synthetic datasets. As expected, we observed an increase in the performance with finetuning.

C.8 Anti-commonsense context results

In this experiment, we essentially flipped the arrows of the DAG. We observed that the LLM showed a decrease in performance showing its reliance on background parametric knowledge for reasoning tasks.

	Source	Sink	Parent	Child	Mediator	Confounder
JSON	0.30	0.04	0.58	0.20	0.21	0.19
JSON FT	0.63	0.36	0.73	0.42	0.33	0.44
Adjacency	0.36	0.15	0.26	0.56	0.28	0.31
Adjacency FT	0.53	0.44	0.71	0.46	0.41	0.52
Adjacency-M	0.07	0.16	0.11	0.10	0.09	0.10
Adjacency-M FT	0.47	0.31	0.30	0.38	0.33	0.37
GraphML	0.18	0.21	0.31	0.59	0.46	0.61
GraphML FT	0.47	0.42	0.55	0.73	0.68	0.73
Multi node	0.37	0.25	0.24	0.45	0.31	0.42
Multi node FT	0.66	0.68	0.48	0.71	0.62	0.53
Single node	0.50	0.22	0.44	0.43	0.33	0.20
Single node FT	0.81	0.58	0.69	0.75	0.53	0.49

Table 8: Effect of finetuning Mistral 7b model for different graph encoding.

Enc	**Model**	**Source**		**Sink**		**Parent**		**Child**	
		w/o	anti	w/o	anti	w/o	anti	w/o	anti
Insurance	GritLM	0.55	0.38	0.43	0.21	0.40	0.18	0.35	0.17
	Mistral	0.66	0.58	0.21	-0.01	0.43	0.21	0.50	0.31
	Mixtral	0.66	0.51	0.32	0.17	0.36	0.18	0.49	0.26
	GPT-3.5	0.48	0.22	0.40	0.12	0.39	0.10	0.42	0.18
	Gemini	0.72	0.66	0.65	0.56	0.57	0.40	0.73	0.67
	GPT-4	0.68	0.57	0.83	0.74	0.75	0.58	0.88	0.96
Alarm	GritLM	0.52	0.45	0.47	0.40	0.36	0.18	0.52	0.43
	Mistral	0.33	0.25	0.46	0.29	0.31	0.28	0.46	0.34
	Mixtral	0.66	0.51	0.32	0.17	0.36	0.18	0.49	0.26
	GPT-3.5	0.60	0.44	0.69	0.54	0.38	0.28	0.42	0.46
	Gemini	0.77	0.70	0.68	0.54	0.71	0.73	0.49	0.43
	GPT-4	0.82	0.81	0.78	0.67	0.66	0.50	0.77	0.86

Table 9: Performance of different models across Alarm and Insurance graphs. w/o - without context anti - with reversed contextual variables. The results are averages across the encodings.

D Causal Query explanation

A source node in a causal graph is a variable that does not have any incoming edges, meaning it is not caused by any other variable in the graph.

A sink node in a causal graph is a variable that does not have any children in the graph, meaning it is not caused by any other variables in the system.

A direct mediator in a causal graph is a variable that lies on the direct path between two other immediate variables. Only consider mediators that exist in the direct causal path (not mediated via other mediators).

A confounder in a causal graph is a variable that influences both the cause and the effect variables. It is a common cause for both the dependent and independent variables.

What nodes are the direct causes of Node X?

What nodes are directly caused by Node X?

D.1 Prompt

For further prompt templates, please check the codebase.

Hello. You will be given a causal graph. The causal relationships in this causal graph are - [causal-graph-based-encoding]. Now answer using this causal graph only, name all of the [node-type] in the graph. [node-type-description]. Think step by step. Give reasoning and then give answer within <Answer> [a1,a2,a3..] </Answer>, if Null then return <Answer>Null</Answer>.

Hello. You will be given a causal graph. The causal relationships in this causal graph are - [causal-graph-encodingbased]. Now answer using this causal graph only, is [nodeX] a [node-type] in the graph. [node-type-description]. Think step by step. Give reasoning and then give answer within <Answer> Yes/No </Answer>.

E Graphs

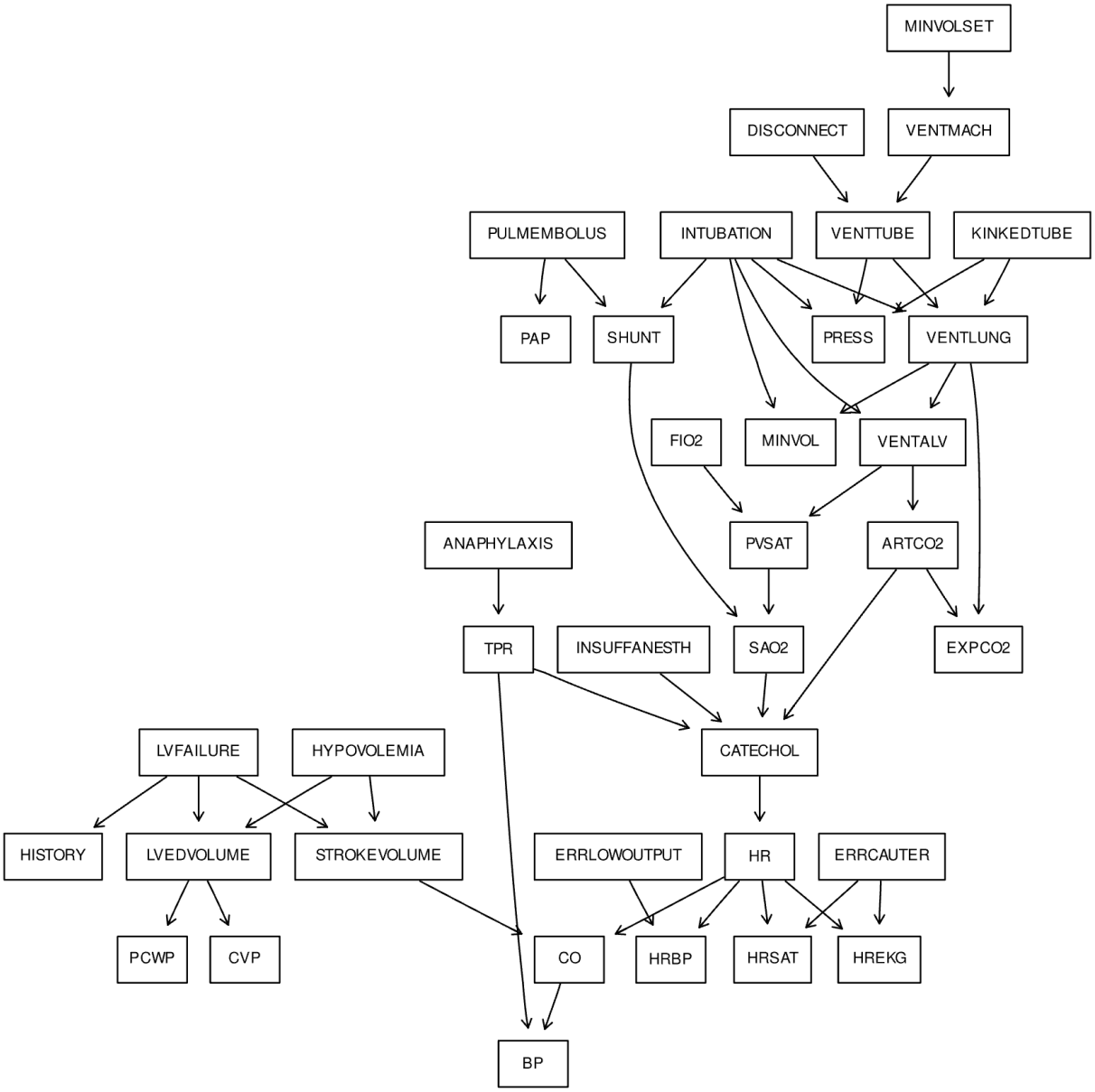


Figure 11: Alarm causal graph

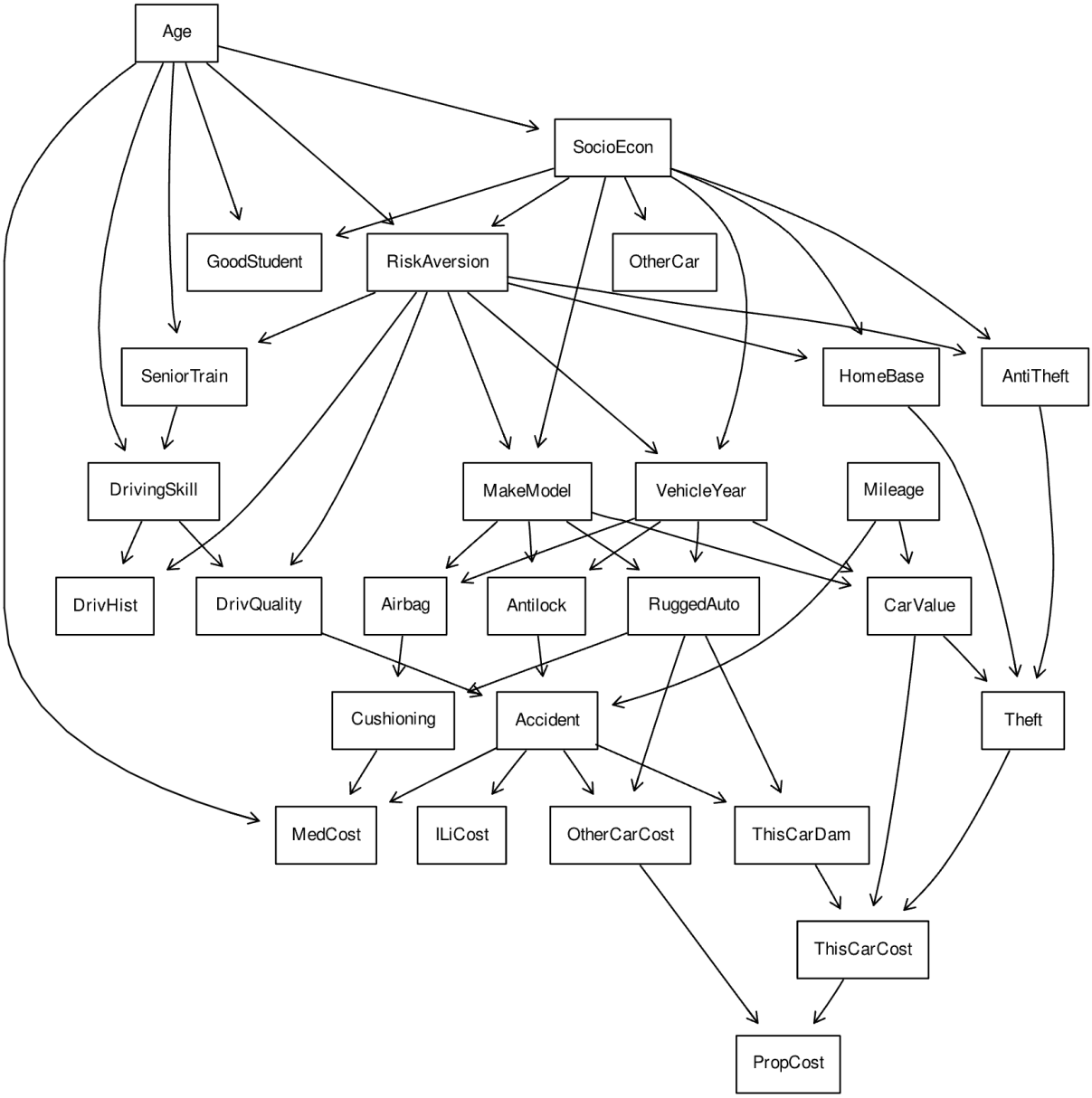


Figure 12: Insurance causal graph