CARE: Turning LLMs Into Causal Reasoning Expert

Juncheng Dong[*]
Duke University
Durham, NC
juncheng.dong@duke.edu

Yiling Liu[*]
Duke University
Durham, NC
yiling.liu@duke.edu

Ahmed Aloui
Duke University
Durham, NC
ahmed.aloui@duke.edu

Vahid Tarokh[†]
Duke University
Durham, NC
vahid.tarokh@duke.edu

David Carlson[†]
Duke University
Durham, NC
david.carlson@duke.edu

Abstract

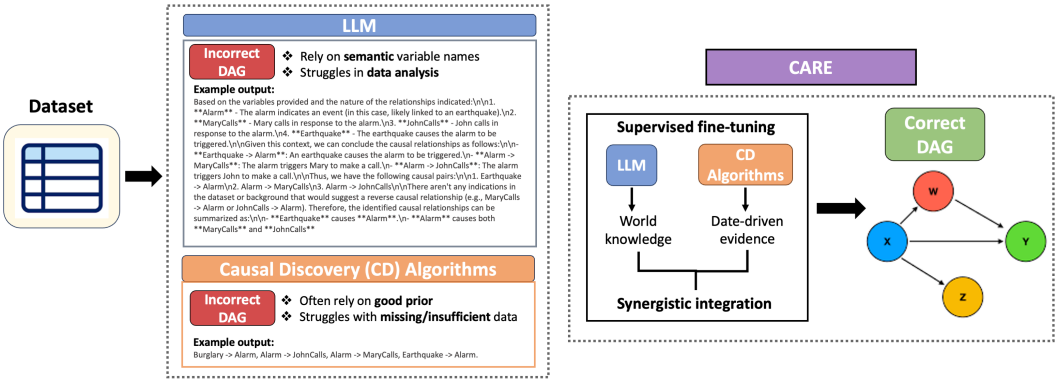
Large language models (LLMs) have recently demonstrated impressive capabilities across a range of reasoning and generation tasks. However, research studies have shown that LLMs lack the ability to identify causal relationships, a fundamental cornerstone of human intelligence. We first conduct an exploratory investigation of LLMs' behavior when asked to perform a causal-discovery task and find that they mostly rely on the semantic meaning of variable names, *ignoring the observation data*. This is unsurprising, given that LLMs were never trained to process structural datasets. To first tackle this challenge, we prompt the LLMs with the outputs of established causal discovery algorithms designed for observational datasets. These algorithm outputs effectively serve as the sufficient statistics of the observation data. However, quite surprisingly, we find that prompting the LLMs with these sufficient statistics *decreases* the LLMs' performance in causal discovery. To address this current limitation, we propose CARE, a framework that enhances LLMs' causal-reasoning ability by teaching them to effectively utilize the outputs of established causal-discovery algorithms through supervised fine-tuning. Experimental results show that a finetuned Qwen2.5-1.5B model produced by CARE significantly outperforms both traditional causal-discovery algorithms and state-of-the-art LLMs with over a thousand times more parameters, demonstrating effective utilization of its own knowledge and the external algorithmic clues.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across diverse application domains, demonstrating astonishing capabilities in medicine [32], natural language understanding [1], complex code generation [9], and even accelerating scientific discovery [37]. While these advancements reflect substantial progress in language understanding and reasoning, a critical examination reveals persistent challenges when LLMs perform causal discovery [7]. Causal discovery, which infers causal relationships and structures from data, is fundamental to scientific understanding and human reasoning [25, 31]. Despite this, there is an opportunity to use LLMs to improve traditional causal discovery algorithms that use data alone, as they can significantly benefit from external prior knowledge to reduce the hypothesis space, especially with limited data. Thus, the extensive world knowledge embedded in LLMs presents a rich resource for providing crucial guiding priors [12, 13].

[*]Equal contribution, first authors listed alphabetically.
[†]Corresponding author.

Accepted at the Foundations of Reasoning in Language Models, NeurIPS 2025 workshop in San Diego, California



(a) LLM versus Causal Discovery (CD) Algorithms

(b) Motivation of the CARE Framework

Figure 1: **Synergizing LLM Knowledge with Strengths of Causal Discovery Algorithms**. **(a)** Standalone approaches often falter: LLMs can misinterpret data due to over-reliance on pre-trained knowledge, while traditional Causal Discovery (CD) algorithms may struggle with insufficient data or without strong guiding priors, *both potentially leading to incorrect causal graphs*. **(b)** CARE uses supervised fine-tuning to *synergistically integrate* the extensive world knowledge of LLMs with the data-driven evidence from CD algorithm outputs, aiming to produce *more accurate* causal discoveries.

However, although LLMs can sometimes generate plausible narratives about potential causal links, their capacity for accurately identifying causal graphs and distinguishing direct from indirect causes based on provided data frequently falls short [4, 23]. Specifically, recent studies show that LLMs often lack genuine causal reasoning capabilities [10, 16, 36]. Rather than conducting genuine causal thinking, they frequently engage in *causal mimicry*: relying on recalling memorized facts, associative patterns, or plausible structures learned during pretraining on vast amounts of real-world causal information, and simply paraphrasing the knowledge contained in its pretraining corpora. This limitation becomes particularly apparent in data-driven tasks like causal discovery. For example, when tasked with discovering the causal relationships between INCOME, EDUCATION, SMOKING, and LIFE EXPECTANCY from a given observational dataset, LLMs mostly rely on the semantic meaning of these four variables and recite facts they have memorized during pretraining, such as "SMOKING causes LIFE EXPECTANCY" and "EDUCATION causes INCOME". This behavior is the **opposite** of conducting **genuine** causal discovery based on the given observational data as done by established causal discovery algorithms such as LiNGAM [28, 29]. In a recent work [7] and Section 4, we showcase that *LLMs fail to effectively utilize the given observation dataset*, mainly relying on the semantic meaning of variable names for causal discovery. Consequently, while performance might appear competent when variable names offer strong semantic clues, their discovered causal structures are often unreliable when genuine analysis of novel or complex data dependencies is required.

This observation is perhaps not very surprising, given that LLMs are primarily trained for next-token prediction and natural language understanding, not for interpreting structured (e.g., tabular) data. Yet, the ability to perform genuine causal discovery from data is *crucial* for applying LLMs to domains where understanding cause-and-effect is essential. To address this challenge, our key insight is to prompt LLMs with the outputs of established causal discovery algorithms, which are *inherently* designed for structured data, essentially treating these outputs as *concise and sufficient summaries of the underlying statistical evidence* in a given observation dataset. Moreover, as shown in Figure 1, these algorithmic outputs are much more aligned with natural language representations, making them significantly easier for LLMs to interpret and utilize effectively.

However, quite surprisingly, we observe that LLMs cannot effectively use the algorithm output when they are injected as part of the prompts. Empirically, including algorithm outputs even sometimes *degrades* the causal discovery performance of LLMs. To bridge these gaps, we propose CARE (CAusal Reasoning Experts), a supervised fine-tuning (SFT) framework designed to empower LLMs to perform more robust and data-grounded causal structure learning. CARE transforms pretrained LLMs by teaching them to *synergistically integrate* their extensive **world knowledge** with the **structured outputs** of established causal discovery algorithms. Our key technical contributions are twofold:

- ❖ **We investigate LLMs' capability in performing genuine causal discovery and their ability to employ algorithm outputs for self-improvement.** To this end, we consider multiple challenging scenarios (detailed in Figure 3 and Section 5.2) designed to test LLM biases, such as over-reliance on variable semantics and sensitivity to data ordering. **For example**, our benchmark includes a variable permuted version of the *EARTHQUAKE* dataset with 5 variables including EARTHQUAKE and ALARM. While both the variables names and the statistical patterns within the observation dataset strongly indicate "EARTHQUAKE causes ALARM" in the original dataset, we create a different version of the earthquake dataset by exchanging the dataset columns corresponding to EARTHQUAKE and ALARM. By doing so, the statistical patterns within the dataset suggest that "ALARM causes EARTHQUAKE" while the variable names suggest vice-versa.
- ✯ **We propose a novel SFT paradigm for improved causal discovery.** CARE is specifically designed to enhance the causal discovery performance of LLMs. Unlike approaches that rely solely on LLMs' internal knowledge or attempt direct data analysis, CARE trains LLMs to effectively interpret and utilize the structured outputs (e.g., candidate graphs, conditional independencies) generated by established causal discovery algorithms. With CARE, LLM learns to treat these algorithm outputs as sufficient evidence for data and correct these algorithms' biases with its own world knowledge. **For example**, when discovering a gene network, CARE receives the graph proposed by an established causal discovery method based on expression data and learns to reason from that graph, potentially refining it using its world knowledge. This synergistic approach allows CARE to outperform both baseline pretrained LLMs (which struggle with data-driven analysis) and often the standalone causal discovery algorithms themselves (which lack broader world knowledge for refinement), as demonstrated through our extensive experiments.

2 Related Work

Causal discovery aims to to learn causal relationships from observational and interventional data. There has been a wide array of algorithms that have been proposed, with different statistical assumptions and computational trade-offs.

Constraint-based methods. Constraint-based algorithms try to recover the causal graph by exploiting a set of conditional independence tests. For example, the PC algorithm [31] is one of the first algorithm that has been proposed and it uses conditional independence tests to (e.g., Fisher's Z-test for Gaussian data or G-square for categorical data) to infer the skeleton of a causal graph and orient edges.

Score-based methods. Score-based algorithms approach causal discovery by searching over the space of possible graphs to find the one that maximizes a predefined scoring function, which quantifies the goodness-of-fit between the graph structure G and the observed data [11]. This typically involves balancing model complexity with data likelihood to avoid overfitting. Common scores are the Akaike Information Criterion (AIC) [2], Bayesian Information Criterion (BIC), and the Bayesian Dirichlet sparse (BDs) [26]. Recent score-based methods include BOSS (Best Order Score Search) [3], which leverages Grow-shrink trees [24] for efficient exploration of the DAG space.

Functional methods. These methods restricts the space of the Structural Causal Models function class. Notably, ICA-LiNGAM [28] leverages the assumption of linear, non-Gaussian data generating processes to uniquely identify DAGs via independent component analysis. DirectLiNGAM [29] extends this by sequentially identifying exogenous variables based on their statistical independence from residuals, enabling more scalable recovery of DAGs in practice.

Causal Discovery and LLMs The emergence of transformer-based architectures [33] has led to a revolution in NLP and general AI capabilities. Modern LLMs, trained on massive text corpora, exhibit emergent reasoning capabilities and have shown promise in various symbolic and statistical reasoning tasks. Recent efforts have investigated whether LLMs can complement or even replace classical causal discovery tools. Some works explore LLMs' capacity to understand causal relationships in language [4, 16], revealing partial causal reasoning but limited structural understanding. Others aim to explicitly use LLMs to generate or assist with causal graphs [23, 15], bridging data-driven methods with prior knowledge encoded in language. Critically, debates remain as to whether LLMs can internalize or reason over structural causal models [10].

While LLMs show promise in democratizing access to causal tools, their integration with traditional causal discovery algorithms remains an open frontier. Recent work [19] categorizes the role of LLMs in causal discovery into three main paradigms: (a) direct inference from metadata, (b) posterior

refinement of graphs produced by traditional algorithms, and (c) integration of LLM-derived priors into the discovery process. Within the direct inference paradigm, two primary approaches have emerged. The first focuses on causal order prediction using direct LLM queries combined with chain-of-thought prompting [35, 18, 34]. The second aims at full or partial causal graph discovery through iterative pairwise comparisons [23]. However, this latter approach can incur substantial computational overhead, which has been addressed in more recent work on efficient inference strategies [17, 30]. In the posterior correction literature, LLMs are used as expert judges to refine and correct the outputs of traditional causal discovery algorithms. Given the set of all conditional independence relations, [22] proposes a method that treats the LLM as an imperfect expert to progressively narrow down the set of possible causal structures within the Markov equivalence class, while mitigating the risk of incorrect edge orientations. However, its scalability to larger datasets remains unproven, likely due to the method's computational complexity and design limitations. LLMs can also be used to provide prior knowledge by extracting metadata from textual descriptions or domain-specific sources. One approach feeds variable descriptions to an LLM to infer direct causal relationships based on semantic understanding [5]. Prior knowledge is typically incorporated using either hard or soft constraints. Hard constraints enforce strict exclusions of edges, but lack flexibility, errors in prior assumptions cannot be corrected during learning. To address this, some methods propose mechanisms for detecting and correcting flawed LLM-derived priors [8].

3 Preliminary

3.1 Causal Discovery

In this section, we briefly review the causal discovery problem from a Structural Causal Model perspective [25]. Let V denote a finite set of random variables, and $G = (V, E)$ be a directed acyclic graph (DAG), where an edge $(i, j) \in E$ encodes a direct causal influence from variable X_i to X_j. Each node $X_i \in V$ is generated according to a structural causal model (SCM):

$$X_i = f_i(\mathrm{PA}_i, N_i),$$

where $\mathrm{PA}_i \subseteq V \setminus \{X_i\}$ denotes the parents of X_i in G, and N_i is an exogenous noise variable independent of all other noise terms. Under standard assumptions, this implies a factorization,

$$P(X) = \prod_{i=1}^{|V|} P(X_i \mid \mathrm{PA}_i). \tag{1}$$

The goal of causal discovery is to identify the true DAG, i.e., edges in E, from an observation dataset containing sampled according to the data generating process (DGP) specified above.

Evaluation Metrics To assess the quality of the learned causal graph, we treat each edge in the ground-truth set E as a target to be predicted, and compare it against the predicted edge set $\hat{E}$. We choose the standard **F1 Score** as our main evaluation metrics, defined as

$$\mathrm{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

where Precision $= \frac{|\hat{E} \cap E|}{|\hat{E}|}$, and Recall $= \frac{|\hat{E} \cap E|}{|E|}$.

3.2 Large Language Models and Supervised Fine-Tuning

LLMs are transformer-based architectures trained on large-scale text corpora using the objective of next-token prediction. Formally, given a sequence of tokens $x = (x_1, x_2, \ldots, x_T)$ and model parameters θ, the model is trained to maximize the likelihood,

$$\max_\theta \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}), \tag{2}$$

where $x_{<t}$ represents the sequence prior to time step t. This autoregressive training objective enables LLMs to model complex dependencies in natural language and generate coherent text continuations.

Despite their impressive generalization capabilities, pretrained LLMs are often misaligned with task-specific goals or user preferences. To address this, a common practice is *Supervised Fine-Tuning*

(SFT), where a pretrained model is optimized on instruction-response pairs $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $x^{(i)}$ is typically a prompt or instruction, and $y^{(i)}$ is the desired output. The SFT objective is,

$$\max_\theta \sum_{i=1}^{N} \log p_\theta(y^{(i)} \mid x^{(i)}), \tag{3}$$

SFT refines the model's behavior to better follow human instructions, generate accurate task-specific outputs, and align with downstream usage scenarios. It plays a central role in adapting general-purpose LLMs to specialized domains such as reasoning, programming, and scientific discovery.

4 Prompting LLMs with Causal Discovery Algorithm Outputs

Motivation. We begin by asking whether *prompting alone* can boost LLMs' causal discovery capabilities. Concretely, we study two questions:

- **Q1.** Does providing more observational data improve LLMs' ability to recover the underlying causal graph?
- **Q2.** Can LLMs, *without* any additional finetuning, effectively leverage the outputs of established causal discovery algorithms when included in the prompt?

In particular, answering **Q1** tells us whether LLMs can directly transform raw observational data into causal structure. Answering **Q2** tests their capacity to *integrate and refine* external algorithmic knowledge. In particular, we are interested in whether LLMs can correct their and/or algorithmic biases to achieve a *combined* effect where *LLM+algorithm* outperforms either component alone.

Experiments Setup. We evaluate an array of LLMs: **(i)** *Qwen2.5-1.5B*, **(ii)** *gpt-4.1-mini*, **(iii)** *gpt-4o-mini*, and **(iv)** *o4-mini*, using the classic *ASIA* benchmark with 8 variables.

To address **Q1**, we query the LLMs with prompts containing observational data samples drawn from the ASIA data distribution with increasing sample sizes $N \in \{0, 50, 200\}$. Performance is evaluated based on the F1-score comparing the LLM-generated causal graph against the ground truth.

To address **Q2**, we first run a collection of established causal discovery algorithms with the observational data and collect their algorithm outputs including an estimated causal graph. We consider an assortment of representative algorithms: `PC`, `GES`, `ICA-LiNGAM`, `DirectLiNGAM`, `FCI`, `GRaSP`, `BOSS`. Details of these algorithms are discussed in Section 2.

For each sample size $N \in \{0, 50, 200\}$, we conduct a controlled comparison between two conditions: (a) LLMs prompted *with* the algorithm outputs and (b) LLMs prompted *without* them.

Controlling for Prior Knowledge in Variable Semantics. A critical consideration when evaluating LLMs' ability to perform genuine causal analysis is their potential reliance on pre-existing knowledge of variable semantics [7]. To isolate this factor and rigorously assess the models' capacity for data-driven reasoning, we design experiments under three **variable naming conditions**, each presenting a distinct challenge:

- ***Original*** uses the standard, semantically meaningful names from the ASIA dataset, e.g., VISITASIA, SMOKING, TUBERCULOSIS, etc. *This allows LLMs to leverage their equipped world knowledge through variable name semantics*.
- ***Non-semantic (Random)*** replaces each variable name with a unique, non-semantic identifier consisting of randomly sampled characters (e.g., XRAY $\rightarrow$ 4NWS). This ensures variable names carry no semantic meaning, *forcing the LLMs to rely solely on the data structure and, if applicable, provided algorithm outputs*.
- ***Permuted*** shuffles variable names while leaving the underlying dataset matrix untouched; hence, we remove any reliable linguistic hints that a model might rely on. Moreover, this creates a challenging scenario for LLMs, as they need to *counteract the biases associated with the semantic knowledge*. Success in this condition demands reasoning purely from the observation data while refraining from the potentially misleading variable names.

Results. We summarize the key results in Table 1, with several important observations:

- **O1. Reliance on semantic priors.** With original, semantically meaningful variable names in Table 1, LLMs often achieve *high F1-scores*, *even when with **no** observation data* nor algorithm outputs. This suggests dominant reliance on memorized semantic associations rather than data-driven analysis.

O2. Performance collapse without semantic hints. When variable names are rendered non-informative—either via permuted variable names or non-semantic random names in Table 1, *LLM performance plummets*, particularly when without algorithm outputs (often near zero F1-score). This highlights their inability to perform causal discovery from data patterns alone.

O3. Ineffective use of observational data (Q1). In semantically-neutral conditions (permuted variable names and non-semantic names), providing or *increasing* observational data (N=0 to N=200) offers *minimal and inconsistent improvement* – LLMs do not effectively learn from raw data via prompting alone.

O4. Limited and non-synergistic use of algorithm outputs (Q2). While providing algorithm outputs improves performance over data-only prompting (without algorithm) in semantically-neutral settings, LLM performance typically only approaches or lags behind the input algorithm's standalone performance (e.g., Table 1, LLM scores mostly < 0.4 vs. Algorithm Performance 0.413). There's no clear evidence of LLMs refining or synergistically improving upon these outputs. With original names, algorithm outputs can even *degrade* LLM performance if they conflict with LLMs' strong semantic priors.

Table 1: Performance with and without Algorithm Outputs (F-1 score).

Model	Conditions	Original			Random			Permuted		
		N=0	N=50	N=200	N=0	N=50	N=200	N=0	N=50	N=200
Algorithm Performance		–	–	0.362	–	–	0.413	–	–	0.361
Qwen2.5-1.5B	with Algo	0.609	0.496	0.423	0.327	0.328	0.344	0.357	0.300	0.338
	w/o Algo	0.554	0.404	0.351	0.027	0.143	0.053	0.085	0.074	0.063
GPT-4.1-mini	with Algo	0.812	0.858	0.840	0.314	0.315	0.314	0.132	0.137	0.127
	w/o Algo	0.977	0.988	0.984	0.000	0.175	0.209	0.126	0.099	0.107
GPT-4o-mini	with Algo	0.461	0.497	0.568	0.381	0.380	0.376	0.353	0.367	0.374
	w/o Algo	0.688	0.693	0.637	0.000	0.129	0.132	0.133	0.136	0.125
o4-mini	with Algo	0.986	0.994	0.994	0.293	0.219	0.271	0.164	0.152	0.152
	w/o Algo	1.000	0.996	1.000	0.000	0.078	0.052	0.160	0.149	0.153

Summary. This exploratory investigation suggests that prompting LLMs for causal discovery is highly unreliable when semantic meanings are unavailable. LLMs primarily use variable name semantics, struggle to analyze observational data effectively, and do not consistently or synergistically enhance outputs from established causal discovery algorithms. The absence of robust data-driven reasoning and critical engagement with algorithmic outputs highlight *the inadequacy of prompting alone for this complex task*. These findings motivates us to use SFT to teach LLMs the specific skills required for genuine causal discovery from diverse information sources.

5 CARE: Transforming LLMs into Causal Reasoning Experts

Given the limitations of prompting identified in Section 4, particularly the surprising finding that providing outputs from strong causal discovery algorithms can decrease LLM performance, we aim for an approach to transform LLMs into robust causal reasoning experts via SFT. To this end, we propose CARE (CAusal Reasoning Experts). See Figure 2 for the workflow of our framework.

5.1 Framework Overview

Core Objective. CARE trains an LLM to effectively combine its internal semantic knowledge with external algorithm outputs, specifically for the purpose of discovering more accurate causal graphs. Specifically, we leverage SFT to teach LLMs how to synthesize these diverse information sources while ensuring resilience against data variations.

Training objective. The LLM is trained to output a *refined causal graph structure* that represents the optimal synthesis of the LLM's internal priors and external algorithmic evidence.

SFT Instances. Our SFT procedure follows the standard instruction-following template with each SFT instance consisting of an *instruction* and a *target*. The **instruction** includes **(a)** the context or query defining the discovery task, e.g., "Find the causal graph for variables V1, V2,...”; **(b)**

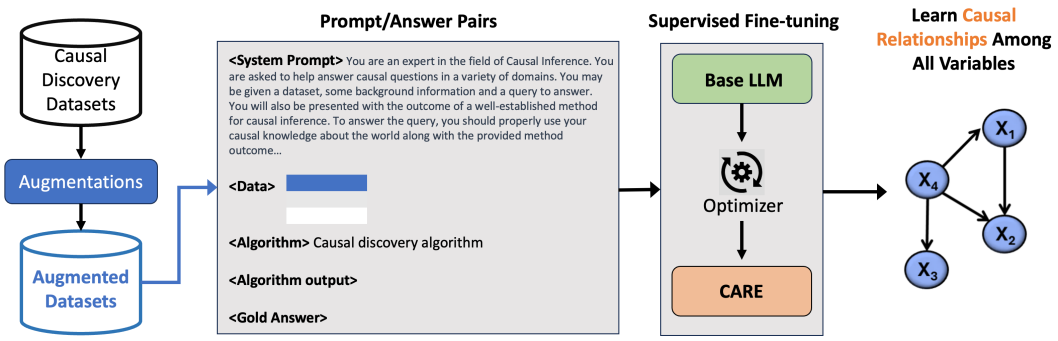


Figure 2: **CARE Framework Overview.** CARE processes causal discovery datasets, augments them to create *diverse training scenarios* (**left**), and uses these to construct *prompt/answer pairs* (**middle-left**). These pairs are then used for *supervised fine-tuning* (**middle-right**), enabling LLMs to learn and output accurate causal relationships among variables (**right**).

an observational dataset; and **(c)** the output graph structures from one or more established causal discovery algorithms (e.g., PC, GES) applied to the dataset. The **target** includes the ground-truth causal graph corresponding to the synthetic data scenario. By training against this target, LLMs autonomously learn *the optimal strategy for weighting and integrating* the semantic context and the provided algorithm outputs to best recover the true structure. We next elaborate on our procedure for generating the SFT data, a central challenge to ensure that LLMs generalize.

5.2 Data Augmentation Strategy for Robust Learning.

Motivation. The findings in Section 4 demonstrate that *prompting alone is insufficient* to elicit robust, data-driven causal discovery from LLMs. Specifically, LLMs tend to over-rely on variable semantics and struggle to effectively integrate observational data or external algorithmic outputs. To transcend these limitations, SFT within our CARE framework requires a training set that actively *challenges these ingrained biases and teaches genuine evidence integration*. In particular, an effective SFT requires a large and diverse dataset of causal discovery problems where combining internal knowledge and external algorithm outputs is beneficial. To this end, we employ a collection of carefully designed data augmentations.

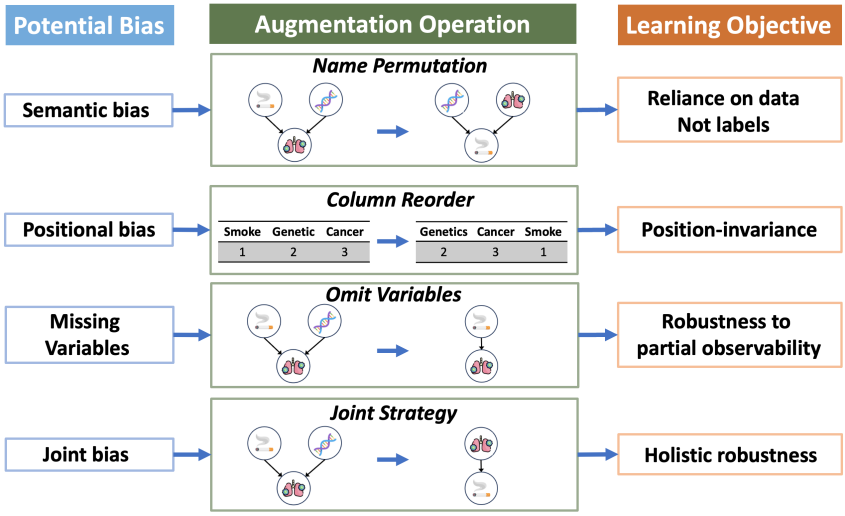


Figure 3: **Purposes of Various Augmentation Methods.** Each row illustrates a potential LLM *bias* (**left**), the corresponding *augmentation* operation designed to address it (**middle**), and the intended *learning objective* for the model (**right**).

Concretely, our SFT data generation strategy is designed to ensure that the LLMs learn to:

- **A1.** Reason from statistical structure *independent of misleading or absent* semantic cues.
- **A2.** Maintain performance irrespective of superficial data presentation characteristics like *column ordering*.
- **A3.** Perform robust discovery even with *incomplete information* due to unobserved variables.
- **A4.** Synergistically combine its internal knowledge with external algorithmic evidence, especially under *complex, multifaceted perturbations*.

To achieve (**A1-A4**), we rely on various data augmentation methods for SFT (see Figure 3). Each augmentation is designed for a specific bias, fostering *more generalizable* causal reasoning. Specifically, we use four types of augmentation, one for each of **A1-A4**:

- **Semantic Variable Name Permutation [A1]:** Original semantic variable names are shuffled; data matrix remain unchanged.
- **Column Order Randomization [A2]:** Variable column sequence is permuted.
- **Variable Omission [A3]:** Remove subset of the variables and update the ground-truth causal graph to reflect the marginalized structure over the remaining variables.
- **Perturbation Combination [A4]:** Compositions of above to ensure holistic robustness. This collection evaluates robustness using A2 as a foundational perturbation, either alone or in combination with other perturbations. Based on *how the perturbations affect ground truth*, the specific scenarios are (i) *Original:* the foundational scenario applying only perturbation A2; (ii) *Omitted:* combining perturbation A3 with the foundational A2; (iii) *Permuted:* combining perturbation A1 with the foundational A2.

By training `CARE` across this diverse corpus with thousands of augmented datasets derived from each benchmark dataset, we teach the model to learn *robust and generalizable causal knowledge*.

A Bayesian Perspective on SFT. We view the pretrained LLM's background knowledge as a *prior* over causal structures: it encodes statistical and semantic associations learned from large-scale text corpora, which can implicitly inform causal relationships. During SFT, this prior is updated into a *posterior* by incorporating new evidence, namely, observational data and the outputs of classical causal discovery algorithms. In this way, the LLM learns to integrate its internal knowledge with relevant data-driven clues, resulting in more accurate and grounded causal discovery predictions.

6 Experiments

We empirically evaluate the effectiveness of our `CARE` framework in enhancing the causal discovery capabilities of LLMs through Supervised Fine-Tuning (SFT). Our fine-tuned model is benchmarked against strong baseline LLMs and the performance of traditional causal discovery algorithms.

Training Mechanism. To manage computational resources, we utilize Parameter-Efficient Fine-Tuning (PEFT), specifically Low-Rank Adaptation (LoRA) [14], enabling efficient adaptation of large base models by training only a small subset of their parameters. In addition, PEFT can help mitigate overfitting and reduce the risk of catastrophic forgetting in LLMs.

6.1 Evaluation Strategy

To evaluate the performance of our fine-tuned models, we employ an LLM-as-a-judge approach, leveraging a capable model for automated assessment. Here, we use `GPT-4.1-mini`. The evaluation process for each test instance is as follows::

1. **Structured comparison:** The evaluation LLM receives the ground-truth causal graph and the graph predicted by the model under test. It is instructed via a detailed prompt (see Appendix C.1) to parse directed edges from both, handling formatting variations and prioritizing specifically marked answer sections in model outputs.
2. **Edge metrics calculation:** The judge LLM identifies and counts true positives (TP, correctly recovered edges), false negatives (FN, missed true edges), and false positives (FP, incorrectly predicted extra edges).
3. **Performance metric:** From the TP, FP, and FN counts, we calculate the F1-score, a standard metric for graph discovery accuracy.

This automated pipeline ensures consistent evaluation across different models, datasets, and augmentation types.

6.2 Experimental Setup

Datasets. Our experiments utilize established benchmark networks from the `bnlearn` repository[3]. These include: *ASIA* (8 nodes)[21], *SURVEY* (6 nodes)[27], *EARTHQUAKE* (5 nodes)[20], and the larger *ALARM* network (37 nodes)[6]. Each network provides a known ground-truth Directed Acyclic Graph (DAG), allowing for precise quantitative assessment. Further details on these datasets are in Appendix B.

Synthetic observational data is sampled from the true conditional probability distributions of these networks. This data serves as the foundation for our augmentation strategy (detailed in Section 5.2 and Figure 3), which applies Name Permutation, Column Reordering, Variable Omission, and Joint Perturbations to create a diverse and challenging corpus for robust training and testing.

A crucial aspect of our evaluation methodology is that for all test scenarios, we generate ***new, unseen observational datasets*** sampled from the ground-truth graphs. While in-distribution evaluation utilizes graph structures encountered during SFT, testing on these distinct data realizations ensures we assess genuine generalization to novel data instances rather than mere memorization of the training samples. This significantly increases the credibility of our assessment.

Benchmark Models. We evaluate the following models:

- **Baseline LLMs:** We directly prompt the base `Qwen2.5-1.5B`, `gpt-4.1-mini`, `gpt-4o-mini`, `o4-mini` with algorithm outputs *with no additional fine-tuning*.
- **Algorithm:** The F1-score achieved by the collection of traditional causal discovery algorithms whose outputs serve as input to the LLMs. This acts as a non-LLM baseline.
- **CARE:** Our *fine-tuned* `Qwen2.5-1.5B` model, using the SFT strategy described in Section 5.

6.3 Results and Analysis

The main results comparing our fine-tuned `Qwen2.5-1.5B` model (CARE) against baselines are presented in Table 2. We analyze performance across the four benchmark datasets under three augmentation conditions: Original variable names, Omitted variables, and Permuted variable names as detailed in Section 5.2. We demonstrate that our framework CARE achieves state-of-the-art performance compared to baselines.

Table 2: F1 score on benchmark datasets. Higher is better. Best results per condition are bolded.

	Asia			Survey			Earthquake			Alarm		
Model	Original	Omitted	Permuted	Original	Omitted	Permuted	Original	Omitted	Permuted	Original	Omitted	Permuted
Algorithm	0.362	0.371	0.361	0.282	0.299	0.271	0.419	0.481	0.385	0.301	0.314	0.326
GPT-4.1-mini	0.858	0.824	0.137	0.268	0.269	0.269	**1.000**	0.979	0.220	0.425	0.395	0.446
GPT-4o-mini	0.497	0.380	0.367	0.313	0.308	0.293	0.447	0.549	0.437	0.420	0.458	0.529
o4-mini	0.994	0.978	0.152	0.319	0.301	0.246	**1.000**	0.971	0.242	0.487	0.507	0.415
Qwen-2.5-1.5B	0.496	0.336	0.300	0.306	0.293	0.268	0.397	0.475	0.404	0.400	0.362	0.409
CARE	**1.000**	**0.991**	**0.460**	**0.979**	**0.997**	**0.792**	0.953	**1.000**	**0.623**	**0.990**	**0.971**	**0.618**

7 Conclusion

We introduced CARE, an SFT framework designed to overcome LLMs' inherent limitations in causal discovery. By fine-tuning with diverse data augmentations that challenge semantic biases and promote data-driven analysis, CARE enables LLMs to effectively synthesize their world knowledge with outputs from established causal algorithms. Our results demonstrate that a CARE-finetuned Qwen2.5-1.5B model achieves state-of-the-art performance, surpassing both traditional methods and larger LLMs on challenging causal discovery tasks, particularly when variable names are permuted or information is partial. This highlights the efficacy of targeted SFT in building more capable and reliable causal reasoning agents from LLMs.

[3] `https://www.bnlearn.com/bnrepository/`

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 2003.

[3] B. Andrews, J. Ramsey, R. Sanchez Romero, J. Camchong, and E. Kummerfeld. Fast scalable and accurate discovery of dags using the best order score search and grow shrink trees. *Advances in neural information processing systems*, 36:63945–63956, 2023.

[4] S. Ashwani, K. Hegde, N. R. Mannuru, D. S. Sengar, M. Jindal, K. C. R. Kathala, D. Banga, V. Jain, and A. Chadha. Cause and effect: can large language models truly understand causality? In *Proceedings of the AAAI Symposium Series*, volume 4, pages 2–9, 2024.

[5] T. Ban, L. Chen, X. Wang, and H. Chen. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data. *arXiv preprint arXiv:2306.16902*, 2023.

[6] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. 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, 1989.

[7] H. Cai, S. Liu, and R. Song. Is knowledge all large language models needed for causal reasoning? *arXiv preprint arXiv:2401.00139*, 2023.

[8] L. Chen, T. Ban, X. Wang, D. Lyu, and H. Chen. Mitigating prior errors in causal structure learning: Towards llm driven prior knowledge. *arXiv preprint arXiv:2306.07032*, 2023.

[9] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[10] H. Chi, H. Li, W. Yang, F. Liu, L. Lan, X. Ren, T. Liu, and B. Han. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670, 2024.

[11] D. M. Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[12] V.-A. Darvariu, S. Hailes, and M. Musolesi. Large language models are effective priors for causal graph discovery. *arXiv preprint arXiv:2405.13551*, 2024.

[13] U. Hasan and M. O. Gani. Kcrl: A prior knowledge based causal discovery framework with reinforcement learning. In *Machine Learning for Healthcare Conference*, pages 691–714. PMLR, 2022.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[15] H. Jiang, L. Ge, Y. Gao, J. Wang, and R. Song. Llm4causal: Democratized causal tools for everyone via large language model. In *First Conference on Language Modeling*, 2023.

[16] Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. Lyu, K. Blin, F. Gonzalez Adauto, M. Kleiman-Weiner, M. Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:31038–31065, 2023.

[17] T. Jiralerspong, X. Chen, Y. More, V. Shah, and Y. Bengio. Efficient causal graph discovery using large language models. *arXiv preprint arXiv:2402.01207*, 2024.

[18] E. Kiciman, R. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2023.

[19] E. Kiciman, R. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=mqoxLkX210. Featured Certification.

[20] K. B. Korb and A. E. Nicholson. *Bayesian artificial intelligence*. CRC press, 2010.

[21] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

[22] S. Long, A. Piché, V. Zantedeschi, T. Schuster, and A. Drouin. Causal discovery with language models as imperfect experts. *arXiv preprint arXiv:2307.02390*, 2023.

[23] S. Long, T. Schuster, and A. Piché. Can large language models build causal graphs? *arXiv preprint arXiv:2303.05279*, 2023.

[24] D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. *Advances in neural information processing systems*, 12, 1999.

[25] J. Pearl. *Causality*. Cambridge university press, 2009.

[26] M. Scutari. An empirical-bayes score for discrete bayesian networks. In *Conference on probabilistic graphical models*, pages 438–448. PMLR, 2016.

[27] M. Scutari and J.-B. Denis. *Bayesian networks: with examples in R*. Chapman and Hall/CRC, 2021.

[28] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

[29] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvarinen, Y. Kawahara, T. Washio, P. O. Hoyer, K. Bollen, and P. Hoyer. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research-JMLR*, 12(Apr):1225–1248, 2011.

[30] A. Sokolov, F. Sabelli, W. Li, L. A. Seco, et al. Towards automating causal discovery in financial markets and beyond. *Behzad and Li, Wuding and Seco, Luis A., Towards Automating Causal Discovery in Financial Markets and Beyond (December 27, 2023)*, 2023.

[31] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

[32] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[35] M. Willig, M. Zečević, D. S. Dhami, and K. Kersting. Can foundation models talk causality? *arXiv preprint arXiv:2206.10591*, 2022.

[36] M. Zečević, M. Willig, D. S. Dhami, and K. Kersting. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067*, 2023.

[37] Y. Zhang, X. Chen, B. Jin, S. Wang, S. Ji, W. Wang, and J. Han. A comprehensive survey of scientific large language models and their applications in scientific discovery. *arXiv preprint arXiv:2406.10833*, 2024.

A Supervised Fine-tuning

Here we discuss the Supervised Fine-tuning (SFT) used within the CARE framework. As established in Section 4, prompting LLMs directly for causal discovery tasks, even with the inclusion of observational data or outputs from established causal discovery algorithms, yields unreliable performance and does not effectively address their inherent biases (e.g., over-reliance on variable name semantics). SFT within CARE is therefore crucial for explicitly teaching the LLM to:

- Robustly reason about causal structures from diverse data presentations.
- Synergistically integrate its internal world knowledge with the evidence provided by external causal discovery algorithms.
- Mitigate common biases observed in pre-trained models when faced with causal tasks.

The SFT process involves data preparation by using the augmentation strategies detailed in Section 5.2, to structure causal problems into a conversational format. This is then followed by model training using established fine-tuning approaches.

A.1 System Prompts for SFT

Here we use two distinct system prompts to set the context and expected behavior for the model during SFT. The choice between these prompts is determined by whether an external causal discovery method's output is provided as part of the input for a given training instance. These prompts guide the LLM on how to approach the causal discovery task, emphasizing the use of provided information.

System Prompt (with algorithm output)

You are an expert in the field of Causal Inference. You are asked to help answer causal questions in a variety of domains. You may be given a dataset, some background information and a query to answer. You will also be presented with the outcome of a well-established method for causal inference. To answer the query, you should properly use your causal knowledge about the world along with the provided method outcome. Note the outcome of the causal inference method may not always be correct. Your goal is to provide the correct answer to the query. Do NOT invent new variables. If you invent new variables, your answer will be WRONG.

System Prompt (without algorithm output)

You are an expert in the field of Causal Inference. You are asked to help answer causal questions in a variety of domains. You may be given a dataset, some background information and a query to answer. To answer the query, you should properly use your causal knowledge about the world along with the provided dataset if applicable. Your goal is to provide the correct answer to the query. Do NOT invent new variables. If you invent new variables, your answer will be WRONG.

A.2 Data Preparation for Fine-tuning

The foundation of SFT lies in generating high-quality training examples that mimic a dialogue format. Each example consists of a system directive (as detailed in Section A.1), a user query presenting a causal problem, and an ideal assistant response.

A.2.1 User Message Construction: Presenting the Causal Problem

User messages are formulated to present the causal problem to the model. This involves assembling various pieces of information from the source data:

- **Background Information and Query**: The core context and the specific causal question to be answered.

- **Dataset (Optional)**: If applicable, variable names are extracted, and a specified number of sample data rows from the dataset can be included.
- **Causal Discovery Method Output (Optional)**: If provided in the source data, the output from a causal discovery algorithm (typically a graph structure) is included, after stripping away any performance metrics or extra details.

We want to highlight that important metadata fields like `'Correct Answer'` or `'Augmentation Type'` are **intentionally excluded** from the user prompt.

A.2.2 Assistant Message Construction: Defining the Target Output

The assistant's message in each training example represents the desired output. This is constructed by taking the pre-defined correct answer for the causal problem and formatting it, typically by prefixing it with a label like `***Answer***`.

A.2.3 Tokenization and Labeling Strategy

Once the conversational messages (system, user, assistant) are constructed, they are tokenized. This process converts the text into a sequence of numerical IDs that the language model can process. A standard chat template is applied to format the entire conversation into a single input sequence.

For training, two key sequences are generated:

- `input_ids`: The tokenized representation of the full conversation.
- `labels`: These are initially a copy of the `input_ids`. A crucial step here is the option for "answer-only" training. If this mode is active, the parts of the `labels` corresponding to the system prompt and the user message (including the assistant's role identifier before its actual content) are masked (typically by setting their values to -100). This masking ensures that the model's learning (loss calculation) is focused solely on predicting the tokens of the assistant's answer, rather than the prompt itself.

A.2.4 Dataset Handling

Each data instance is then processed through the message construction and tokenization steps described previously, transforming it into the required `input_ids` and `labels` format for the trainer. The **number of data samples** included in the `Dataset` field in the user prompt is configurable.

A.2.5 Model and Tokenizer Setup

- **Tokenizer**: A tokenizer corresponding to a specified base model is loaded. It is configured with the end-of-sequence (EOS) token also serving as the padding token, and padding is applied to the left side of sequences. This is common for decoder-only models to ensure correct attention masking.
- **Model Loading**: The core causal language model is loaded from a model identifier.
  - **Quantization (Optional)**: To manage memory and potentially improve efficiency, an option exists to load the model with 4-bit quantization. This uses techniques like NF4 quantization and `bfloat16` for computation, which can be particularly beneficial for training larger models on hardware with limited memory.
  - If quantization is not enabled, the model is loaded in its standard precision.

A.2.6 Fine-tuning Strategies

Here we use **Parameter-Efficient Fine-Tuning (PEFT)** with Low-Rank Adaptation (LoRA) [14]. Specifically LoRA enables efficient adaptation of large base models by training only a small subset of their parameters. In addition, PEFT can help mitigate overfitting and reduce the risk of catastrophic forgetting in LLMs.

A.2.7 Trainer Configuration and Execution

A specialized trainer, such as the `SFTTrainer` from the TRL library, is configured with various hyperparameters. These include:

- Batch size per device and gradient accumulation steps (which together determine the effective batch size).
- Learning rate.
- Number of training epochs.
- Maximum sequence length to handle potentially long inputs.
- Settings for mixed-precision training (e.g., using FP16 or BF16) to accelerate training and reduce memory usage.

The trainer is then initialized with the prepared model (potentially adapted with LoRA), the tokenized dataset, and the training configuration. The training loop is initiated, and upon completion, the final trained model (or the LoRA adapter weights if PEFT was used) is saved.

A.3 SFT Configuration Used

For our experiments, the Supervised Fine-tuning (SFT) process utilized the `Qwen2.5-1.5B-Instruct` model as both the base for fine-tuning and for its corresponding tokenizer.

The model was trained for 30 epochs. A key aspect of this configuration was the use of a very long maximum sequence length, set to 15,000 tokens, enabling the model to process extensive contextual information within each training example. Furthermore, Parameter-Efficient Fine-Tuning (PEFT) was employed using Low-Rank Adaptation (LoRA) to optimize the training process by only updating a small subset of the model's parameters.

Other SFT training parameters, such as the learning rate, per-device batch size, gradient accumulation steps, and specific LoRA configurations (e.g., LoRA rank `r` and `lora_alpha`), were set to their default values as defined within the SFT script and the underlying TRL library configurations. For instance, the LoRA rank (`r`) was 8, and `lora_alpha` was 16. Training was performed with a batch size of 7 per device and 4 gradient accumulation steps.

B Datasets

The experiments conducted to evaluate the `CARE` framework utilize several well-established benchmark Bayesian networks. These networks provide known ground-truth Directed Acyclic Graphs (DAGs), which are essential for quantitatively assessing the performance of causal discovery methods. The primary source for these networks is the `bnlearn` repository[4]. The ground-truth DAG structures for these benchmarks are illustrated in Figures 4 through 8.

The benchmark datasets used are:

- ***ASIA***: A small Bayesian network with 8 nodes, originally proposed by Lauritzen and Spiegelhalter [21]. It represents a stylized medical diagnostic scenario involving variables such as recent travel to Asia, smoking, and respiratory diseases. The network models dependencies among factors like tuberculosis, lung cancer, bronchitis, and symptoms such as dyspnoea and X-ray results.
- ***SURVEY***: A network with 6 nodes, described by Scutari and Denis [27]. It models socioeconomic and demographic factors, including age (A), sex (S), education (E), occupation (O), residence (R), and preferred mode of transport (T). All variables are categorical.
- ***EARTHQUAKE***: A compact network with 5 nodes, often attributed to Korb and Nicholson [20]. It models a burglary direction scenario where an alarm may be triggered either by a burglary or an earthquake, and two neighbors (John and Mary) may or may not call the police based on hearing the alarm, each with a different reliability.
- ***ALARM***: A significantly larger network with 37 nodes, developed by Beinlich et al. [6]. It was designed as a diagnostic support system for patient monitoring in intensive care, encoding relationships among physiological variables, diseases, and treatments. The network includes 37 nodes spanning diagnoses, findings, and intermediate variables to facilitate probabilistic reasoning under uncertainty.

[4] `https://www.bnlearn.com/bnrepository/`

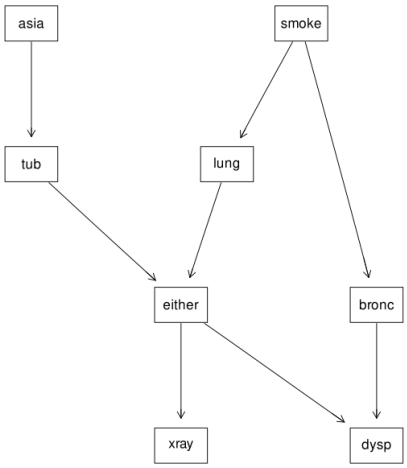


Figure 4: Ground-truth DAG for the *ASIA* network (8 nodes) [21].

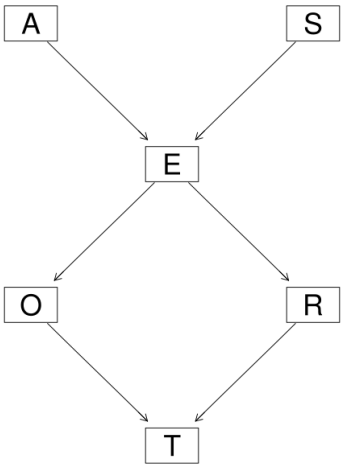


Figure 5: Ground-truth DAG for the *SURVEY* network (6 nodes) [27].

Figure 6: Ground-truth DAGs for the *ASIA* and *SURVEY* benchmark networks.

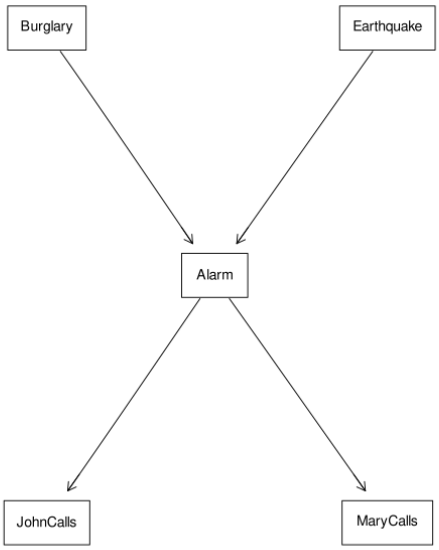


Figure 7: Ground-truth DAG for the *EARTHQUAKE* network (5 nodes) [20].

For each of these benchmark networks, their true conditional probability distributions are used to sample synthetic observational data. This sampled data forms the basis for the Supervised Fine-tuning (SFT) instances and the test scenarios. As detailed in Section 5.2, we applied various augmentation techniques (Name Permutation, Column Reordering, Variable Omission, and Joint Perturbations) to create a diverse and challenging corpus. This corpus is then used to construct the prompt-answer pairs for fine-tuning `CARE` and for evaluating its performance against baseline models.

A critical aspect of our evaluation, as mentioned in Section 6, is that for all test scenarios, **new, unseen observational datasets** are sampled from the respective ground-truth graphs. While the graph structures themselves might be encountered during SFT (in their original or augmented forms), testing

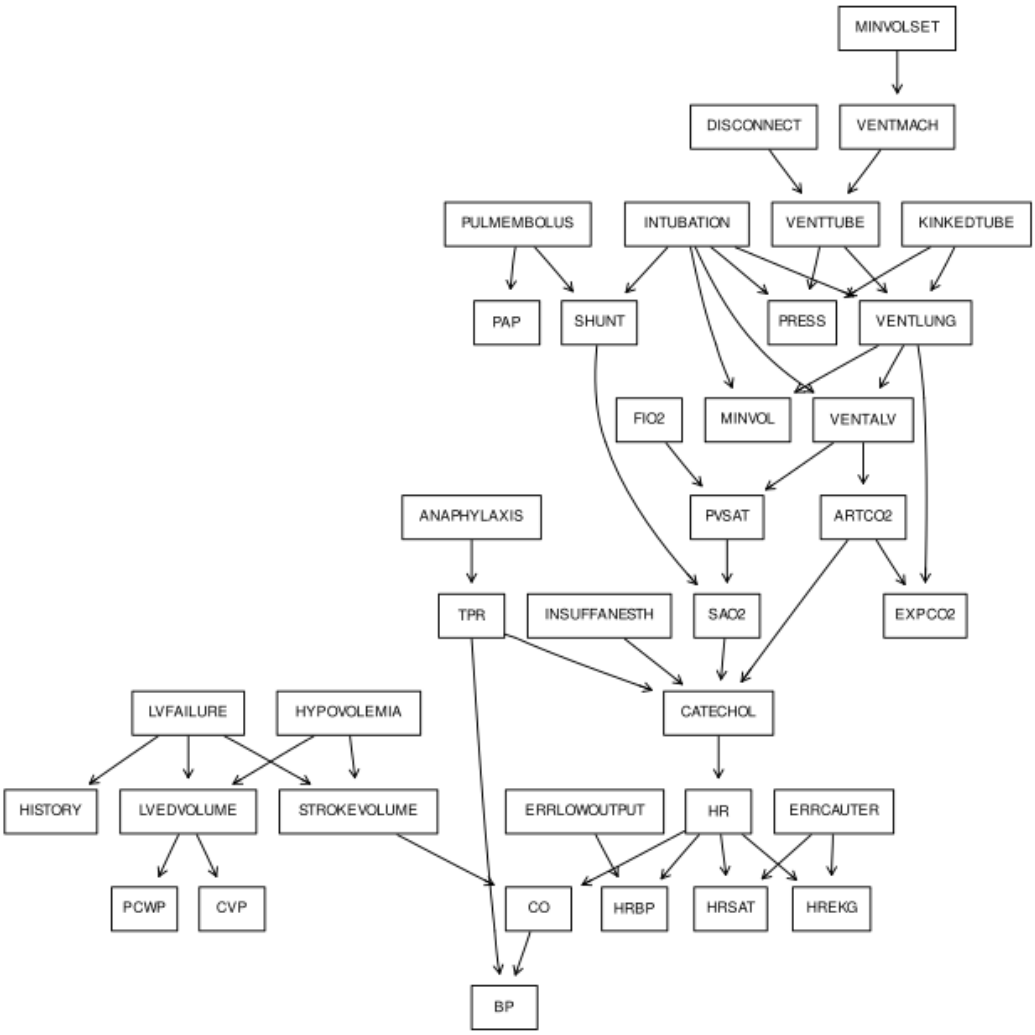


Figure 8: Ground-truth DAG for the *ALARM* network (37 nodes) [6]. A medical diagnostic network designed for patient monitoring, representing various physiological states, diseases, and measurements.

on distinct data realizations ensures that we assess the models' genuine generalization capabilities to novel data instances, rather than mere memorization of the specific training samples. This approach significantly strengthens the credibility of our performance assessments.

C Experiments Details

Here, we provide further details regarding the experimental setup, evaluation procedures, and implementation specifics for the results.

C.1 Evaluation Methodology: LLM-as-Judge

As outlined in Section 6.1, we employ an LLM-as-a-judge approach for the automated and consistent evaluation of causal graphs predicted by different models. Specifically, `GPT-4.1-mini` is utilized as the judge. The following prompt is provided to the judge LLM for each test instance. The placeholders `gold_answer_str` and `model_output_str` are replaced with the ground-truth causal graph and the predicted graph from the model under evaluation, respectively. This structured approach ensures that

the F1 score, Precision, and Recall metrics reported in the main paper are derived consistently and objectively based on the parsed edges.

C.2 Implementation Details

Baseline LLMs: The baseline LLM evaluations (Table 1 and Table 2) for `Qwen2.5-1.5B`, `gpt-4.1-mini`, `gpt-4o-mini`, and `o4-mini` were performed using their respective publicly available APIs or Hugging Face model checkpoints at the time of experimentation (specify dates/versions if crucial). Prompts were constructed as described in Section 4, including observational data (for $N = 50, 200$) and outputs from a suite of classical causal discovery algorithms.

Baseline Causal Discovery Methods: The specific algorithms whose outputs were provided include `PC`, `GES`, `ICA-LiNGAM`, `DirectLiNGAM`, `FCI`, `GRaSP`, and `BOSS`. These were executed using standard Python libraries such as `cdt` (Causal Discovery Toolbox) and `pgmpy`. For each baseline LLM, the temperature or other generation parameters were set to ensure deterministic or near-deterministic output for reproducibility (e.g., temperature set to 0 or a very low value).

C.3 Computational Resources

All Supervised Fine-tuning experiments for `CARE` and baseline LLM evaluations (where local models were run) were conducted on a compute cluster equipped with 8 NVIDIA A6000 GPUs, each with 48GB of VRAM. The specific GPU allocation per run varied depending on model size and batch configuration. For example, fine-tuning the `Qwen2.5-1.5B` model with LoRA typically utilized a single A6000 GPU. The total training time for `CARE` across all epochs and dataset augmentations was approximately 40 hours. Evaluations involving API-based models like GPT-4 series incurred costs associated with token usage.

C.4 Further Details on Augmentation Scenarios (Perturbation Combinations)

As mentioned in Section 5.2, the "Perturbation Combination" category for data augmentation, used to generate SFT instances and test scenarios, is built upon a foundational perturbation of Column Order Randomization (**A2**). This ensures the model is not sensitive to the input order of variables. The specific scenarios tested in Table 2 under the main "Original", "Omitted", and "Permuted" column groups implicitly include this foundational column reordering:

- **Original (in Table 2)**: Refers to scenarios where the original semantic variable names are used, but the *columns in the provided observational dataset are randomly reordered* (**A2**). The ground-truth DAG remains the same as the benchmark's original DAG.

- **Omitted (in Table 2)**: These scenarios start with Variable Omission (**A3**) to create a new, smaller set of variables and a corresponding marginalized ground-truth DAG. Then, the columns for these remaining variables in the observational dataset are *randomly reordered* (**A2**).

- **Permuted (in Table 2)**: These scenarios apply Semantic Variable Name Permutation (**A1**), where the names are shuffled but the data matrix initially is not. Crucially, after this name permutation, the columns in the observational dataset (now associated with misleading names) are also *randomly reordered* (**A2**). This creates a particularly challenging scenario where the model must discern true relationships despite both misleading semantic cues and arbitrary data ordering. The ground-truth DAG corresponds to the true relationships in the underlying (now reordered) data, not the permuted names.

This **consistent application of column reordering across conditions** ensures that improvements **are not due to the model learning spurious positional clues**.

LLM-as-Judge Evaluation Prompt

Analyze the following causal relationship information and provide a structured comparison in JSON format.
Ground Truth Causal Relationships (Gold Standard): gold_answer_str
Model Generated Output: model_output_str
Instructions:

1. **Parse the Gold Standard:** Identify all individual directed causal edges (e.g., `"A -> B"`, `"C <- D"` which is equivalent to `"D -> C"`) from the Gold Standard section. List them using only `"->"` notation.

2. **Parse the Model Output:** Carefully extract all causal relationships from the text block labeled "Model Generated Output". Look for patterns like `"X -> Y"`, `"X <- Y"` (which is equivalent to `"Y -> X"`), `"X causes Y"`, or `"X leads to Y"`. Consider both explicit statements like "A causes B" and structured lists (with bullet points, numbered items, etc.). Ignore introductory phrases, confidence scores, apologies, or explanations unless they define relationships. Convert all relationships to `"X -> Y"` notation - for example `"X <- Y"` should be recorded as `"Y -> X"`. Extract all relationships you find, even if they form cycles (like A->B->C->A). While proper causal graphs should be acyclic, faithfully report what the model output contains.
   CRITICAL INSTRUCTION: If the text contains a section marked with `"***Answer:***"` or similar indicator (like `"Final Answer:"`, `"Answer:"`, etc.), ONLY extract edges from that section and IGNORE all other parts of the text. The relationships listed in this final answer section are the definitive edges that should be extracted, superseding any relationships mentioned earlier in the text.

3. **Compare:** Compare the set of edges extracted from the model output against the set of edges from the Gold Standard.

4. **Calculate Metrics:**
   - `recovered_edges`: List the edges present in *both* the Gold Standard and the Model Output.
   - `missed_edges`: List the edges present in the Gold Standard but *missing* from the Model Output.
   - `extra_edges`: List the edges present in the Model Output but *not* in the Gold Standard.
   - `recovered_count`: Count of `recovered_edges`.
   - `missed_count`: Count of `missed_edges`.
   - `extra_count`: Count of `extra_edges`.

5. **Format Output:** Return *only* a single JSON object containing the following keys: `gold_edges_parsed`, `model_edges_extracted`, `recovered_edges`, `missed_edges`, `extra_edges`, `recovered_count`, `missed_count`, `extra_count`. Ensure edge representation is consistent (e.g., always use `"->"`).

Example Edge Parsing:

- `"A -> B"` is one edge.
- `"C <- D"` should be treated as `"D -> C"`.
- `"E -> F -> G"` implies edges `"E -> F"` and `"F -> G"`.
- `"- H -> I"` in a list implies edge `"H -> I"`.
- Phrases like `"X causes Y"` should be converted to `"X -> Y"`.
- A statement that `"X influences Y and Z"` implies both `"X -> Y"` and `"X -> Z"`.

D Limitations and Discussions

While CARE demonstrates strong performance with a 1.5B parameter model, we identify several directions for future exploration and potential improvements:

- **Impact of Base Model Scale and Computational Resources:** Our current study utilized the Qwen2.5-1.5B model, guided by available computational resources. Applying the CARE SFT methodology to larger base LLMs (e.g., 7B, 70B, or beyond) is anticipated to yield further substantial gains in causal reasoning capabilities, leveraging their inherently greater knowledge and capacity. Exploring this scaling is a promising direction for future work.
- **Scalability to Extremely Large and Complex Graphs:** While effective on benchmarks like ALARM (37 nodes), investigating CARE's performance on problems with extremely large numbers of variables (hundreds or thousands), especially when combined with larger base models as mentioned above, remains an important area. This includes managing the complexity of algorithmic outputs and potential LLM context window considerations.
- **Resource Considerations for SFT and Data Curation:** Effective SFT, as employed by CARE, benefits from a diverse corpus of training instances. The process of generating and curating these augmented datasets, alongside the general computational costs of SFT (even with PEFT), are practical considerations for adapting CARE to new, specialized domains.