Causal-LLM: A Unified One-Shot Framework for Prompt- and Data-Driven Causal Graph Discovery

Anonymous ACL submission

Abstract

Current causal discovery methods using Large Language Models (LLMs) often rely on pairwise or iterative strategies, which fail to capture global dependencies, amplify local biases, and reduce overall accuracy. This work introduces a unified framework for one-step full causal graph discovery through: (1) **Prompt-based discovery** with in-context learning when node metadata is available, and (2) **Causal_llm**, a data-driven method for settings without metadata. Empirical results demonstrate that the prompt-based approach outperforms state-of-the-art models (GranDAG, GES, ICA-LiNGAM) by approximately 40% in edge accuracy on datasets like Asia and Sachs, while maintaining strong performance on more complex graphs (ALARM, HEPAR2). Causal_llm consistently excels across all benchmarks, achieving 50% faster inference than reinforcement learning-based methods and improving precision by 25% in fairness-sensitive domains such as legal decision-making. We also introduce two domain-specific DAGs—one for bias propagation and another for legal reasoning under the Bhartiya Nyaya Sanhita—demonstrating LLMs' capability for systemic, real-world causal discovery.

1 Introduction

"LLMs are good at manipulating language, but not at thinking."

— Yann LeCun

Large Language Models (LLMs) have demonstrated remarkable linguistic proficiency, yet their ability to perform structured reasoning—particularly in causal discovery—remains largely unexplored. Current methods rely on pairwise or iterative approaches, which fragment systemic interactions, propagate local biases, and fail to capture higher-order dependencies. These limitations lead to error accumulation, computational inefficiencies, and reduced accuracy in causal inference.

This raises a fundamental question:

Can LLMs Discover Full Causal Graphs in One Step?

We address this challenge by introducing a unified framework that leverages:

• **Prompt-based full-graph discovery**: Utilizing in-context learning (ICL) when node metadata is available (refer Section 3.1).

• **Data-driven causal modeling (causal_llm)**: Extracting causal structures directly from data when metadata is absent (refer Section 3.2).

Empirical results demonstrate that the prompt-based method significantly outperforms existing causal discovery models in datasets like **Asia, Lucas, and Sachs**, achieving higher true positives per nonzero (TP/NNZ) and maintaining low false discovery rates (FDR). As the number of nodes increases (**ALARM, HEPAR2**), its performance declines but remains competitive (refer Section 4.4). Conversely, our data-driven causal_llm model consistently performs well across all datasets, excelling in large-scale and metadata-absent settings such as **DREAM and synthetic datasets**. In fairness-sensitive domains like legal decision-making, **causal_llm (DeepSeek)** (refer Appendix D.1) surpasses existing models, achieving $\approx 25\%$ higher precision in detecting true causal edges and mitigating systemic biases.

Key Contributions

- **Unified causal-LLM:** prompt-based full-graph generation with metadata (App. Figures 6 to 8); `causal_llm` for end-to-end data-driven inference (Sec. 4).
- **Cycle-free, scalable inference:** no iterative/pairwise queries, avoids spurious cycles, handles large graphs across domains.
- **Domain DAGs:** Bias Formation & Propagation; Legal Decision Process (BNS) (App. Figures 6 to 8, Figure 9).

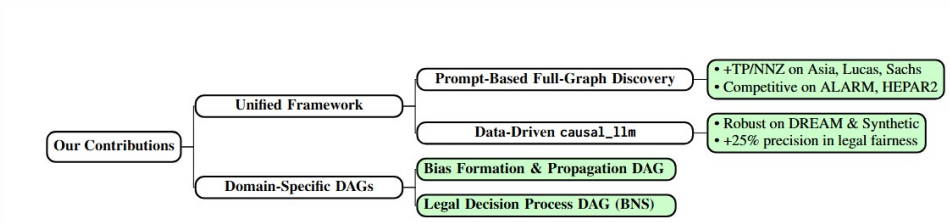

Figure 1: Our key contributions in causal-LLM: unified one-step graph discovery (both prompt-based and data-driven) plus two domain-specific DAG proposals.

By combining global context reasoning with data-driven learning, our framework establishes LLMs as powerful tools for systemic causal discovery—pushing them beyond language tasks toward structured, domain-aware reasoning with real-world impact.

Organization: The paper is structured as follows: we review related work in Section 2, present our approach in Section 3, and detail experiments, including baselines, datasets, and metrics, in Section 4. Major insights and key takeaways are discussed in Section 5 and Section 6, and we conclude with a summary and open directions in Section 7.

2 Related Works

This work investigates the causal discovery capabilities of large language models (LLMs), specifically focusing on the construction of the complete causal graph (see Figure 2). Although prior studies have explored general causal reasoning (Hobbhahn et al., 2022; Zhang et al., 2023), cause-effect inference (Zhiheng et al., 2022), and correlation-to-causation transitions (Jin et al., 2023), they do not address full graph discovery.

Most LLM-based approaches rely on pairwise causal edge detection (Willig et al., 2022; Long et al., 2023) or iterated querying across all node pairs (Kıcıman et al., 2023; Zečević et al., 2023; Kampani et al., 2024), which scale poorly due to quadratic complexity and often introduce cycles (Antonucci et al., 2023). Some mitigate this via post-processing or causal ordering with voting (Vashishtha et al., 2023), but these are typically restricted to small graphs ($\leq$22 nodes). Some works explore breadth-first querying for more scalable graph discovery (Jiralerspong et al., 2024), or generate domain knowledge graphs from text (Arsenyan et al., 2023), but without benchmarking against ground-truth DAGs. Recent efforts in single-shot generation (Naik et al., 2024) show promise, yet remain limited in scope.

Crucially, these methods are *prompt-based* and rely on node metadata—making them unsuitable for purely *data-driven* causal discovery. Existing work using LLMs as auxiliary tools (Ban et al., 2023; Cohrs et al., 2024) typically generate priors—e.g., pairwise edge constraints, causal orders, or adjacency matrices—which guide conventional algorithms rather than enabling direct inference. Attempts to elicit direct causal structure from data via prompting (Zhang et al., 2023) have not succeeded.

To fill this gap, we propose and benchmark a unified framework (refer Section 3): (i) prompt-based full-graph discovery when metadata is available, and (ii) `causal_llm`, a novel LLM-based method for end-to-end causal graph inference directly from data—evaluated on diverse datasets with up to 100 nodes.

3 Methodology

Prior works in LLM-based causal discovery have largely explored either: (i) **prompt-based querying**, which relies on external metadata and human-readable descriptions to elicit causal knowledge from language models (Willig et al., 2022; Tu et al., 2023; Kampani et al., 2024), or (ii) **data-driven causal discovery**, grounded in statistical principles and algorithms such as PC, GES, or ICA-LiNGAM. However, these two strands have been treated independently, and the literature lacks a unified framework that combines both capabilities—especially at scale.

Our work (please ref Figure 3) addresses this gap by proposing a dual-mode framework that evaluates and compares: (1) A **prompt-based approach** that performs causal graph generation directly from node metadata (refer Appendix A), enabling an LLM to reason based on its pre-trained knowledge; (2) A **data-driven model**, `causal_llm` (refer Algorithm 1), that learns causal structure purely from observational data using LLMs pretrained trans-

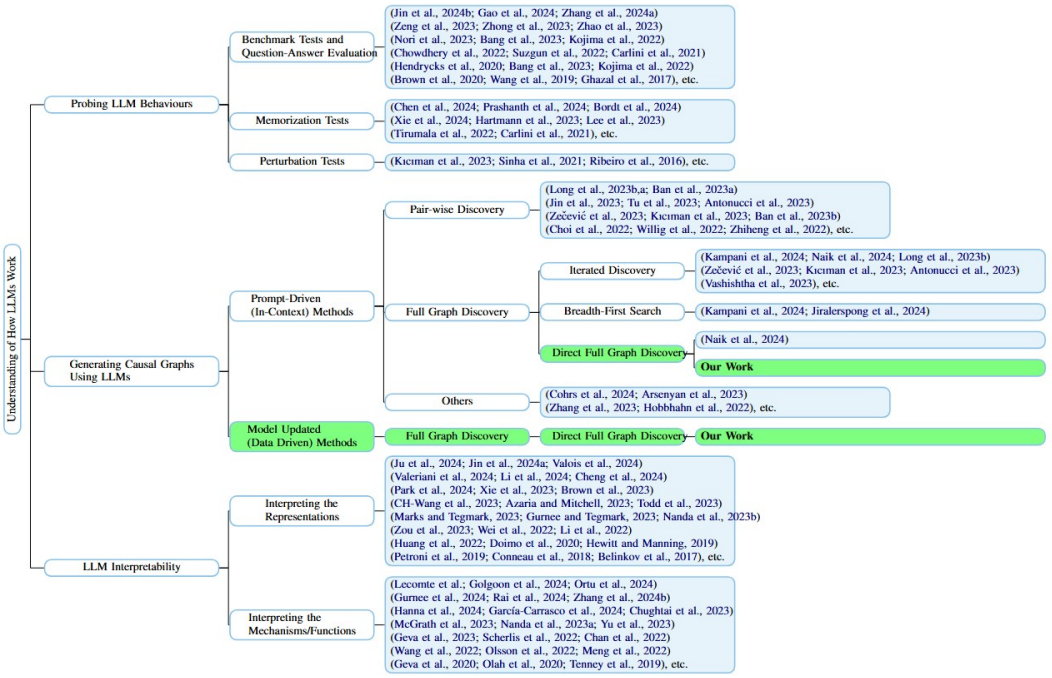

Figure 2: Overview of LLM-understanding research: This taxonomy categorizes studies on LLM behavior probing, causal graph discovery, and interpretability. The causal discovery methods include prompt-driven and model-updated approaches, highlighting pairwise, iterative, and full graph discovery techniques. **Our Work** (marked in green) contributes to direct full graph discovery in both paradigms.

former architecture.

This combination allows us to assess: (a) the ability of LLMs to perform causal discovery when metadata is available, and (b) their capacity to learn graph structure from data in a scalable and generalizable way. The key motivation behind our data-driven model is to move beyond using LLMs solely as promptable knowledge bases (as in (Ban et al., 2023; Cohrs et al., 2024)) toward direct end-to-end inference from data—a path that remains underexplored. A key question we address is: *Do we have node metadata for In-Context Learning?* If so, we employ a **prompt-based method**; otherwise, we use a **data-driven approach**, as shown in Equation (1).

$$\mathbf{A} = \begin{cases} \text{Parse}\left(f_p(LLM(\cdot), \mathcal{P}(\mathcal{T}, \text{M}(x)))\right), & \text{if M}(x) \neq \phi \\ \text{PostProcess}\left(f_d(LLM(\cdot), x)\right), & \text{otherwise} \end{cases} \tag{1}$$

Where:

- $\mathbf{A}$ is the **Adjacency matrix**.
- x is the **Dataset**.
- $\text{M}(x)$ extracts **Node Metadata** from the dataset.
- $\mathcal{T}$ is the **Prompt Template** (refer Appendix A).
- $\mathcal{P}(\mathcal{T}, \text{M}(x))$ generates a **dataset-specific prompt**.
- $LLM(\cdot)$ is the Large Language Model
- $\text{Parse}(\cdot)$ extracts the adjacency matrix from the LLM output.
- $f_p(\cdot)$ is the prompt-based approach (refer Section 3.1).
- $f_d(\cdot)$ is the data-driven model causal_llm (refer Algorithm 1).
- $\text{PostProcess}(\cdot)$ ensures DAG validity and prunes weak edges (Algorithms 3 and 4).

The **prompt-based approach** leverages modern LLMs' extended context lengths to perform full-graph causal discovery in a single pass, overcoming dependency loss in traditional pairwise iterative methods. It uses a carefully designed prompt (refer Appendix A) to ensure accuracy, scalability, and interoperability.The prompt defines the LLM's role as an **intelligent causal discovery agent** and sets the dataset's context, specifying its domain (e.g., medical, financial, or biochemical). It establishes the objective: identifying causal relationships between features to construct a **Directed Acyclic Graph (DAG)**. This framing ensures clarity and focus in the task.

The prompt incorporates detailed rules to guide the discovery process and provides metadata for **features (nodes)**, including descriptions and roles.

This metadata offers essential context, enabling the LLM to reason effectively about causal relationships.The output is structured in a standardized format, listing causal edges as pairs (e.g., (A, B)) with detailed explanations. This format ensures interpretability and enables automated post-processing using `regex` to extract the **adjacency matrix**, which precisely represents the causal structure.

This **prompt-based approach** is holistic and scalable, leveraging the increasing context length of LLMs to analyze larger datasets with higher-order nodes. The output explanations enhance the *interpretability* and *reliability* of the discovered DAG, ensuring robustness and efficiency in causal discovery.

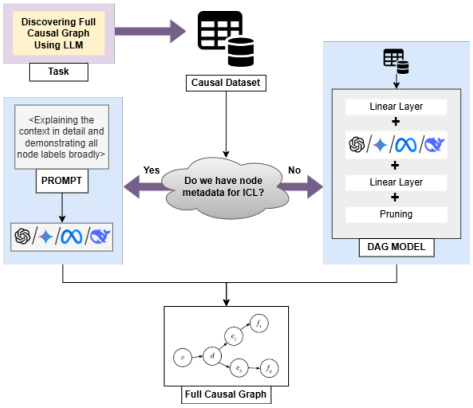

Figure 3: Overview of our causal discovery approach: If metadata is available, **prompt-based full-graph discovery (ICL)** is applied; otherwise, **data-driven causal_llm** extracts causal structures directly from the dataset.

Algorithm 1 LLM-Assisted DAG Discovery

Require: Data $X \in \mathbb{R}^{n \times d}$, pre-trained LLM, epochs E, sparsity weight λ, threshold τ

Freeze LLM parameters
Initialize projection matrices $W_{\text{in}} \in \mathbb{R}^{d \times h}$, $W_{\text{out}} \in \mathbb{R}^{h \times d}$
for $e \leftarrow 1$ to E **do**
  $Z \leftarrow X\, W_{\text{in}}$ ▷ Project inputs into h-dim space
  $H \leftarrow \text{LLM}(Z)$ ▷ Obtain contextual embeddings
  $A_{\text{logits}} \leftarrow H\, W_{\text{out}}$ ▷ Compute edge logits
  $A \leftarrow \sigma(A_{\text{logits}})$ ▷ Edge probabilities via sigmoid

$$\mathcal{L} = -\sum_{i,j} \log(1 - A_{ij}) + \lambda \sum_{i,j} |A_{ij}|$$

▷ Push probabilities to zero + enforce sparsity
  Update $W_{\text{in}}, W_{\text{out}}$ by backpropagating $\nabla \mathcal{L}$
end for

Enforce Acyclicity:
1: Remove the smallest-weight edge in any detected cycle
2: Repeat until the graph is acyclic

Prune Edges:

$$\text{Drop edge } (i, j) \text{ if } |\beta_{ij}| < \tau$$

return Adjacency matrix A of the resulting DAG

3.1 Prompt-based Approach

3.2 Data-Driven Approach

DAG Model: Our DAG model, **causal_llm** (refer Algorithm 1), utilizes a Large Language Model (LLM) to extract meaningful representations for causal discovery. It consists of three components: an **input projection layer**, the **LLM**, and an **output projection layer**. The **input projection layer** maps input data of dimension d_{input} to a higher-dimensional space compatible with the LLM's hidden size.The projected input, Z, is processed by the **LLM**, generating **contextualized hidden representations** that capture input dependencies. The LLM produces a hidden state matrix, $\mathbf{H}$, which the **output projection layer** maps to a $d \times d$ causal adjacency matrix. A sigmoid activation ensures values in $[0, 1]$, representing edge probabilities. By freezing LLM parameters and training only input and output layers, the model efficiently leverages LLM's feature extraction capabilities for accurate causal discovery with minimal computational overhead.

Model Training: The model operates in a synthetic environment, where each state corresponds to a dataset sample. Through forward passes and loss minimization, it predicts an adjacency matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$, constrained to be **acyclic** to satisfy DAG properties (refer Algorithm 2). The training loss comprises: (1) **binary cross-entropy loss** to measure the difference between predicted edge probabilities $\mathbf{A}_{ij}$ and a null matrix, and (2) an **L1 regularization term** to promote sparsity. As the model refines $\mathbf{A}$, edge weights $\mathbf{A}_{ij}$ update dynam-

ically, with the environment providing new states for learning. Over multiple epochs, the decreasing average loss indicates convergence to an optimal causal graph that balances sparsity and essential relationships.

Theoretical Justification of the loss function $\mathcal{L}(A)$

Let $A = (A_{ij})$ with each $A_{ij} \in (0, 1)$. Define

$$\mathcal{L}(A) = -\sum_{i,j} \log(1 - A_{ij}) + \lambda \sum_{i,j} |A_{ij}|.$$

1. **MAP derivation.**
   - *Likelihood:* $Y_{ij} \sim \text{Bernoulli}(1 - A_{ij})$, observe $Y_{ij} = 1$. $-\log P(Y = 1) = -\sum_{i,j} \log(1 - A_{ij})$.
   - *Prior:* $p(A_{ij}) \propto e^{-\lambda |A_{ij}|}$ gives $-\log p = \lambda |A_{ij}|$.
2. **Convexity & uniqueness.** For $a \in (0, 1)$, $f(a) = -\log(1 - a)$ with $f''(a) = 1/(1 - a)^2 > 0$, and $|a|$ is convex. Thus $\mathcal{L}$ is strictly convex.
3. **Exact sparsity.** $\partial |a||_0 = [-1, 1]$ blocks small gradients unless $a = 0$.

Hence $\mathcal{L}$ is the convex MAP-estimate with exact sparsity.

Post-Processing: To ensure a valid DAG, cycles are removed by iteratively deleting the lowest weight edge in each cycle (Algorithm 3). The resulting graph is further refined by **pruning weak edges** using linear regression: each node is regressed on its potential parents, and edges with coefficients below a threshold τ (set as the d-th largest weight for d nodes) are discarded (Algorithm 4). This process enhances the quality of the adjacency matrix by eliminating spurious and low-confidence connections.

4 Experimental Setup

4.1 Baselines

To benchmark our approach, we employ established causal structure discovery methods, including constraint-based approaches like the PC algorithm, Functional Causal Model (FCM)-based methods such as ICA-LiNGAM, and score-based techniques like Greedy Equivalence Search (GES) and RL-BIC. Additionally, we incorporate gradient-based methods, including Gradient-Based Neural DAG Learning (GraNDAG). These diverse algorithms provide a comprehensive foundation for evaluating our model's performance (Zhang et al., 2021). For details on the parameter settings of the baseline methods, refer to Appendix F.

4.2 Metrics

We use standard metrics to evaluate causal discovery algorithms (refer to *Evaluation Metrics for Causal Discovery* in (Hasan et al., 2023)).

Additionally[1], we introduce two new metrics designed to assess the precision of true edge identification by causal algorithms.

True Positives per Non-Zero Predictions (TP/NNZ): This metric calculates the proportion of true positives relative to all predicted edges (non-zero entries). This is an indicator on the precision of the model in detecting the true edges out of all its edge predictions. Higher values indicate better performance in predicting true edges without excess.

$$\text{TP/NNZ} = \frac{\text{TP}}{\text{NNZ}}$$

where, TP: Number of true positives, NNZ: Number of predicted edges (non-zero entries).

Relative Performance (RP): RP compares the TP/NNZ of a model against the best-performing model. A lower RP indicates that the model's performance is closer to the best.

$$\text{RP} = \frac{\text{Best(TP/NNZ)} - \text{TP/NNZ}}{\text{Best(TP/NNZ)}}$$

where, Best(TP/NNZ): Best value of TP/NNZ across models TP/NNZ: True positives per non-zero predictions for the current model.

4.3 Datasets

Causal discovery methods analyze datasets from real-world observations or synthetic sources. Real data comes from *medical trials, economic surveys, and genomics experiments*, while synthetic datasets are generated using known or artificial causal structures.

In our experiments, we used both real and publicly available datasets, alongside synthetic datasets generated from *domain knowledge-based Directed Acyclic Graphs (DAGs)*. For publicly available datasets, we utilize the bnlearn repository (Scutari, 2009) and the Causal Discovery Toolbox (CDT) (Kalainathan et al., 2020).

Publicly available datasets: SACHS, DREAM, ASIA, ALARM, LUCAS, HEPAR2 (refer Appendix C.1).

Synthetic datasets: • Linear models with Gaussian/non-Gaussian noise (refer Appendix C.2.1) • Non-linear quadratic models

[1] refer Appendix E

with Gaussian/non-Gaussian noise (refer Appendix C.2.2) • Non-linear Gaussian process models with Gaussian noise (refer Appendix C.2.3)

Domain Specific Dags

We have also constructed two DAGs from Domain Expert Knowledge and used it to generate synthetic data (refer Appendix C.2.4).
• A DAG representing **bias formation and propagation** (refer Apendix Figures 6 to 8)
• A DAG representing **legal decision processes under the Bhartiya Nyaya Sanhita (BNS) scheme** (refer Appendix Figure 9)

4.4 Results

In this section, we present **dataset-wise results** comparing the performance of all baseline models against our proposed model (refer Appendix Figure 12). This structured comparison allows us to evaluate the effectiveness of our model across different datasets (refer Figure 4).

PUBLICLY AVAILABLE DATASETS (see Figure 4)

In **SACHS**, the prompt-based method achieves superior FDR and RP, likely due to semantically rich metadata that aligns well with LLM pretraining. In contrast, traditional algorithms like PC exhibit high TPR but suffer from high FDR and SHD, indicating overprediction. The data-driven **causal_llm** maintains balanced performance across all metrics, demonstrating robustness without metadata reliance.

For **ASIA**, the strong performance of all prompt-based LLMs—some matching the ground truth exactly—suggests the dataset or similar structures may have been encountered during LLM training. Traditional methods like GES are competitive but slightly hampered by higher FDR. **Causal_llm** underperforms, possibly due to the dataset's simplicity and low variance.

In **LUCAS**, GES aligns perfectly with the ground truth, benefiting from efficient structure scoring in small graphs. Prompt-based models perform nearly as well, with **causal_llm** offering stable, if not top-tier, performance. GranDAG underperforms due to limited edge predictions, struggling with sparse structures.

For **ALARM**, a mid-sized graph, prompt-based models outperform symbolic approaches by achieving better trade-offs between TPR and FDR. PC and GES have higher TPR but also elevated FDR and SHD, indicating noise. **Causal_llm** struggles in this transitional regime, highlighting limitations in medium-scale structures.

In **HEPAR2**, as the node count increases, symbolic models face combinatorial challenges and often fail to converge. Prompt-based methods excel across all metrics, leveraging global metadata. **Causal_llm** remains competitive, showing resilience in node-dense settings.

In the high-dimensional **DREAM** dataset, most models fail due to complexity. **Causal_llm (GPT)** stands out with the best RP and TPR, demonstrating the effectiveness of LLMs in metadata-absent, large-scale settings. GranDAG's low SHD is undermined by high FDR, indicating excessive regularization.

In **Bias & Legal** datasets (Appendix Figure 11), prompt-based methods dominate, particularly where node labels encode sociocultural or legal context. **Causal_llm** also performs well, especially in the *implicit-to-explicit* and *Legal* cases, revealing its ability to capture fairness-related dependencies directly from data.

SYNTHETIC DATASETS (see Appendix Figure 10)

For **10-node** graphs, **causal_llm (GPT)** and prompt-based methods excel. ICA-LiNGAM and GES perform well but are limited to low-dimensional settings. At **40 nodes**, **causal_llm (Gemini)** leads on linear graphs, while ICA-LiNGAM excels in GP settings, highlighting its non-linear modeling capacity.

On **70-node** graphs, most models degrade, but **causal_llm** maintains effective detection of causal edges, demonstrating scalability. For **100-node** graphs, **causal_llm (Llama, GPT)** are among the few viable models, outperforming others by handling dimensionality and noise robustly.

Overall results suggest that **prompt-based method** using LLMs outperform data-driven approaches, especially when node metadata is available, achieving high accuracy in edge detection. Among data-driven models, **causal_llm** consistently performs best, particularly in larger datasets. **GES** and **ICA-LiNGAM** excel in specific cases (e.g., ASIA, LUCAS), but their effectiveness is limited by high FDR and SHD. GranDAG underperforms across datasets, often failing to capture causal relationships. As the number of nodes increases, most models decline in performance, but causal_llm remains consistent overall.

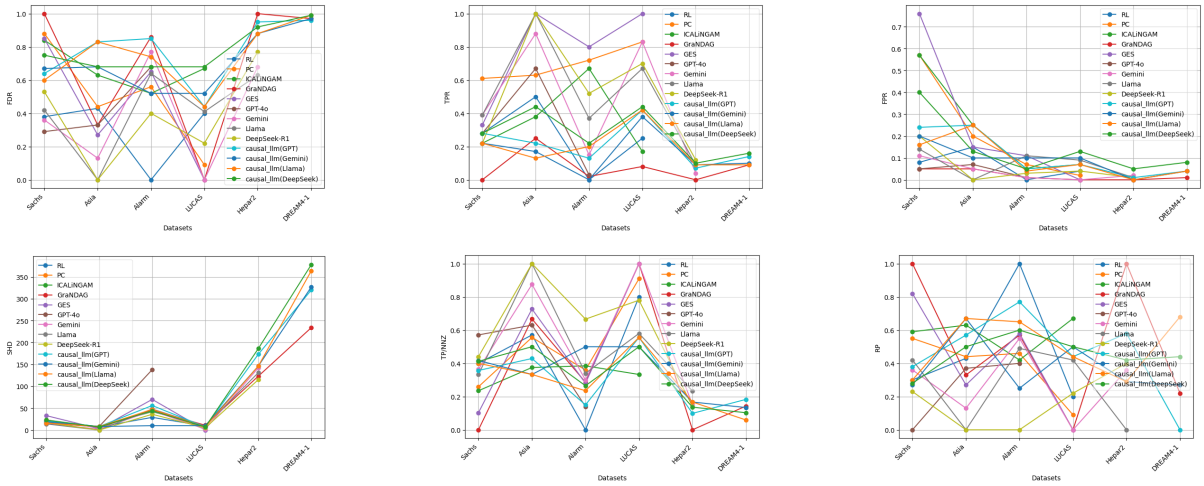

Figure 4: FDR, TPR, FPR, SHD, TP/NNZ and RP metrics for RL, PC, ICALiNGAM, GraNDAG, GES, GPT-4o, Gemini, Llama, DeepSeek-R1 and causal_llm (with models GPT, Gemini, Llama and DeepSeek) plotted for the publicly available datasets SACHS, ASIA, ALARM, LUCAS, HEPAR2, and DREAM4.

5 Discussion

In this paper, we have argued that the question of causal understanding is equivalent to the understanding of how LLM functions, that is, whether LLM follows any causation while generating the output. Our experimental results rigorously validate the effectiveness of both the **prompt-based method** and the **data-driven causal_llm model**, while also delineating their respective strengths and limitations. Below, we synthesize these findings through systematic analysis:

Prompt-Based Method: Leveraging Node Metadata for Superior Accuracy The prompt-based approach, which utilizes node metadata, demonstrates measurable advantages (refer Appendix Figure 12):

• **Edge Accuracy:** On datasets like **ASIA** and **LUCAS**, the prompt-based method achieves an average of $\approx 40\%$ higher edge accuracy compared to data-driven methods, highlighting its ability to leverage metadata for precise causal discovery.

• **Fairness-Critical Domains:** In fairness-critical domains such as legal systems, the prompt-based method improves precision in identifying true causal edges by $\approx 25\%$, effectively addressing systemic biases often overlooked by pairwise methods.

• **Limitation in Metadata-Absent Scenarios:** On datasets like **DREAM41**, where metadata is unavailable, the prompt-based method **cannot be used**, emphasizing its reliance on node metadata for optimal results.

Data-Driven Approach: Competitive Performance and Efficiency The **causal_llm** model, which integrates LLMs for causal discovery purely from data, demonstrates competitve performance and scalability (refer Appendix Figure 12):

• **Runtime Efficiency:** On the **Sachs dataset**, causal_llm achieves inference in $\approx 50\%$ less runtime on average compared to RL-based and continuous optimization-based methods, showcasing its computational efficiency.

• **Scalability:** In synthetic scenarios with larger graphs (e.g., **70-node** and **100-node datasets**), causal_llm scales seamlessly, offering $\approx 20\%$ faster inference while maintaining competitive accuracy.

• **Limitation in Metadata-Rich Scenarios:** While competitive, causal_llm's performance **lags behind the prompt-based method** in datasets where metadata plays a crucial role in guiding causal discovery.

Comparative Analysis: Strengths and Trade-offs:The prompt-based method excels in metadata-rich settings, delivering high accuracy and addressing fairness in sensitive domains (see Figure 11). In contrast, the data-driven **causal_llm** model offers a scalable, efficient alternative with competitive performance and faster runtime. Together, they showcase the potential of LLMs in causal discovery, providing robust solutions for both metadata-driven and data-only scenarios while balancing accuracy, efficiency, and fairness.

These complementary strengths establish the **prompt-based** and **data-driven** approaches as effective, versatile tools for modern causal discovery (refer Appendix Figures 10 to 12), with demon-

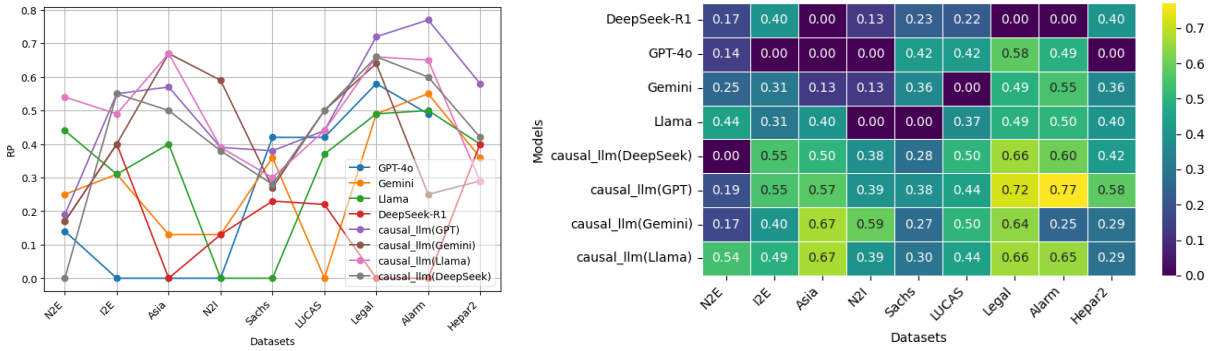

Figure 5: A comparative plot of the **relative performance (RP metric)** of LLMs (**prompt-based approach** and **causal_llm**) on the metadata-rich datasets (SACHS, ASIA, ALARM, LUCAS, HEPAR2, and BIAS and LEGAL datasets), in increasing order of number of nodes.

strated success across domains ranging from small biological networks to large-scale gene regulatory systems.

6 Key Takeaways

In this section, we compare the prompt-based approach and the data-driven approach to determine their respective advantages (refer Figure 5).

- In datasets such as **Asia, Lucas, and Sachs**, where the number of nodes is small and node metadata is available, the prompt-based method outperforms all other causal algorithms by achieving better true positives per nonzero (TP/NNZ) and maintaining a low false discovery rate (FDR). In the **ALARM** dataset, as the number of nodes increases, the prompt-based approach remains competitive with other causal algorithms in terms of true positive rate (TPR) while still maintaining a low FDR, making it a consistent method. As the number of nodes increases further, such as in the **HEPAR2** dataset, the performance of the prompt-based approach declines but it still remains competitive with other causal models.
- In datasets like **DREAM** and **synthetic datasets**, where **node metadata is unavailable, the prompt-based approach cannot be applied**. Despite this limitation, our data-driven method, **causal_llm**, remains competitive across all datasets. It excels particularly in large-scale datasets and those without metadata, offering a robust alternative to state-of-the-art causal algorithms. Notably, in the **neutral to explicit dataset**, **causal_llm (DeepSeek)** (refer Appendix D.1) outperforms all others, including prompt-based methods, in detecting true edges, as shown by its high TP/NNZ ratio and low false positive rate (RP), highlighting its effectiveness across diverse scenarios.
- Therefore, when node metadata is available, the **prompt-based approach** is preferred due to its exceptional performance, while in cases where metadata is unavailable, the data-driven model **causal_llm** emerges as a consistent and reliable choice.

7 Conclusion

Overall, the **prompt-based method** excels in metadata-rich settings, ensuring high accuracy and fairness in critical domains. The **data-driven causal_llm model** emerges as a scalable and efficient alternative, delivering competitive performance with reduced runtime. This highlights LLMs' capability for full graph discovery, positioning them as strong contenders in causal discovery for both metadata-rich and data-only scenarios.

Limitations

Despite its strong performance, our framework has some limitations. The prompt-based approach depends heavily on prompt quality and metadata completeness, which can affect accuracy. Token limits and attention constraints challenge scalability on large graphs. In the data-driven model, freezing the LLM backbone improves efficiency but reduces adaptability to domain-specific contexts. Real-world evaluation is limited by the absence of ground truth, and post-processing steps involve heuristics that may introduce variability across datasets.

References

Alessandro Antonucci, Gregorio Piqué, and Marco Zaffalon. 2023. Zero-shot causal graph extrapolation from text via llms. *arXiv preprint arXiv:2312.14670*.

Vahan Arsenyan, Spartak Bughdaryan, Fadi Shaya, Kent Small, and Davit Shahnazaryan. 2023. Large language models for biomedical knowledge graph construction: Information extraction from emr notes. *arXiv preprint arXiv:2301.12473*.

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

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.

Kai-Hendrik Cohrs, Gherardo Varando, Emiliano Diaz, Vasileios Sitokonstantinou, and Gustau Camps-Valls. 2024. Large language models for constrained-based causal discovery. *arXiv preprint arXiv:2406.07378*.

Uzma Hasan, Emam Hossain, and Md Osman Gani. 2023. A survey on causal discovery methods for iid and time series data. *arXiv preprint arXiv:2303.15027*.

Marius Hobbhahn, Tom Lieberum, and David Seiler. 2022. Investigating causal understanding in llms. In *NeurIPS ML Safety Workshop*.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. 2023. Can large language models infer causation from correlation? *arXiv preprint arXiv:2306.05836*.

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*.

Diviyan Kalainathan and Olivier Goudet. 1903. Causal discovery toolbox: Uncover causal relationships in python; 2019. *URL https://arxiv. org/abs*.

Diviyan Kalainathan, Olivier Goudet, and Ritik Dutta. 2020. Causal discovery toolbox: Uncovering causal relationships in python. *Journal of Machine Learning Research*, 21(37):1–5.

Shiv Kampani, David Hidary, Constantijn van der Poel, Martin Ganahl, and Brenda Miao. 2024. Llm-initialized differentiable causal discovery. *arXiv preprint arXiv:2410.21141*.

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

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. 2019. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*.

Steffen L Lauritzen and David J Spiegelhalter. 1988. 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.

Stephanie Long, Tibor Schuster, and Alexandre Piché. 2023. Can large language models build causal graphs? *arXiv preprint arXiv:2303.05279*.

Peter JF Lucas, Linda C Van der Gaag, and Ameen Abu-Hanna. 2004. Bayesian networks in biomedicine and health-care.

Narmada Naik, Ayush Khandelwal, Mohit Joshi, Madhusudan Atre, Hollis Wright, Kavya Kannan, Scott Hill, Giridhar Mamidipudi, Ganapati Srinivasa, Carlo Bifulco, and 1 others. 2024. Applying large language models for causal structure learning in non small cell lung cancer. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pages 688–693. IEEE.

Agnieszka Onisko. 2003. Probabilistic causal models in medicine: Application to diagnosis of liver disorders. In *Ph. D. dissertation, Inst. Biocybern. Biomed. Eng., Polish Academy Sci., Warsaw, Poland*.

Jonas Peters and Peter Bühlmann. 2014. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. 2014. Causal discovery with continuous additive noise models.

Marco Scutari. 2009. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. 2006. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10):2003–2030.

Ruibo Tu, Chao Ma, and Cheng Zhang. 2023. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis. *arXiv preprint arXiv:2301.13819*.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. 2023. Causal inference using llm-guided discovery. *arXiv preprint arXiv:2310.15117*.

Moritz Willig, Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. 2022. Can foundation models talk causality? *arXiv preprint arXiv:2206.10591*.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067*.

Cheng Zhang, Stefan Bauer, Paul Bennett, Jiangfeng Gao, Wenbo Gong, Agrin Hilmkil, Joel Jennings, Chao Ma, Tom Minka, Nick Pawlowski, and 1 others. 2023. Understanding causality with large language models: Feasibility and opportunities. *arXiv preprint arXiv:2304.05524*.

Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. 2021. gcastle: A python toolbox for causal discovery. *arXiv preprint arXiv:2111.15155*.

LYU Zhiheng, Zhijing Jin, Rada Mihalcea, Mrinmaya Sachan, and Bernhard Schölkopf. 2022. Can large language models distinguish cause from effect? In *UAI 2022 Workshop on Causal Representation Learning*.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. 2020. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*.

A Prompt Used for Single-Step Full Graph Discovery

PROMPT TEMPLATE

You are an intelligent causal discovery agent tasked with mapping the causal relationships between features in the **[*Dataset Name]** dataset. This dataset models [brief description of the domain, e.g., medical conditions, social biases, biochemical signaling*]. Your goal is to identify how these features influence one another and construct a Directed Acyclic Graph (DAG) that represents these causal relationships.

—

Important Rules:

1. ***Multiple Incoming Edges:*** Each feature may have multiple incoming edges to reflect its dependency on upstream causes.
2. ***Root Causes:*** Some features act as root causes (independent variables) that initiate the causal chain.
3. ***Intermediate Variables:*** Other features act as intermediaries, propagating the effects of root causes and influencing downstream outcomes.
4. ***Outcome Variables:*** Observable outcomes should only receive causal inputs from relevant upstream features.
5. ***Acyclic Structure:*** Ensure the DAG is acyclic and aligns with domain knowledge.

—

Features (Nodes):

- ***[Feature 1]:*** [Brief description of the feature].
- ***[Feature 2]:*** [Brief description of the feature]. - ...

—

Step 1: Finding the Edges

Identify the causal relationships between the features. Focus on how upstream features influence downstream ones. For example:

1. ***Edge (Feature A → Feature B):*** [Explanation of why Feature A causes Feature B].
2. ***Edge (Feature C → Feature D):*** [Explanation of why Feature C causes Feature D].
3. ...

—

Step 2: Reflect Back on Each Edge

Review each edge to ensure it aligns with domain knowledge. Refine the causal relationships if necessary.

—

Output Format:

Provide a final list of edges in the following format:

1. (A, B) : Explanation of why A causes B.
2. (C, D) : Explanation of why C causes D. ...

B Algorithms

B.1 DAG Model: causal_llm

Model Architecture The architecture comprises three primary components: an **input projection layer**, the **Large Language Model**, and an **output projection layer**. The **input projection layer** takes input data of dimension d_{input} and projects it into a higher-dimensional feature space compatible with the LLM's hidden size. The projected input, $\mathbf{x}_{\text{projected}}$, is then passed through the **LLM**, which generates **contextualized hidden representations** that encapsulate the dependencies in the input. The output of the LLM is a hidden state matrix, $\mathbf{H}$. These hidden states are processed by the **output projection layer**, which maps the high-dimensional representations to an $d \times d$ causal adjacency matrix, where d is the number of

nodes in the causal graph. A sigmoid activation function is applied to ensure the adjacency matrix values are in the range $[0, 1]$, representing edge probabilities. By freezing the pre-trained LLM parameters and training only the input and output layers, the model efficiently adapts to the causal discovery, leveraging LLM's strong feature extraction capabilities without increased computational burden to extract accurate causal relationships from the dataset.

Algorithm 2 causal_llm Training and Inference

Require: d_{in}, d_{out}
Ensure: Trained model and inferred adjacency matrix
$\mathcal{M} \leftarrow$ causal_llm$(d_{\text{in}}, d_{\text{out}})$
$\mathcal{O} \leftarrow$ Adam($\mathcal{M}$.parameters(), lr $= 2e-5$)
$\mathcal{L} \leftarrow$ BCE Loss
function LEARN($\mathcal{D}$, $\mathcal{E}$, $\mathcal{B}$, ϵ)
    $\mathcal{G} \leftarrow$ SyntheticEnvironment($\mathcal{D}$)
    for $e = 1$ to $\mathcal{E}$ **do**
        $\mathcal{L}_{\text{epoch}} \leftarrow []$
        for $b = 1$ to $\mathcal{B}$ **do**
            $s \leftarrow \mathcal{G}$.get_next_state()
            $\mathbf{s} \leftarrow$ tensor(s)
            $\mathbf{a} \leftarrow \sigma(\mathcal{M}(\mathbf{s}))$
            if random ϵ **then**
                $\mathbf{a} \leftarrow$ random tensor
            end if
            $\mathbf{A} \leftarrow$ Reshape$(\mathbf{a})$
            $\mathbf{A} \leftarrow$ RemoveCycles$(\mathbf{A})$
            $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}(\mathbf{A}, \mathbf{0}) + 0.01\|\mathcal{M}\|$
            Backpropagate: $\mathcal{O}$.step()
            Store $\mathcal{L}_{\text{epoch}}$
        end for
        $\mathcal{L}_{\text{avg}} \leftarrow$ mean$(\mathcal{L}_{\text{epoch}})$
    end for
    if $\mathcal{P}$ exists **then**
        Save $\mathcal{M}$ to $\mathcal{P}$
    end if
end function
function CAUSALMATRIX($\mathcal{D}$)
    $\mathbf{D} \leftarrow$ tensor$(\mathcal{D})$
    $\mathbf{s} \leftarrow$ mean$(\mathbf{D}, 0)$
    Set $\mathcal{M}$ to eval mode
    $\mathbf{A} \leftarrow \sigma(\mathcal{M}(\mathbf{s}))$
    $\mathbf{A} \leftarrow \mathbf{A} \cdot (1 - I)$
    $\mathbf{A} \leftarrow$ PruneWeakEdges$(\mathbf{A})$
    $\mathbf{A}_{\text{final}} \leftarrow$ RemoveCycles$(\mathbf{A})$
    return $\mathbf{A}_{\text{final}}$
end function

B.2 Helper Functions

B.2.1 RemoveCycles

This functions transforms a directed graph containing loops into a Directed Acyclic Graphs(DAGs). Starting with a weighted adjacency matrix (where entries represent connection strengths between nodes), it first constructs the graph. It then iteratively looks for cycles, removes them by eliminating the weakest link in each loop.To minimize structural damage, the function prioritizes removing edges with the smallest weights, ensuring stronger, more critical connections are preserved. When multiple edges in a cycle share the same minimal weight, it breaks ties randomly to avoid unintended bias. This process repeats until all cycles are eliminated, producing a directed acyclic graph (DAG) that retains the original graph with most of the relevant edges.

Algorithm 3 RemoveCycles

Require: Adjacency matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$
Ensure: Acyclic adjacency matrix $\mathbf{A}_{\text{acyclic}}$
Step 1: Initialize Graph
Create directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from $\mathbf{A}$:
for all $i, j \in [1, d]$ **do**
  if $i \neq j$ and $\mathbf{A}[i, j] > 0$ **then**
    Add edge (i, j) with weight $\mathbf{A}[i, j]$ to $\mathcal{G}$
  end if
end for
Step 2: Remove Cycles
while $\mathcal{G}$ contains cycles **do**
  Detect cycles: $\mathcal{C} \leftarrow \text{FindCycle}(\mathcal{G})$
  Initialize minimum weight: $w_{\text{min}} \leftarrow \infty$
  Initialize candidate edges: $\mathcal{E}_{\text{min}} \leftarrow []$
  for all $(u, v, \text{direction}) \in \mathcal{C}$ **do**
    $w \leftarrow \mathcal{G}[u][v]['\text{weight}']$
    if $w < w_{\text{min}}$ **then**
      $\mathcal{E}_{\text{min}} \leftarrow [(u, v)]$
      $w_{\text{min}} \leftarrow w$
    else if $w == w_{\text{min}}$ **then**
      Add (u, v) to $\mathcal{E}_{\text{min}}$
    end if
  end for
  Randomly select edge: $(u_{\text{min}}, v_{\text{min}}) \sim \mathcal{E}_{\text{min}}$
  Remove edge: $\mathcal{G}$.remove_edge$(u_{\text{min}}, v_{\text{min}})$
  Update $\mathbf{A}[u_{\text{min}}, v_{\text{min}}] \leftarrow 0$
end while
return $\mathbf{A}_{\text{acyclic}}$

B.2.2 PruneWeakEdges

This function is designed to refine a given graph by pruning weak connections based on regression coefficients derived from the dataset. It begins by initializing variables, including the graph structure, node count, and a weight matrix to store regression coefficients. For each node in the graph, the algorithm identifies its connected nodes, extracts the corresponding features and target values from the dataset, and performs linear regression to compute the coefficients. These coefficients, representing the strength of connections, are stored in a weight matrix.The algorithm calculates a threshold based on the sorted absolute values of the coefficients, ensuring that at least one strong connection per node is preserved. Finally, edges in the graph are pruned by retaining only those connections with coefficient magnitudes greater than or equal to the threshold.

Algorithm 4 PruneWeakEdges

Require: Graph batch $\mathbf{G}$, Dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$
Ensure: Pruned graph $\mathbf{G}_{\text{pruned}} \in \{0, 1\}^{d \times d}$
Step 1: Initialize Variables
Number of nodes: $d \leftarrow \text{len}(\mathbf{G})$
Initialize weight matrix: $\mathbf{W} \leftarrow []$ ▷ To store regression coefficients
Step 2: Compute Regression Coefficients
for $i = 1$ to d **do**
    Select column: col $\leftarrow |\mathbf{G}[i, :]| > 0.5$
    if $\sum(\text{col}) == 0$ **then**
        Append zeros: $\mathbf{W}$.append($\mathbf{0}_d$)
        Continue
    end if
    Extract features: $\mathbf{X}_{\text{train}} \leftarrow \mathbf{X}[:, \text{col}]$
    Extract target: $\mathbf{y} \leftarrow \mathbf{X}[:, i]$
    Fit linear regression: reg.fit($\mathbf{X}_{\text{train}}, \mathbf{y}$)
    Obtain coefficients: $\mathbf{c} \leftarrow$ reg.coef_
    Initialize zero vector: $\mathbf{c}_{\text{new}} \leftarrow \mathbf{0}_d$
    Assign coefficients: $\mathbf{c}_{\text{new}}[\text{col}] \leftarrow \mathbf{c}$
    Append to weight matrix: $\mathbf{W}$.append($\mathbf{c}_{\text{new}}$)
end for
Step 3: Calculate Threshold
Sort: $\mathbf{W}_{\text{sorted}} \leftarrow \text{sort}(|\mathbf{W}|.\text{flatten}())$
Determine threshold index: $d_{\text{idx}} \leftarrow \min(d - 1, \text{len}(\mathbf{W}_{\text{sorted}}) - 1)$
Calculate threshold: th $\leftarrow \mathbf{W}_{\text{sorted}}[d_{\text{idx}}]$
Step 4: Prune Graph
Prune edges: $\mathbf{G}_{\text{pruned}} \leftarrow (|\mathbf{W}| \geq \text{th})$
return $\mathbf{G}_{\text{pruned}}$

C Datasets

C.1 Publicly available datasets

Publicly available causal datasets are commonly used to benchmark algorithms in *causal discovery, machine learning, and statistical modeling*. These datasets often stem from interventional experiments across real-world domains such as *biology, medicine, environment, and education*. We evaluate our method using datasets from the *bnlearn* repository (Scutari, 2009) and the *Causal Discovery Toolbox (CDT)* (Kalainathan et al., 2020).

SACHS (Zhang et al., 2021): This dataset captures causal relationships between genes based on known biological pathways. It has **11 nodes** with well known ground truth.

DREAM (Kalainathan and Goudet, 1903): DREAM (Dialogue on Reverse Engineering Assessments and Methods) challenges provide simulated and real biological datasets to test methods for inferring gene regulatory networks.We have used the dataset DREAM4-1, consisting of **100 nodes.**

ALARM (Beinlich et al., 1989): This dataset simulates a medical monitoring system for patient status in intensive care, including variables such as heart rate, blood pressure, and oxygen levels.It consists of **37 nodes** and is widely used in benchmarking algorithms in the medical domain.

ASIA (Lauritzen and Spiegelhalter, 1988): Asia dataset models a causal network of variables related to lung diseases and the likelihood of visiting Asia. This is a small dataset consisting of only **8 nodes**.

LUCAS (Lucas et al., 2004): LUCAS (LUng CAncer Simple) is a synthetic dataset designed for causal discovery benchmarking in medical contexts. It simulates causal relationships related to lung cancer, incorporating variables such as smoking habits, exposure to pollution, genetic predisposition, and disease outcomes. The dataset consists of **11 nodes** and is often used to evaluate causal structure learning algorithms in the medical domain.

HEPAR2 (Onisko, 2003): HEPAR2 dataset is a probabilistic Bayesian network model representing causal relationships in the diagnosis of liver disorders. It consists of **70 nodes** and **123 edges**, making it a comprehensive benchmark for testing causal discovery algorithms in the healthcare domain.

C.2 Synthetic datasets

We generate synthetic datasets using methods from **ICA-LiNGAM** (Shimizu et al., 2006) and **RL-based causal discovery** (Zhu et al., 2020). These datasets, derived from both domain knowledge-based and purely synthetic DAGs, enable us to explore diverse causal structures and benchmark our model against *state-of-the-art* causal algorithms.

We generate four types of Datasets:

- **Linear model** with Gaussian and non-Gaussian noise
- **Non-linear quadratic model** with Gaussian and non-Gaussian noise
- **Non-linear Gaussian process** with Gaussian noise
- **Bias and Legal** datasets from Domain knowledge (refer Appendix C.2.4)

We employ the same initialization method used in the ICA-LiNGAM (Shimizu et al., 2006) and RL-based causal discovery (Zhu et al., 2020) papers to generate synthetic datasets. For the Bias and Legal datasets, we create synthetic data using a linear model with the same initialization approach.

C.2.1 Linear Model with Gaussian and Non-Gaussian Noise

To generate synthetic data, we start by creating a $d \times d$ upper triangular adjacency matrix representing the graph structure, where the upper triangular entries are independently sampled from a Bernoulli distribution - Bern(0.5). Next, we assign edge weights from the uniform distribution Unif($[-2, -0.5] \cup [0.5, 2]$), forming a weight matrix, $W \in \mathbb{R}^{d \times d}$.

Using this setup, we generate data samples according to

$$x = W^T x + n,$$

where $n \in \mathbb{R}^d$ represents noise. Both Gaussian and non-Gaussian noise models are used. For the non-Gaussian case, we adopt the approach from **ICA-LiNGAM** (Shimizu et al., 2006) , where Gaussian noise samples are transformed via a power non-linearity to induce non-Gaussianity. In both cases, unit noise variances are used.

We generate $n = 5000$ samples and randomly permute the variables to create the final datasets. This procedure aligns with approaches used in prior works such as NOTEARS and DAG-GNN, where the true causal graphs are known to be identifiable (Shimizu et al., 2006; Peters and Bühlmann, 2014). We repeat this process for $d = 10, 40, 70, 100$ nodes and use it benchmark against *state-of-the-art* causal algorithms.

C.2.2 Non-linear Quadratic Model with Gaussian and Non-Gaussian Noise

In this method, we investigate nonlinear causal relationships using quadratic functions. The graph structure is generated by creating an upper triangular adjacency matrix, following a similar procedure as the first method. For each node i, the parent variables $x_{\text{pa}(i)} = [x_{i1}, x_{i2}, \dots]^T$ are expanded to include both first-order and second-order features. The coefficients for these features are either set to zero or sampled from the uniform distribution

$$\text{Unif}([-1, -0.5] \cup [0.5, 1])$$

with equal probability. If a parent variable does not contribute to any feature term with a non-zero coefficient, the corresponding edge is removed from the causal graph.

Data is generated for graphs with $d = 10, 40, 70$, and 100 nodes, with 5,000 samples for each case. We consider both Gaussian and non-Gaussian noise models. For the non-Gaussian case, noise is generated by transforming Gaussian samples using a power nonlinearity to induce non-Gaussianity. However, large variable values can sometimes occur in the quadratic model, which can cause computational problems in quadratic regression. Such extreme samples are treated as outliers.

This approach allows us to study the identifiability of nonlinear causal graphs across varying graph sizes and noise models while addressing computational challenges.

C.2.3 Non-Linear Model with Gaussian Processes

This method involves studying nonlinear causal relationships in randomly generated causal graphs. Each causal relationship f_i is modeled as a nonlinear function sampled from a Gaussian process with a Radial

Basis Function (RBF) kernel, where the bandwidth is set to one. The use of the RBF kernel ensures smoothness and flexibility in the functional form of f_i, allowing it to model complex dependencies between variables.

The additive noise n_i in the system is drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$, where the noise variance σ^2 is sampled uniformly across a predefined range. This variability in noise strength across different relationships influences the complexity of causal inference. The setup adheres to conditions under which the true causal graph is identifiable, as established by (Peters et al., 2014).

For this experiment, we adopt a framework inspired by GraN-DAG (Lachapelle et al., 2019). Specifically, we generate causal graphs with 10 nodes and 40 directed edges, ensuring a dense and complex network of dependencies. The data consists of 1,000 samples, allowing for robust statistical inference and testing of causal discovery methods.

This setup is particularly valuable for benchmarking algorithms designed for nonlinear causal discovery, as it captures realistic complexities while maintaining identifiability.

C.2.4 Bias and Legal DAGs

To construct the **Bias dataset**, we undertook an in-depth literature review on implicit bias, analyzing the factors contributing to unconscious biases and their subtle manifestations in language. This process guided the development of three **Directed Acyclic Graphs (DAGs)** that depict how bias propagates and evolves in linguistic contexts. These diagrams were validated by domain experts.For the **Legal dataset**, we collaborated with a legal expert to create a DAG that models the **legal decision-making processes under the Bhartiya Nyaya Sanhita (BNS) scheme**.This DAG models the structured reasoning and causal pathways used to determine outcomes such as murder, culpable homicide, or non-culpable homicide under the BNS framework. The nodes in this graph represent critical legal factors and decision points in the judicial process.After obtaining the DAG, we generate the weighted adjacency matrix by sampling the weights randomly from the uniform distribution $\text{Unif}([-2, -0.5) \cup (0.5, 2])$. The data is then generated in the same way as described in the first method.

Neutral to Implicitly Biased Sentences (N2I)

This DAG captures the transition from neutral language to implicitly biased sentences. The transformation is influenced by the following factors:

• **Social Identity**: The speaker's or listener's sense of belonging to a particular group.
• **Stereotype**: Preconceived notions or generalized beliefs about a group.
• **Stereotype Activation**: The subconscious triggering of stereotypes in response to specific cues.
• **Cognitive Dissonance**: The discomfort from holding conflicting beliefs, which can subtly shape language.
• **Ambiguous Language**: Words or phrases with multiple interpretations, leaving room for implicit bias.
• **Unprotected Features**: Attributes not safeguarded against discrimination, potentially amplifying bias.
• **Social Desirability**: The tendency to conform to socially acceptable norms, sometimes leading to veiled biases.
• **Protected Features**: Characteristics shielded under anti-discrimination policies that may still influence bias indirectly.

Neutral to Explicitly Biased Sentences (N2E))

This DAG models how neutral language transforms into overtly biased statements, driven by:

• **Social Identity**: The speaker's or listener's sense of belonging to a particular group.
• **Stereotype**: The direct incorporation of generalized beliefs into speech.
• **Conscious Stereotyping**: Deliberate application of stereotypes in communication.
• **Protected Features**: Characteristics (e.g., race, gender) that become focal points in biased discourse.
• **Motivated Reasoning**: The use of reasoning aligned with one's goals or biases to justify explicit statements.

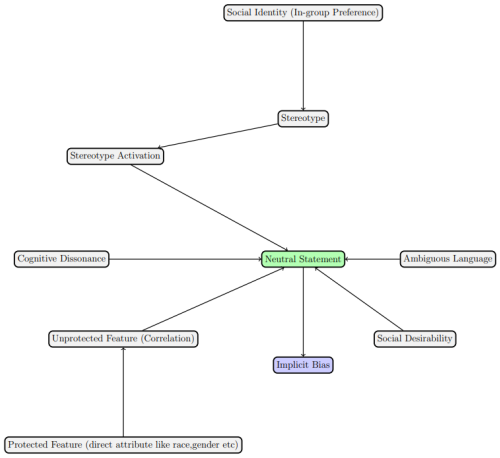

Figure 6: A DAG representing the causal pathway for a neutral sentence being transformed into an implicitly biased sentence

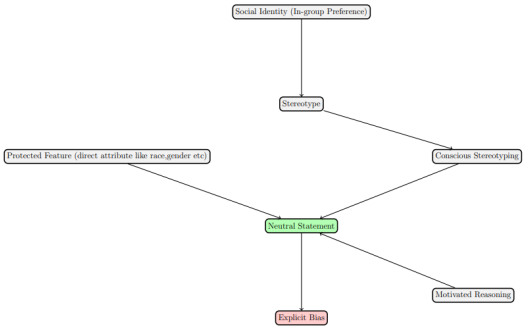

Figure 7: A DAG representing the causal pathway for a neutral sentence being transformed into an explicitly biased sentence

Implicit to Explicitly Biased Sentences (I2E))

This DAG explains the progression from implicit to explicit bias in language.

Key factors include **Social Identity**, which reflects the influence of group affiliation on decision-making and language; **Stereotype**, representing generalized beliefs about groups that shape perceptions and behavior; **Conscious Stereotyping**, which involves the deliberate application of stereotypes; **Protected Features**, referring to characteristics safeguarded under anti-discrimination policies that can still influence biases; and **Motivated Reasoning**, where reasoning is aligned with personal goals or biases to justify certain conclusions or actions.

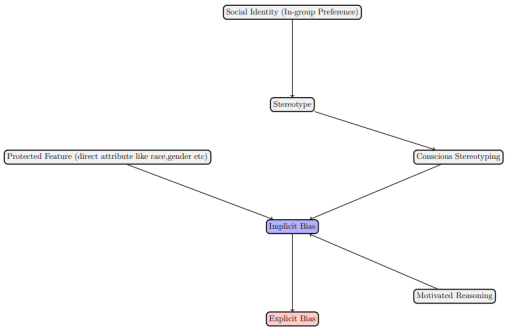

Figure 8: A DAG representing the causal pathway for a implicitly biased sentence being transformed into an explicitly biased sentence

Legal decision process under BNS scheme

The **key nodes** in the legal reasoning DAG under the Bhartiya Nyaya Sanhita (BNS) framework include the following: **Death Established**(D), which determines if a death has occurred, and **Intention to Cause Death**(ID), which assesses whether there was a clear intent to cause death, along with its counterpart, **No Intention to Cause Death**(!ID), for cases lacking such intent. Other critical nodes include **Falls Under Exceptions of BNS**(BNS) and **Does Not Fall Under Exceptions of BNS**(!BNS), which evaluate whether the act qualifies for legal exceptions. Additional nodes like **Intention to Cause Bodily Injury Likely to Cause Death**(IB) and **No Intention to Cause Bodily Injury Likely to Cause Death**(!IB) explore intent regarding bodily harm. The DAG also considers **Knowledge That Injury Is Likely to Cause Death**(KTI) versus **No Knowledge That Injury Is Likely to Cause Death**(!KTI), assessing the accused's awareness of fatal consequences. Severity is analyzed through nodes like **Injury Sufficient to Cause Death**(SD) and **Injury Not Sufficient to Cause Death**(!SD), as well as **High Probability That Death Would Be Caused**(HP) and **Not Very Likely to Cause Death**(!HP), which evaluate the likelihood of fatality. Finally, the outcomes are classified into **Murder**(M), **Culpable Homicide**(C), and **Non-Culpable Homicide**(NC), based on the interplay of intent, knowledge, and other factors.

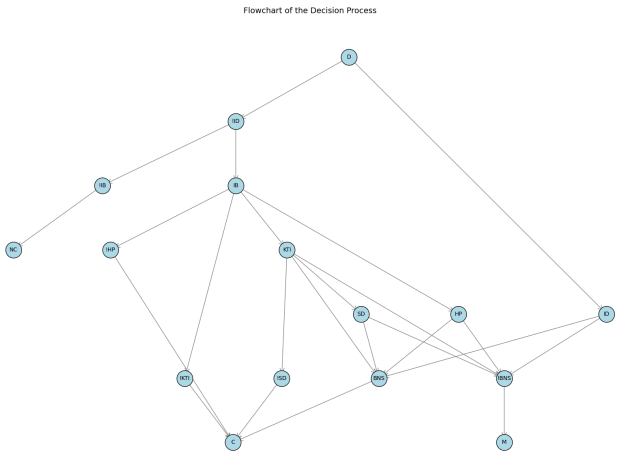

Figure 9: A DAG representing the causal pathway for legal decision process under BNS scheme

D Results and Analysis

In this section, we present the plots of the results for our framework: prompt-based and **causal_llm**, applied to synthetic (refer Figure 10) and bias & legal (refer Figure 11) datasets. The synthetic datasets cover varying complexities, including 10, 40, 70, and 100 nodes, with each set being evaluated under three different types of causal relationships: linear, quadratic, and Gaussian process (GP). These datasets serve as a benchmark for assessing the causal_llm model's ability to uncover causal structures across different levels of graph complexity and non-linearity. The following plots showcase the key performance metrics used to compare our framework with existing state-of-the-art causal discovery methods, offering a comprehensive analysis of the model's strengths and limitations (refer Figure 10).

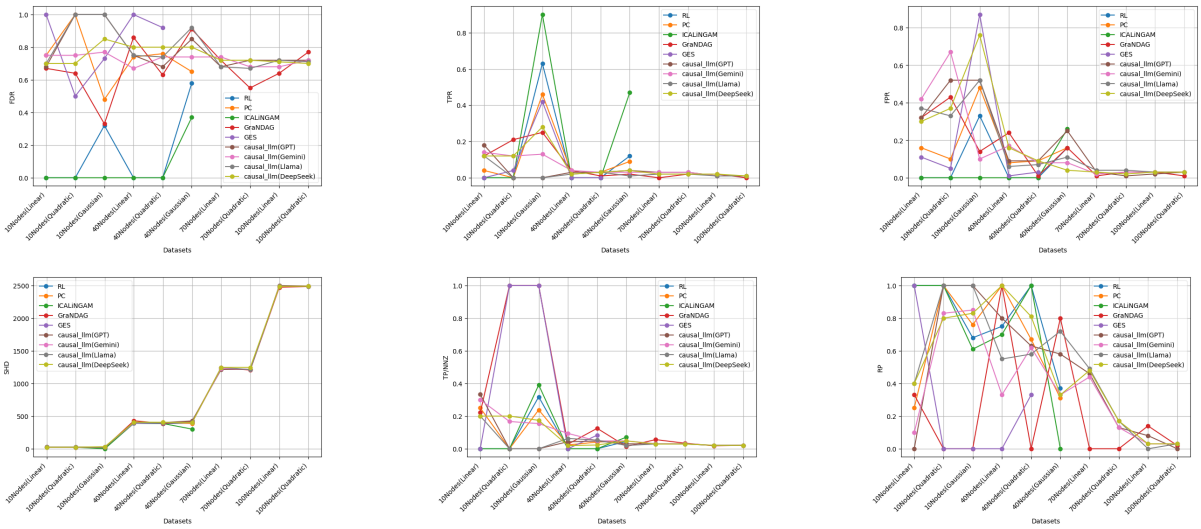

Figure 10: FDR, TPR, FPR, SHD, TP/NNZ and RP metrics for RL, PC, ICALiNGAM, GraNDAG, GES and causal_llm (with models GPT, Gemini, Llama and DeepSeek), plotted for the synthetic datasets (10, 40, 70 and 100 nodes for linear, quadratic and Gaussian models.

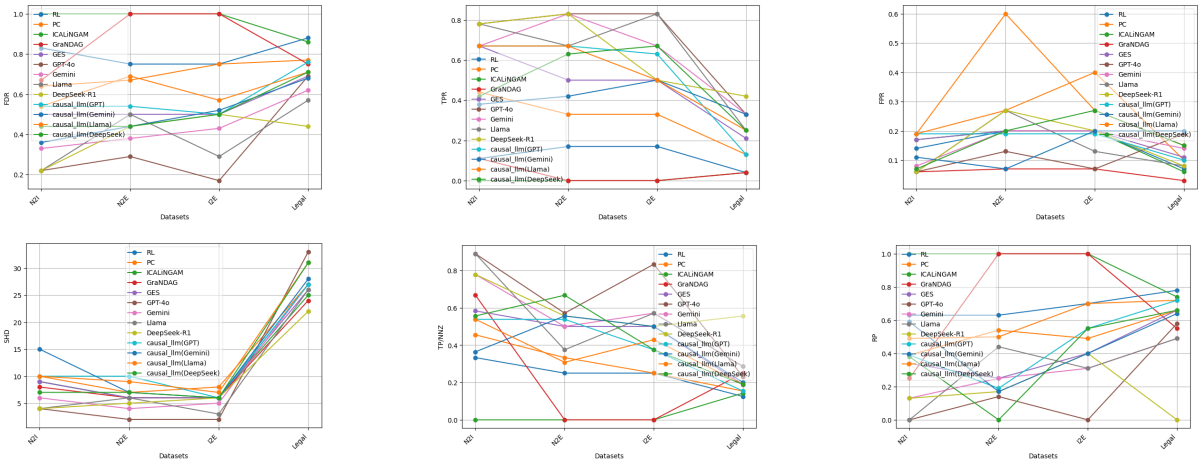

Figure 11: FDR, TPR, FPR, SHD, TP/NNZ and RP metrics for RL, PC, ICALiNGAM, GraNDAG, GES, GPT-4o, Gemini, Llama, DeepSeek-R1 and causal_llm (with models GPT, Gemini, Llama and DeepSeek), plotted for the synthetic datasets obtained from the domain knowledge causal graphs - BIAS (N2I, N2E and I2E) and LEGAL.

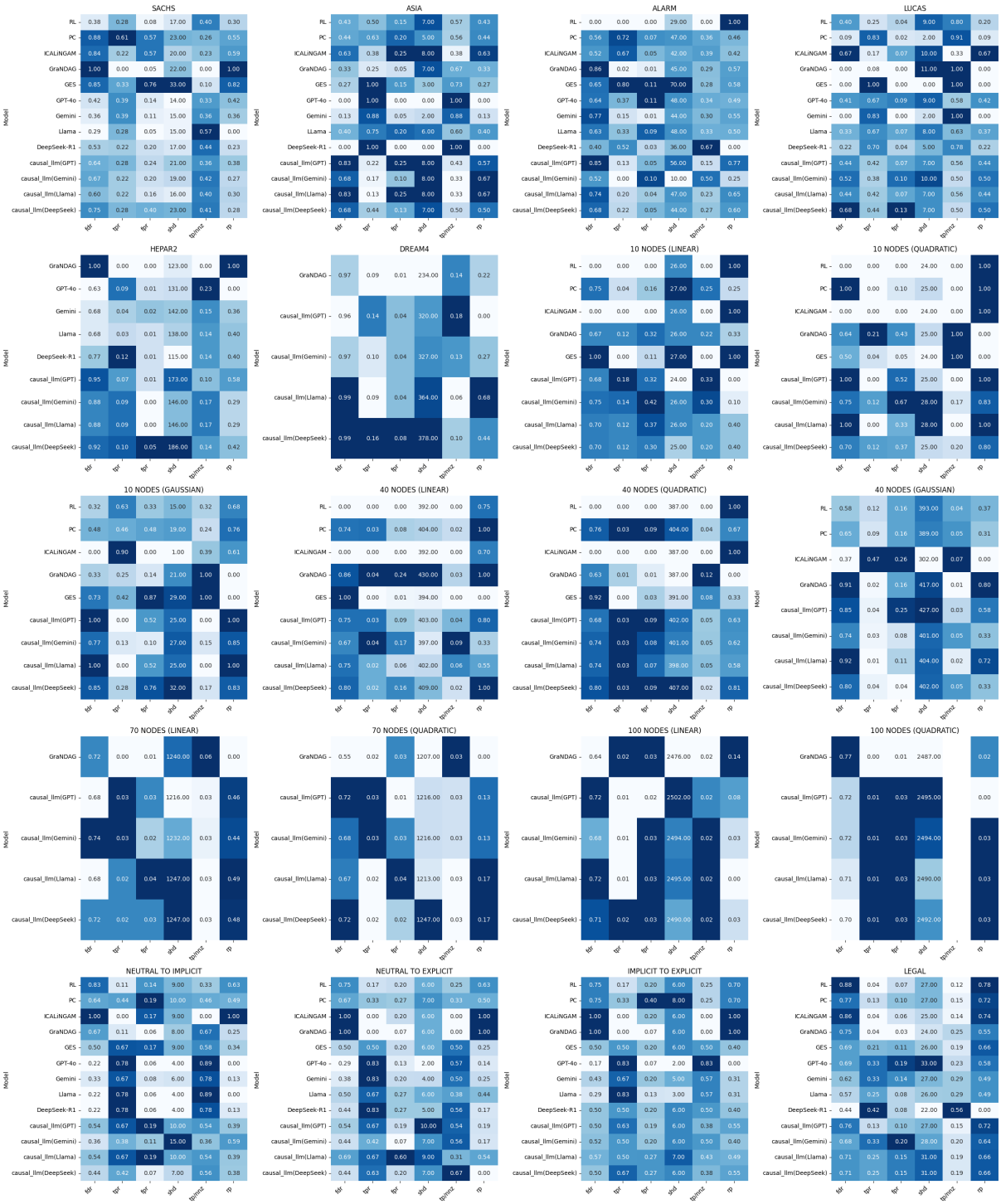

Figure 12: Performance metrics for all causal algorithms, including our causal_llm model and large language models (LLMs) like GPT, Gemini, Llama, and DeepSeek, are evaluated and plotted, comparing their performance on both publicly available and synthetic datasets.

D.1 GPT-4o vs DeepSeek: A Comparison

Among the prompt-based approaches, **GPT-4o** and **DeepSeek** consistently emerge as top-performing models (refer Figure 5).

In the **Sachs** dataset, GPT-4o exhibits superior performance with a high TPR, low FDR, and the best TP/NNZ, making it the most effective model for detecting true edges, while DeepSeek remains competitive.In the **Lucas** dataset, DeepSeek slightly outperforms GPT-4o by achieving a better TPR, lower FDR, and higher TP/NNZ. In the **Asia** dataset, both DeepSeek and GPT-4o achieve perfect metrics, producing the exact ground truth DAG.

As node complexity increases in datasets such as **ALARM and HEPAR2**, GPT-4o experiences a slight decline in performance, whereas DeepSeek remains consistent, achieving a higher TPR and lower FDR, particularly for higher-order nodes.

In conclusion, both **GPT-4o** and **DeepSeek** excel in prompt-based causal discovery. **GPT-4o** performs best on lower-order datasets like **Sachs**, while **DeepSeek** outperforms in **Lucas** and maintains consistency in higher-order datasets such as **ALARM** and **HEPAR2**, where **GPT-4o** declines slightly. **DeepSeek** proves to be more robust to increasing node order, making it a reliable choice for complex causal structures.

E New Metrics

Why these Metrics?

These metrics specifically assess the proportion of predicted edges that are actually true edges, unlike traditional precision, which accounts for both edge and non-edge predictions. In real-world datasets, ground truth causal graphs are typically sparse, meaning true edges are rare. As a result, traditional precision can be skewed by correctly identified non-edges, obscuring the model's performance in detecting actual causal relationships. By focusing solely on edge predictions, these metrics offer a more precise evaluation of the model's ability to uncover genuine causal links.

F Parameter Settings

We used various causal discovery methods based on constraints, functional causal model (FCM) based, score based, reinforcement learning based, and gradient based techniques, each configured with appropriate hyperparameters.We have used parameter initialization from *gcastle* causal discovery package (Zhang et al., 2021).

Parameter Settings for Baseline Causal Algorithms

Constraint-based approaches:
PC = PC(variant='original', alpha=0.05, ci_test='fisherz', priori_knowledge=None)

FCM-based methods:
ICA-LiNGAM = ICALiNGAM(random_state=None, max_iter=1000, thresh=0.3)

Score-based techniques:
GES = GES(criterion='bic', method='scatter', k=0.001, N=10)

RL-BIC= RL(encoder_type: str = 'TransformerEncoder', hidden_dim: int = 64, num_heads: int = 16, num_stacks: int = 6, residual: bool = False, decoder_type: str = 'SingleLayerDecoder', decoder_activation: str = 'tanh', decoder_hidden_dim: int = 16, use_bias: bool = False, use_bias_constant: bool = False, bias_initial_value: bool = False, batch_size: int = 64, input_dimension: int = 64, normalize: bool = False, transpose: bool = False, score_type: str = 'BIC', reg_type: str = 'LR', lambda_iter_num: int = 1000, lambda_flag_default: bool = True, score_bd_tight: bool = False, lambda2_update: int = 10, score_lower: float = 0, score_upper: float = 0, seed: int = 8, nb_epoch: int = 10, lr1_start: float = 0.001, lr1_decay_step: int = 5000, lr1_decay_rate: float = 0.96, alpha: float = 0.99, init_baseline: float = -1, l1_graph_reg: float = 0, verbose: bool = False, device_type: str = 'gpu', device_ids: int = 0)

Gradient-based methods:
GraNDAG = GraNDAG(input_dim, hidden_num: int = 2, hidden_dim: int = 10, batch_size: int = 64, lr: float = 0.001, iterations: int = 10000, model_name: str = 'NonLinGaussANM', nonlinear: str = 'leaky-relu', optimizer: str = 'rmsprop', h_threshold: float = 1e-7, device_type: str = 'cpu', device_ids: int = 0, use_pns: bool = False, pns_thresh: float = 0.75, num_neighbors: Any | None = None, normalize: bool = False, random_seed: int = 42, jac_thresh: bool = True, lambda_init: float = 0, mu_init: float = 0.001, omega_lambda: float = 0.0001, omega_mu: float = 0.9, stop_crit_win: int = 100, edge_clamp_range: float = 0.0001, norm_prod: str = 'paths', square_prod: bool = False)