LLM-initialized Differentiable Causal Discovery

Shiv Kampani
SandboxAQ
shiv.kampani@sandboxaq.com

David Hidary
SandboxAQ
david.hidary@sandboxaq.com

Constantijn van der Poel
SandboxAQ
constantijn@sandboxaq.com

Martin Ganahl
SandboxAQ
martin.ganahl@sandboxaq.com

Brenda Miao
SandboxAQ
brenda.miao@sandboxaq.com

Abstract

The discovery of causal relationships between random variables is an important yet challenging problem that has applications across many scientific domains. Differentiable causal discovery (DCD) methods are effective in uncovering causal relationships from observational data; however, these approaches often suffer from limited interpretability and face challenges in incorporating domain-specific prior knowledge. In contrast, Large Language Models (LLMs)-based causal discovery approaches have recently been shown capable of providing useful priors for causal discovery but struggle with formal causal reasoning. In this paper, we propose LLM-DCD, which uses an LLM to initialize the optimization of the maximum likelihood objective function of DCD approaches, thereby incorporating strong priors into the discovery method. To achieve this initialization, we design our objective function to depend on an explicitly defined adjacency matrix of the causal graph as its only variational parameter. Directly optimizing the explicitly defined adjacency matrix provides a more interpretable approach to causal discovery. Additionally, we demonstrate higher accuracy on key benchmarking datasets of our approach compared to state-of-the-art alternatives, and provide empirical evidence that the quality of the initialization directly impacts the quality of the final output of our DCD approach. LLM-DCD opens up new opportunities for traditional causal discovery methods like DCD to benefit from future improvements in the causal reasoning capabilities of LLMs.

1 Introduction

Discovering causal relationships is a fundamental task across scientific fields including epidemiology, genetics, and economics. For a variety of reasons – ethical, logistical, legal – it may not be possible to conduct controlled experiments or interventional studies to generate Causal Graphical Models (CGM) that allow for causal inference. Consequently, there has been a shift towards the development of *causal discovery* methods that infer CGMs from observations about the variables alone.

1.1 Background

Causal discovery is the problem of learning a CGM from a set of observed data points. Each observation $n, n = 1, \ldots, N$, can be characterized as a specific realization of d random variables

NeurIPS 2024 Workshop on Causality and Large Models (CaLM).

$V = \{v_1, \ldots, v_d\}$, where the variable v_j can take on discrete or continuous values. In this work we restrict ourselves to discrete random variables. We use $\mathbf{x}^n$ to denote the vector of d discrete values $x_j^n, j = 1, \ldots, d$ of the n-th observation, and denote the table of all N observations as $X \equiv \{\mathbf{x}^n\}_{n=1}^N$.

A CGM for the variables V consists of two components: 1) a directed acyclic graph (DAG) $G = (V, E)$ with nodes V and *directed* edges $E \equiv \{e_1, \ldots e_M\} \subseteq V \times V$ of (ordered) pairs $e \equiv (v_a, v_b)$, with the order implying causation, i.e. v_a causes v_b in this notation. 2) a set of d conditional probability distributions $p(\, v_j \mid \mathrm{Pa}(v_j, G))$, where $\forall j \in [\, d\,]$, $\mathrm{Pa}(v_j, G) \subseteq V$is the set of *causal parents* or direct causes of v_j according to G. In this work we do not consider interventions, although they can be incorporated into our framework.

Since causal discovery is an NP-hard problem (Chickering et al. [2004]), a variety of approaches have been studied to improve the efficiency and accuracy of causal discovery. These methods often fall into three categories - score-based methods (SBMs), differentiable causal discovery (DCD), and Large Language Model (LLM)-based approaches.

Score-based methods. SBMs formulate causal discovery as the maximization of a log-likelihood objective function

$$\mathcal{L}(G, \theta; X) = \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{d} \log p(v_j = x_j^n \,;\, X, \theta, G) \tag{1}$$

with respect to parameters θ and the graph G (regularization penalties, typically added to the cost function, are omitted here for simplicity). The functions $p(v_j^n \,;\, X, \theta, G)$ are ansatz functions for the conditional probability distributions $p(\, v_j \mid \mathrm{Pa}(v_j, G))$. Brouillard et al. [2020] showed that, under certain regularity assumptions (see Appendix A.5), the maximizer of this objective function is *Markov equivalent* to the ground-truth CGM.

In this approach, acyclicity and directedness can be enforced by applying suitable constraints to the discrete optimization problem. An advantage is that the optimization is amenable to standard combinatiorial optization methods; however, due to the super-exponential growth of the solution space with the size of the graph, score-based approaches become quickly intractable (Meek [1997]).

Differentiable causal discovery. More recent approaches formulate causal discovery as a continuous optimization problem over the adjacency matrix A_θ of the graph G and parameters θ. The notation A_θ here means that the adjacency matrix can be an arbitrary differentiable function of the parameter θ. Acyclicity and directedness constraints of the graph are incorporated into the optimization by a differentiable penalty function $h(A_\theta)$, i.e.

$$\mathcal{L}(A_\theta, \theta; X) = \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{d} \log p(v_j = x_j^n \,;\, X, \theta, A_\theta) - \beta h(A_\theta) \tag{2}$$

The function $h(A)$ needs to satisfy the conditions (Nazaret et al. [2024])

$$h(A) \geq 0 \tag{3}$$

$$h(A) = 0 \iff G \text{ a directed acyclic graph.} \tag{4}$$

The objective is then maximized using standard gradient ascent optimization for parameters A_θ and θ. Acyclicity and directedness are in this case not strictly enforced during the optimization, but approached during the maximization of the objective function.

Zheng et al. [2018] were the first to propose NOTEARS, a differentiable causal discovery approach based on a trace-exponential acyclicity constraint $h(A_\theta) = \mathrm{Tr}(\exp(A_\theta)) - d$. Bello et al. [2023] later on introduced DAGMA, a direct improvement over NOTEARS owing to an alternate log-det-based acyclicity constraint. Nazaret et al. [2024] proposed a more stable acyclicity constraint in the form of the spectral radius of A_θ, i.e. $h(A_\theta) = |\lambda_d|$, with λ_d the largest eigenvalue of the matrix. Note that the spectral radius of the adjacency matrix of a DAG is identically 0 (A is a nilpotent matrix in this case). We employ the spectral acyclicity constraint in this work.

LLM-based approaches. LLMs have shown promise in being able to evaluate pairwise causal relationships between variables of interest (Kıcıman et al. [2023], Liu et al. [2024]). However, LLM-based approaches are often unable to perform formal causal reasoning and can exhibit inconsistencies; it is also difficult to distinguish true causal reasoning from memorization (Jin et al. [2024]). Also, unlike previous approaches, LLMs are not designed to leverage observational data effectively.

1.2 Problem setup.

Recent approaches have sought to merge the advantages of LLM-based and SBM approaches. Darvariu et al. [2024] showed that LLMs can provide effective priors to improve score-based algorithms. LLMs have also been used specify constraints for SBMs (Ban et al. [2023]), by orienting edges in partial CGMs discovered by other numerical approaches (Long et al. [2023]), or by providing a "warm-start" or initial point for combinatorial search-based methods (Vashishtha et al. [2023]). These results suggest **LLMs may also be able to complement and improve the performance of state-of-the-art DCD methods**. However, integrating LLM-based causal discovery methods with DCD is challenging due to *non-interpretable* adjacency matrices.

The function $p(\cdot)$ used to model conditional distributions in the DCD loss, Eq. (2), is represented using a neural network with parameters θ. The adjacency matrix W_θ of the CGM DAG is then implicitly defined from θ, by taking a norm over the first layer of the network. Waxman et al. [2024] show that when $p(\cdot)$ is modelled using a Multi-Layer Perceptron (MLP) with sigmoid or ReLU activation (as in NOTEARS, DAGMA, and SDCD), the implicit adjacency matrix, W_θ, from the first layer of the network may be arbitrarily different from the true causal relationships found by taking derivatives over all model layers. The MLP parameters θ do not directly relate to the implied adjacency matrix W_θ, so the implicit adjacency matrix W_θ is **non-interpretable**.

Said another way, suppose an LLM (correctly) reasons that variable v_a directly causes v_b. It is clear that $(W_\theta)_{ab} = 1$. However, it is not clear how to modify the MLP parameters, θ, to encode this information and leverage information from LLMs: hence the non-interpretability.

1.3 Contributions

In this work we propose a novel combination of large language models (LLM) with DCD. The two key aspects of our approach are:

1. an ansatz function $p(v_j = x_j^n\,;\, X, A)$ in Eq.(9) [`MLE-INTERP`], which depends on elements $a_{jk} \geq 0$ of an explicitly defined adjacency matrix A as the only variational parameters. `MLE-INTERP` does not rely on an MLP or neural network to model conditional distributions.
2. the usage of LLMs for parameter initialization of the adjacency matrix A. This is possible because we use an ansatz function depending on an explicitly defined adjacency matrix.

We make the following assumptions in our approach:

1. *Faithfulness, causal Markov condition:* conditional independence holds in the observational data table X if and only if the corresponding d-separation holds in the CGM.
2. *Causal sufficiency:* there are no hidden or latent confounders.

We coin our approach LLM-initialized Differentiable Causal Discovery (LLM-DCD). Our method is applicable to datasets with discrete-valued variables, and we do not make any assumptions about conditional distributions. This addresses a limitation in DCD methods like SDCD that assume that conditionals are normally-distributed. To our knowledge, LLM-DCD is the first method to integrate LLMs with differentiable causal discovery.

We prove that our model satisfies the standard regularity assumptions for DCD (Brouillard et al. [2020], Appendix A.5) and benchmark the performance of LLM-DCD on five datasets ranging from 5-70 discrete-valued variables and limited observational data. LLM-DCD outperforms existing methods and scales reasonably with time. We also empirically show that the quality of LLM-DCD depends on the quality of its initialization; thus, the development of higher quality LLM-based causal discovery methods is expected to have direct impact on the quality of LLM-DCD.

All code from this work is available at: `https://github.com/sandbox-quantum/llm-dcd`

2 LLM-initialized Differentiable Causal Discovery

We consider the differentiable objective function

$$\mathcal{L}(A; X) = \frac{1}{n} \sum_{n=1}^{N} \sum_{j=1}^{d} \log \texttt{MLE-INTERP}(x_j^n; X, A) - \alpha ||A||_1 - \beta_t |\lambda_d| \,. \tag{5}$$

with an ansatz function $\texttt{MLE-INTERP}(x_i^n; X, A)$ to model the conditional probability $p(v_i = x_i | \{v_k = x_k\}_{k \neq i}) \equiv \frac{p(x_i, \{x_k\}_{k\neq i})}{p(\{x_k\}_{k\neq i})}$ of observation x_i given values $\{x_k\}_{k\neq i}$ for the variables $\{v_k\}_{k\neq i}$ from the set of training data X.

Specifically, we use a maximum-likelihood based estimator of this conditional probability which is computed from ratios of frequency counts of observations in the training data. In the following we use $\text{cnt}(x_{j_1}, x_{j_2}, \dots x_{j_M})$ to denote the number of samples in X with values $x_{j_1}, x_{j_2}, \dots x_{j_M}$.

Consider as an example the case of two variables v_1, v_2, with a_{21} the only possible non-zero element of A. We use the following ansatz for the conditional distribution $p(v_1 = x_1 | v_2 = x_2)$ is in this case:

$$\texttt{MLE-INTERP}(x_1; X, A) = \frac{\text{cnt}(x_1)(1 - a_{21}) + \text{cnt}(x_1, x_2)a_{21}}{N(1 - a_{21}) + \text{cnt}(x_2)a_{21}}. \tag{6}$$

In the limits of $a_{21} = 1$ and $a_{21} = 0$, the function reduces to the expected expressions $\frac{\text{cnt}(x_1,x_2)}{\text{cnt}(x_2)}$ and $\frac{\text{cnt}(x_1)}{N}$, respectively. For values $0 < a_{21} < 1$ the nominator and denominator are taken to be a linear interpolation between the counts of the two edge cases. This ansatz can be straightforwardly generalized to d variables:

$$\texttt{MLE-INTERP}(x_i; X, A) = \frac{\text{cnt}(x_i) \prod_{k\neq i}^{d}(1 - a_{ki}) + \sum_{j\neq i} \text{cnt}(x_i, x_j) a_{ji} \prod_{k\neq\{i,j\}}(1 - a_{ki}) + \dots}{N \prod_{k\neq i}(1 - a_{ki}) + \sum_j \text{cnt}(x_j) a_{ji} \prod_{k\neq j}(1 - a_{ki}) + \dots}. \tag{7}$$

We emphasize that despite the seeming complexity, this function can be evaluated in $\mathcal{O}(Nd)$ time; see also Algorithm 1 in the Appendix. Eq.(7) can be rewritten in a more succinct form: using the notation $\bar{x}_j$ to denote all values different from x_j, such that e.g. $\text{cnt}(x_1, \bar{x}_2)$ are the number of samples in X with values $v_1 = x_1$ and values $v_2 \neq x_2$, and with the shorthand

$$\delta_k(x_j) = \begin{cases} x_j \text{ if } k = 0 \\ \bar{x}_j \text{ if } k = 1 \end{cases} \tag{8}$$

Eq.(7) can be recast into

$$\texttt{MLE-INTERP}(x_i; X, A) = \frac{\sum_{\{i_k\}_{k=1\dots d-1}\in\{0,1\}} \text{cnt}(x_i, \{\delta_{i_k}(x_{j_k})\}_{j_k\neq i}) \prod_{k=1}^{d-1}(1 - i_k a_{j_k i})}{\sum_{\{i_k\}_{k=1\dots d-1}\in\{0,1\}} \text{cnt}(\{\delta_{i_k}(x_{j_k})\}_{j_k\neq i}) \prod_{k=1}^{M_i}(1 - i_k a_{j_k i})}. \tag{9}$$

where $\{x_k\}_{k\neq i}$ denotes all variables x_k with $k \neq i$.

Eq.(9) evidently is a rational function of the the elements of the adjacency matrix A, for which gradients w.r.t. a_{ji} can be computed efficiently. To improve stability during the optimization, we replace factors $(1 - i_k a_{j_k i})$ in Eq.(9) with a third order polynomial $g(1 - i_k a_{j_k i})$. We chose g as follows because it is a smooth function that satisfies $g(0) = 0, g(1) = 1$, and $\forall x \in [0, 1], g'(x) \neq 0$.

$$g(x) = 0.15x + 2.55x^2 - 1.7x^3. \tag{10}$$

The objective function Eq.(5) is maximized using an Adam optimizer (Kingma and Ba [2015]) with mini-batch gradient ascent. Note that spectral acyclicity, $h(A) = 0$, is included in the objective via the penalty $\beta_t \cdot h(A)$ with a step-dependent coefficient β_t. The optimization is carried out over two stages, as in described in Nazaret et al. [2024]. In the first stage, β_t is kept 0 for t_1 iterations, with the goal of maximizing log-likelihood. In the second stage, β_t is incremented by δ at every timestep, with the goal of ensuring acyclicity, for at least t_2 iterations, until A converges.

2.1 LLM-initialization

Since the model parameters are exactly the adjacency matrix A of the CGM, we may choose to initialize the optimization process with a "warm start" adjacency matrix A_0 provided by an LLM. We consider "warm starts" provided both by the pairwise (PAIR) and breadth-first-search (BFS):

1. **Pairwise LLM queries.** This is a straightforward LLM-based solution to the problem of causal discovery introduced by Kıcıman et al. [2023]. For every pair of variables (v_a, v_b), an LLM is asked to reason whether v_a directly causes v_b, v_b directly causes v_a, or that there is no direct causal relationship between the two variables.

2. **LLM breadth-first-search (BFS).** The number of queries made by the pairwise LLM method scales as $O(d^2)$, where d is the number of variables. Jiralerspong et al. [2024] proposed a three-stage BFS-based algorithm that requires only $O(d)$ queries.

For both methods, we also provided the LLMs with pairwise correlational coefficients where relevant. In Section 4, we provide empirical evidence that LLM-initialization improves the quality of LLM-DCD over previous state-of-the-art methods.

3 Experimental Setup

We benchmark the performance of LLM-DCD against previous score-based method GES, differentiable methods SDCD and DAGMA, and the aforementioned LLM-based methods (PAIR and BFS) on the following CGM datasets from the `bnlearn` package (Scutari [2010]): `cancer` (5 variables, 4 causal edges), `sachs` (11 variables, 17 causal edges), `child` (20 variables, 25 causal edges), `alarm` (37 variables, 46 causal edges), and `hepar2` (70 variables, 123 causal edges).

All observational data ($n = 1000$ observations for each experiment) are generated by independent forward-sampling from the ground-truth CGM joint distribution. We run LLM-DCD with a random initialization, initialized with the output of PAIR (denoted LLM-DCD-P or LLM-DCD (PAIR)), and initialized with the output of BFS (denoted LLM-DCD-B or LLM-DCD (BFS)). Each method was run with three global seeds (0,1,2) on each dataset. Consistent with previous works, we run LLM-DCD with a fixed set of hyperparameters (Appendix A.2) in all experiments.

Metrics We report performance of LLM-DCD and baseline models using structural Hamming distance (SHD) between the predicted and true CGMs. SHD is a standard causal discovery metric and is measured as the number of causal edge insertions, deletions, or flips required to transform the current CGM DAG into the ground-truth CGM DAG. We also provide results based on precision, recall, F1-score, and runtime (seconds). Results are reported as means $\pm$ standard deviation.

4 Results

Theoretical results. In Appendix A.5, we prove that the `MLE-INTERP` model used by LLM-DCD satisfies regularity assumptions introduced by Brouillard et al. [2020]. As a corollary, in limit of infinite samples ($n \to \infty$), LLM-DCD outputs a CGM that is *Markov equivalent* to the ground-truth CGM. We also provide an analysis of the time-complexity of LLM-DCD in Appendix A.3. The current implementation of LLM-DCD calculates $h'(A)$ naïvely in $O(d^3)$-time, but future implementations are likely to benefit from utilizing the recently developed $O(d^2)$-time power-iteration algorithm from Nazaret et al. [2024].

Table 1: Performance (SHD) of causal discovery methods on benchmarking datasets

	Dataset				
Method	Cancer (n=5)	Sachs (n=11)	Child (n=20)	Alarm (n=37)	Hepar2 (n=70)
SDCD	2.00 ± 0.00	$\mathbf{12.3 \pm 2.08}$	35.7 ± 0.58	$131. \pm 11.6$	$230. \pm 13.6$
GES	2.67 ± 1.15	$\mathbf{14.3 \pm 1.53}$	$\mathbf{26.7 \pm 3.51}$	60.5 ± 0.71	–
DAGMA	4.00 ± 0.00	25.3 ± 1.53	32.7 ± 0.58	88.3 ± 9.29	$272. \pm 6.43$
PAIR	4.00 ± 0.00	31.0 ± 0.00	97.3 ± 10.2	$207. \pm 9.54$	–
BFS	$\mathbf{1.00 \pm 1.00}$	16.3 ± 2.08	36.3 ± 3.51	64.0 ± 5.29	$173. \pm 13.1$
LLM-DCD	3.67 ± 0.58	23.7 ± 9.02	49.7 ± 5.51	58.0 ± 1.73	$318. \pm 8.89$
LLM-DCD-P	2.67 ± 0.58	21.0 ± 3.46	41.0 ± 3.61	60.3 ± 18.8	–
LLM-DCD-B	$\mathbf{1.00 \pm 1.00}$	$\mathbf{14.3 \pm 0.58}$	$\mathbf{23.7 \pm 0.58}$	$\mathbf{36.3 \pm 5.51}$	$\mathbf{140. \pm 6.66}$

Evaluation Results Table 1 shows the performance of LLM-DCD-B on several benchmarking datasets of varying sizes. Lower SHD indicates better performance. LLM-DCD (BFS) outperformed all baseline SBM, DCD, and LLM based approaches in the Alarm and Hepar2 datasets, and achieved results that were comparable to the top-performing models on the Cancer, Sachs, and Child datasets.

Results are not reported for methods with intractable runtimes, including GES, PAIR, and LLM-DCD (Pair) on the Hepar2 dataset.

In Figure 4, we observe similar trends for F1-score, Precision, and Recall, with LLM-DCD-B consistently outperforming other methods across datasets. The runtime of LLM-DCD scales worse than that of SDCD, which remains roughly constant, although LLM-DCD (BFS) was still more efficient than GES, PAIR, and DAGMA (Figure 4).

We further show that initialization of the adjacency matrix in LLM-DCD affects performance, with higher quality initializations tending to result to better performance across metrics and datasets. LLM-DCD (BFS) tended to outperform LLM-DCD (PAIR) and the randomly initialized LLM-DCD. Additionally, we show that the randomly initialized LLM-DCD method has performance that is comparable to existing DCD methods like DAGMA and SDCD.

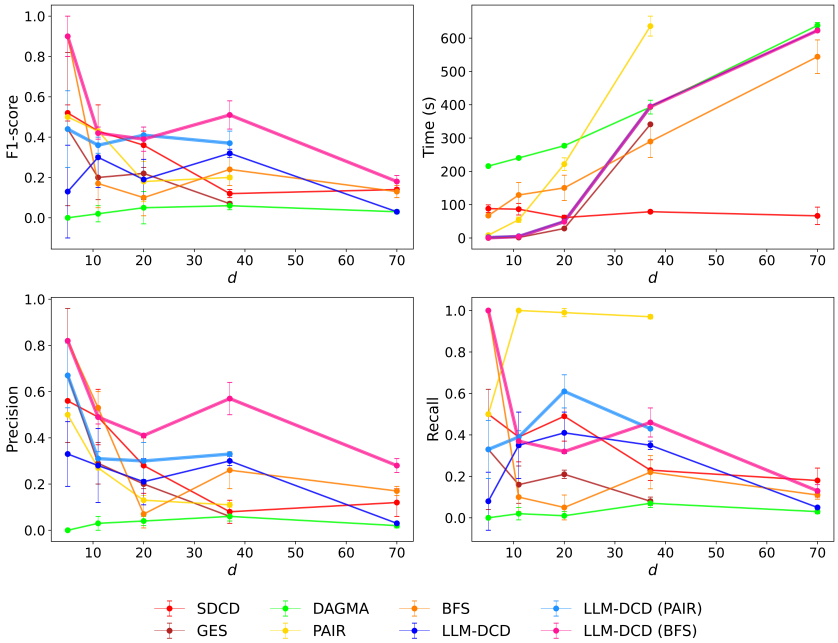

Figure 1: F1-score, precision, recall, and runtime of causal discovery methods on observational datasets of different sizes (d).

5 Conclusion

We developed LLM-DCD, integrating LLMs with DCD to take advantage of the prior knowledge learned by LLMs while maintaining the performance and computational efficiency of DCD approaches for causal discovery. LLM-DCD outperforms previous state-of-the-art approaches across several causal discovery benchmarking datasets. Although not as efficient as the recent developed SDCD, LLM-DCD is more scalable than several other baseline SBM and DCD methods. Future implementations of LLM-DCD may be able to directly integrate the computational optimization of SDCD Nazaret et al. [2024] to improve scalability while preserving state-of-the-art performance.

Because our approach also directly optimizes an explicitly defined adjacency matrix, LLM-DCD also provides a more interpretable approach to causal discovery. Additionally, LLM-DCD directly benefits from higher quality initializations of this adjacency matrix, and can take advantage of future advancements in LLM reasoning and causal inference capabilities. Other future work may investigate how the size and reasoning capabilities of various LLMs can affect initialization of the adjacency matrix and downstream performance of LLM-DCD. As LLM-based causal discovery evolves, the performance of LLM-DCD is only expected to improve as well.

References

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, 2023. URL https://arxiv.org/abs/2306.16902.

K. Bello, B. Aragam, and P. Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization, 2023. URL https://arxiv.org/abs/2209.08037.

P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin. Differentiable causal discovery from interventional data, 2020. URL https://arxiv.org/abs/2007.01754.

D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *J. Mach. Learn. Res.*, 5:1287–1330, dec 2004. ISSN 1532-4435.

V.-A. Darvariu, S. Hailes, and M. Musolesi. Large language models are effective priors for causal graph discovery, 2024. URL https://arxiv.org/abs/2405.13551.

Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. Lyu, K. Blin, F. G. Adauto, M. Kleiman-Weiner, M. Sachan, and B. Schölkopf. Cladder: Assessing causal reasoning in language models, 2024. URL https://arxiv.org/abs/2312.04350.

T. Jiralerspong, X. Chen, Y. More, V. Shah, and Y. Bengio. Efficient causal graph discovery using large language models, 2024. URL https://arxiv.org/abs/2402.01207.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

E. Kıcıman, R. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality, 2023. URL https://arxiv.org/abs/2305.00050.

X. Liu, P. Xu, J. Wu, J. Yuan, Y. Yang, Y. Zhou, F. Liu, T. Guan, H. Wang, T. Yu, J. McAuley, W. Ai, and F. Huang. Large language models and causal inference in collaboration: A comprehensive survey, 2024. URL https://arxiv.org/abs/2403.09606.

S. Long, A. Piché, V. Zantedeschi, T. Schuster, and A. Drouin. Causal discovery with language models as imperfect experts, 2023. URL https://arxiv.org/abs/2307.02390.

C. Meek. Graphical Models: Selecting causal and statistical models. 1 1997. doi: 10.1184/R1/22696393.v1. URL https://kilthub.cmu.edu/articles/thesis/Graphical_Models_Selecting_causal_and_statistical_models/22696393.

A. Nazaret, J. Hong, E. Azizi, and D. Blei. Stable differentiable causal discovery, 2024. URL https://arxiv.org/abs/2311.10263.

M. Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v035i03.

A. Vashishtha, A. G. Reddy, A. Kumar, S. Bachu, V. N. Balasubramanian, and A. Sharma. Causal inference using llm-guided discovery, 2023. URL https://arxiv.org/abs/2310.15117.

D. Waxman, K. Butler, and P. M. Djurić. Dagma-dce: Interpretable, non-parametric differentiable causal discovery. *IEEE Open Journal of Signal Processing*, 5:393–401, 2024. ISSN 2644-1322. doi: 10.1109/ojsp.2024.3351593. URL http://dx.doi.org/10.1109/OJSP.2024.3351593.

X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018. URL https://arxiv.org/abs/1803.01422.

Algorithm 1 `MLE-INTERP`$(x_j^i\,;\,X, A)$

Input: i, j, X, A
`num, den` $\leftarrow 0$
for $k \in [n]$ **do**
  `numprod, denprod` $\leftarrow 1$
  for $m \in [d]$ **do**
    `numprod` $\leftarrow$ `numprod` $\cdot \big((1 \text{ if } \mathbf{x}_m^k = \mathbf{x}_m^i \text{ else } g(1-a_{mj})) \text{ if } \mathbf{x}_j^k = \mathbf{x}_j^i \text{ else } 0\big)$
    `denprod` $\leftarrow$ `denprod` $\cdot\, (1 \text{ if } \mathbf{x}_m^k = \mathbf{x}_m^i \text{ or } m = j \text{ else } g(1-a_{mj}))$
  end for
  `num` $\leftarrow$ `num` $+$ `numprod`
  `den` $\leftarrow$ `den` $+$ `denprod`
end for
return `(num / den)`

A Appendix

A.1 Implementation details of `MLE-INTERP` and gradient ascent optimization

Algorithm (1) details our implementation of Eq.(9) in the main text.

The maximization of the objective function is carried out using the `LLM-DCD` algorithm summarized in Algorithm (2). For more details on hyperparameter settings we refer the reader to Appendix A.2.

Algorithm 2 `LLM-DCD`

Input: $X, A_0, i_1, i_2, \text{lr}, \alpha, \delta, \gamma_1, \gamma_2, \delta, \varepsilon$
$A_t \leftarrow A_0$
$M_t, V_t \leftarrow 0^{d\times d}$
$t, \beta_t \leftarrow 0$
while $t \leq i_1 + i_2$ and *not converged* **do**
  $t \leftarrow t + 1$
  $\mathtt{D} \leftarrow$ gradient of objective function $\mathcal{L}(X; A)$ with respect to W
  $M_t \leftarrow (\gamma_1 M_t + (1-\gamma_1)\mathtt{D})/\gamma_1^t$
  $V_t \leftarrow (\gamma_2 V_t + (1-\gamma_2)(\mathtt{D} \odot \mathtt{D}))/\gamma_2^t$
  $A_t \leftarrow \text{clip}(A_t + \text{lr} \cdot \mathtt{divide}(M_t, \sqrt{V_t} + \varepsilon), 0, 1)$
  $\beta_t \leftarrow \beta_t + (\delta \text{ if } t > i_1 \text{ else } 0)$
end while

In `LLM-DCD`, $\odot$ denotes element-wise multiplication and $\mathtt{divide}(\cdot)$ denotes element-wise division. The gradient, `D`, of the objective function can be computed using `D-MLE-INTERP`$(x_j^i\,;\,C, A)$, summed over all i, j and by computing the derivative of the spectral acyclicity constraint, $h(A)$. For an algorithm to calculate `D-MLE-INTERP` with respect to W, see Appendix (A.4).

A.2 LLM-DCD Hyperparameters

For the LLM-DCD algorithm specified in Algorithm 2, we use the following hyperparameters: $i_1 = 200, i_2 = 400, \text{lr} = 0.025, \alpha = 0.120, \delta = 0.040, \gamma_1 = 0.80, \gamma_2 = 0.90, \varepsilon = 10^{-8}$. The only change across all datasets was in the minibatch-size (b) selected ($b = 125$ for `hepar2`, $b = 250$ for `alarm` and $b = 500$ for other datasets). Black-box implementations for all other methods were used with no modifications, except for DAGMA, for which we ran $T = 1$ iterations instead of $T = 4$, to keep runtimes comparable with other methods. For LLM methods, we used OpenAI GPT-4 (`gpt-4-0613`) with default settings. All experiments were conducted on a virtual machine provided by Google Colaboratory, utilizing an NVIDIA A100 Tensor Core GPU.

A.3 Time-complexity analysis

$\forall i, j,$ `D-MLE-INTERP`$(x_j^i; X, A)$ (Appendix A.4) can be computed in $O(nd^3)$-time. Since `D` requires a sum over all i, j, $O(n^2d^4)$-time is needed. This computation, however, is easily parallelizable; using

a GPU, only $O(nd)$ steps need to be carried out sequentially. Using mini-batch gradient descent, we can reduce the time-complexity to $O(b^2d^4)$-time (with only $O(bd)$ sequential steps) (b = mini-batch size). Computing $h'(A)$ naïvely requires $O(d^3)$-time in our current implementation, although we plan to use the $O(d^2)$-time power-iteration algorithm fromNazaret et al. [2024] in future studies.

A.4 Derivative of `MLE-INTERP`

The derivative of `MLE-INTERP` (`D-MLE-INTERP`) with respect to W.

Algorithm 3 `D-MLE-INTERP`$(x_j^i\,;\,X, A)$

```
Input: i, j, X, A
DLL ← 0^{d×d}
for a ∈ [d] do
    for b ∈ [d] : a ≠ b do
        num, den, dnum, dden ← 0
        for k ∈ [n] do
            numprod, denprod, dnumprod, ddenprod ← 1
            for m ∈ [d] do
                numterm ← ((1 if x_m^k = x_m^i else g(1 − a_mj)) if x_j^k = x_j^i else 0)
                denterm ← (1 if x_m^k = x_m^i or m = j else g(1 − a_mj))
                numprod ← numprod · numterm
                denprod ← denprod · denterm
                if a = m and b = j then
                    dnumprod ← dnumprod · (0 if x_m^k = x_m^i or x_j^k = x_j^i else g'(1 − a_mj))
                    ddenprod ← ddenprod · (0 if x_m^k = x_m^i or m = j else g'(1 − a_mj))
                else
                    dnumprod ← dnumprod · numterm
                    ddenprod ← ddenprod · denterm
                end if
            end for
            num ← num + numprod
            den ← den + denprod
            dnum ← dnum + dnumprod
            dden ← dden + ddenprod
        end for
        DLL[a, b] ← (dnum / num − dden / den)
    end for
end for
return DLL
```

A.5 Regularity assumptions

Theorem (Brouillard et al. [2020])**.** *Under the following **regularity assumptions** (these assumptions have been simplified from the original assumptions listed in Brouillard et al. [2020], since we do not consider interventions):*

1. $n \geq 0$
2. *Faithfulness; causal Markov condition; causal sufficiency.*
3. *The joint-distribution of the latent or ground-truth CGM and the all distributions from the model class (`MLE-INTERP`) have strictly-positive density.*
4. *The model class is able to express the latent CGM conditional distributions in the limit of infinite samples ($n \to \infty$). This assumption is implicit in their work.*

Any CGM that maximizes of the DCD objective function is equivalent to the ground-truth CGM upto a Markov equivalence class (i.e., the CGM is Markov equivalent to the ground-truth CGM).

Theorem (Regularity of `LLM-DCD`)**.** *The model used by `LLM-DCD` satisfies regularity assumptions, in the limit of infinite observational samples ($n \to \infty$).*

Proof. Assumption 1 is trivially satisfied, since we always include a non-empty observational data table X. Assumption 2 is included in the assumptions of this work. For Assumption 3, refer to the implementation of `MLE-INTERP`. The numerator term (`num`) is non-zero whenever there is at least one row in X where every variable of interest takes on every one of its possible values from its finite, discrete set of values. If the joint-distribution of the latent or ground-truth CGM has strictly positive density (as is the case for all chosen datasets), Assumption 3 holds true for all distributions from the model class `MLE-INTERP` in the limit of infinite samples ($n \to \infty$). Finally, we must show that Assumption 4 holds true for `MLE-INTERP`.

Let W^* be the adjacency matrix of the latent CGM DAG, and W be the adjacency matrix used in `LLM-DCD`. We will show that when $W = W^*$, `MLE-INTERP` models the latent CGM joint distribution. Based on the implementation of `MLE-INTERP`:

$$\texttt{MLE-INTERP}(x_j^i; X, A^*) = \frac{\text{cnt}\left(\mathbf{x}_j^i, \text{Pa}(v_j, A^*) = \mathbf{x}^i_{\text{Pa}(v_j, A^*)}\right)}{\text{cnt}\left(\text{Pa}(v_j, A^*) = \mathbf{x}^i_{\text{Pa}(v_j, A^*)}\right)}$$
$$\to \Pr[v_j = \mathbf{x}_j^i \mid \text{Pa}(v_j, A^*) = \mathbf{x}^i_{\text{Pa}(v_j, A^*)}] \quad (\text{as } n \to \infty)$$

Recall that $\text{Pa}(v_j, A^*)$ is defined as the causal parents or direct causes of v_j in the latent CGM (according to A^*). count_X represents the number of observations or rows in the table X that satisfy the specified condition. Thus, we have shown that, our specified model `MLE-INTERP` is able to model the conditional distributions in the latent CGM (as $n \to \infty$). □

A.6 Other relevant metrics

The following tables shows how relevant metrics scale with d for all methods. The results follow the same trends as for SHD in Figure 1. Comparable results within the margin of error of the best performing algorithm are bolded. Note that the Recall values for the PAIR method are particularly high: this is because the pairwise LLM method tends to prioritize false positives over false negatives based on the provided correlation coefficients. This trade-off may be adjusted in future work.

Table 2: Performance comparison of methods on the cancer dataset ($d = 5$)

Method	Precision	Recall	F1-score	Time (s)
SDCD	0.56 ± 0.00	0.50 ± 0.00	0.52 ± 0.04	88.1 ± 11.4
GES	$\mathbf{0.67 \pm 0.29}$	0.33 ± 0.29	0.44 ± 0.38	$\mathbf{0.09 \pm 0.00}$
DAGMA	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$215. \pm 1.82$
PAIR	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	8.52 ± 1.31
BFS	$\mathbf{0.82 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.90 \pm 0.10}$	67.2 ± 8.81
LLM-DCD	0.33 ± 0.14	0.08 ± 0.14	0.13 ± 0.23	2.14 ± 0.19
LLM-DCD-P	$\mathbf{0.67 \pm 0.14}$	0.33 ± 0.14	0.44 ± 0.19	1.38 ± 0.21
LLM-DCD-B	$\mathbf{0.82 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.90 \pm 0.10}$	0.50 ± 0.07

Table 3: Performance comparison of methods on the sachs dataset ($d = 11$)

Method	Precision	Recall	F1-score	Time (s)
SDCD	$\mathbf{0.49 \pm 0.12}$	0.39 ± 0.12	$\mathbf{0.43 \pm 0.13}$	86.5 ± 17.5
GES	0.29 ± 0.09	0.16 ± 0.09	0.20 ± 0.11	$\mathbf{1.49 \pm 0.16}$
DAGMA	0.03 ± 0.03	0.02 ± 0.03	0.02 ± 0.04	$240. \pm 2.64$
PAIR	0.27 ± 0.00	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.43 \pm 0.00}$	54.9 ± 8.01
BFS	$\mathbf{0.53 \pm 0.07}$	0.10 ± 0.07	0.17 ± 0.12	$129. \pm 37.4$
LLM-DCD	0.28 ± 0.16	0.35 ± 0.16	$\mathbf{0.30 \pm 0.15}$	5.26 ± 1.91
LLM-DCD-P	0.31 ± 0.03	0.39 ± 0.03	$\mathbf{0.36 \pm 0.04}$	$\mathbf{4.75 \pm 2.06}$
LLM-DCD-B	$\mathbf{0.49 \pm 0.03}$	0.37 ± 0.03	$\mathbf{0.42 \pm 0.03}$	$\mathbf{4.22 \pm 2.05}$

Table 4: Performance comparison of methods on the child dataset ($d = 20$)

Method	Precision	Recall	F1-score	Time (s)
SDCD	$\mathbf{0.28 \pm 0.12}$	0.49 ± 0.12	$\mathbf{0.36 \pm 0.07}$	61.3 ± 7.00
GES	0.20 ± 0.02	0.21 ± 0.02	0.22 ± 0.03	$\mathbf{28.5 \pm 3.35}$
DAGMA	0.04 ± 0.02	0.01 ± 0.02	0.05 ± 0.08	$277. \pm 5.23$
PAIR	0.13 ± 0.02	$\mathbf{0.99 \pm 0.02}$	0.18 ± 0.10	$222. \pm 19.0$
BFS	0.07 ± 0.06	0.05 ± 0.06	0.10 ± 0.09	$150. \pm 38.0$
LLM-DCD	0.21 ± 0.10	0.41 ± 0.10	0.19 ± 0.10	50.1 ± 0.13
LLM-DCD-P	$\mathbf{0.30 \pm 0.08}$	0.61 ± 0.08	$\mathbf{0.41 \pm 0.04}$	49.5 ± 0.11
LLM-DCD-B	$\mathbf{0.41 \pm 0.00}$	0.32 ± 0.00	$\mathbf{0.39 \pm 0.06}$	48.8 ± 0.12

Table 5: Performance comparison of methods on the alarm dataset ($d = 37$)

Method	Precision	Recall	F1-score	Time (s)
SDCD	0.08 ± 0.05	0.23 ± 0.05	0.12 ± 0.02	$\mathbf{78.9 \pm 4.35}$
GES	0.06 ± 0.02	0.08 ± 0.02	0.07 ± 0.01	$341. \pm 2.10$
DAGMA	0.06 ± 0.02	0.07 ± 0.02	0.06 ± 0.02	$393. \pm 20.7$
PAIR	0.11 ± 0.01	$\mathbf{0.97 \pm 0.01}$	0.20 ± 0.01	$636. \pm 29.9$
BFS	0.26 ± 0.08	0.22 ± 0.08	0.24 ± 0.08	$290. \pm 47.9$
LLM-DCD	0.30 ± 0.02	0.35 ± 0.02	0.32 ± 0.02	$395. \pm 0.25$
LLM-DCD-P	0.33 ± 0.00	0.43 ± 0.00	0.37 ± 0.06	$395. \pm 0.21$
LLM-DCD-B	$\mathbf{0.57 \pm 0.07}$	0.46 ± 0.07	$\mathbf{0.51 \pm 0.07}$	$394. \pm 0.20$

Table 6: Performance comparison of methods on the hepar2 dataset ($d = 70$)

Method	Precision	Recall	F1-score	Time (s)
SDCD	0.12 ± 0.06	$\mathbf{0.18 \pm 0.06}$	$\mathbf{0.14 \pm 0.04}$	$\mathbf{66.5 \pm 26.2}$
GES	–	–	–	–
DAGMA	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	$638. \pm 8.75$
PAIR	–	–	–	–
BFS	0.17 ± 0.02	$\mathbf{0.11 \pm 0.02}$	$\mathbf{0.13 \pm 0.03}$	$544. \pm 50.4$
LLM-DCD	0.03 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	$623. \pm 0.26$
LLM-DCD-P	–	–	–	–
LLM-DCD-B	$\mathbf{0.28 \pm 0.03}$	$\mathbf{0.13 \pm 0.03}$	$\mathbf{0.18 \pm 0.03}$	$623. \pm 0.38$