Evaluating Causal Discovery under Nonlinear Gaussian Process Structural Models

DTU Special Course – Department of Technology, Management and Economics

Filip Arthur Blaafjell
Author

Francisco Camara Pereira
Supervisor

Francisco Madaleno Ferreira Santos
Supervisor

I. Introduction

Causal discovery aims to recover the directed acyclic graph (DAG) that underlies a structural causal model (SCM) [1]. Classical approaches such as PC and GES rely on conditional independence testing or score-based search, and they are known to perform poorly when the data-generating process is non-linear or non-additive [1]. This has pushed a shift towards continuous optimisation methods that model general functional relations while enforcing no cycles. Two widely used examples are NOTEARS-MLP [6], which solves the problem as a smooth optimisation over neural network parameters, and GraN-DAG [5], which trains a neural generator and uses an augmented Lagrangian penalty to remove the cycles.

To evaluate these methods, we use a synthetic setup commonly adopted in the causal discovery literature, based on random sparse DAG generation with known topological order and controlled structural mechanisms [1]. We generate random sparse DAGs with a known topological order and define an SCM on top of them. Each structural equation is a smooth non-linear function drawn from a Gaussian process (GP) prior with an RBF kernel, following the standard construction in Rasmussen and Williams (put bibtex: 2006). This GP-based design has been used in previous causal discovery research because it provides flexible, non-parametric causal mechanisms while keeping the ground truth controlled [5] [6].

The framework samples both Erdős–Rényi (ER) random graphs and scale-free (SF) graphs in order to compare performance across different structures. ER graphs have relatively uniform degree distributions and are therefore considered easier for causal discovery (soruce), whereas SF graphs contain heavy-tailed hub structures that create stronger dependencies and more challenging recovery conditions. By evaluating methods on both, we can see whether gradient-based causal discovery algorithms degrade when confronted with hub-dominated causal structures rather than the more homogeneous patterns of ER graphs.

Across different graph sizes and sample sizes, we generate observational data, estimate the causal graph, and compare the recovered structure to the ground truth using standard metrics: adjacency precision and recall, arrowhead precision and recall, Structural Hamming Distance (SHD), and Structural Intervention Distance (SID).

The aim of this report is to examine: (i) how continuous non-linear causal discovery methods behave under SCMs with GP-based mechanisms, and (ii) how the scale free graphs data affects performance. Although this is a learning-focused project rather than a methodological contribution, it gives practical insight into the reliability and limitations of modern gradient-based causal discovery models in realistic non-linear settings.

II. Background

A. Causal Graphs

A causal graph is a directed acyclic graph where each edge represents a cause–effect relation. Reversing an edge in a causal graph changes the underlying data-generating mechanism, unlike in ordinary directed graphical models where distributions may remain unchanged [1].

Each variable is produced by a structural equation of the form

$$X_j = f_j(\mathrm{Pa}(X_j)),$$

where $\mathrm{Pa}(X_j)$ are the direct causes of X_j. The graph therefore reflects the order in which the system generates its variables.

B. Structural Causal Models

A structural causal model (SCM) extends the graph by specifying functional mechanisms and noise. Each variable satisfies

$$X_j = f_j(\mathrm{Pa}(X_j), U_j),$$

where U_j is an exogenous noise term. The SCM is determined by the graph, the functions f_j, and the noise distribution. Observational data are generated by this system, and causal discovery methods aim to recover the graph from samples of X [1].

C. Evaluation Metrics

To evaluate a causal discovery algorithm, the estimated adjacency matrix is compared to the true one.

Adjacency precision (AP) measures the proportion of predicted edges that are correct:

$$\mathrm{AP} = \frac{TP}{TP + FP}.$$

Adjacency recall (AR) measures the proportion of true edges that are recovered:

$$\mathrm{AR} = \frac{TP}{TP + FN}.$$

Arrowhead precision/recall evaluate correctness of edge directions.

Structural Hamming Distance (SHD) counts the number of edge additions, deletions, or reversals required to match the true graph.

Structural Intervention Distance (SID) measures how many intervention effects the estimated graph gets wrong.

Adjacency metrics assess connections, arrow metrics assess direction, SHD measures structural error, and SID evaluates whether the graph leads to incorrect causal conclusions [1].

III. Data Generation

A. Overview

To evaluate causal discovery methods under controlled nonlinear conditions, I generate data from a structural causal model (SCM) where each causal mechanism is a sample from a Gaussian process (GP). The process consists of four steps:

1) sampling a random DAG,
2) assigning one GP function to every edge,
3) sampling node-specific noise,
4) generating data in topological order.

All data used in the experiments come from this SCM, and only the observational setting is considered.

B. Random DAG Sampling

A random DAG is constructed by first sampling a permutation of the node indices to define a topological ordering. For any pair of nodes (i, j) consistent with the ordering, an edge $i \to j$ is included independently with probability

$$p = \frac{2}{d-1},$$

where d is the number of nodes. This produces sparse DAGs with expected in-degree approximately 2.

The final adjacency matrix is stored as

$$A_{ij} = \begin{cases} 1 & \text{if } i \to j \text{ is present,} \\ 0 & \text{otherwise.} \end{cases}$$

C. Gaussian Process Mechanisms

Each causal mechanism is modelled as a univariate Gaussian process. For every edge (i, j) in the DAG, I sample a GP prior

$$f_{ij} \sim \mathcal{GP}(0, k), \qquad k(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2}\right),$$

using an RBF kernel with $\ell = 1.0$. A dense grid on the interval $[-4, 4]$ is used to draw a prior sample, and a GaussianProcessRegressor is then fitted for interpolation. This gives a smooth nonlinear function f_{ij} that defines the causal effect of X_i on X_j.

D. Noise Terms

Each node receives an independent Gaussian noise term with node-specific scale:

$$\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2), \qquad \sigma_j \sim \mathrm{Uniform}(0.1, 0.5).$$

The noise scales are sampled once per dataset.

E. Sampling the SCM

Given the adjacency matrix, the GP functions, and the noise terms, the data are generated in topological order. For each node X_j with parent set $\mathrm{Pa}(j)$, the structural equation is

$$X_j = \sum_{i \in \mathrm{Pa}(j)} f_{ij}(X_i) \; + \; \varepsilon_j.$$

If a node has no parents, it is generated as pure noise,

$$X_j = \varepsilon_j.$$

Stacking all samples produces the final observational dataset

$$X \in \mathbb{R}^{n \times d},$$

where n is the sample size and d is the number of nodes.

F. Summary

This generator produces smooth nonlinear SCMs with GP-based causal mechanisms, independent noise, and sparse random DAG structure. Because the true graph and functional mechanisms are known, the synthetic data allow controlled benchmarking of causal discovery algorithms.

IV. Assumptions and Limitations

The synthetic data are generated under a specific class of structural causal models. These design choices make the evaluation controlled and reproducible, but they also impose restrictions on what the results represent.

A. Assumptions

- **Acyclicity.** The graph is a DAG and is sampled by imposing a random topological ordering. No feedback loops or cyclic mechanisms are allowed.
- **Additive noise.** Each variable is generated by a sum of GP-based parent functions and an independent Gaussian noise term. This follows the additive noise model but excludes more general structural forms.
- **Independent noise.** Noise terms are independent across nodes. There are no hidden confounders or correlated disturbances.
- **GP mechanisms.** Parent–child relations are smooth functions drawn from an RBF Gaussian process prior. This assumes smooth nonlinear effects and excludes non-smooth or discontinuous mechanisms.
- **Sparse graphs.** The edge probability is fixed at $2/(d-1)$, producing graphs with low expected in-degree.

B. Limitations

- **No latent confounding.** All variables are observed. Real systems often include unobserved factors, which most algorithms struggle with.
- **No interventions.** Only observational data are used. Algorithms that rely on interventional variation are not evaluated in this setting.
- **Limited functional diversity.** Although GP samples cover a wide class of smooth functions, they do not represent all realistic nonlinearities, such as discontinuities, heteroscedastic effects, or strong interactions.
- **Favouring smooth models.** Smooth GP functions align more naturally with kernel-based or nonparametric methods, and may disadvantage methods that assume piecewise-linear or sharply nonlinear relationships.
- **Finite sample behaviour.** Synthetic data do not capture issues like non-stationarity, structural breaks, or heavy tails often present in real-world systems.
- **Low amount of nodes.** In these experiments, the amount of nodes are restricted to a maximum of 15 due to computational constrains. Therefore, results must be interpreted as such. Especially in scale-free graphs, the low amount of nodes will not properly give the emergence of hubs as intended.

These assumptions and limitations should be considered when interpreting algorithmic performance. The goal of this setup is to provide a controlled benchmark, not to replicate all complexities of real data.

V. Algorithms

This study evaluates both classical and neural causal discovery methods. The algorithms are selected to represent fundamentally different approaches to learning directed acyclic graphs (DAGs) from observational data, allowing comparison across modelling assumptions, optimisation strategies, and scalability regimes.

A. Constraint- and score-based methods: PC and GES

a) PC algorithm.: The PC algorithm is a constraint-based causal discovery method introduced by [2]. It recovers the Markov equivalence class of the underlying causal DAG by performing a sequence of conditional independence (CI) tests on the observed variables.

PC proceeds in two main stages. First, it learns an undirected graph (the skeleton) by iteratively removing edges between variables that are found to be conditionally independent given some conditioning set. Second, it orients edges using a set of logical rules derived from causal graph theory, such as the identification of v-structures and propagation rules that preserve acyclicity.

The PC algorithm assumes causal sufficiency (no latent confounders), faithfulness, and that the observed data distribution satisfies the causal Markov condition. In practice, PC relies on statistical CI tests, most commonly partial correlation tests, which implicitly assume linear relationships and Gaussian noise. As a result, PC is sensitive to violations of these assumptions and to finite-sample errors in CI testing.

Despite these limitations, PC is widely used due to its interpretability, theoretical guarantees under ideal conditions, and relatively low computational cost for small to medium-sized graphs.

b) GES algorithm.: Greedy Equivalence Search (GES) is a score-based causal discovery algorithm proposed by [3]. Rather than testing conditional independencies directly, GES searches over the space of Markov equivalence classes of DAGs and selects the graph that optimises a predefined scoring function.

GES operates in two phases. In the forward phase, edges are greedily added to improve the score, while maintaining acyclicity. In the backward phase, unnecessary edges are removed to further refine the graph. The algorithm typically uses decomposable scores such as the Bayesian Information Criterion (BIC), which balance data fit and model complexity.

Like PC, GES assumes causal sufficiency and faithfulness. When used with Gaussian likelihoods and BIC scoring, it further assumes linear structural equations with Gaussian noise. Under these assumptions and in the large-sample limit, GES is consistent and can recover the correct Markov equivalence class.

GES is included as a baseline because it represents a fundamentally different paradigm from constraint-based methods. While PC relies on local independence tests, GES performs a global optimisation over graph structures, often exhibiting different failure modes and robustness characteristics in finite samples.

c) Rationale for inclusion.: PC and GES serve as strong classical baselines for causal discovery. They are well-understood, theoretically grounded, and widely used in the literature [1]. Comparing neural methods against PC and GES allows us to assess whether increased model flexibility and optimisation complexity translate into empirical gains, particularly under nonlinear data-generating processes and limited sample sizes.

B. Continuous optimisation methods: NOTEARS-MLP

a) NOTEARS framework.: NOTEARS is a continuous optimisation approach to causal discovery introduced by Zheng et al. [4]. Unlike constraint- or score-based methods that search over discrete graph structures, NOTEARS formulates DAG learning as a continuous constrained optimisation problem. The key insight is to encode the acyclicity constraint as a smooth, differentiable function of the weighted adjacency matrix, enabling the use of gradient-based optimisation.

Specifically, NOTEARS represents the causal graph by a weighted adjacency matrix and enforces acyclicity through a trace-based constraint derived from matrix exponential properties. This allows the originally combinatorial DAG learning problem to be solved using standard optimisation techniques.

b) Nonlinear NOTEARS (NOTEARS-MLP).: The original NOTEARS formulation assumes linear structural equation models (SEMs). To relax this assumption, Zheng et al. [6]

extend the framework to nonlinear SEMs by modelling each structural equation using a multilayer perceptron (MLP). In this setting, each variable is expressed as a nonlinear function of its parents plus noise, while the acyclicity constraint is imposed on the induced dependency structure of the neural network. The NOTEARS-MLP variant thus combines neural network flexibility with explicit DAG constraints.

c) Rationale for inclusion.: NOTEARS-MLP represents a bridge between classical causal discovery and modern machine learning approaches. It provides a principled baseline for assessing whether continuous optimisation and nonlinear function approximation offer empirical advantages over constraint- and score-based methods, particularly in settings with nonlinear causal mechanisms.

C. Variational neural methods: GraNDAG

a) GraNDAG framework.: GraNDAG (Gradient-based Neural DAG learning) is a neural causal discovery method introduced by Lachapelle et al. [5]. Like NOTEARS, GraNDAG formulates causal structure learning as a continuous optimisation problem, but it differs fundamentally in how the graph structure and functional relationships are parameterised and optimised.

GraNDAG models each structural equation using a neural network while explicitly parameterising the adjacency matrix through learnable gating variables. These gates control whether a directed edge between two variables is active, allowing the model to jointly learn graph structure and nonlinear causal mechanisms. A differentiable acyclicity constraint is imposed on the learned adjacency matrix to ensure that the resulting graph is a DAG.

b) Optimisation and modelling assumptions.: The optimisation objective in GraNDAG combines a data-fitting loss, typically based on a likelihood or squared error, with regularisation terms that promote sparsity and enforce acyclicity. The method uses gradient-based optimisation throughout, enabling end-to-end training of both the graph structure and the neural functional components.

GraNDAG assumes causal sufficiency, independent noise terms, and that the true causal mechanisms can be approximated by the chosen neural network architecture. Unlike constraint-based methods, it does not rely on conditional independence testing, and unlike score-based methods, it does not require explicit enumeration or greedy search over graph structures.

c) Rationale for inclusion.: GraNDAG represents a newer neural approach to causal discovery that combines nonlinear modelling capacity with explicit structural constraints. Including GraNDAG allows us to assess whether highly expressive, optimisation-based neural methods provide empirical advantages over both classical approaches (PC, GES) and simpler continuous optimisation methods such as NOTEARS-MLP, particularly in settings with nonlinear data-generating processes.

VI. Experiments

A. Experimental Setup

Synthetic datasets are generated using the SCM described in the data generation section. Only observational data are used.

I vary two factors:

$$d \in \{5, 10, 15\}, \qquad n \in \{500, 1000, 2000\}.$$

For each pair (d, n), I run 20 independent trials with different random seeds. Each trial produces:

1) one synthetic dataset,
2) one estimated DAG from each algorithm.

All methods take the same data matrix as input and return a directed graph (CPDAG for PC and GES).

B. Hyperparameters

I use standard hyperparmeter configurations:

- **PC:** Gaussian CI test, $\alpha = 0.01$.
- **GES:** BIC score with Gaussian likelihood.
- **NOTEARS-MLP:** two-layer MLP ($[d, 10, 1]$), $\lambda_1 = \lambda_2 = 0.01$, edge threshold 0.3.
- **GraNDAG:** default architecture and parameters from the gcastle implementation.

PC and GES are impelemented using the casual-learn python implementation [9]. Notears-MLP is implemented using its source code from [6], and GraNDAG is implemented using the gcastle python library [7].

C. Evaluation

Each estimated graph is compared to the true DAG using:

- adjacency precision/recall/F1,
- arrowhead precision/recall/F1,
- Structural Hamming Distance (SHD),
- Strutural Intervention Distance (SID),
- runtime (local Intel i7 10th Gen CPU)

Metrics are implemented using the casual-learn [9] and cdt [8] python libraries.

Reported values are the mean and standard deviation over the 20 trials.

VII. Results

A. PC and GES Results (Baseline)

Tables I and II report baseline results for PC and GES on ER and SF graphs generated from nonlinear GP-based SCMs, averaged over 20 Monte Carlo runs. Both algorithms output CPDAGs rather than fully directed DAGs, and all results should be interpreted in this context.

Across all settings, adjacency recovery is substantially stronger than arrow recovery. This is expected, as CPDAGs represent Markov equivalence classes in which many edge directions are not identifiable from observational data alone. Consequently, low arrow precision, recall, and F1 do not necessarily indicate incorrect inference, but rather reflect the presence of unoriented edges in the estimated equivalence class.

On ER graphs, PC and GES achieve similar adjacency performance across node sizes and sample sizes. Differences between the two algorithms are small and not systematic. Arrow metrics are uniformly low for both methods, consistent with the limited number of compelled edge directions in sparse random graphs. SHD and SID increase with the number of nodes, reflecting the larger equivalence classes and increased structural ambiguity in higher dimensions.

On SF graphs, adjacency performance is comparable in magnitude to ER graphs but exhibits higher variability. PC generally recovers more directed edges than GES, leading to higher arrow F1 scores, although absolute values remain low. SHD and SID are consistently higher than for ER graphs at comparable sizes, indicating that recovering the correct equivalence class is more challenging for graphs with hub structures.

Increasing the sample size leads to modest improvements in adjacency recall and F1 in some settings, but has little effect on arrow metrics. This suggests that additional data primarily help stabilise adjacency decisions, while the size of the Markov equivalence class remains largely unchanged.

PC is consistently faster than GES across all configurations, with the runtime gap increasing as the number of nodes grows, particularly for SF graphs.

Overall, these results establish baseline CPDAG recovery performance for PC and GES under nonlinear GP-based data generation, with moderate adjacency accuracy and limited edge orientation due to the fundamental identifiability constraints [1].

B. NOTEARS-MLP

Tables III and IV report the results for NOTEARS-MLP on ER and SF graphs generated from the GP-based SCM. In contrast to PC and GES, NOTEARS directly estimates a fully directed DAG, and the arrow metrics therefore reflect explicit orientation decisions rather than partially directed equivalence classes.

Across both graph types, NOTEARS achieves adjacency recovery that is broadly in line with the baseline methods. Adjacency precision is generally high, while recall is more limited, especially as the number of nodes increases. As a result, adjacency F1 scores remain moderate and do not show a strong or consistent dependence on sample size. SHD increases with graph size, as expected, reflecting the growing difficulty of recovering larger graphs.

A clear difference compared to PC and GES is observed in edge orientation. NOTEARS consistently attains substantially higher arrow precision, recall, and F1 across all settings. This is not unexpected, as the method enforces acyclicity and returns a fully directed graph, rather than leaving edges unoriented when directionality is not identifiable. Arrow performance is relatively stable across sample sizes, suggesting that additional data do not fundamentally change the learned directions under the present nonlinear mechanisms.

Performance is similar between ER and SF graphs in terms of both adjacency and arrow metrics, although SHD and SID are systematically higher for SF graphs. This indicates that, while NOTEARS maintains comparable recovery rates, errors in directed structure are more consequential for power-law graphs.

NOTEARS is substantially more computationally expensive than PC and GES. Runtime increases with both graph size and sample size, with particularly noticeable growth for larger SF graphs. This highlights a trade-off between producing fully directed solutions and computational cost.

Overall, NOTEARS-Nonlinear provides a directed baseline with stable orientation performance and moderate adjacency recovery under nonlinear GP-based data generation, at the expense of significantly higher runtime.

C. GraNDAG Results

Tables V and VI report results for GraNDAG on ER and SF graphs generated from the GP-based SCM. As with NOTEARS, GraNDAG estimates a fully directed DAG, and arrow metrics therefore correspond to explicit orientation decisions rather than equivalence classes.

On ER graphs, GraNDAG achieves the strongest adjacency precision among the tested methods, consistently exceeding 0.9 across most settings. Adjacency recall is lower and decreases as the number of nodes increases, leading to moderate adjacency F1 scores overall. This pattern suggests that GraNDAG tends to recover a relatively sparse directed structure, favouring precision over recall. SHD increases with graph size but remains comparable to NOTEARS, while SID follows a similar trend.

Edge orientation performance is consistently high relative to the other methods. GraNDAG achieves high arrow precision across all configurations and maintains moderate arrow recall, resulting in arrow F1 scores around 0.5 for most ER settings. Unlike NOTEARS, arrow performance does not improve with increasing sample size and, in some cases, slightly degrades for larger graphs, indicating that estimation difficulty grows with dimensionality despite additional data.

On SF graphs, adjacency precision remains high, but adjacency recall drops more sharply as graph size increases. This results in a clearer decline in adjacency F1 for larger SF graphs compared to ER graphs. Arrow metrics follow a similar pattern: while arrow precision remains relatively high, arrow recall decreases substantially for 15-node graphs, leading to lower arrow F1 scores. Correspondingly, SHD and SID are markedly higher on SF graphs, reflecting the increased difficulty of recovering directed structures in the presence of hub nodes.

GraNDAG is substantially more computationally expensive than both PC/GES and NOTEARS. Runtime is high even for small graphs and increases further with graph size, with especially large runtimes observed for SF graphs. Compared to NOTEARS, GraNDAG trades increased computational cost for higher precision in both adjacency and arrow recovery.

Overall, GraNDAG produces highly precise directed graphs under nonlinear GP-based mechanisms, but with limited recall and significant computational overhead. Performance degrades

for larger and more heterogeneous graph structures, particularly in the scale-free setting.

VIII. Discussion

IX. Discussion

This work evaluated constraint-based, score-based, and continuous optimisation approaches to causal discovery under nonlinear Gaussian process (GP) structural causal models. While the experimental setting is controlled and synthetic, it reflects a commonly studied class of nonlinear additive noise models where, under ideal conditions, the full DAG is identifiable from observational data.

a) Constraint-based baselines.: The results for PC and GES are consistent with their theoretical guarantees and practical limitations. Both algorithms recover adjacency structure more reliably than edge orientations, producing CPDAGs rather than fully directed graphs. This behaviour aligns with the causal discovery literature, where constraint-based and score-based greedy methods are known to identify Markov equivalence classes under faithfulness assumptions, rather than unique DAGs, when only observational data are available [1].

In this context, low arrow precision and recall should not be interpreted as failure to orient edges correctly, but rather as a consequence of equivalence class ambiguity. The modest improvements in adjacency recovery with increasing sample size, alongside relatively stable arrow metrics, suggest that additional data primarily reduce conditional independence estimation noise without substantially shrinking the equivalence class. This pattern is consistent with the discussion of identifiability and CPDAG recovery in observational settings [1].

b) NOTEARS-Nonlinear.: NOTEARS-Nonlinear directly estimates a fully directed DAG by solving a continuous constrained optimisation problem [6]. As a result, arrow metrics are inherently higher than for PC and GES, since the method must commit to an orientation for every recovered edge. In the experiments, NOTEARS achieves arrow F1 scores that are substantially higher than the CPDAG-based baselines, while adjacency recovery remains moderate and comparable.

This behaviour is aligned with the design of NOTEARS. The method optimises a global score under an acyclicity constraint, rather than relying on conditional independence testing, and therefore produces a single DAG even in settings where multiple DAGs are observationally equivalent. As noted by Zheng et al., the optimisation problem is non-convex and solutions correspond to stationary points of the empirical objective, rather than guaranteed recovery of the true graph in finite samples [6]. The relatively weak dependence on sample size observed in the results is consistent with this optimisation-driven behaviour.

c) GraNDAG.: GraNDAG extends the continuous optimisation framework of NOTEARS by introducing neural networks to model nonlinear conditional distributions [5]. In the experiments, GraNDAG exhibits high adjacency and arrow precision across both ER and SF graphs, but lower recall, particularly as graph size increases. This indicates a tendency toward sparse, conservative graph estimates.

This precision–recall trade-off is consistent with the strong regularisation induced by the acyclicity constraint and the thresholding procedures described in the GraNDAG framework [5]. While the underlying data-generating process satisfies the assumptions under which nonlinear additive noise models are identifiable, GraNDAG still relies on solving a challenging non-convex optimisation problem with finite data. The observed degradation in recall for larger and scale-free graphs is therefore unsurprising, particularly given the increased difficulty of correctly identifying parent sets for hub nodes.

GraNDAG's higher computational cost relative to NOTEARS and the baseline methods also aligns with the literature, as the method optimises multiple neural networks jointly under an acyclicity constraint [5].

d) Overall takeaways.: Taken together, the results are broadly consistent with existing theory and empirical findings. Constraint-based methods provide reliable adjacency recovery but are fundamentally limited to equivalence classes under observational data. Continuous optimisation methods such as NOTEARS and GraNDAG produce fully directed graphs and achieve higher arrow metrics, but at the cost of increased computational complexity and sensitivity to optimisation and regularisation choices.

Importantly, higher arrow accuracy for DAG-based methods does not necessarily imply better causal identification in finite samples, as orientations may reflect optimisation bias rather than identifiable causal directions. Overall, the experiments highlight the trade-offs between identifiability, computational cost, and structural conservatism that are well documented in the causal discovery literature [1].

References

[1] A. Zanga, E. Ozkirimli, and F. Stella, "A Survey on Causal Discovery: Theory and Practice," arXiv preprint arXiv:2305.10032, 2025.

[2] P. Spirtes, C. Glymour, and R. Scheines, "Causation, Prediction, and Search," MIT Press, Cambridge, MA, 2nd ed., 2000.

[3] D. M. Chickering, "Optimal Structure Identification with Greedy Search," *Journal of Machine Learning Research*, vol. 3, pp. 507–554, 2002.

[4] X. Zheng, B. Aragam, P. Ravikumar, and E. Xing, "DAGs with NO TEARS: Continuous Optimization for Structure Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[5] S. Lachapelle, O. Bengio, and E. Lacoste-Julien, "Gradient-Based Neural DAG Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. P. Xing, "Learning Sparse Nonparametric DAGs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] K. Zhang, S. Zhu, M. Kalander, I. Ng, J. Ye, Z. Chen, and L. Pan, "gCastle: A Python Toolbox for Causal Discovery," *arXiv preprint arXiv:2111.15155*, 2021.

[8] FEN Technologies, "Causal Discovery Toolbox (CDT): Documentation," *Online documentation*, available at https://fentechsolutions.github.io/CausalDiscoveryToolbox/html/index.html, accessed 2025.

[9] CausalLearn developers, "CausalLearn: Causal Learning in Python," *Online documentation*, available at https://causal-learn.readthedocs.io/en/latest/, accessed 2025.

APPENDIX A
PC AND GES RESULTS

TABLE I
PC AND GES RESULTS ON ER GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	PC	0.737±0.384	0.477±0.292	0.555±0.296	0.232±0.329	0.143±0.198	0.171±0.237	4.600±1.655	12.450±5.427	0.013±0.005
5	500	GES	0.642±0.405	0.385±0.259	0.468±0.296	0.113±0.211	0.063±0.128	0.080±0.157	5.000±1.378	13.600±5.034	0.166±0.052
5	1000	PC	0.754±0.389	0.505±0.313	0.582±0.318	0.260±0.359	0.163±0.215	0.188±0.245	4.350±1.352	12.200±5.741	0.013±0.006
5	1000	GES	0.769±0.354	0.482±0.251	0.576±0.270	0.228±0.315	0.144±0.188	0.171±0.226	4.600±1.594	13.200±5.758	0.169±0.055
5	2000	PC	0.670±0.364	0.467±0.295	0.531±0.299	0.188±0.305	0.117±0.174	0.135±0.202	4.900±1.546	13.050±5.025	0.012±0.005
5	2000	GES	0.698±0.386	0.513±0.280	0.577±0.305	0.258±0.333	0.190±0.238	0.216±0.275	4.550±1.910	12.300±6.017	0.111±0.039
10	500	PC	0.880±0.118	0.520±0.127	0.641±0.106	0.135±0.153	0.098±0.126	0.111±0.134	9.350±1.931	40.850±15.932	0.040±0.016
10	500	GES	0.835±0.150	0.570±0.163	0.662±0.137	0.115±0.139	0.084±0.098	0.095±0.111	9.950±2.479	41.950±16.436	1.379±0.538
10	1000	PC	0.823±0.119	0.598±0.145	0.680±0.112	0.182±0.198	0.142±0.167	0.158±0.177	9.350±2.032	40.050±15.838	0.047±0.020
10	1000	GES	0.838±0.149	0.614±0.138	0.696±0.122	0.096±0.126	0.081±0.109	0.086±0.115	10.150±3.038	44.350±15.170	1.680±0.821
10	2000	PC	0.831±0.130	0.645±0.135	0.715±0.114	0.174±0.154	0.138±0.120	0.152±0.133	9.550±2.765	41.350±14.664	0.054±0.026
10	2000	GES	0.798±0.124	0.596±0.136	0.672±0.112	0.068±0.110	0.059±0.103	0.062±0.105	10.450±2.459	46.200±15.835	1.644±0.568
15	500	PC	0.773±0.125	0.522±0.128	0.614±0.114	0.209±0.186	0.149±0.135	0.172±0.155	14.050±3.626	65.950±38.888	0.081±0.029
15	500	GES	0.827±0.110	0.501±0.145	0.611±0.114	0.170±0.195	0.099±0.100	0.122±0.127	14.150±3.838	68.450±36.590	6.698±2.581
15	1000	PC	0.776±0.130	0.563±0.137	0.639±0.113	0.221±0.175	0.174±0.147	0.191±0.153	13.950±3.542	65.150±35.188	0.089±0.043
15	1000	GES	0.803±0.143	0.559±0.168	0.641±0.143	0.131±0.153	0.100±0.105	0.112±0.120	14.700±4.100	71.050±36.320	7.218±2.588
15	2000	PC	0.786±0.121	0.621±0.146	0.678±0.113	0.189±0.161	0.161±0.127	0.172±0.139	14.350±4.041	70.000±37.652	0.109±0.048
15	2000	GES	0.791±0.151	0.581±0.147	0.650±0.110	0.132±0.198	0.085±0.104	0.099±0.128	15.700±5.832	75.150±41.790	7.325±3.226

TABLE II
PC AND GES RESULTS ON SF GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	PC	0.657±0.443	0.525±0.370	0.572±0.386	0.115±0.217	0.100±0.200	0.104±0.200	3.800±0.748	13.450±4.577	0.008±0.002
5	500	GES	0.592±0.460	0.450±0.367	0.504±0.396	0.046±0.152	0.037±0.119	0.041±0.133	4.050±0.740	13.800±3.842	0.078±0.016
5	1000	PC	0.688±0.354	0.562±0.305	0.612±0.319	0.056±0.156	0.050±0.127	0.052±0.139	4.450±0.921	14.800±3.709	0.010±0.002
5	1000	GES	0.662±0.424	0.487±0.340	0.554±0.366	0.029±0.088	0.025±0.075	0.027±0.081	4.150±0.654	14.400±3.137	0.091±0.025
5	2000	PC	0.713±0.342	0.550±0.292	0.611±0.300	0.067±0.226	0.050±0.170	0.057±0.194	4.400±1.158	14.400±4.247	0.011±0.003
5	2000	GES	0.613±0.432	0.487±0.358	0.531±0.373	0.025±0.079	0.025±0.075	0.024±0.074	4.250±0.622	14.250±3.192	0.093±0.024
10	500	PC	0.759±0.249	0.544±0.225	0.602±0.206	0.184±0.200	0.150±0.162	0.157±0.166	9.350±1.459	50.800±14.586	0.050±0.026
10	500	GES	0.867±0.193	0.506±0.184	0.603±0.160	0.066±0.131	0.039±0.073	0.046±0.083	9.850±1.931	58.400±10.656	1.139±0.293
10	1000	PC	0.691±0.238	0.611±0.242	0.633±0.229	0.172±0.197	0.172±0.197	0.171±0.193	9.700±2.431	52.850±15.701	0.080±0.067
10	1000	GES	0.801±0.171	0.622±0.215	0.662±0.140	0.076±0.111	0.100±0.157	0.085±0.127	10.100±1.729	60.200±10.524	1.519±0.520
10	2000	PC	0.722±0.182	0.606±0.151	0.633±0.117	0.158±0.167	0.172±0.187	0.161±0.170	10.200±2.135	51.950±13.017	0.078±0.043
10	2000	GES	0.766±0.208	0.594±0.196	0.629±0.146	0.079±0.144	0.083±0.149	0.078±0.140	10.850±3.410	60.050±14.565	1.615±0.580
15	500	PC	0.721±0.192	0.589±0.156	0.639±0.149	0.206±0.097	0.175±0.092	0.187±0.089	15.150±2.780	99.800±23.634	0.366±0.233
15	500	GES	0.770±0.175	0.582±0.171	0.632±0.134	0.081±0.107	0.082±0.109	0.081±0.106	16.100±2.278	119.100±30.191	17.899±3.722
15	1000	PC	0.696±0.193	0.589±0.146	0.620±0.142	0.122±0.146	0.125±0.163	0.122±0.153	16.600±3.007	112.250±31.218	0.441±0.192
15	1000	GES	0.729±0.200	0.629±0.186	0.647±0.154	0.094±0.132	0.104±0.145	0.097±0.135	16.750±3.727	119.350±26.411	20.143±5.642
15	2000	PC	0.698±0.182	0.700±0.154	0.681±0.128	0.184±0.124	0.211±0.154	0.193±0.136	16.200±3.385	105.550±26.297	0.625±0.367
15	2000	GES	0.677±0.209	0.661±0.175	0.640±0.148	0.056±0.077	0.071±0.104	0.062±0.087	18.600±3.583	128.450±29.292	21.335±8.638

APPENDIX B
NOTEARS-MLP AND GRANDAG RESULTS

TABLE III
NOTEARS-NONLINEAR RESULTS ON ER GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	NL	0.750±0.356	0.447±0.301	0.529±0.300	0.642±0.393	0.366±0.298	0.437±0.303	3.500±1.746	8.050±5.152	1.287±1.042
5	1000	NL	0.778±0.359	0.449±0.322	0.527±0.311	0.558±0.427	0.338±0.321	0.391±0.324	3.550±1.830	8.200±5.144	1.343±1.199
5	2000	NL	0.777±0.351	0.463±0.309	0.543±0.301	0.595±0.422	0.356±0.307	0.416±0.317	3.500±1.746	8.350±5.003	1.630±1.447
10	500	NL	0.867±0.217	0.458±0.235	0.567±0.229	0.824±0.244	0.427±0.222	0.532±0.220	5.900±1.972	16.100±7.726	3.649±2.359
10	1000	NL	0.872±0.219	0.481±0.244	0.587±0.234	0.830±0.242	0.453±0.240	0.555±0.234	5.650±2.128	16.100±8.105	4.225±2.617
10	2000	NL	0.852±0.215	0.459±0.235	0.564±0.222	0.809±0.245	0.431±0.235	0.530±0.224	6.050±2.179	16.900±11.077	5.955±3.843
15	500	NL	0.826±0.241	0.392±0.170	0.513±0.190	0.768±0.241	0.363±0.162	0.475±0.182	9.600±2.396	31.300±14.741	5.353±3.018
15	1000	NL	0.854±0.148	0.422±0.166	0.541±0.168	0.797±0.170	0.390±0.154	0.500±0.158	9.550±2.747	30.900±15.843	6.631±3.365
15	2000	NL	0.835±0.138	0.436±0.167	0.547±0.165	0.714±0.219	0.389±0.163	0.487±0.172	9.750±2.662	31.700±16.586	10.972±5.181

TABLE IV
NOTEARS-NONLINEAR RESULTS ON SF GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	NL	0.680±0.418	0.412±0.309	0.482±0.320	0.601±0.412	0.350±0.278	0.414±0.294	2.900±1.136	6.250±3.345	1.225±1.134
5	1000	NL	0.818±0.358	0.450±0.269	0.553±0.278	0.638±0.413	0.362±0.279	0.440±0.301	2.700±1.054	6.500±4.129	1.435±1.336
5	2000	NL	0.746±0.396	0.463±0.309	0.541±0.312	0.529±0.439	0.338±0.319	0.389±0.331	2.900±1.221	6.900±4.194	1.702±1.485
10	500	NL	0.780±0.245	0.483±0.190	0.575±0.196	0.645±0.277	0.400±0.200	0.474±0.213	6.650±2.104	26.150±14.044	4.325±2.613
10	1000	NL	0.772±0.246	0.478±0.190	0.565±0.191	0.643±0.278	0.406±0.215	0.474±0.219	6.700±1.926	25.300±13.576	5.025±3.408
10	2000	NL	0.778±0.242	0.478±0.158	0.577±0.167	0.652±0.270	0.406±0.184	0.486±0.198	6.600±1.908	25.550±13.589	7.527±4.266
15	500	NL	0.746±0.161	0.454±0.169	0.542±0.143	0.667±0.227	0.407±0.185	0.487±0.180	10.750±2.844	45.200±22.704	7.544±3.001
15	1000	NL	0.764±0.176	0.439±0.160	0.532±0.134	0.650±0.271	0.379±0.190	0.457±0.197	11.050±2.889	47.700±24.270	9.202±4.344
15	2000	NL	0.758±0.156	0.457±0.167	0.550±0.137	0.687±0.225	0.411±0.179	0.495±0.170	10.550±2.765	44.250±22.228	14.953±6.669

TABLE V
GRANDAG RESULTS ON ER GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	GraNDAG	0.892±0.197	0.593±0.276	0.644±0.213	0.794±0.285	0.540±0.281	0.582±0.244	3.200±1.661	6.050±4.105	25.993±1.883
5	1000	GraNDAG	0.929±0.129	0.554±0.280	0.637±0.221	0.873±0.189	0.503±0.253	0.586±0.210	3.100±1.546	6.450±4.201	27.266±2.223
5	2000	GraNDAG	0.916±0.187	0.578±0.289	0.634±0.219	0.867±0.218	0.540±0.282	0.592±0.210	3.100±1.513	6.200±4.366	44.338±13.841
10	500	GraNDAG	0.876±0.121	0.419±0.188	0.539±0.169	0.801±0.242	0.382±0.194	0.492±0.194	6.550±2.202	17.750±11.013	55.227±9.109
10	1000	GraNDAG	0.937±0.103	0.415±0.159	0.556±0.163	0.891±0.194	0.392±0.164	0.526±0.179	6.100±2.142	17.500±10.916	36.195±11.388
10	2000	GraNDAG	0.915±0.123	0.363±0.137	0.502±0.147	0.849±0.256	0.335±0.149	0.464±0.178	6.700±1.952	19.450±11.737	33.362±4.694
15	500	GraNDAG	0.912±0.102	0.349±0.141	0.484±0.136	0.825±0.218	0.335±0.157	0.461±0.170	9.900±3.048	33.850±20.031	39.209±2.353
15	1000	GraNDAG	0.913±0.091	0.378±0.146	0.514±0.148	0.823±0.225	0.364±0.163	0.491±0.182	9.500±2.975	31.550±15.929	39.502±1.872
15	2000	GraNDAG	0.931±0.090	0.379±0.159	0.516±0.164	0.850±0.223	0.365±0.173	0.494±0.192	9.250±2.791	31.300±17.321	39.824±0.348

TABLE VI
GRANDAG RESULTS ON SF GRAPHS.

Nodes	Samples	Algo	Adjacency			Arrow			SHD	SID	Time
			AP	AR	F1	HP	HR	F1			
5	500	GraNDAG	0.866±0.215	0.512±0.201	0.601±0.150	0.785±0.295	0.475±0.208	0.556±0.200	2.750±1.135	4.850±3.229	22.802±0.200
5	1000	GraNDAG	0.912±0.184	0.537±0.198	0.639±0.157	0.820±0.278	0.500±0.224	0.588±0.205	2.450±1.023	4.850±3.321	24.374±1.167
5	2000	GraNDAG	0.904±0.202	0.512±0.167	0.616±0.138	0.829±0.288	0.487±0.201	0.579±0.196	2.600±1.114	4.650±3.087	25.295±0.856
10	500	GraNDAG	0.888±0.171	0.467±0.207	0.573±0.182	0.805±0.193	0.428±0.203	0.523±0.183	6.050±1.857	23.150±11.913	33.290±2.797
10	1000	GraNDAG	0.865±0.185	0.444±0.222	0.542±0.204	0.770±0.201	0.411±0.233	0.495±0.221	6.350±2.264	23.700±11.091	32.601±2.207
10	2000	GraNDAG	0.858±0.182	0.439±0.206	0.541±0.178	0.786±0.197	0.411±0.217	0.502±0.192	6.350±1.956	24.000±11.777	37.381±11.908
15	500	GraNDAG	0.871±0.175	0.286±0.168	0.398±0.177	0.696±0.297	0.250±0.170	0.344±0.195	11.400±2.223	55.750±25.901	57.303±17.529
15	1000	GraNDAG	0.902±0.124	0.325±0.165	0.447±0.174	0.706±0.286	0.286±0.169	0.389±0.198	10.750±2.095	55.050±28.628	45.354±1.831
15	2000	GraNDAG	0.898±0.152	0.279±0.139	0.407±0.159	0.746±0.292	0.250±0.146	0.362±0.181	11.000±2.025	57.800±26.149	44.369±0.427

APPENDIX C
GITHUB

The code implementation can be found in the following GitHub repository: GitHub repository