Causal Discovery under Unmeasured Confounding

Danish Shakeel Mohammad
Student Num. *240830263*
Supervisor *Dr. Anthony Constantinou*
Queen Mary University of London
MSc in Artificial Intelligence

***Abstract*—The process of discovering causal structures from observational data is complicated by the presence of latent confouders. The common graphical structures of DAGs are no longer representative of such data and we must instead use Acyclic Directed Mixed Graphs (ADMGs), which use bidirected edges to represent confounded variables. This study investigates the structure learning problem within the domain of Linear Gaussian Structural Causal Models (SCMs), targeting ancestral and bow-free ADMGs, which exhibit global and almost-everywhere identifiability, respectively, in the limit of infinite samples. We introduce a novel, modular framework built on top of two surjective mappings which project a continuous vector into the combinatorial spaces of bow-free and ancestral ADMGs. ADMG search then becomes a continuous optimization task with BIC minimization objective. We systematically evaluate the efficacy of searching this high-dimensional Euclidean space using two distinct stochastic optimization methodologies: Relcadilac, which employs Proximal Policy Optimization (PPO), and a derivative-free approach utilizing the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), on synthetic datasets and real-world data.**

I. Introduction

Causal discovery, which involves discovering cause-effect relationships [1], plays a significant role in scientific enquiry in the domains of economics [2], medicine [3], biology [4], and climate science [5], among others. Randomised Control Trials (RCT) are considered to be the standard technique for determining causal relationships [6] [7]. However, they might turn out to be prohibitively expensive, impractical or immoral to carry out in the real world. This is where algorithms for causal discovery, solely from observational data are highly useful.

A number of causal discovery methods assume *causal sufficiency*, where no common cause of the set of variables of interest is extraneous to the set [8]. This assumption is often violated in practice [9] and algorithms relying on this assumption might predict spurious relationships between variables. Acyclic Directed Mixed Graphs (ADMGs) [cite paper that defines ADMGs], which utilize directed edges to depict cause-effect relationships between variables and bidirected edges to indicate the presence of confounders, are often used to model data containing unobserved confounders [cite some papers that use ADMGs].

Score-based methods: Score-based causal discovery methods work by assigning a score $S(G)$ to each candidate graph G and find the best graph G^* in some class of graphs by minimising the score while searching over the space of all possible graphs in that class. Common scores used for this purpose include the Akaike Information Criterion (AIC) [10], the Bayesian Information Criterion (BIC) [11] (which we use), and the Bayesian Dirichlet equivalent uniform (BDeu) [12].

$$G^* = \underset{G \in \text{ADMG}}{\operatorname{argmin}} \; S(G) \tag{1}$$

In general, the problem in 1 is NP-hard add citation. We can restrict the search space to ancestral or bow-free ADMGs to make the problem more tractable. Ancestral ADMGs can capture all regular conditional independencies, but not more general non-parametric equality constraints called Verma Constraints; while bow-free ADMGs can capture both [cite]. By further restricting the data generation model to the linear gaussian setting, ancestral ADMGs become globally identifiable while bow-free ADMGs become almost-everywhere identifiable, both parametrically and structurally, in the limit of infinite samples. This allows us to use the BIC criterion (imperfectly for bow-free ADMGs) to search through the space of ancestral and bow-free ADMGs to find the score-minimizing graph structure.

Deep Learning (DL) based approaches for Causal Discovery gained traction with the introduction of the smooth acyclicity constraint in [13] which allowed 1 to be modelled as a continuous optimization problem, enabling the use of neural networks for causal discovery like in [14] and [15]. DL, however, requires differentiable loss functions, making it difficult to directly use some graph scoring metrics. Phrase it differently since RL is a part of DL and it also requires differentiable loss functions. Since the reward which is being maximised in Reinforcement Learning (RL) does not have to be differentiable, and it has a built-in exploration-exploitation mechanic, it is often used as a search tool. They handle acyclicity through incorporating the regularization term into the reward [cite] or through flow-based auto-regressive methods that sequentially add edges, checking for acyclicity at every step. The former fails to entirely prevent cycles while the latter is slow since it generates one graph at a time.

Based on [16], who develop a one-step graph generation function which maps a real vector of appropriate size into the space of all DAGs, we present get a better word than present two VEC2ADMG functions which map to all bow-free and ancestral ADMGs, respectively. This allows us to search through continuous euclidean space to find the optimal graph in discrete ADMG space. The negative of the BIC metric, computed using the RICF algorithm for bow-free ADMGs, acts as the reward function, while we use the Proximal Policy Optimization (PPO) [cite] algorithm to search through a high-dimensional vector space. The instability and sample inefficiency of the RL algorithm causes it to struggle in the high dimensional piece-wise constant search space, thus the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) algorithm was also tried, in that, being a genetic algorithm, it might be more sample efficient, which does turn out to be the case.

II. Related Work

The main constraint based algorithm for causal discovery under latent confounding is the FCI [17] algorithm, which uses conditional independence (CI) tests to determine edge existence. Although it is asymptotically consistent, it is super-exponential in the worst case since it needs to condition on all possible separating sets to determine $d-$separation. RFCI [18] trades of effectiveness for speed by performing a restricted set of CI tests compared to FCI. GFCI [19] starts by executing the FGES [20] score based algorithm for determining edge adjacencies under the assumption of causal sufficiency. Then it discards the causal sufficiency assumption and uses the logic from FCI to prune and orient edges. These algorithms, however, rely on the quality of the CI tests and become unsuitable for large or dense graphs. Additionally, they cannot act on the more general class of bow-free ADMG graphs.

Score based algorithms for causal discovery with latent confounders often rely on the RICF [21] algorithm, which allows computing the Maximum Likelihood Estimate for bow-free ADMGs. The greedyBAP algorithm [22] leverages RICF along with greedy hill climbing to maximise a penalised likelihood score. Their method, however, gets stuck in local optima, which they try to mitigate with repeated random restarts. DCD [23] is another algorithm that uses a modification of RICF along with algebraic constraints which restrict the search to bow-free, arid, and ancestral ADMGs (separate constraints for each graph class) and a differentiable version of the BIC score to finally execute a dual descent algorithm to find an optimal ADMG graph. The SPOT algorithm [24] follows a two-phase approach. In the first phase they estimate a posterior distribution over the graph skeleton parameterised by a neural network trained through supervised learning on simulated data followed by domain adaptation. In the second phase, they incorporate the skeleton posterior into the differentiable optimization approach in the DCD algorithm. They demonstrate that they are able to scale the approach to 100 node graphs as well.

Building on top of the work of [25], who introduce a node potential vector p to characterise an implicit causal ordering, [16] propose the Vec2DAG operator which is an unconstrained parameterization for DAGs and allows one-step DAG generation from a high dimensional Euclidean space. They use the PPO algorithm to search through this high dimensional space, along with a Gaussian Process based method to compute the BIC score for DAGs. They demonstrate an SHD smaller than 3 for DAGs with upto 200 nodes. In [26], the authors establish that under conditions of non-linear additive noise SCMs, bow-free ADMGs, and observable and unobservable variable non-modulation, the ADMG is identifiable. They transform the ADMG into a DAG through ”magnification” where they introduce a latent variable for each confounded edge, parameterize the non-linear functions using neural networks, use variational inference to approximate the posterior over the graphs and the latent variables, and use a differentiable constraint to enforce acyclicity, similar to the constraint used in DCD and SPOT.

In our proposed algorithms, we use the RICF to compute the BIC score and bypass the requirement for a differentiable acyclicity constraint or a differentiable score by using a vector-to-admg formulation inspired by Vec2DAG for both bow-free and ancestral ADMGs.

III. Background

A. Linear Gaussian Structural Causal Models

A Structural Causal Model (SCM) is defined by the 4-tuple $\mathbf{M} = (\mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{U}))$ where $\mathbf{V} = \{V_1, \dots, V_d\}$ are the observable random variables and $\mathbf{U} = \{U_1, \dots, U_d\}$ are the set of unobservable random variables such that $\mathbf{V} \cap \mathbf{U} = \varnothing$. Each function, $f_i : (\mathbf{V} \cup \mathbf{U})^p \rightarrow \mathbf{V}$ in the set of functions $\mathcal{F}$, specifies a structural equation between the observed and unobserved causes (parents) of $V_i \in \mathbf{V}$. $P(\mathbf{U})$ is a joint probability distribution over $\mathbf{U}$. Under the assumption of causal sufficiency, the distribution $P_V(\mathbf{V})$ induces a DAG G over $\mathbf{V}$ that is Markov wrt P_V in that the conditional independences in P_V are implied in G through d-separation. In the linear Gaussian confounded variables setting, each f_i is a linear transformation of its inputs and $P(\mathbf{U})$ is a multivariate zero-mean Gaussian with symmetric positive definite covariance matrix $\mathbf{\Omega}$, meaning the error terms U_i are not necessarily mutually independent. Thus, the structural equations can be written as:

$$V_i = \sum_{V_j \in \text{Pa}(V_i)} \theta_{ij} V_j + U_i \Rightarrow \mathbf{V} = \mathbf{\Theta V} + \mathbf{U}; \mathbf{U} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega}) \tag{2}$$

B. Acyclic Directed Mixed Graphs

Acyclic Directed Mixed Graphs (ADMGs), which use directed edges ($\rightarrow$) to capture direct causal relationships and bidirected edges ($\leftrightarrow$) to capture the presence of confounded variables, are a superset of DAGs, and are used to represent potentially causally insufficient SCMs. Each ADMG can be represented by two binary adjacency matrices, D and B where $D_{ij} = 1 \Leftrightarrow \mathbf{\Theta}_{ij} \neq 0$ (and so, $V_j \in \text{Pa}(V_i)$) and $B_{ij} = B_{ji} = 1 \Leftrightarrow \mathbf{\Omega}_{ij} \neq 0$.

An ADMG $\mathcal{G}(V, E)$ is said to be *ancestral* if no strictly directed path $V_i \rightarrow \dots \rightarrow V_j$ is also joined by a bidirected edge $V_i \leftrightarrow V_j$ for any pair $V_i, V_j \in V$. An ADMG $\mathcal{G}(V, E)$ is said to be *bow-free* if no pair $V_i, V_j \in V$ is simultaneously connected by both a directed edge $V_i \rightarrow V_j$ and a bidirected edge $V_i \leftrightarrow V_j$. All ancestral graphs are bow-free but not all bow-free graphs are ancestral.

We also assume causal faithfulness where, if in the SCM for the probability distribution over $\mathbf{V}$, $\mathbf{\Theta}_{ij} = \mathbf{\Theta}_{ji} = \mathbf{\Omega}_{ij} = \mathbf{\Omega}_{ji} = 0$ then V_i and V_j are not connected by an edge in $\mathcal{G}$. Thus, "cancellation effects" in the SCM do not spuriously alter the edges in $\mathcal{G}$. This statement might not be rigorous and the causation might go the other way.

C. Bayesian Information Criterion

Given some data $X \in \mathbb{R}^{n \times d}$ and a candidate ADMG $\mathcal{G}$, the Bayesian Information Criterion (BIC), is a parametric score over a model family, and is defined as:

$$\text{BIC}(X, \mathcal{G}) = -2 \ln p(X|\hat{\theta}, \mathcal{G}) + k \ln(n) \tag{3}$$

where p is the likelihood function and $\hat{\theta}$ are the model parameters that maximise that likelihood, and k is the number of model parameters and for the graph $\mathcal{G}(V, E)$, $k = |V| + |E|$, the sum of the number of directed edges, the number of bidirected edges, and the number of vertices. If the candidate models form a smooth curved exponential family, then in the limit of infinite data ($n \to \infty$), the BIC score is consistent in that it assigns the minimum value to the ground truth model. Linear Gaussian ancestral ADMGs are globally identifiable and form smooth curved exponential families. Linear Gaussian bow-free ADMGs are only almost-everywhere identifiable and the BIC score is not consistent for them. However, as we show empirically, the BIC metric functions well enough in practice, even for bow-free ADMGs.

D. Residual Iterative Conditional Fitting

The Residual Iterative Conditional Fitting (RICF) algorithm [21] is designed to compute the Maximum Likelihood Estimate (MLE) for bow-free acyclic linear structural equation graphs. This is the same graph class as bow-free ADMG models, only with a different name. Unlike other optimization methods like Newton-Raphson, it ensures that the positive definiteness of the covariance matrix is maintained. By exploiting the conditional independence structure of bow-free ADMGs, it is able to compute the MLE only using least square regressions, without requiring costly Hessian computations. In the code, we modify and use an implementation of the RICF algorithm from the `ananke-causal` python library [27]. The modifications leverage the `numba`[1] python library along with specific adaptations like pre-computing the covariance matrix before invoking the algorithm to achieve a $\sim 50\%$ speedup, which is significant over the tens of thousands of invocations per run for the proposed algorithms.

E. Proximal Policy Optimization

Proximal Policy Optimization (PPO) [28] is a stochastic gradient descent optimizer functioning in the framework of Markov Decision Processes (MDPs), which are given by the tuple (S, A, P, R, γ) where S is the set of states an agent can operate in, A is the set of actions the agent can take, P is the distribution over state transitions, R is the reward distribution and γ is the discount factor. The goal of the agent is to find a policy π_θ, parameterised by the policy neural network, that maximises the expected discounted reward, $J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_t]$. The algorithm is designed to function well in high-dimensional continuous control problems, and does this by penalising large changes to the policy. It is generally robust the hyperparameter configurations and avoids performance collapse in non-convex optimization problems, which was the main motivating factor in its choice as optimizer for our algorithm. One of its drawbacks is that it is an on-policy algorithm, meaning that it discards trajectory samples (action sequences obtained by following the policy) after every policy update. This generally means that it is less sample efficient than some off-policy algorithms.

F. Covariance Matrix Adaptation Evolutionary Strategy

The Covariance Matrix Adaptation - Evolutionary Strategy (CMA-ES) [29] is a continuous genetic algorithm used for black-box optimization in non-convex and ill-behaved optimization landscapes. It functions in an iterative manner, sampling a population of candidate solutions from a multivariate normal gaussian $\mathcal{N}(\mu, \sigma^2\mathbf{C}$, ranking the fitness of the population individuals through an objective function, and updating the mean, μ, the step-size σ and the covariance matrix $\mathbf{C}$. By approximating the inverse Hessian of the objective function, ensuring that the population does not collapse into smaller subspaces, and preventing premature convergence through step size control, the CMA-ES algorithm reduced the number of required function evaluations and functions well even in search spaces with discontinuities, sharp bends, and local optima [30]. For these reasons, it was chosen as one of the black-box optimizers in the proposed algorithm.

IV. Methodology

A. Overview

The proposed algorithm, given an input observational dataset, works by repeatedly executing the below three steps for a specified number of iterations:

1) Use a continuous optimization algorithm (PPO and CMA-ES in our case) to search through a high dimensional Euclidean space.
2) Use the surjective vector-to-admg mappings to convert the sampled points into bow-free or ancestral ADMG graphs.
3) Use the RICF algorithm to compute the BIC score of the graph on the provided dataset. This is the score that the optimization algorithm attempts to minimize.

The modular nature of the algorithm allows one to swap-in different components, like optimization algorithms or goodness-of-fit metrics, allowing the algorithm to be easily adapted and extended to different use-cases as long as the conditions for identifiability and consistency are satisfied, and a suitably fast mapping to the intended class of graphs can be found.

[1] https://numba.pydata.org

B. Vector to ADMG Mappings

In order to search the space of ADMGs effectively, we propose two $\mathcal{O}(d^2)$ mappings, Φ^{BF} and Φ^{AN}. These are surjective mappings from $\mathbb{R}^{d^2}$ to the space of bow-free ADMGs, $\mathbb{S}_{BF}$ and ancestral ADMGs, $\mathbb{S}_{AN}$, respectively. The mappings follow a similar pattern to the VEC2DAG mapping found in [16], extended to the more complicated space of ADMGs. Each of the mappings return a pair of binary adjacency matrices, D and B, for directed and bidirected edges, respectively.

Given any real vector, $z \in \mathbb{R}^{d^2}$, where d is the number of observed variables, we can split it into three components: the node potential vector, p, made of the first d elements of z, and which determines the causal order of the nodes; the strictly lower triangular directed edge potential matrix, $E_{\rightarrow} \in \mathbb{R}^{d\times d}$, made from the subsequent $\frac{d(d-1)}{2}$ elements of z, and which determines the existence of directed edges while respecting the causal order; and the strictly lower triangular bidirected edge potential matrix, $E_{\leftrightarrow} \in \mathbb{R}^{d\times d}$, derived from the final $\frac{d(d-1)}{2}$ elements of z, and which determines the existence of bidirected edges while ensuring that the derived graph remains bow-free or ancestral, as required.

Definition 1. For all $d \in \mathbb{N}^+$ and $z \in \mathbb{R}^{d^2}$

$$\Phi_d^{BF}(z)[D] := H\left(E_{\rightarrow} + E_{\rightarrow}^{\top}\right) \odot H\left(\mathrm{grad}(p)\right)$$
$$\Phi_d^{BF}(z)[B] := H\left(E_{\leftrightarrow} + E_{\leftrightarrow}^{\top}\right) \odot (I - D) \odot \left(I - D^{\top}\right)$$
$$\Phi_d^{AN}(z)[D] = \Phi_d^{BF}(z)[D]$$
$$\Phi_d^{AN}(z)[B] := H\left(E_{\leftrightarrow} + E_{\leftrightarrow}^{\top}\right) \odot \left(I - D^{+}\right) \odot \left(I - (D^{+})^{\top}\right)$$

where $H(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$ *is the Heaviside step function,* $\odot$ *is the element-wise or Hadamard product* ($(A \odot B)_{ij} = A_{ij}B_{ij}$), $\mathrm{grad}(u)_{ij} := u_j - u_i$ *is the gradient flow operator,* $I \in \mathbb{R}^{d\times d}$ *is the identity matrix, and* A^+ *is the transitive closure of a directed graph:* $A^+_{ij} = 1$ *iff there is a non-empty directed path from node* i *to* j *in the graph whose adjacency matrix is given by* A.

The $H(E+E^{\top})$ terms in definition 1 control edge existence. E is used to refer to both the edge potential matrices since they behave similarly. $E_{ij} = E_{ji} > 0 \Rightarrow H(E+E^{\top}) = 1$ means there *could* be an edge between nodes i and j, determined by the remaining terms in the equation. $E_{ij} = E_{ji} \leq 0 \Rightarrow H(E+E^{\top}) = 0$ means no edge can exist between nodes i and j. The Heaviside step function applied on top of the gradient operator means that a directed edge $i \rightarrow j$ is allowed to exist only if $p_i < p_j$: $H(\mathrm{grad}(p))_{ij} = 1 \Leftrightarrow p_i < p_j$. The causal order of the nodes is the order of the indices of the elements of p when they are considered in ascending order. The first term in the derivation of the directed edge adjacency matrices, D controls whether there is an edge between two nodes, while the second term filters out some of the edges to enfore a causal ordering. D, thus represents the DAG component of the ADMGs.

To enforce the bow-free nature of the graphs, $(I - D)$ and $(I - D^{\top})$ act as filters, removing bidirected edges suggested through the $H(E_{\leftrightarrow} + E_{\leftrightarrow}^{\top})$ term, between nodes which already have directed edges between them. $(I - D)$ prevents the $i \leftrightarrow j$ edge when the edge $i \rightarrow j$ exists while $(I - D^{\top})$ prevents the same edge when $j \rightarrow i$ exists.

The transitive closure of a DAG, $\mathcal{G}$ is just another DAG, $\mathcal{G}^+$ with an edge $i \rightarrow j$ whenever there is a directed path from i to j in D. To enforce ancestrality in an ADMG we must make sure that a bidirected edge does not connect two ends of a non-empty directed path. Based on the definition of the transitive closure, this can be ensured by preventing a bidirected edge from existing between two nodes in $\mathcal{G}$ when they are joined by a single directed edge in $\mathcal{G}^+$. This is exactly what is done while deriving the bidirected edge adjacency matrix for the VEC2AN mapping, using the same logic as in the bow-free derivation, only substituting the directed edge adjacency matrix, D with its transitive closure, D^+.

Theorem 1 (Surjectivity)**.** *For all* $d \in \mathbb{N}^+$*, let* $\Phi_d : \mathbb{R}^{d^2} \rightarrow \{0,1\}^{d\times d} \times \{0,1\}^{d\times d}$ *represent both the mappings defined in definition 1, and let* $\mathbb{S}_d$ *denote the set of all* d *node ADMGs belonging to the corresponding target class, then* $\forall\, \mathcal{G} \in \mathbb{S}_d \;\exists\; z \in \mathbb{R}^{d^2}$ *such that* $\Phi_d(z) = \mathcal{G}$.

Theorem 1 (proved in the appendix), based on an analogous theorem from [16] for DAGs, establishes the surjective nature of the ADMG mappings, allowing the use of any continuous optimization algorithm to search the combinatorial space of all bow-free or ancestral ADMGs, without the need for acyclicity constraints.

Lemma 1 (Scale Invariance)**.** *For all* $d \in \mathbb{N}^+$*, let* Φ_d*,* $\mathbb{S}_d$ *be defined as in theorem 1, then* $\forall\, z \in \mathbb{R}^{d^2}, \lambda \in \mathbb{R}, \lambda > 0, \Phi_d(\lambda z) = \Phi_d(z)$.

Lemma 2. *For all* $d \in \mathbb{N}^+$*, let* Φ_d*,* $\mathbb{S}_d$ *be defined as in theorem 1, then if* $\mathcal{U} \in \mathbb{R}^{d^2}$ *is an open set containing the origin, we have that* $\Phi_d(\mathcal{U}) = \mathbb{S}_d$.

C. Optimization with PPO

The policy for the PPO algorithm is parameterised by a multivariate normal distribution with a diagonal covariance matrix: $\pi_\theta(z) = \mathcal{N}(z; \mu_\theta, \mathrm{diag}(\sigma_\theta^2))$. The covariance matrix is restricted to be diagonal due to the high dimensionality, d^2 and the axis-aligned edge transitions (edges are added or removed when crossing $z_i = 0, d < i < d^2$ hyperplanes) in the search space, although the causal ordering of the nodes changes when crossing $z_i = z_j, i \neq j, \{i,j\} \leq d$ hyperplanes.

The reward for each action, $z \sim \pi_\theta$ is a scaled negative BIC score: $R(z) = -\frac{1}{n}\mathrm{BIC}(X, \mathrm{VEC2ADMG}(z))$. RL algorithms are designed to maximise the reward and we wish to minimise the BIC score, hence the reward is the negative of the BIC score. The BIC score is scaled to ensure that it does not suffer from large variance in magnitude, helping with the stability of the algorithm. Based on lemma 2, we restrict the search region to be a axis-aligned hypercube of side length 2, centered at the origin.

The environment for the PPO algorithm is a one-step environment since every sampling from the search space immediately provides a valid ADMG graph through the use of the VEC2ADMG mappings. In this aspect, the environment is like a continuous generalization of the multi-armed bandit problem.

Policy Gradient approaches only guarantee convergence to local optima, but we can incentivise the model to exit local minima by adding a entropy regularization term to the PPO loss. Since we actually only care about the best action (the z value with the least BIC score) found so far, rather than the final policy to which the model converges, we can additionally cycle the entropy coefficient so that when its value is high, the magnitude of the variance term increases and the model explores more, while when its value is low, the variance decreases and the model can converge to a local optima.

D. Optimization with CMA-ES

The objective function to be minimized by the CMA-ES algorithm was a the BIC score modified with a margin-maximization term:

$$f_d(z) = \text{BIC}\left(X, \Phi_d(z)\right) - \gamma\Gamma(z) \tag{4}$$

$$\Gamma(z) = \underbrace{\sum_{i<j}^{\{i,j\}<d} \min(|z_i - z_j|, \delta)}_{\Gamma_1(z)-\text{order stability}} + \underbrace{\sum_{k,k\geq d} \min(|z_k|, \delta)}_{\Gamma_2(z)-\text{edge stability}} \tag{5}$$

The BIC was left unscaled since the CMA-ES algorithm does not rely on the magnitude of the fitness values, only their ranking. The search space consists of large regions where the BIC score does not change since the graph remains the same. If there is no gradient in the values of the objective function then the ranking of the individuals in the population becomes arbitrary and the CMA-ES algorithm might become a random walk. To prevent this outcome, we add a margin-maximization term, $\Gamma(z)$ to the objective function, which helps prioritize solutions that are away from the transition boundaries, adding stability by preventing the population from inadvertently getting polluted by poor fitness individuals. Altering a single edge can change the BIC score by about $\ln(n)$, $\Gamma_1(z)$ by $d \times \delta$, and $\Gamma_2(z)$ by δ. Thus, by using $\gamma = 0.2\frac{\ln(n)}{d\delta}$, we can ensure that it does not violate the consistency of the BIC score while still providing a gradient to the CMA-ES algorithm.

The causal ordering of the nodes is derived from the order of the elements of the node potential vector. $\Gamma_1(z)$, thresholded by δ, tries to maximise the difference in element values so that the order of elements does not change with small perturbations. Edge existence is controlled by crossing the $z_i = 0, i \geq d$ hyperplane, so $\Gamma_2(z)$ tries to move z_i away from the edge change boundary to add stability, again with a threshold of δ.

The CMA-ES algorithm terminates when, among other reasons, the difference in the best objective function values over the last few generations falls below a set threshold, or the elements of the standard deviation matrix of the search distribution fall below a set threshold. To allow the CMA-ES algorithm to escape these local minima, just like the entropy cycling of the PPO algorithm, we allow the CMA-ES algorithm to do restarts, each time with slightly larger starting standard deviation, until the limit of the number of function evaluations is reached.

V. EXPERIMENTS

A. Algorithms and Implementations

The version of the algorithm that uses PPO is labelled *Relcadilac* (Reinforcement Learning for Causal Discovery under Latent Confounding) and the version using CMA-ES is called *CMA-ES*. Comparison is made with the *DCD* algorithm cite and the *GFCI* algorithm cite. The authors' original implementation[2] was used for the DCD algorithm while for the GFCI algorithm we use the `py-tetrad`[3] python library [31] which is a wrapper around the Tetrad suite of programs [32]. The hyperparameters for Relcadilac and CMA-ES are provided in Appendix D, while default configurations were used for DCD and GFCI, except for allowing one restart for DCD. The Stable-Baselines3 [33] python library was used for the PPO algorithm while the `cma`[4] [34] python library was leveraged for their implementation of the CMA-ES algorithm.

B. Data Generation

Graphs are generated using a modification of Erdős-Rényi random graph generation model [35] to account for the presence of bidirected edges and to ensure that the directed edges form a DAG, in addition to the requirements for generating bow-free and ancestral graphs. The presence of these varied constraints mean that an individual generated graph might not have the exact required number of edges or the requested ratio of directed to bidirected edges, however, over several instances, the statistical expected values align closely with the requested values.

The graph generator accepts the number of nodes needed in the graph (d), the average degree of the graph skeleton ($\bar{\rho}$), and the fraction of directed edges ($f^{\rightarrow}$) required as inputs. It produces a graph with $|\hat{E}|$ actual edges and $f^{\hat{\rightarrow}}$ actual fraction of directed edges. By requiring the generator to regenerate the graph, we guarantee that it satisfies the following constraints: $|\hat{E}| = \bar{\rho}d/2 \pm 5$ and $f^{\hat{\rightarrow}} = f^{\rightarrow} \pm 0.1$.

The elements of the structural coefficients matrix $\mathbf{\Theta}$ are sampled from the uniform distribution $\mathcal{U}(\pm[0.5, 2])$, while the upper triangular off-diagonal elements of the error covariance matrix $\mathbf{\Omega}$ are sampled from $\mathcal{U}(\pm[0.4, 0.7])$ and mirrored on the lower triangle to ensure symmetry. The diagonal elements of $\mathbf{\Omega}$ are the sum of the off-diagonal elements and samples from $\mathcal{U}([0.7, 1.2])$ to ensure that $\mathbf{\Omega}$ is positive definite. The data itself is then sampled from $\mathcal{N}\left(\mathbf{0}, (I - \mathbf{\Theta})^{-1}\mathbf{\Omega}(I - \mathbf{\Theta})^{-\top}\right)$ where the covariance matrix is derived from 2.

C. Experiments

In order to evaluate the performance of the proposed algorithms across a wide range of input graphs, four parameters were chosen and varied: the number of nodes, the sample size,

[2] https://gitlab.com/rbhatta8/dcd

[3] https://github.com/cmu-phil/py-tetrad

[4] https://github.com/CMA-ES/pycma

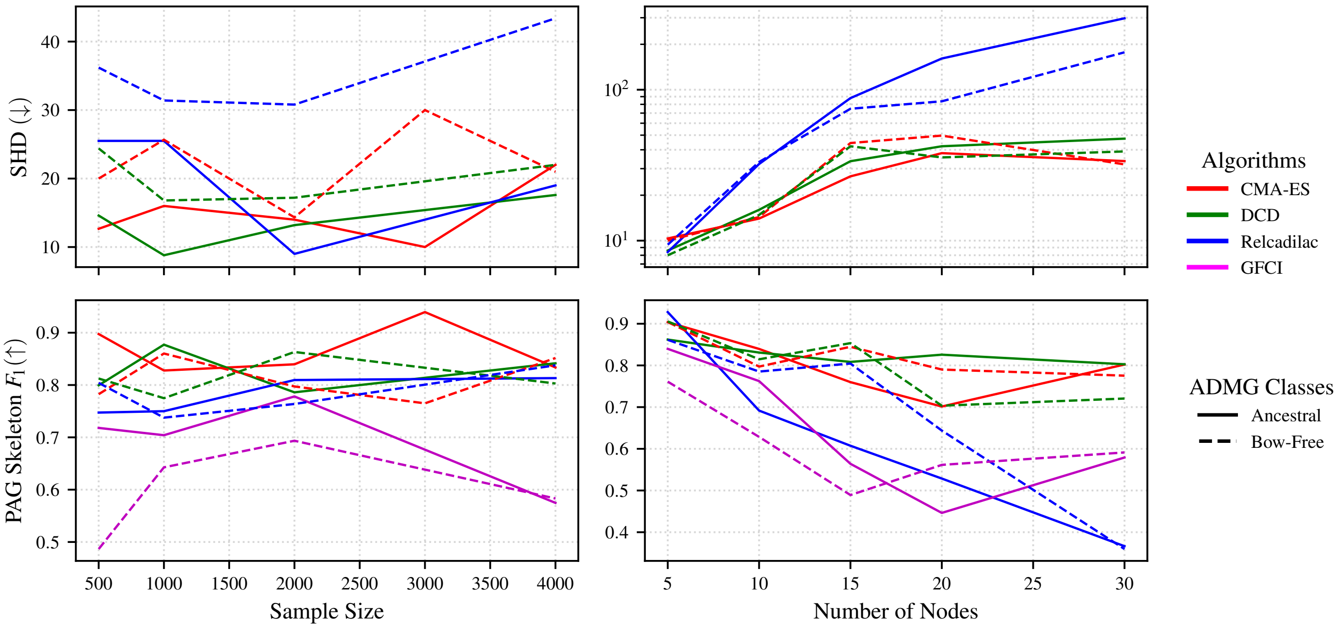

Fig. 1. **Performance of Causal Discovery Algorithms on Linear Gaussian Ancestral and Bow-Free ADMGs** We compare Structural Hamming Distance (SHD, lower is better) against the ground truth ADMG graph and F_1 scores on the skeleton of the Partial Ancestral Graph (higher is better). The algorithm compared are the CMA-ES implementation of our algorithm (CMA-ES), the PPO implementation of our algorithm (Relcadilac), DCD [23], and GFCI [19]. For all runs, the fraction of directed edges was 0.6 and the average degree was 4. When sample size was varied, the number of nodes was kept to 10 (so we can expect 20 edges per graph on average). When the number of nodes was varied, the sample size was kept to 2000. The graphs depict the average value over 5 runs. The SHD scores on the top-right graph are in log-scale to accommodate the Relcadilac scores without losing legibility for the scores of the other algorithms. All other plots are in linear scale.

the average degree, and the fraction of directed edges. Each algorithm was run with the same set of parameters at least three times and the data averaged to ensure more representative results. For each set of parameters, the algorithms were run on both ancestral and bow-free graph classes.

The identifiablity of ancestral ADMGs means that we can directly compare ADMG predictions from the algorithms with the ground truth ADMG graph. Bow-free ADMGs, being only almost-everywhere identifiable, means that there might be more than one ADMG which is observationally equivalent to the ground truth ADMG. Owing to this, following the same logic as [23], our algorithm first converts the proposed ADMG to a MAG, then converts the MAG to a PAG using the java-based Tetrad library. Our algorithm reports both the proposed bow-free or ancestral ADMG and the derived PAG. While the DCD algorithm also provides both the ADMG and the corresponding PAG, the GFCI algorithm only outputs a PAG. Thus, the GFCI algorithm is only included when comparing metrics derived from the PAG.

The primary metrics we compare are the Structural Hamming Distance against the ground truth ADMG graph, and the F_1 score for edge adjacencies computed on the PAG skeleton. The SHD score computes the number of addition and deletion operations (reversels therefore count as two operations) that need to be performed on the edges of the proposed graph to convert it to the ground truth graph. Additions or removal of bidirected edges only count as one operation. Graphs for other metrics like fractional BIC excess, which computes how much larger the predicted BIC score is than the true BIC score as a fraction of the true BIC score, and runtime graphs are presented in appendix E. The graphs comparing the performance of the algorithms with variation in fraction of directed edges and in the average degree are also delegated to appendix E due to lack of space.

To move beyond synthetic data, we evaluate the proposed algorithms on the real world flow cytometry dataset called the Sachs dataset [36]. The ground truth graph of the Sachs dataset is not consistent despite its widespread use (like in [37] and [38], so we use the graph from Figure 2 of the original paper (see reproduction in appendix F). The total number of edges in the Sachs dataset is 16 where one of the edges is bidirected, and it has 11 nodes. The observational component of the Sachs dataset is only 853 samples. This is likely the reason why the BIC scores for the proposed graphs are all lower than the BIC

TABLE I
PERFORMANCE OF ALGORITHMS ON SACHS DATASET

	SHD (↓)	$\lvert E \cap \hat{E}\rvert / \lvert \hat{E}\rvert^1$(↑)	**PAG F$_1$**^{3}(↑)	τ^2(↓)
DCD	53	3 / 43	0.47	249.5
CMA-ES	22	1 / 8	0.58	105.7
Relcadilac	23	1 / 9	0.48	1029.2

[1] $\lvert E \cap \hat{E}\rvert$ is the number of correct predicted edges, and $\lvert \hat{E}\rvert$ is the total number of predicted edges.
[2] τ is the runtime of the algorithm in seconds.
[3] The F_1 score is computed on the skeleton of the PAG.

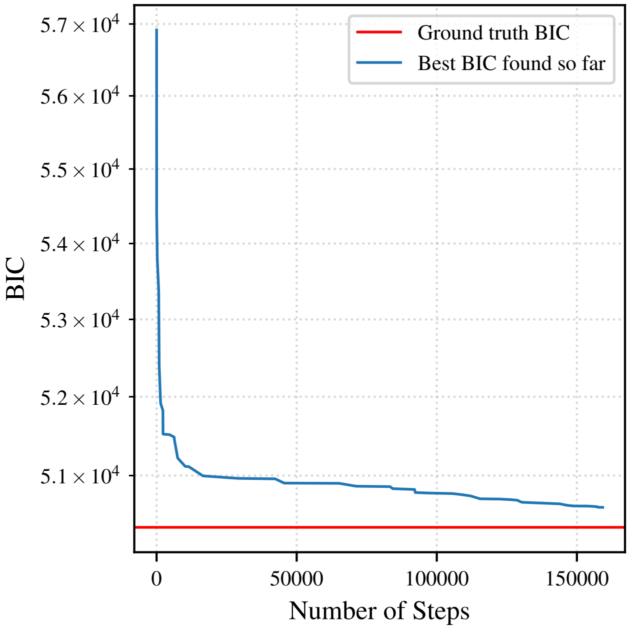

Fig. 2. The graph plots the best BIC score found at each time step for the Relcadilac algorithm. This run was made for a 20 node ancestral graph, 4 average degree (meaning 40 edges on average), 2000 sample size, running for 2 hours and 43 minutes. The final SHD score obtained for the 20-node graph was 35, which is in contrast to the 161 average 20-node SHD seen in figure 1. The graph is intended to show the large sample requirement (number of environment interactions) of the PPO algorithm before it provides results comparable to CMA-ES and DCD, especially on larger node graphs.

scores for the true graph. See table I for the results.

VI. Results

The Relcadilac algorithm (where we use PPO for optimization) often performs poorly on most experiments when compared to the performance of the CMA-ES and DCD algorithms. It mostly outperforms the GFCI algorithm on comparable metrics. The likely cause for this is the sample inefficiency of the PPO algorithm [39] [40]. The authors in [16], who use PPO for DAG search, run their algorithm for 1.28×10^6 function evaluations. Since they only need to compute the BIC score for DAGs, they are able to use the Sum of Square Residuals for the calculation:

$$S_{BIC}(X, \mathcal{G}) = -\left(n \sum_{i=1}^{d} \ln \frac{\text{SSR}_i}{n} + |\mathcal{G}| \ln n \right)$$

This likelihood term in this score can be decomposed on a per-node basis, and thus, the contribution of each node and potential parent set is fixed, given the dataset. This allows partial likelihood scores to be saved and re-used throughout the learning process, considerably speeding up the computation, especially for sparse or small graphs.

The RICF-based BIC computation for bow-free ADMGs does not permit a per-node decomposition, and has a time complexity of $\mathcal{O}(nd^3 + d^4)$. This massively slows down each BIC computation, only allowing $\sim$ 50 BIC evaluations per second on available resources. Thus, the number of BIC evaluations for Relcadilac was restricted to 8×10^4 which still requires $\sim$ 30 minutes for 10 node, 4 average degree, 2000 sample datasets. This decrease, coupled with the more complicated search space likely resulted in the observed poor performance of Relcadilac. See figure 2 for illustration.

The CMA-ES algorithm shows comparable or better performance than the DCD algorithm [23] across metrics. It is considerably faster than the PPO algorithm due to being far more sample efficient.

None of the algorithms compared are able to consistently able to recover the ground truth ADMGs on synthetic or real world data. However, the CMA-ES algorithm shows that it is able to recover the PAG skeleton reasonably well on synthetic data, and has the smallest SHD and highest PAG Skeleton F_1 score on the Sachs dataset among the compared algorithms, while also being the fastest.

VII. Conclusion

In this work, we propose an adaptable, modular framework for Causal Discovery on Linear Gaussian data in the presence of Latent Confounding. The framework leverages the proposed one-step characterizations of bow-free and ancestral ADMGs to bypass the requirement for differentiable constraints for acyclicity and for the aforementioned graph classes. The framework was evaluated with two black-box optimization algorithms: PPO and CMA-ES, whose performance was evaluated across several synthetic graph parameters and on real-world datasets.

Future work can explore using a more sample efficient off-policy reinforcement learning algorithm like SAC [39] to improve the convergence time of the framework. Another potential avenue for speeding up the algorithm is to use a decomposition of the RICF algorithm score computation over bidirected connected components in the graph (like in [22]). The partial scores for these components can then be cached and re-used. These partial scores will likely be more beneficial on a sequential graph generation algorithm where successive graphs are closely related to one another than in the current framework which, between distribution updates, essentially samples graphs i.i.d.

Linear Gaussian arid ADMGs, which are subsets of bow-free ADMGs, have strong identifiability guarantees in contrast to bow-free ADMGs which are only almost-everywhere identifiable. Arid ADMGs, however, unlike ancestral ADMGs, are able to encode Verma Constraints. Future work might try and come up with vector-to-arid-admg formulations in line with the proposals for bow-free and ancestral ADMGs in this paper.

References

[1] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. USA: Cambridge University Press, 2009.

[2] P. Garg and T. Fetzer, “Causal claims in economics,” *ArXiv*, vol. abs/2501.06873, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:275470763

[3] P. Sanchez, J. P. Voisey, T. Xia, H. I. Watson, A. Q. O’Neil, and S. A. Tsaftaris, “Causal machine learning for healthcare and precision medicine,” *Royal Society Open Science*, vol. 9, no. 8, p. 220638, 2022.

[4] V. Lagani, S. Triantafillou, G. Ball, J. Tegnér, and I. Tsamardinos, *Probabilistic Computational Causal Discovery for Systems Biology*. Cham: Springer International Publishing, 2016, pp. 33–73. [Online]. Available: https://doi.org/10.1007/978-3-319-21296-8_3

[5] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí, E. H. van Nes, J. Peters, R. Quax, M. Reichstein, M. Scheffer, B. Schölkopf, P. Spirtes, G. Sugihara, J. Sun, K. Zhang, and J. Zscheischler, "Inferring causation from time series in earth system sciences," *Nature Communications*, vol. 10, no. 1, p. 2553, Jun 2019. [Online]. Available: https://doi.org/10.1038/s41467-019-10105-3

[6] H. O. Stolberg, G. Norman, and I. Trop, "Randomized controlled trials," *American Journal of Roentgenology*, vol. 183, no. 6, pp. 1539–1544, 2004, pMID: 15547188. [Online]. Available: https://doi.org/10.2214/ajr.183.6.01831539

[7] G. W. Imbens and D. B. Rubin, *CLASSICAL RANDOMIZED EXPERIMENTS*. Cambridge University Press, 2015, p. 45–46.

[8] A. Zanga, E. Ozkirimli, and F. Stella, "A survey on causal discovery: Theory and practice," *International Journal of Approximate Reasoning*, vol. 151, pp. 101–129, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888613X22001402

[9] H. Kesteloot, S. Sans, and D. Kromhout, "Dynamics of cardiovascular and all-cause mortality in western and eastern europe between 1970 and 2000," *Eur. Heart J.*, vol. 27, no. 1, pp. 107–113, Jan. 2006.

[10] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[11] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978. [Online]. Available: http://www.jstor.org/stable/2958889

[12] D. Geiger and D. Heckerman, "Learning gaussian networks," in *Uncertainty in Artificial Intelligence*, R. L. de Mantaras and D. Poole, Eds. San Francisco (CA): Morgan Kaufmann, 1994, pp. 235–243. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558603325500353

[13] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "Dags with no tears: Continuous optimization for structure learning," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/e347c51419ffb23ca3fd5050202f9c3d-Paper.pdf

[14] M. Nauta, D. Bucur, and C. Seifert, "Causal discovery with attention-based convolutional neural networks," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 312–340, 2019. [Online]. Available: https://www.mdpi.com/2504-4990/1/1/19

[15] Y. Wang, V. Menkovski, H. Wang, X. Du, and M. Pechenizkiy, "Causal discovery from incomplete data: A deep learning approach," *ArXiv*, vol. abs/2001.05343, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210701213

[16] B. Duong, H. Le, B. Huang, and T. Nguyen, "Reinforcement learning for causal discovery without acyclicity constraints," *Transactions on Machine Learning Research*, 2025. [Online]. Available: https://openreview.net/forum?id=sNzBi8rZTy

[17] P. Spirtes, C. Glymour, S. N., and Richard, *Causation, Prediction, and Search*. Mit Press: Cambridge, 1993.

[18] D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson, "Learning high-dimensional directed acyclic graphs with latent and selection variables," *The Annals of Statistics*, vol. 40, no. 1, pp. 294–321, 2012. [Online]. Available: http://www.jstor.org/stable/41713636

[19] J. M. Ogarrio, P. Spirtes, and J. Ramsey, "A hybrid causal search algorithm for latent variable models," in *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, ser. Proceedings of Machine Learning Research, A. Antonucci, G. Corani, and C. P. Campos, Eds., vol. 52. Lugano, Switzerland: PMLR, 06–09 Sep 2016, pp. 368–379. [Online]. Available: https://proceedings.mlr.press/v52/ogarrio16.html

[20] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour, "A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images," *International Journal of Data Science and Analytics*, vol. 3, no. 2, pp. 121–129, Mar 2017. [Online]. Available: https://doi.org/10.1007/s41060-016-0032-z

[21] M. Drton, M. Eichler, and T. S. Richardson, "Computing maximum likelihood estimates in recursive linear models with correlated errors," *Journal of Machine Learning Research*, vol. 10, no. 81, pp. 2329–2348, 2009. [Online]. Available: http://jmlr.org/papers/v10/drton09a.html

[22] C. Nowzohour, M. Maathuis, R. Evans, and P. Bühlmann, "Distributional equivalence and structure learning for bow-free acyclic path diagrams," *Electronic Journal of Statistics*, vol. 11, pp. 5342–5374, 01 2017.

[23] R. Bhattacharya, T. Nagarajan, D. Malinsky, and I. Shpitser, "Differentiable causal discovery under unmeasured confounding," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, Apr 2021, pp. 2314–2322. [Online]. Available: https://proceedings.mlr.press/v130/bhattacharya21a.html

[24] P. Ma, R. Ding, Q. Fu, J. Zhang, S. Wang, S. Han, and D. Zhang, "Scalable differentiable causal discovery in the presence of latent confounders with skeleton posterior," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2141–2152. [Online]. Available: https://doi.org/10.1145/3637528.3672031

[25] Y. Yu, T. Gao, N. Yin, and Q. Ji, "Dags with no curl: An efficient dag structure learning approach," 2021. [Online]. Available: https://arxiv.org/abs/2106.07197

[26] M. Ashman, C. Ma, A. Hilmkil, J. Jennings, and C. Zhang, "Causal reasoning in the presence of latent confounders via neural ADMG learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=dcN0CaXQhT

[27] J. J. Lee, R. Bhattacharya, R. Nabi, and I. Shpitser, "Ananke: A python package for causal inference using graphical models," *arXiv preprint arXiv:2301.11477*, 2023.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[29] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, June 2001.

[30] N. Hansen, "The cma evolution strategy: A tutorial," 2023. [Online]. Available: https://arxiv.org/abs/1604.00772

[31] J. Ramsey and B. Andrews, "Py-tetrad and rpy-tetrad: A new python interface with r support for tetrad causal search," in *Proceedings of the 2023 Causal Analysis Workshop Series*, ser. Proceedings of Machine Learning Research, E. Kummerfeld, S. Ma, E. Rawls, and B. Andrews, Eds., vol. 223. PMLR, Aug 2023, pp. 40–51. [Online]. Available: https://proceedings.mlr.press/v223/ramsey23a.html

[32] J. Ramsey, K. Zhang, M. Glymour, R. S. Romero, B. Huang, Immé, Ebert-Uphoff, S. M. Samarasinghe, E. A. Barnes, and C. Glymour, "Tetrad - a toolbox for causal discovery," in *8th international workshop on climate informatics*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:201054499

[33] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[34] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma on Github," Zenodo, DOI:10.5281/zenodo.2559634, Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2559634

[35] P. L. Erdős and A. Rényi, "On the evolution of random graphs," *Transactions of the American Mathematical Society*, vol. 286, pp. 257–257, 1984. [Online]. Available: https://api.semanticscholar.org/CorpusID:6829589

[36] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, 2005. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1105809

[37] J. Ramsey and B. Andrews, "Fask with interventional knowledge recovers edges from the sachs model," *ArXiv*, vol. abs/1805.03108, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:13686921

[38] J. M. Mooij, S. Magliacane, and T. Claassen, "Joint causal inference from multiple contexts," *Journal of Machine Learning Research*, vol. 21, pp. 99:1–99:108, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:126017772

[39] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a

stochastic actor," 2018. [Online]. Available: https://openreview.net/forum?id=HJjvxl-Cb

[40] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[41] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH '08. New York, NY, USA: Association for Computing Machinery, 2008. [Online]. Available: https://doi.org/10.1145/1401132.1401152

Appendix A
Proof of Theorem 1

Theorem 2 (Surjectivity)**.** *For all $d \in \mathbb{N}^+$, let $\Phi_d : \mathbb{R}^{d^2} \rightarrow \{0,1\}^{d \times d} \times \{0,1\}^{d \times d}$ represent both the mappings defined in definition 1, and let $\mathbb{S}_d$ denote the set of all d node ADMGs belonging to the corresponding target class, then $\forall\ \mathcal{G} \in \mathbb{S}_d \ \exists\ z \in \mathbb{R}^{d^2}$ such that $\Phi_d(z) = \mathcal{G}$.*

In order to show the surjectivity of the Φ_d mappings, we can first show that every point in $\mathbb{R}^{d^2}$ maps to a valid ADMG of the corresponding graph class and then we can show that for every valid graph, there exists a vector that maps to it.

Lemma 3. *For all $d \in \mathbb{N}^+$, let $\Phi_d : \mathbb{R}^{d^2} \rightarrow \{0,1\}^{d \times d} \times \{0,1\}^{d \times d}$ and $\mathbb{S}_d$ be as defined in theorem 1. Then, $\forall\ z \in \mathbb{R}^{d^2}, \Phi_d(z) \in \mathbb{S}_d$.*

Proof. For the directed component of $\Phi_d(z)$, we can follow the same argument as [25] and show that it forms a DAG. Let $D = \Phi_d(z) = H(E_{\rightarrow} + E_{\rightarrow}^{\top}) \odot H(\text{grad}(p))$. Then, $p_i > p_j \implies p_j - p_i < 0 \implies \text{grad}(p)_{ij} < 0 \implies H(\text{grad}(p)_{ij}) = 0$. Thus, $p_i > p_j$ means that the edge $i \rightarrow j$ does not exist. So, $p_i < p_j$ is a necessary condition for the edge $i \rightarrow j$ to exist. Now, if we assume that a cycle exists in the graph represented by D: $(i_1 \rightarrow \ldots \rightarrow i_k \rightarrow i_1)$, we have that $p_{i_1} < \ldots < p_{i_k} < p_{i_1}$. This is a violation of the total-ordering property of $\mathbb{R}$ and a contradiction. Thus, $\Phi_d(z)[D]$ must be acyclic and represents a DAG.

For the bidirected component of $\Phi_d(z)$, let $B = \Phi_d(z) = H(E_{\leftrightarrow} + E_{\leftrightarrow}^{\top}) \odot (I - M) \odot (I - M^{\top})$ where $M = D$ or $M = D^{+}$ depending on the mapping. $M_{ij} = 1 \implies (I - M)_{ij} = 0 \implies B_{ij} = 0$. Again for the symmetric component: $M_{ij} = 1 \implies (M^{\top})_{ji} = 1 \implies (I - M^{\top})_{ji} = 0 \implies B_{ji} = 0$. Thus, $M_{ij} = 0$ is a necessary condition for the bidirected edge $i \leftrightarrow j$ to exist. Following a parallel argument for $M^{\top}$, $M_{ji} = 0$ is also a necessary condition for bidirected edge $i \leftrightarrow j$ to exist. If $M = D$, the absence of directed edges $i \rightarrow j$ and $j \rightarrow i$ are necessary conditions for bidirected edge $i \leftrightarrow j$ to exist. Thus, Φ_d^{BF} is bow-free. If $M = D^{+}$, from the definition of the transitive closure, the absence of directed paths between $i \rightarrow \ldots \rightarrow j$ and $j \rightarrow \ldots \rightarrow i$ are necessary conditions for bidirected edge $i \leftrightarrow j$ to exist. Thus, Φ_d^{AN} is ancestral. □

Lemma 4. *For all $d \in \mathbb{N}^+$, let Φ_d and $\mathbb{S}_d$ be as defined in theorem 1. Then, $\forall\ \mathcal{G} \in \mathbb{S}_d \ \exists\ z \in \mathbb{R}^{d^2}$ such that $\Phi_d(z) = \mathcal{G}$.*

Proof. Let the graph $\mathcal{G}$ be represented by the tuple of binary adjacency matrices (D, B) such that the edge $i \rightarrow j$ exists if and only if $D_{ij} = 1$, and the edge $i \leftrightarrow j$ exists if and only if $B_{ij} = B_{ji} = 1$. The directed component of the ADMG, $\mathcal{G}$ contains no cycles. Thus, there exists a topological ordering of its vertices. Let $\{\pi_1, \pi_2, \ldots, \pi_d\}$ be a permutation of $\{1, \ldots, d\}$ that represents that topological order such that $D_{ij} = 1 \implies \pi_i < \pi_j$. We can construct $p, E_{\rightarrow}$, and $E_{\leftrightarrow}$ (and therefore, z) as given in definition 1, such that $\Phi_d(z) = \mathcal{G}$

as follows (a similar proof is provided in [16] for the case of DAGs):

$$p_k = \pi_k \tag{6}$$

We construct the lower-triangular edge potential matrices as follows:

$$(E_\to)_{ij} = \begin{cases} +1, & \text{if } i > j \text{ and } D_{ij} + D_{ji} = 1 \\ -1, & \text{otherwise} \end{cases} \tag{7}$$

$$(E_\leftrightarrow)_{ij} = \begin{cases} +1, & \text{if } i > j \text{ and } B_{ij} = B_{ji} = 1 \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

We now verify that $p, E_\to, E_\leftrightarrow$, when mapped through $\Phi_d(z)$ as shown in definition 1, give the exact adjacency matrices D and B.

If the edges $i \to j$ or $j \to i$ are present in $\mathcal{G}$, we have that $D_{ij} + D_{ji} = 1$ (since $\mathcal{G}$ is an acyclic graph, only one of the edges can be present). Since $E_\to$ is a lower-triangular matrix, only one of $(E_\to)_{ij}$ and $(E_\to)_{ji}$ can be 1 depending on whether $i > j$ or $j > i$ respectively. Thus, if there is a directed edge between nodes i and j we have that $(E_\to)_{ij} + (E_\to)_{ji} = 1 > 0$, and $(E_\to)_{ij} + (E_\to)_{ji} = -1 < 0$ otherwise. Thus, $H(E_\to + E_\to^\top)$ is the undirected version of D.

For any directed edge in $\mathcal{G}$, $i \to j$, we have that $\pi_i < \pi_j$ (ancestors before descendants in topological order) which means that $p_i < p_j \implies p_j - p_i > 0 \implies \text{grad}(p)_{ji} > 0$. Thus, $H(\text{grad}(p)$ correctly encodes the direction of all the directed edges in $\mathcal{G}$. So, through the elementwise product, $H(E_\to + E_\to^\top)$ masks the edges in $H(\text{grad}(p))$ that don't exist in G, and $H(\text{grad}(p))$ gives direction to the edges from $H(E_\to + E_\to^\top)$, giving the exact adjacency matrix as D.

Let $\hat{D}$ be the directed component of $\Phi_d(z)$ derived from p and $E_\to$ as above, and shown to be identical to D. If the edge $i \leftrightarrow j$ exists in $\mathcal{G}$, we have that $B_{ij} = B_{ji} = 1 \implies (E_\leftrightarrow)_{ij} + (E_\leftrightarrow)_{ji} = 1 > 0 \implies H(E_\leftrightarrow + E_\leftrightarrow^\top) = 1$. If the edge $i \leftrightarrow j$ does not exist in $\mathcal{G}$, we have that $B_{ij} = B_{ji} = 0 \implies (E_\leftrightarrow)_{ij} + (E_\leftrightarrow)_{ji} = -1 < 0 \implies H(E_\leftrightarrow + E_\leftrightarrow^\top) = 0$. Thus, $H(E_\leftrightarrow + E_\leftrightarrow^\top) = B$. We now show that in this construction, the rest of the terms in $\Phi_d(z)[B] = \hat{B}$ don't change $H(E_\leftrightarrow + E_\leftrightarrow^\top)$ which would mean that the value of z, as constructed, when passed through the mappings, produces the same bidirected adjacency matrix $\hat{B}$ as the bidirected adjacency matrix, B of $\mathcal{G}$.

If $\mathcal{G}$ is bow-free, when $B_{ij} = B_{ji} = 1$ we must have, from the definition of being bow-free, that $D_{ij} = D_{ji} = 0 \implies (I - D)_{ij} = (I - D^\top)_{ij} = (I - D)_{ji} = (I - D^\top)_{ji} = 1$. Since $\hat{D} = D$, the same statement can be made for $\hat{D}$ as well. Thus, $\hat{B} = H(E_\leftrightarrow + E_\leftrightarrow^\top) \odot (I - \hat{D}) \odot (I - \hat{D}^\top) = H(E_\leftrightarrow + E_\leftrightarrow^\top) = B$. So, for bow-free ADMGs, we have that $\hat{D} = D$ and $\hat{B} = B$.

The same parallel argument can be made for the case where $\mathcal{G}$ is ancestral. The presence of a bidirected edge between two nodes implies the absence of a directed path between the same two nodes. Through the definition of the transitive closure, this implies the absence of a direct single edge between the two nodes. Thus, $B_{ji} = B_{ij} = 1$ means that $[(I - \hat{D}^+) \odot (I - \hat{D}^{+\top})]_{ij} = [(I - \hat{D}^+) \odot (I - \hat{D}^{+\top})]_{ji} = 1$. When $B_{ij} = B_{ji} = 0$, the full three-term product for the corresponding indices will also be 0 since we are taking element-wise products. Thus, $\hat{B} = H(E_\leftrightarrow + E_\leftrightarrow^\top) \odot (I - \hat{D}^+) \odot (I - \hat{D}^{+\top}) = H(E_\leftrightarrow + E_\leftrightarrow^\top) = B$. So, for ancestral ADMGs, we have that $\hat{D} = D$ and $\hat{B} = B$. □

Appendix B
Proof of Lemma 1

Proof. If z decomposes into $p, E_\to, E_\leftrightarrow$, then, by construction, λz decomposes into $p, E_\to, E_\leftrightarrow$. The heaviside step function is scale invariant for positive scaling since it only cares about the sign of the input term:

$$\forall\ \lambda > 0, H(\lambda x) = H(x) \tag{9}$$

Scaling the input to the gradient flow operator by λ scales its output value by λ:

$$\forall \lambda > 0 \ \text{grad}(\lambda u) = \lambda u_j - \lambda u_i = \lambda(u_j - u_i) = \lambda \ \text{grad}(u) \tag{10}$$

Let $\Phi_d(\lambda z)[D] = D_\lambda$ and $\Phi_d(z)[D] = D$. Then using 9 and 10, $D_\lambda = H(\lambda E_\to + \lambda E_\to^\top) \odot H(\text{grad}(\lambda p)) = H(E_\to + E_\to^\top) \odot H(\lambda \ \text{grad}(p)) = H(E_\to + E_\to^\top) \odot H(\text{grad}(p)) = D$. Since $D = D_\lambda$ and $H(\lambda E_\leftrightarrow + \lambda E_\leftrightarrow^\top) = H(E_\leftrightarrow + E_\leftrightarrow^\top)$ we have that $\Phi_d^{BF}(\lambda z)[B] = \Phi_d^{BF}(z)[B]$. Since $D = D_\lambda$, their transitive closures are also identical, and through a parallel argument for the ancestral mapping, we have that $\Phi_d^{AN}(\lambda z)[B] = \Phi_d^{AN}(z)[B]$. Thus, $\Phi_d(\lambda z) = \Phi_d(z) \ \forall \lambda > 0$. □

Appendix C
Proof of Lemma 2

Let $\mathcal{U} \subset \mathbb{R}^{d^2}$ be an open set containing the origin. Then, there exists $\epsilon > 0$ such that the open ball $\mathcal{B}_\epsilon(0) \subset \mathcal{U}$. Let $G \in \mathbb{S}_d$ be an arbitrary valid graph. Then by lemma 4, there exists $z \in \mathbb{R}^{d^2}$ such that $\Phi_d(z) = G$. Consider the scaled vector $z* = \frac{\epsilon}{2\|z\|} z$. $\|z*\| = \frac{\epsilon}{2} < \epsilon \implies z* \in \mathcal{B}_\epsilon(0) \subset \mathcal{U}$. From lemma 2, by setting $\lambda = \frac{\epsilon}{2\|z\|} > 0$, we have that $\Phi_d(z*) = \Phi_d(z) = G$. Thus, for every $\mathcal{G} \in \mathbb{S}_d$ there exists $z* \in \mathcal{U}$ such that $\Phi_d(z*) = \mathcal{G}$. Therefore, $\Phi_d(\mathcal{U}) = \mathbb{S}_d$.

Appendix D
Algorithm Hyperparameters

All algorithms were run on a HP Omen 16 laptop with a Ryzen 7 7840HS CPU and a Nvidia RTX 4060 GPU. The GPU was only used for the neural network updates for the PPO algorithm through the use of the cuda toolkit [41]. All other computation was performed on the CPU.

A. PPO

The hyperparameters for the PPO algorithm are in table II. The entropy value is revised for every update to the policy, using the below formula:

$$S_t = \alpha_{\text{min}} + \frac{1}{2}(\alpha_0 - \alpha_{\text{min}}) S_t^{\cos} \exp\left(-\lambda \frac{t}{10 T_{\text{cycle}}}\right)$$

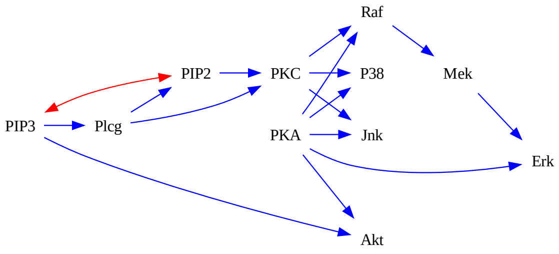

Fig. 3. The ground graph of the Sachs Dataset [36] derived from Fig 2. in the original paper.

where

$$S_t^{\cos} = \left(1 + \cos\left(\frac{2\pi(t \mod T_{\text{cycle}})}{T_{\text{cycle}}}\right)\right)$$

The α_0 and α_{min} terms keep the entropy within described ranges, the cosine term cycles the entropy up and down, and the exponential term decays the value of the entropy.

B. CMA-ES

The hyperparameters of the CMA-ES algorithm are given in table III. In the table, d is the number of nodes in the graph and n is the number of samples in the data. The default population size for the CMA-ES algorithm is: $p_{\text{size}}^{\text{default}} = \lfloor 4 + 3\ln(d^2) \rfloor$. Due to the high dimensionality of the search space, this value is scaled by k: $p_{\text{size}} = kp_{\text{size}}^{\text{default}}$. All other hyperparameter values are defaults taken from the `cma`[5] python library [34].

TABLE II
PPO Hyperparameters

Hyperparameter	Value
Batch Size (No. of parallel environments)	8
Training steps (Steps per environment)	10,000
Steps per environment per update	1
Data centered (mean 0)	Yes
Data standardized (mean 0, variance 1)	No
Advantage Normalization	Yes
Number of Epochs	1
Use State Driven Exploration (sde)	Yes
Do Entropy Annealing	Yes
Initial Entropy (α_0)	0.3
Minimum Entropy (α_{min})	0.005
Cycle Length (T_{cycle})	16,000
Damping Factor (λ)	0.5

TABLE III
CMA-ES Hyperparameters

Hyperparameter	Value
Maximum Objective Function Evaluations	40,000
Use Diagonal Covariance Matrix	Yes
Population Size Scale, k	4
Stability Threshold, δ	1
Stability Coefficient, γ	$0.2 \ln n/(\delta d)$
Number of Parallel Workers	12

Appendix E
Performance Plots

The plots comparing the performance of the DCD, Relcadilac, GFCI and CMA-ES algorithms across various metrics are provided in figure 4. The Fractional Excess BIC Score sometimes goes below 0 for the CMA-ES algorithm indicating that it has found a graph with a lower BIC score than the ground truth graph. This is most likely due to the smaller dataset sizes of 2000 samples, since the BIC score is only consistent in the limit of infinite data.

Appendix F
Sachs Dataset

The ground truth graph of the Sachs Dataset is reproduced in figure 3

[5] https://https://github.com/CMA-ES/pycma

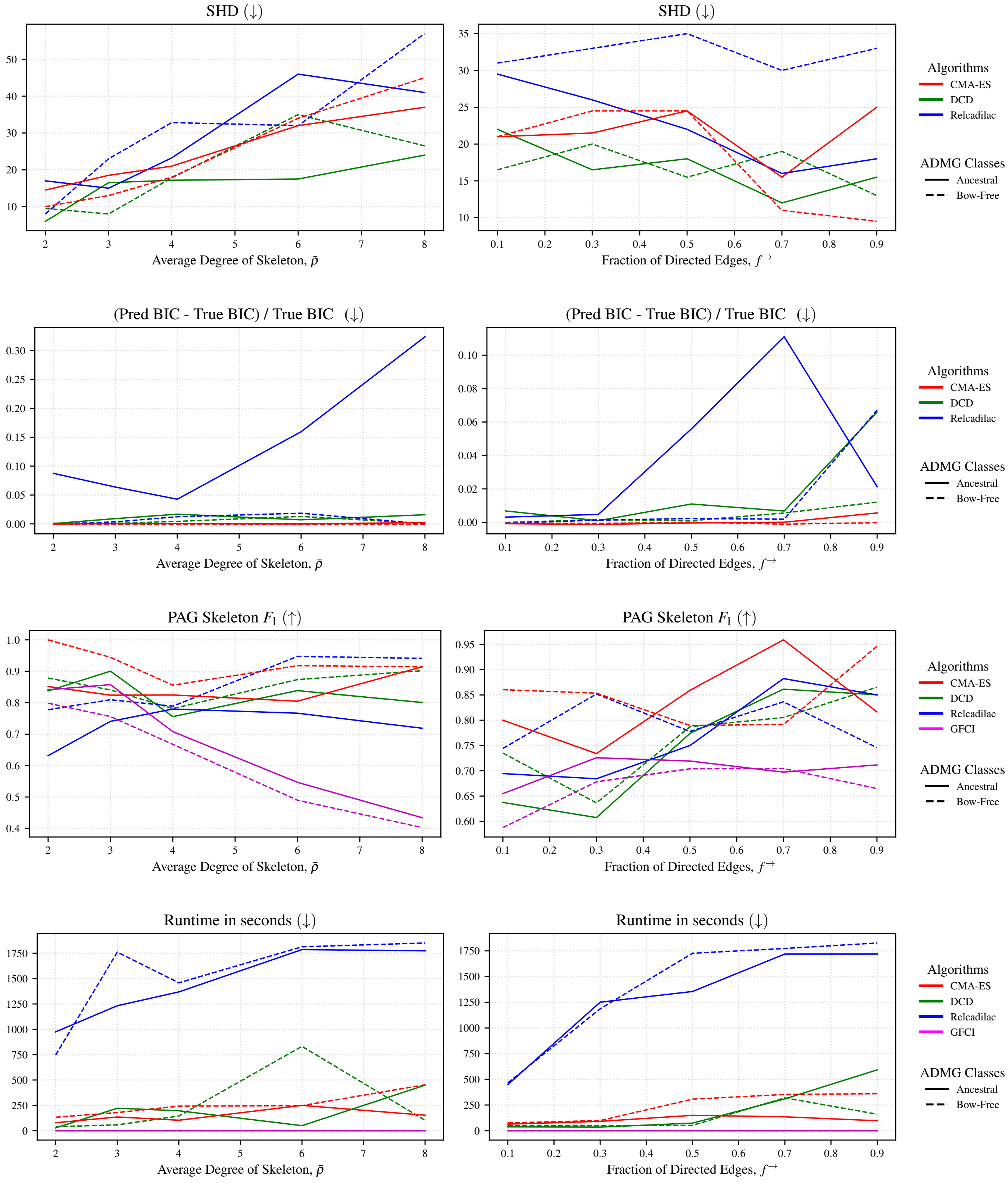

Fig. 4. **Performance of Causal Discovery Algorithms with varying graph degree and fraction of directed edges**. All algorithms are run on 10 node, 2000 sample graphs. When the degree of the graph was varied, the fraction of directed edges was fixed to 0.6. When the fraction of directed edges was varied, the degree of the graph was fixed to be 4. The graphs depict the average value over 5 runs. The metrics compared are the Structural Hamming Distance (lower is better), Fractional BIC Excess (lower is better) which is the difference between the BIC value of the predicted graph and the BIC value of the graph divided by the BIC of the true graph, the PAG Skeleton F_1 score (higher is better) which is the F_1 score of the edges of the skeleton of the PAG, and the runtime (lower is better) of the algorithms.