Journal of Machine Learning Research 26 (2025) 1-30 Submitted 10/24; Revised 9/25; Published 11/25

An Asymptotically Optimal Coordinate Descent Algorithm for Learning Bayesian Networks from Gaussian Models

Tong Xu tongxu2027@u.northwestern.edu
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA

Simge Küçükyavuz simge@northwestern.edu
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA

Ali Shojaie ashojaie@uw.edu
Department of Biostatistics
University of Washington
Seattle, WA 98195-1617, USA

Armeen Taeb ataeb@uw.edu
Department of Statistics
University of Washington
Seattle, WA 98195-4322, USA

Editor: Peter Richtarik

Abstract

This paper studies the problem of learning Bayesian networks from continuous observational data, generated according to a linear Gaussian structural equation model. We consider an ℓ_0-penalized maximum likelihood estimator for this problem, which is known to have favorable statistical properties but is computationally challenging to solve, especially for medium-sized Bayesian networks. We propose a new coordinate descent algorithm to approximate this estimator and prove several remarkable properties of our procedure: The algorithm converges to a coordinate-wise minimum, and despite the non-convexity of the loss function, as the sample size tends to infinity, the objective value of the coordinate descent solution converges to the optimal objective value of the ℓ_0-penalized maximum likelihood estimator. To the best of our knowledge, our proposal is the first coordinate descent procedure endowed with optimality guarantees in the context of learning Bayesian networks. Numerical experiments on synthetic and real data demonstrate that our coordinate descent method can obtain near-optimal solutions while being scalable.

Keywords: Directed acyclic graphs, ℓ_0-penalization, Non-convex optimization, Structural equation models

1. Introduction

Bayesian networks provide a powerful framework for modeling causal relationships among a collection of random variables. A Bayesian network is typically represented by a directed acyclic graph (DAG), where the random variables are encoded as vertices (or nodes), a directed edge from node i to node j indicates that i causes j, and the acyclic property of the graph prevents the occurrence of circular dependencies. If the DAG is known, it can be used to predict the behavior of the system under manipulations or interventions. However, in large systems such as gene regulatory networks, the DAG is not known a priori, making it necessary to develop efficient and rigorous methods to learn the graph from data. To solve this problem using only observational data, we assume that all relevant variables are observed and that we only have access to observational data.

Three broad classes of methods for learning DAGs from data are constraint-based, score-based, and hybrid. Constraint-based methods use repeated conditional independence tests to determine the presence of edges in a DAG. A prominent example is the PC algorithm and its extensions (Spirtes et al., 1993; Tsamardinos et al., 2006). While the PC algorithm can be applied in non-parametric settings, testing for conditional independencies is generally hard (Shah and Peters, 2018). Furthermore, even in the Gaussian setting, statistical consistency guarantees for the PC algorithm are shown under the *strong faithfulness* condition (Kalisch and Bühlmann, 2007), which is known to be restrictive in high-dimensional settings (Uhler et al., 2012). Score-based methods often deploy a penalized log-likelihood as a score function and search over the space of DAGs to identify a DAG with an optimal score. These approaches do not require the strong faithfulness assumption. However, statistical guarantees are not provided for many score-based approaches and solving them exactly suffers from high computational complexity. For example, learning an optimal graph using dynamic programming takes about 10 hours for a medium-size problem with 29 nodes (Silander and Myllymäki, 2006). Several papers (Küçükyavuz et al., 2023; Xu et al., 2025) offer speedup by casting the problem as a convex mixed-integer program, but finding an optimal solution with these approaches can still take an hour for a medium-sized problem. Finally, hybrid approaches combine constraint-based and score-based methods by using background knowledge or conditional independence tests to restrict the DAG search space (Tsamardinos et al., 2006; Nandy et al., 2018).

Several strategies have been developed to make score-based methods more scalable by finding approximate solutions instead of finding optimally scoring DAGs. One direction to find good approximate solutions is to resort to greedy-based methods, with a prominent example being the Greedy Equivalence Search (GES) algorithm (Chickering, 2002). GES performs a greedy search on the space of completed partially directed acyclic graphs (an equivalence class of DAGs) and is known to produce asymptotically consistent solutions (Chickering, 2002). Despite its favorable properties, GES does not provide optimality or consistency guarantees for any finite sample size. Further, the guarantees of GES assume a fixed number of nodes with a sample size going to infinity and do not allow for a growing number of nodes. Another direction is gradient-based approaches (Yu et al., 2019; Zheng et al., 2018), which relax the discrete search space over DAGs to a continuous search space, allowing gradient descent and other techniques from continuous optimization to be applied. However, the search space for these problems is highly non-convex, resulting in limited guarantees for convergence, even to a local minimum. Finally, another notable direction is based on *coordinate descent*; that is iteratively maximizing the given score function over a single parameter, while keeping the remaining parameters fixed and checking that the resulting model is a DAG at each update (Aragam and Zhou, 2015; Aragam et al., 2019; Fu and Zhou, 2013; Ye et al., 2020). While coordinate descent algorithms have shown significant promise in learning large-scale Bayesian networks, to the best of our knowledge, they do not come with convergence and optimality guarantees.

Contributions: We propose a new score-based coordinate descent algorithm for learning Bayesian networks from Gaussian linear structural equation models. Remarkably, unlike prior coordinate descent algorithms for learning Bayesian networks, our procedure provably i) converges to a coordinate-wise minimum, and ii) yields optimally scoring DAGs in the large sample limit despite the non-convex nature of the problem. Moreover, we characterize the convergence rate as a function of both the sample size and the number of nodes. A byproduct of our analysis is a characterization of the optimization landscape: in the large-sample regime, all local minima achieve objective values close to a global minimum, and exactly match it in the limit—a result that may be of independent interest.

As a scoring function for this approach, we deploy an ℓ_0-penalized Gaussian log-likelihood, which implies that optimally-scoring DAGs are solutions to a highly non-convex ℓ_0-penalized maximum likelihood estimator. This estimator is known to have strong statistical consistency guarantees (van de Geer and Bühlmann, 2013), but solving it is, in general, intractable. Thus, our coordinate descent algorithm can be viewed as a scalable and efficient approach to finding approximate solutions to this estimator that are asymptotically optimal (that is, match the optimal objective value of the ℓ_0 penalized maximum-likelihood estimator as the sample size tends to infinity).

We illustrate the advantages of our method over competing approaches via extensive numerical experiments. The proposed approach is implemented in the python package *micodag*, and all numerical results and figures can be reproduced using the code in `https://github.com/AtomXT/coordinate-descent-for-bayesian-networks.git`.

2. Problem Setup

Consider an unknown DAG whose m nodes correspond to observed random variables $X \in \mathbb{R}^m$. We denote the DAG by $\mathcal{G}^\star = (V, E^\star)$ where $V = \{1, \ldots, m\}$ is the vertex set and $E^\star \subseteq V \times V$ is the directed edge set, where $(i, j) \in E^\star$ indicates $i \to j$ in graph $\mathcal{G}^\star$. We assume that the random variables X satisfy the linear structural equation model (SEM):

$$X = B^{\star\mathrm{T}} X + \epsilon, \tag{1}$$

where $B^\star \in \mathbb{R}^{m \times m}$ is the connectivity matrix with zeros on the diagonal and $B^\star_{jk} \neq 0$ if $(j, k) \in E^\star$. In other words, the sparsity pattern of $B^\star$ encodes the true DAG structure. Further, $\epsilon \sim \mathcal{N}(0, \Omega^\star)$ is a random Gaussian noise vector with zero mean and independent coordinates so that $\Omega^\star$ is a diagonal matrix. We use the Bachman-Landau symbols $\mathcal{O}$ to describe the limiting behavior of a function. Furthermore, we denote $z \asymp 1$ to express $z = \mathcal{O}(1)$ and $1/z = \mathcal{O}(1)$. Assuming, without loss of generality, that all random variables are centered, each variable X_j in this model can be expressed as the linear combination of its parents—the set of nodes with directed edges pointing to j—plus independent Gaussian noise. By the SEM (1) and the Gaussianity of ϵ, the random vector X follows the Gaussian distribution $\mathcal{P}^\star = \mathcal{N}(0, \Sigma^\star)$, with $\Sigma^\star = (I - B^\star)^{-\mathrm{T}} \Omega^\star (I - B^\star)^{-1}$. Throughout, we assume that the distribution $\mathcal{P}^\star$ is non-degenerate, or equivalently, $\Sigma^\star$ is positive definite. Our objective is to estimate the matrix $B^\star$, or as we describe next, an equivalence class when the underlying model is not identifiable.

Multiple SEMs are generally compatible with the distribution $\mathcal{P}^\star$. To formalize this, we need the following definition.

Definition 1 *(Graph $\mathcal{G}(B)$ induced by B) Let $B \in \mathbb{R}^{m\times m}$ with zeros on the diagonal. Then, $\mathcal{G}(B)$ is the directed graph on m nodes where the directed edge from i to j appears in $\mathcal{G}(B)$ if and only if $B_{ij} \neq 0$.*

To see why the model (1) is generally not identifiable, note that there are multiple tuples (B, Ω) where $\mathcal{G}(B)$ is DAG and Ω is a positive definite diagonal matrix with $\Sigma^\star = (I - B)^{-\mathrm{T}}\Omega(I - B)^{-1}$ (van de Geer and Bühlmann, 2013). As a result, the SEM given by (B, Ω) yields an equally representative model as the one given by the population parameters $(B^\star, \Omega^\star)$. When $\mathcal{G}^\star$ is *faithful* with respect to the graph $\mathcal{G}^\star$, the sparsest DAGs that are compatible with $\mathcal{P}^\star$ are precisely $\mathrm{MEC}(\mathcal{G}^\star)$, the *Markov equivalence class* of $\mathcal{G}^\star$ (van de Geer and Bühlmann, 2013). Next, we formally define the Markov equivalence class.

Definition 2 *(Markov equivalence class $\mathrm{MEC}(\mathcal{G})$ Verma and Pearl, 1990) Let $\mathcal{G} = (V, E)$ be a DAG. Then, $\mathrm{MEC}(\mathcal{G})$ consists of DAGs that have the same skeleton and same v-structures as $\mathcal{G}$. The skeleton of $\mathcal{G}$ is the undirected graph obtained from $\mathcal{G}$ by substituting directed edges with undirected ones. Furthermore, nodes i, j, and k form a v-structure if $(i, k) \in E$ and $(j, k) \in E$, and there is no edge between i and j.*

With these preliminaries and definitions in place, we next describe the proposed ℓ_0-penalized maximum likelihood estimator. Consider n independent and identically distributed observations of the random vector X generated according to (1). Let $\hat{\Sigma}$ be the sample covariance matrix obtained from these observations. Further, consider a Gaussian SEM parameterized by connectivity matrix B and noise variance Ω with $D = \Omega^{-1}$. The parameters (B, D) specify the following precision, or inverse covariance, matrix $\Theta := \Theta(B, D) := (I - B)D(I - B)^{\mathrm{T}}$. The negative log-likelihood of this SEM is proportional to $\ell_n(\Theta) = \mathrm{trace}(\Theta\hat{\Sigma}) - \log\det(\Theta)$. Naturally, we seek a model that not only has a small negative log-likelihood but is also specified by a sparse connectivity matrix containing few nonzero elements. Thus, we deploy the following ℓ_0-penalized maximum likelihood estimator with a regularization parameter $\lambda \geq 0$:

$$\min_{B\in\mathbb{R}^{m\times m}, D\in\mathbb{D}^m_{++}} \ell_n\left((I - B)\, D\, (I - B)^{\mathrm{T}}\right) + \lambda^2\|B\|_{\ell_0} \quad \text{s.t.} \quad \mathcal{G}(B) \text{ is a DAG}. \tag{2}$$

Here, $\mathbb{D}^m_{++}$ denotes the collection of positive definite $m \times m$ diagonal matrices and $\|B\|_{\ell_0}$ denotes the number of non-zeros in B. Note that the ℓ_0 penalty is generally preferred over the ℓ_1 penalty or minimax concave penalty (MCP) for penalizing the complexity of the model. In particular, ℓ_0 regularization exhibits the important property that equivalent DAGs—those in the same Markov equivalence class—have the same penalized likelihood score, while this is not the case for ℓ_1 or MCP regularization (van de Geer and Bühlmann, 2013). Indeed, this lack of score invariance with ℓ_1 regularization partially explains the unfavorable properties of some existing methods (see Section 6).

The Markov equivalence class $\mathrm{MEC}(\mathcal{G}(\hat{B}^{\mathrm{opt}}))$ of the connectivity matrix $\hat{B}^{\mathrm{opt}}$ obtained from solving (2) provides an estimate of $\mathrm{MEC}(\mathcal{G}^\star)$. van de Geer and Bühlmann (2013) prove that this estimate has desirable statistical properties; however, solving it is, in general, intractable. As stated, the objective function $\ell_n((I - B)D(I - B)^{\mathrm{T}})$ is non-convex and non-linear function of (B, D). Furthermore, the log det function in the likelihood ℓ_n is not amenable to standard mixed-integer programming optimization techniques. To circumvent the aforementioned challenges, Xu et al. (2025) derive the following equivalent optimization model via the change of variables $\Gamma \leftarrow (I - B)D^{1/2}$:

$$\min_{\Gamma \in \mathbb{R}^{m \times m}} f(\Gamma) \quad \text{s.t.} \quad \mathcal{G}\left(\Gamma - \operatorname{diag}(\Gamma)\right) \text{ is a DAG.} \tag{3}$$

Here $f(\Gamma) := \sum_{i=1}^{m} -2\log(\Gamma_{ii}) + \operatorname{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}) + \lambda^2\|\Gamma - \operatorname{diag}(\Gamma)\|_{\ell_0}$, and $\operatorname{diag}(\Gamma)$ is the diagonal matrix formed by taking the diagonal entries of Γ. The optimal solutions of (2) and (3) are directly connected: Letting $(\hat{B}^{\mathrm{opt}}, \hat{D}^{\mathrm{opt}})$ be an optimal solution of (2), then $\hat{\Gamma}^{\mathrm{opt}} = (I - \hat{B}^{\mathrm{opt}})(\hat{D}^{\mathrm{opt}})^{1/2}$ is an optimal solution of (3). Furthermore, the sparsity pattern of $\hat{\Gamma}^{\mathrm{opt}} - \operatorname{diag}(\hat{\Gamma}^{\mathrm{opt}})$ is the same as that of $\hat{B}^{\mathrm{opt}}$; in other words, the Markov equivalence class $\mathrm{MEC}(\mathcal{G}(\hat{B}^{\mathrm{opt}}))$ is the same as the Markov equivalence class $\mathrm{MEC}(\mathcal{G}(\hat{\Gamma}^{\mathrm{opt}} - \operatorname{diag}(\hat{\Gamma}^{\mathrm{opt}})))$.

Xu et al. (2025) recast the optimization problem (3) as a convex mixed-integer program and provide algorithms to solve (3) to optimality. However, solving (3) is, in general, NP-hard, and obtaining optimality certificates may take an hour for a problem with 20 nodes (Xu et al., 2025).

3. A Coordinate Descent Algorithm for DAG Learning

In this section, we develop a cyclic coordinate descent approach to find a heuristic solution to problem (3). The coordinate descent solver is fast and can be scaled to large-scale problems. As we demonstrate in Section 4, it provably converges and produces an asymptotically optimal solution to (3). Given the quality of its estimates, the proposed coordinate descent algorithm can also be used as a warm start for the mixed-integer programming framework in Xu et al. (2025) to obtain optimal solutions.

3.1 Parameter Update Without Acyclicity Constraints

Let us first ignore the acyclicity constraint in (3), and consider solving problem (3) with respect to a single variable Γ_{uv}, for $u, v = 1, \ldots, m$, with the other coordinates of Γ fixed. Specifically, we are solving

$$\min_{\Gamma_{uv} \in \mathbb{R}} g(\Gamma_{uv}) \coloneqq \sum_{i=1}^{m} -2\log(\Gamma_{ii}) + \operatorname{tr}\left(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}\right) + \lambda^2\|\Gamma - \operatorname{diag}(\Gamma)\|_{\ell_0}, \tag{4}$$

with Γ_{ij} being fixed for $i \neq u, j \neq v$.

Proposition 3 *The solution to problem* (4), *for $u, v = 1, \ldots, m$ and $v \neq u$ is given by*

$$\hat{\Gamma}_{uv} = \begin{cases} \frac{-A_{uv}}{2\hat{\Sigma}_{uu}}, & \text{if } \lambda^2 \leq \frac{A_{uv}^2}{4\hat{\Sigma}_{uu}}, \\ 0, & \text{otherwise.} \end{cases} \quad ; \quad \hat{\Gamma}_{uu} = \frac{-A_{uu} + \sqrt{A_{uu}^2 + 16\hat{\Sigma}_{uu}}}{4\hat{\Sigma}_{uu}},$$

where $A_{uu} = \sum_{j \neq u} \Gamma_{ju}\hat{\Sigma}_{ju} + \sum_{k \neq u} \Gamma_{ku}\hat{\Sigma}_{uk}$ *and* $A_{uv} = \sum_{j \neq u} \Gamma_{jv}\hat{\Sigma}_{ju} + \sum_{k \neq u} \Gamma_{kv}\hat{\Sigma}_{uk}$.

Proof The original objective function g with ℓ_0-norm is nonconvex and discontinuous. To find the optimal solution, we compare $g(0)$ with $g(\hat{\gamma}_{uv})$ for some $\hat{\gamma}_{uv} \neq 0$.

For any $u \in V$, we have

$$\operatorname{tr}\left(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}\right) = \sum_{i=1}^{m} \Gamma_{ui} \left(\Gamma_{ui}\hat{\Sigma}_{uu} + \sum_{j\neq u} \Gamma_{ji}\hat{\Sigma}_{ju} \right) + \sum_{k\neq u}\sum_{i=1}^{m} \Gamma_{ki} \left(\Gamma_{ui}\hat{\Sigma}_{uk} + \sum_{j\neq u} \Gamma_{ji}\hat{\Sigma}_{jk} \right). \quad (5)$$

We begin by analyzing off-diagonal entries Γ_{uv} (where $u \neq v$). For each nonzero Γ_{uv}, optimizing $g(\Gamma_{uv})$ is equivalent to minimizing $\operatorname{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma})$. From equation (5), the derivative of $\operatorname{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma})$ with respect to Γ_{uv} is given by

$$\frac{\partial \operatorname{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma})}{\partial \Gamma_{uv}} = 2\hat{\Sigma}_{uu}\Gamma_{uv} + \sum_{j\neq u} \Gamma_{jv}\hat{\Sigma}_{ju} + \sum_{k\neq u} \Gamma_{kv}\hat{\Sigma}_{uk} = 2\hat{\Sigma}_{uu}\Gamma_{uv} + A_{uv}.$$

Setting the derivative to zero, and defining $\hat{\gamma}_{uv} := -A_{uv}/(2\hat{\Sigma}_{uu})$, we obtain

$$\arg\min_{\Gamma_{uv}} g(\Gamma_{uv}) = \hat{\Gamma}_{uv} := \begin{cases} \hat{\gamma}_{uv}, & \text{if } g(\hat{\gamma}_{uv}) \leq g(0), \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, we observe that $\frac{\partial^2}{\partial \Gamma_{uv}^2}\operatorname{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}) = 2\hat{\Sigma}_{uu} > 0$. This positive second derivative implies that optimizing $g(\Gamma_{uv})$ for any nonzero off-diagonal entry reduces to solving a convex optimization problem.

Given that $g(\hat{\gamma}_{uv})$ represents the optimal objective value for any nonzero Γ_{uv}, comparing it with $g(0)$ allows us to determine the optimal solution. Note that $g(\hat{\gamma}_{uv}) - g(0) = \hat{\gamma}_{uv}^2\hat{\Sigma}_{uu} + \hat{\gamma}_{uv}A_{uv} + \lambda^2$. Thus, $g(\hat{\gamma}_{uv}) \leq g(0)$ is equivalent to $\lambda^2 \leq A_{uv}^2/(4\hat{\Sigma}_{uu})$.

Now we consider the update of Γ_{uv} when $u = v$. We have:

$$\frac{\partial g(\Gamma_{uu})}{\partial \Gamma_{uu}} = \frac{-2}{\Gamma_{uu}} + 2\hat{\Sigma}_{uu}\Gamma_{uu} + \sum_{j\neq u} \Gamma_{ju}\hat{\Sigma}_{ju} + \sum_{k\neq u} \Gamma_{ku}\hat{\Sigma}_{uk} = \frac{-2}{\Gamma_{uu}} + 2\hat{\Sigma}_{uu}\Gamma_{uu} + A_{uu}.$$

Setting $\partial g(\Gamma_{uu})/\partial \Gamma_{uu} = 0$, we obtain: $\hat{\Gamma}_{uu} = (-A_{uu} + (A_{uu}^2 + 16\hat{\Sigma}_{uu})^{1/2})/(4\hat{\Sigma}_{uu})$. ■

3.2 Accounting for Acyclicity and Full Algorithm Description

Algorithm 1 fully describes our procedure. The input to our algorithm is the sample covariance $\hat{\Sigma}$, regularization parameter $\lambda \in \mathbb{R}_+$, a super-structure graph E_{super} that is a superset of edges that contains the true edges, a positive integer C, and a topological ordering O that is a permutation of $\{1, 2, \ldots, m\}$. We allow the user to restrict the set of possible edges to be within a user-specified *super-structure* set of edges E_{super}. A natural choice of the superstructure is the moral graph, which can be efficiently and accurately estimated via existing algorithms such as the graphical lasso (Friedman et al., 2007), neighborhood selection (Meinshausen and Buhlmann, 2006), or ℓ_0-regularized pseudo-likelihood-based estimator (Behdin et al., 2023). This superstructure could also be the complete graph if a reliable superstructure estimate is unavailable. The input topological ordering O governs the order in which the coordinates of Γ are updated in our coordinate descent algorithm (see Remark 4 for more discussion).

We start by initializing Γ according to input Γ^{init} and permute the rows and columns of $\hat{\Sigma}$ according to O. Then, for each pair of indices u and v ranging from 1 to m, we update Γ_{uv} based on specific rules. If $u = v$ (a diagonal entry), we update it directly according to Proposition 3. Among the off-diagonal entries, we only update those within the superstructure. Specifically, if $u \neq v$, and (u, v) is in the superstructure, we check if setting Γ_{uv} to a nonzero value violates the acyclicity constraint. (We use the breadth-first search algorithm (see Ellis and Wong, 2008; Fu and Zhou, 2013) to check for acyclicity.) If it does not, we update Γ_{uv} as per Proposition 3; otherwise, we set Γ_{uv} to 0. We refer to a full sequence of coordinate updates as a full loop. The loop is repeated until convergence, when the objective values no longer improve after a complete loop. We keep track of the support of Γs encountered during the algorithm. When the occurrence count of a particular support of Γs reaches a predefined threshold, C, a spacer step (Bertsekas, 2016; Hazimeh and Mazumder, 2020) is initiated, during which we update every nonzero coordinate iteratively. Note that in the spacer step, we use $\hat{\gamma}_{uv}$, which is the optimal update without considering the sparsity penalty, that is, we use $\lambda^2 = 0$. The use of spacer steps stabilizes the behavior of updates and ensures convergence. After finishing the spacer step, we reset the counter of the support of the current solution.

Remark 4 *The effect of the input ordering O is that Algorithm 1 iteratively updates the coordinates of Γ in a particular order: updating row by row in the order given by O, and within each row, according to O. In Section 4, we show that regardless of the input ordering O, as well as the initialization Γ_{init}, Algorithm 1 converges, and is asymptotically optimal (matches the optimal objective value as the sample size tends to infinity). However, while different orderings result in similar final objective values, the associated Markov equivalence classes can vary significantly. Thus, a given ordering or initialization may result in a Markov equivalence class that is far from the population one. In Section 5, we discuss this challenge, and advocate for using an existing algorithm to find a suitable ordering, and use this ordering as input to Algorithm 1. Finally, we note that the input topological ordering O is simply an initialization to Algorithm 1. Indeed, the topological ordering corresponding to the output $\hat{\Gamma}$ can in general be different than O. Finally, we note that the default initialization $\Gamma^{\text{init}} = I$ tends to perform well in practice and is comparable to other initialization values, especially when a good ordering O is selected; see Section 6.*

4. Convergence and Optimality Guarantees

We provide convergence and optimality guarantees for our coordinate descent procedure (Algorithm 1). Specifically, we follow a similar proof strategy as Hazimeh and Mazumder (2020) to show that Algorithm 1 converges. Remarkably, we also prove the surprising result that the objective value attained by our coordinate descent algorithm provably converges to the optimal objective value of (3). Throughout, we assume the super-structure E_{super} that is supplied as input to Algorithm 1 satisfies $E^\star \subseteq E_{\text{super}}$ where $E^\star$ denotes the true edge set; see Xu et al. (2025) for a discussion on how the graphical lasso can yield super-structures that satisfy this property with high probability.

Algorithm 1 Cyclic coordinate descent algorithm with spacer steps

Input: Sample covariance $\hat{\Sigma}$, regularization parameter $\lambda \in \mathbb{R}_+$, super-structure E_{super}, positive integer C, any topological ordering O that is a permutation of $\{1, 2, \ldots, m\}$, initial matrix Γ^{init} (default: identity matrix).
Initialize: $\Gamma^0 \leftarrow \Gamma^{\text{init}}$; $t \leftarrow 1$;
$\hat{\Sigma} \leftarrow \hat{\Sigma}$ with rows and columns permuted according to O.
while objective function $f(\Gamma^t)$ continue decreasing **do**
  for $u = 1$ to m **do**
    $\Gamma^t_{uu} = \hat{\Gamma}_{uu}$, where $\hat{\Gamma}_{uu}$ is calculated from Proposition 3 using the recently updated Γ^t.
    for $v = 1$ to m such that $(u, v) \in E_{\text{super}}$ **do**
      If $\Gamma^t_{uv} \neq 0$ violates acyclicity constraints, set $\Gamma^t_{uv} = 0$.
      If $\Gamma^t_{uv} \neq 0$ would not violate acyclicity constraints, set $\Gamma^t_{uv} = \hat{\Gamma}_{uv}$.
      $t \leftarrow t + 1$.
      Count[support(Γ^t)] $\leftarrow$ Count[support(Γ^t)] $+ 1$.
      if Count[support(Γ^t)] $= Cm^2$ **then**
        $\Gamma^{t+1} \leftarrow$ SpacerStep(Γ^t). (Algorithm 2)
        Count[support(Γ^t)] $= 0$.
       $t \leftarrow t + 1$.
      end if
    end for
  end for
end while
$\hat{\Gamma} \leftarrow \Gamma^t$ with rows and columns reordered back to the original variable order.
Output: $\hat{\Gamma}$ and the Markov equivalence class MEC($\mathcal{G}(\hat{\Gamma} - \text{diag}(\hat{\Gamma}))$).

Algorithm 2 SpacerStep

Input: Γ^t
for $(u, v) \in$ support(Γ^t) **do**
  Set $\Gamma^{t+1}_{uv} \leftarrow \hat{\gamma}_{uv}$
end for
Output: Γ^{t+1}

Our analysis relies on the notion of coordinate-wise minimum (Hazimeh and Mazumder, 2020), which represent solutions Algorithm 1 obtains under different input orderings O.

Definition 5 *(Coordinate-wise (CW) minimum) A connectivity matrix* $\Gamma^{\text{CW}} \in \mathbb{R}^{m \times m}$ *of a DAG is the CW minimum of problem* (3) *if for every* $(u, v), u, v = 1, \ldots, m$, Γ^{CW}_{uv} *is a minimizer of* $g(\Gamma_{uv})$ *with other coordinates of* Γ^{CW} *held fixed.*

4.1 Convergence

Our convergence analysis requires an assumption on the sample covariance matrix:

Assumption 1 *(Positive definite sample covariance) The sample covariance matrix $\hat{\Sigma}$ is positive definite.*

Assumption 1 is satisfied almost surely if $n \geq m$ and the samples of the random vector X are generated from an absolutely continuous distribution. Under this mild assumption, our coordinate descent algorithm provably converges to a coordinate-wise minimum.

Theorem 6 *(Convergence of Algorithm 1) Let $\{\Gamma^t\}_{t=1}^{\infty}$ be the sequence of estimates generated by Algorithm 1. Suppose that Assumption 1 holds. Then, for input topological ordering O or initial point Γ^{init} to Algorithm 1:*

1. *the sequence $\{\text{support}(\Gamma^t)\}_{t=1}^{\infty}$ stabilizes after a finite number of iterations; that is, there exists a positive integer M and a support set $\hat{E} \subseteq \{(i,j) : i,j = 1,2,\ldots,m\}$ such that $\text{support}(\Gamma^t) = \hat{E}$ for all $t \geq M$.*

2. *the sequence $\{\Gamma^t\}_{t=1}^{\infty}$ converges to a coordinate-wise minimum Γ with $\text{support}(\Gamma) = \hat{E}$.*

The proof of Theorem 6 is provided in Appendix A and adapts the analysis in Hazimeh and Mazumder (2020), which was originally developed for coordinate descent in variable selection, to our problem setting.

4.2 Optimality guarantees

Let $d_{\max} := \max_i |\{j : (j,i) \in E_{\text{super}}\}|$, the maximum degree of a node in E_{super}. Our analysis for optimality guarantees requires an assumption on the population model. For the set $E \subseteq \{(i,j) : i,j = 1,2\ldots,m\}$, consider the optimization problem

$$\Gamma_E^{\star} = \operatorname*{arg\,min}_{\Gamma \in \mathbb{R}^{m\times m}} \sum_{i=1}^{m} -2\log(\Gamma_{ii}) + \operatorname{tr}\left(\Gamma\Gamma^{\mathrm{T}}\Sigma^{\star}\right) \qquad \text{s.t.} \qquad \text{support}(\Gamma) \subseteq E. \tag{6}$$

Assumption 2 *There exists constants $\bar{\kappa}, \underline{\kappa} > 0$ such that $\sigma_{min}(\Gamma_E^{\star}) \geq \underline{\kappa}$ and $\sigma_{max}(\Gamma_E^{\star}) \leq \bar{\kappa}$ for every $E \subseteq E_{\text{super}}$ where the graph (V,E) is a DAG, where $\sigma_{min}(\cdot)$ and $\sigma_{min}(\cdot)$ are the smallest and largest eigenvalues respectively.*

Theorem 7 *Let $\hat{\Gamma}$ be the solution of Algorithm 1 with any input topological ordering O or initial point Γ^{init}, and let $\hat{\Gamma}^{\text{opt}}$ be an optimal solution of (3). Suppose Assumption 2 holds and let the regularization parameter be chosen so that $\lambda^2 = \mathcal{O}(\log m/n)$ where m and n denote the number of nodes and number of samples, respectively. Then,*

1. *$f(\hat{\Gamma}) - f(\hat{\Gamma}^{\text{opt}}) \to_P 0$ as $n \to \infty$,*

2. *if $n/\log(n) \geq \mathcal{O}(m^2 \log m)$, with probability greater than $1 - 1/\mathcal{O}(n)$, we have $0 \leq f(\hat{\Gamma}) - f(\hat{\Gamma}^{\text{opt}}) \leq \mathcal{O}(\sqrt{d_{max}^2 m^4 \log m/n})$.*

In other words, regardless of the input ordering O, the objective value of the coordinate descent solution converges in probability to the optimal objective value as $n \to \infty$. Further, assuming the sample size n is sufficiently large, with high probability, the difference in objective value is bounded by $\mathcal{O}(\sqrt{d_{max}^2 m^4 \log m/n})$.

We prove Theorem 7 by showing the results for any coordinate-wise minimum of problem (3) (see Definition 5). Our proof relies on the following lemmas. Throughout, we let $\hat{E}$ be the support of $\hat{\Gamma}$, that is, $\hat{E} = \{(i,j), \hat{\Gamma}_{ij} \neq 0\}$.

Lemma 8 *Let $\hat{\Gamma}, \hat{\Gamma}^{\text{opt}}$ be the solution of Algorithm 1 and optimal solution of* (3)*, respectively. Then, i) for any $u, v = 1, 2, \ldots, m$, $A_{uv} + 2\hat{\Gamma}_{uv}\hat{\Sigma}_{uu} = 2(\hat{\Sigma}\Gamma)_{uv}$ where A_{uv} is defined in Proposition 3. ii) if $\hat{\Gamma}_{uv} \neq 0$, then $(\hat{\Sigma}\hat{\Gamma})_{uv} = 0$, and iii) the matrix $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}$ has ones on the diagonal. Moreover, these properties also hold for the optimal solution $\hat{\Gamma}^{\text{opt}}$.*

Proof [Proof of Lemma 8] For $u, v = 1, \ldots, m$, by the definition of A_{uv}, $A_{uv} + 2\Gamma_{uv}\hat{\Sigma}_{uu} = 2(\hat{\Sigma}\Gamma)_{uv}$, proving item i. Since any solution from Algorithm 1, $\hat{\Gamma}$ satisfies Proposition 3, for any $(u,v) \in \hat{E}$, $(4\hat{\Sigma}_{uu}\hat{\Gamma}_{uu} + A_{uu})^2 = A_{uu}^2 + 16\hat{\Sigma}_{uu}$ and $A_{uv} = -2\hat{\Gamma}_{uv}\hat{\Sigma}_{uu}$. Combining the previous relations, we conclude that $(\hat{\Sigma}\hat{\Gamma})_{uv} = 0$. Therefore, for any $(u,v) \in \hat{E}$, we have $\hat{\Gamma}_{uv} \neq 0$ and $(\hat{\Sigma}\hat{\Gamma})_{uv} = 0$, resulting in $\hat{\Gamma}_{uv}(\hat{\Sigma}\hat{\Gamma})_{uv} = 0$. This proves item ii. Plugging A_{uu} into the previous relations, we arrive at $\hat{\Gamma}_{uu}(\hat{\Sigma}\hat{\Gamma})_{uu} = 1$. Thus, $(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{ii} = \sum_{j=1}^{m}\hat{\Gamma}_{ij}(\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{ji} = \hat{\Gamma}_{ii}(\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{ii} = 1$, proving item iii. Since $\hat{\Gamma}^{\text{opt}}$ is an optimal solution, it must satisfy the conditions stated in Proposition 3; otherwise, we could update $\hat{\Gamma}^{\text{opt}}$ to achieve a better objective value. Therefore, the same arguments apply to $\hat{\Gamma}^{\text{opt}}$. ■

Lemma 9 *Let $E \subseteq \{(i,j) : i, j = 1, 2, \ldots, m\}$ be any set where the graph indexed by tuple (V, E) is a DAG. Consider the estimator:*

$$\hat{\Gamma}_E = \operatorname*{arg\,min}_{\Gamma \in \mathbb{R}^{m\times m}} \sum_{i=1}^{m} -2\log(\Gamma_{ii}) + \operatorname{tr}\left(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}\right) \quad s.t. \quad \operatorname{support}(\Gamma) \subseteq E. \tag{7}$$

Suppose that $4\sqrt{m}\bar{\kappa}\|\hat{\Sigma} - \Sigma^\star\|_2 \leq \min\{\underline{\kappa}/(8\sqrt{m}\bar{\kappa}^2), \frac{1}{2\bar{\kappa}}\}$ and that $\hat{\Sigma}$ is positive definite. Then, $\|\hat{\Gamma}_E - \Gamma_E^\star\|_F \leq 4\sqrt{m}\bar{\kappa}\|\hat{\Sigma} - \Sigma^\star\|_2$.

Proof [Proof of Lemma 9] The proof follows from standard convex analysis and Brouwer's fixed point theorem; we provide the details below. Since Γ follows a DAG structure, the objective of (7) can be written as: $-2\log\det(\Gamma) + \|\Gamma\hat{\Sigma}^{1/2}\|_F^2$. The KKT conditions state that there exists Q with $\operatorname{support}(Q) \cap E = \emptyset$ such that the optimal solution $\hat{\Gamma}_E$ of (7) satisfies $-2\hat{\Gamma}_E^{-1} + Q + 2\hat{\Gamma}_E\hat{\Sigma} = 0$ and $\operatorname{support}(\hat{\Gamma}_E) \subseteq E$. Let $\Delta = \hat{\Gamma}_E - \Gamma_E^\star$. By Taylor series expansion, $\hat{\Gamma}_E^{-1} = (\Gamma_E^\star + \Delta)^{-1} = {\Gamma_E^\star}^{-1} - {\Gamma_E^\star}^{-T}\Delta{\Gamma_E^\star}^{-1} + \mathcal{R}(\Delta)$, where $\mathcal{R}(\Delta) = 2{\Gamma_E^\star}^{-1}\sum_{k=2}^{\infty}(-\Delta\Gamma_E^\star)^k$. For any matrix $M \in \mathbb{R}^{m\times m}$, define the operator $\mathbb{I}^\star$ with $\mathbb{I}^\star(M) := 2{\Gamma_E^\star}^{-T}M{\Gamma_E^\star}^{-1} + 2M\Sigma^\star$. Let $\mathcal{K}$ be the subspace $\mathcal{K} = \{M \in \mathbb{R}^{m\times m} : \operatorname{support}(M) \subseteq E\}$ and let $P_{\mathcal{K}}$ be the projection operator onto subspace $\mathcal{K}$ that zeros out entries of the input matrix outside of the support set E. From the optimality condition of (6), we have $\mathcal{P}_{\mathcal{K}}[2{\Gamma_E^\star}^{-1} - 2\Gamma_E^\star\Sigma^\star] = 0$. Then, the optimality condition of (7) can be rewritten as:

$$\mathcal{P}_{\mathcal{K}}\left[\mathbb{I}^\star(\Delta) + 2\Delta(\hat{\Sigma} - \Sigma^\star) - 2\mathcal{R}(\Delta) + H_n\right] = 0, \tag{8}$$

where $H_n = 2\Gamma_E^*(\hat{\Sigma} - \Sigma^*)$. Since $\hat{\Gamma}_E \in \mathcal{K}$ and $\Gamma_E^\star \in \mathcal{K}$, we have that $\Delta \in \mathcal{K}$. We use Brouwer's theorem to obtain a bound on $\|\Delta\|_F$. We define an operator J as $\mathcal{K} \to \mathcal{K}$:

$$J(\delta) = \delta - (\mathcal{P}_{\mathcal{K}}\mathbb{I}^\star\mathcal{P}_{\mathcal{K}})^{-1}\left(\mathcal{P}_{\mathcal{K}}\left[\mathbb{I}^\star\mathcal{P}_{\mathcal{K}}(\delta) - 2\mathcal{R}(\delta) + H_n + 2\delta(\hat{\Sigma} - \Sigma^\star)\right]\right).$$

Here, the operator $\mathcal{P}_{\mathcal{K}}\mathbb{I}^\star\mathcal{P}_{\mathcal{K}}$ is invertible since $\sigma_{\min}(\mathbb{I}^\star) = \sigma_{\min}({\Gamma_E^\star}^{-1})^2 \geq \frac{1}{\bar{\kappa}^2}$. Notice that any fixed point δ of J satisfies the optimality condition (8). Furthermore, since the objective of (7) is strictly convex, we have that the fixed point must be unique. In other words, the unique fixed point of J is given by Δ. Now consider the following compact set: $\mathcal{B}_r = \{\delta \in \mathbb{R}^{m\times m} : \text{support}(\delta) \subseteq E, \|\delta\|_F \leq r\}$ for $r = 4\sqrt{m}\bar{\kappa}\|\hat{\Sigma} - \Sigma^\star\|_2$. By the assumption, $r \leq \min\{\underline{\kappa}/(8\sqrt{m}\bar{\kappa}^2), \frac{1}{2\bar{\kappa}}\}$. Then, for every $\delta \in \mathcal{B}_r$, we have $\|\delta\Gamma_E^\star\|_F \leq \bar{\kappa}r \leq 1/2$, and $\|\mathcal{R}(\delta)\|_F \leq 2\sqrt{m}\|\Gamma_E^\star\|_2^2/\sigma_{\min}(\Gamma_E^\star)\|\delta\|_2^2\frac{1}{1-\|\delta\Gamma_E^\star\|_2} \leq 2\sqrt{m}\bar{\kappa}^2r^2/\underline{\kappa}\frac{1}{1-r\bar{\kappa}} \leq 4\sqrt{m}\bar{\kappa}^2r^2/\underline{\kappa}$. Since $\|H_n\|_F \leq 2\sqrt{m}\|\Gamma_E^\star\|_2\|\hat{\Sigma} - \Sigma^\star\|_2$ and $\|J(\delta)\|_F \leq \bar{\kappa}^2[\|H_n\|_F + 2\|\mathcal{R}(\delta)\|_F + 2\|\delta(\hat{\Sigma} - \Sigma^\star)\|_F]$, we conclude that $\|J(\delta)\|_F \leq \frac{8\sqrt{m}\bar{\kappa}^2r^2}{\underline{\kappa}} + \frac{\bar{\kappa}^2r}{2} + \frac{r^2}{2\sqrt{m}\bar{\kappa}} \leq r$. In other words, we have shown that J maps $\mathcal{B}_r$ onto itself. Appealing to Brouwer's fixed point theorem, we conclude that the fixed point must also lie inside $\mathcal{B}_r$. Thus, we conclude that $\|\Delta\|_F \leq r$.

■

Lemma 10 *With probability greater than* $1-1/\mathcal{O}(n)$, *we have* $\|\hat{\Sigma}-\Sigma^\star\|_2 \leq \mathcal{O}(\sqrt{m\log(n)/n})$, $\|\hat{\Sigma}\|_\infty \leq 2\bar{\kappa}^2$, $\sigma_{min}(\hat{\Sigma}) \geq \underline{\kappa}^2/2$, $\|\hat{\Gamma}\|_\infty \leq 2\bar{\kappa}$ *and* $\sigma_{min}(\hat{\Gamma}) \geq \underline{\kappa}/2$.

Proof [Proof of Lemma 10] From standard Gaussian concentration results, when $n/\log(n) \geq \mathcal{O}(m)$, with probability greater than $1-\mathcal{O}(1/n)$, we have that $\|\hat{\Sigma}-\Sigma^\star\|_2 \leq \mathcal{O}(\sqrt{m\log(n)/n})$. By Assumption 2, with probability greater than $1 - \mathcal{O}(1/n)$, $\hat{\Sigma}$ is positive definite, with $\|\hat{\Sigma}\|_\infty \leq 2\bar{\kappa}^2$ and $\sigma_{\min}(\hat{\Sigma}) \geq \underline{\kappa}^2 - \mathcal{O}(\sqrt{m\log(n)/n}) \geq \underline{\kappa}^2/2$. Furthermore, appealing to Lemma 9, $\|\hat{\Gamma} - \Gamma_{\hat{E}}^\star\|_F \leq \mathcal{O}(\sqrt{m^2\log(n)/n})$. Thus, $\|\hat{\Gamma}\|_\infty \leq \|\Gamma_{\hat{E}}^\star\|_2 + \bar{\kappa} \leq 2\bar{\kappa}$ and $\sigma_{\min}(\hat{\Gamma}) \geq \underline{\kappa} - \mathcal{O}(\sqrt{m\log(n)/n}) \geq \underline{\kappa}/2$. ■

Proof [Proof of Theorem 7] **Part 1)**. First, observe that by definition, $\hat{\Gamma}^{\text{opt}}$ is the optimal solution of the problem

$$\min_{\Gamma\in\mathbb{R}^{m\times m}} f(\Gamma) \quad \text{s.t.} \quad \mathcal{G}\left(\Gamma - \text{diag}\left(\Gamma\right)\right) \text{ is a DAG},$$

where $f(\Gamma) := \sum_{i=1}^m -2\log(\Gamma_{ii}) + \text{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}) + \lambda^2\|\Gamma - \text{diag}(\Gamma)\|_{\ell_0}$ is the penalized negative log-likelihood function. The negative log-likelihood part of f is convex with the optimal value being $\min_\Theta\{-\log\det(\Theta) + \text{tr}(\Theta\hat{\Sigma})\}$. Clearly, $\sum_{i=1}^m -2\log(\hat{\Gamma}_{ii}^{\text{opt}}) + \text{tr}(\hat{\Gamma}^{\text{opt}}(\hat{\Gamma}^{\text{opt}})^{\mathrm{T}}\hat{\Sigma}) \geq \min_\Theta\{-\log\det(\Theta) + \text{tr}(\Theta\hat{\Sigma})\}$. Therefore,

$$f(\hat{\Gamma}^{\text{opt}}) \geq \min_\Theta\{-\log\det(\Theta) + \text{tr}(\Theta\hat{\Sigma})\} = \log\det(\hat{\Sigma}) + m. \tag{9}$$

Now,

$$0 \leq f(\hat{\Gamma}) - f(\hat{\Gamma}^{\text{opt}}) \leq f(\hat{\Gamma}) - \log\det(\hat{\Sigma}) - m = -\log\det(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}) + \lambda^2\|\hat{\Gamma} - \text{diag}(\hat{\Gamma})\|_0,$$

where the second inequality follows from (9), and the equality follows from appealing to item iii. of Lemma 8 to conclude that $f(\hat{\Gamma}) = -\log\det(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}) + m + \lambda^2\|\hat{\Gamma} - \text{diag}(\hat{\Gamma})\|_0$, because $\text{tr}(\Theta\hat{\Sigma})\} = m$.

Our strategy is to show that as $n \to \infty$, $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}$ converges to a matrix with ones on the diagonal and whose off-diagonal entries induce a DAG. Thus, $\log\det(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}) \to \log\prod_{i=1}^m 1 =$

0 as $n \to \infty$. Since $\lambda^2 \to 0$ as $n \to \infty$ and $\|\hat{\Gamma} - \text{diag}(\hat{\Gamma})\|_0 \leq m^2$, we can then conclude the desired result. For any $u, v = 1, 2, \ldots, m$:

$$(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv} = \sum_{i=1}^{m} \hat{\Gamma}_{ui}(\hat{\Sigma}\hat{\Gamma})_{vi} = \hat{\Gamma}_{uu}(\hat{\Sigma}\hat{\Gamma})_{vu} + \hat{\Gamma}_{uv}(\hat{\Sigma}\hat{\Gamma})_{vv} + \sum_{i \in F_{uv}} \hat{\Gamma}_{ui}(\hat{\Sigma}\hat{\Gamma})_{vi}, \tag{10}$$

where $F_{uv} := \{i \mid i \neq u, i \neq v, (u,i) \in \hat{E}, (v,i) \notin \hat{E}\}$. Here, the second equality is due to item ii. of Lemma 8; note that if $\hat{\Gamma}_{ui}(\hat{\Sigma}\hat{\Gamma})_{vi} \neq 0$, then $i \in F_{uv}$ as otherwise either $\hat{\Gamma}_{ui} = 0$ or $(\hat{\Sigma}\hat{\Gamma})_{vi} = 0$. We consider the two possible settings for $(u,v), u \neq v$: Setting I) $(u,v) \in \hat{E}$ which implies that $(v,u) \notin \hat{E}$ as $\hat{\Gamma}$ specifies a DAG, and Setting II) $(u,v),(v,u) \notin \hat{E}$. (Note that $(u,v),(v,u) \in \hat{E}$ is not possible since $\hat{\Gamma}$ specifies a DAG.)

Setting I: Since $(u,v) \in \hat{E}$ and $(v,u) \notin \hat{E}$, we have

$$(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{vu} = \sum_{i \in F_{vu}} \hat{\Gamma}_{vi}(\hat{\Sigma}\hat{\Gamma})_{ui} = \sum_{i \in F_{vu}} \hat{\Gamma}_{vi}\left(\frac{1}{2}A_{ui} + \hat{\Gamma}_{ui}\hat{\Sigma}_{uu}\right) = \sum_{i \in F_{vu}} \frac{1}{2}\hat{\Gamma}_{vi}A_{ui}.$$

Here, the first equality follows from appealing to (10), and noting that $\hat{\Gamma}_{vu} = 0$ and that $(\hat{\Sigma}\hat{\Gamma})_{uv} = 0$ according to item ii. of Lemma 8; the second equality follows from item i. of Lemma 8; the final equality follows from noting that $\hat{\Gamma}_{ui} = 0$ for $i \in F_{vu}$.

For each $i \in F_{vu}$, Figure 1 (left) represents the relationships between the nodes u, v, i. Here, the directed edge from u to v from the constraint $(u,v) \in \hat{E}$ is represented by a dashed line, the directed edge from v to i from the constraint $i \in F_{vu}$ is represented by a solid line, and the directed edge that is disallowed due to the constraint $i \in F_{vu}$ is represented via a crossed-out solid line.

Since there is a directed path from u to i, to avoid a cycle, a directed path from i to u cannot exist. Thus, adding the edge from u to i to $\hat{E}$ does not violate acyclicity and the fact that it is missing is due to $\lambda^2 > A_{ui}^2/(4\hat{\Sigma}_{uu})$ according to Proposition 3. Then, appealing to Lemma 10, we conclude that with probability greater than $1 - \mathcal{O}(1/n)$: $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{vu}| \leq \sum_{i \in F_{vu}} \frac{1}{2}|\hat{\Gamma}_{vi}| 2\lambda(\hat{\Sigma}_{uu})^{1/2} \leq 4\lambda\bar{\kappa}^2 d_{\max}$. In other words, in this setting, $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{vu}| \to 0$ as $n \to \infty$.

Setting II: Since $(u,v),(v,u) \notin \hat{E}$, we have

$$(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv} = \hat{\Gamma}_{uu}\left(\frac{1}{2}A_{vu} + \hat{\Gamma}_{vu}\hat{\Sigma}_{vv}\right) + \sum_{i \in F_{uv}} \hat{\Gamma}_{ui}\left(\frac{1}{2}A_{vi} + \hat{\Gamma}_{vi}\hat{\Sigma}_{vv}\right) = \sum_{\substack{i \in F_{uv} \\ \cup\{u\}}} \frac{\hat{\Gamma}_{ui}A_{vi}}{2}. \tag{11}$$

Here, the first equality follows from plugging zero for $\hat{\Gamma}_{uv}$ in (10) and appealing to item i. of Lemma 8; the second equality follows from plugging in zero for $\hat{\Gamma}_{vi}$ and $\hat{\Gamma}_{vu}$. Since $\hat{\Gamma}$ specifies a DAG, there cannot simultaneously be a directed path from u to v and from v to u. Thus, either directed edges (u,v) or (v,u) can be added without creating a cycle. We consider the three remaining sub-cases below:

Setting II.1. Adding (u,v) to $\hat{E}$ violates acyclicity but adding (v,u) does not.

For each $i \in F_{uv}$, Figure 1 (middle) represents the relations between nodes u, v, and i. Here, due to the condition of Setting II, nodes u and v are not connected by an edge, which is displayed by a solid, crossed-out, undirected edge. Furthermore, the directed edge from u to i from the constraint $i \in F_{uv}$ is represented via a solid directed edge, the directed edge v to i that is disallowed due to the constraint $i \in F_{uv}$ is represented via a crossed-out solid line. Finally, the directed edge u to v that is disallowed due to acyclicity is represented via a crossed-out dashed line.

Since adding the directed edge (u, v) to $\hat{E}$ creates a cycle, then we have the following implications: i. adding (v, u) to $\hat{E}$ does not violate acyclicity (as both edges $u \to v$ and $v \to u$ cannot simultaneously create cycles) and ii. there must be a directed path from v to u. Implication i. allows us to conclude that $\hat{\Gamma}_{vu}$ must be equal to zero due to the condition $4\hat{\Sigma}_{vv}\lambda^2 > A_{vu}^2$ from Proposition 3. Combining implication ii. and the fact that there is a directed edge from u to i in $\hat{E}$ allows us to conclude that there cannot be a directed path from i to v as we would be creating a direct path from u to itself. Thus, the fact that the directed edge (v, i) is not in $\hat{E}$, or equivalently that $\hat{\Gamma}_{vi} = 0$, is due to $4\hat{\Sigma}_{vv}\lambda^2 > A_{vi}^2$ according to Proposition 3. From (11) and Lemma 10, we conclude with probability greater than $1 - \mathcal{O}(1/n)$, $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv}| \leq \sum_{i \in F_{vu} \cup \{u\}} \frac{1}{2}|\hat{\Gamma}_{vi}| 2\lambda(\hat{\Sigma}_{uu})^{1/2} \leq 4\bar{\kappa}\lambda(1 + d_{\max})$. In other words, in this setting, $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv}| \to 0$ as $n \to \infty$.

Setting II.2. Adding (u, v) or (v, u) to $\hat{E}$ would not violate acyclicity.

For each $i \in F_{uv}$, Figure 1 (right) represents the relations between the nodes u, v, and i. Here, due to the condition of Setting II, nodes u and v are not connected by an edge, so this is displayed by a solid crossed-out undirected edge. Furthermore, the directed edge from u to i from the constraint $i \in F_{uv}$ is represented via a solid directed edge, the directed edge v to i that is disallowed due to the constraint $i \in F_{uv}$ is represented via a crossed-out solid line.

In this setting, recall that the directed edges u to v and v to u are not present in the estimate $\hat{E}$. Since neither of these two edges violates acyclicity according to the condition of this setting, we conclude that $4\hat{\Sigma}_{vv}\lambda^2 > A_{vu}^2$. There cannot be a path from i to v because then there would exist a path from u to v, which contradicts the scenario that an edge from v to u does not create a cycle. As a result, an edge from v to i does not create a cycle and $\hat{\Gamma}_{vi} = 0$ is due to $4\hat{\Sigma}_{vv}\lambda^2 > A_{vi}^2$ according to Proposition 3. Thus, from (11) and Lemma 10, we conclude that, with probability greater than $1 - \mathcal{O}(1/n)$, $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv}| \leq 4\bar{\kappa}\lambda(1 + d_{\max})$. In other words, $|(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv}| \to 0$ as $n \to \infty$.

Setting II.3. Adding (v, u) violates acyclicity but adding (u, v) does not.

In this case, even if $(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma})_{uv}$ does not converge to zero, we have by the setting assumption that adding (u, v) to $\hat{E}$ does not violate DAG constraint. Since $\hat{E}$ specifies a DAG, the off-diagonal nonzero entries of the matrix $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}$ specifies a DAG as well.

Putting Settings I—II together, we have shown that as $n \to \infty$, the nonzero entries in the off-diagonal of $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}$ specify a DAG. Furthermore, according to item iii. of Lemma 8, the diagonal entries of this matrix are equal to one. Since the off-diagonal pattern corresponds to a DAG, we can permute the rows and columns of this matrix to transform it into an upper triangular form without changing its determinant. Given that all diagonal entries are one, the determinant of the matrix is one, and hence $-\log\det(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}) \to 0$ as $n \to \infty$, and, consequently, $f(\hat{\Gamma}) - f(\hat{\Gamma}^{\mathrm{opt}}) \to 0$.

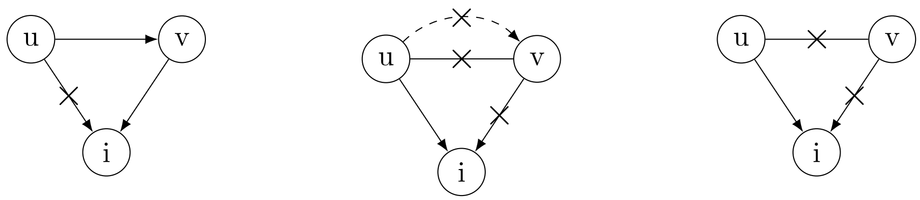


Figure 1: Left: scenario for Setting I, middle: scenario for setting II.1, and right: scenario for setting II.2; solid directed edges represent directed edges that are assumed to be in the estimate $\hat{E}$, crossed out solid directed edges represent directed edges that are assumed to be excluded in the estimate $\hat{E}$, crossed out solid undirected edges indicate that the corresponding nodes are not connected in $\hat{E}$, and crossed out dashed directed edge indicates that the edge is not present in $\hat{E}$ as adding it would create a cycle.

Part 2) When the sample size n is finite, the entries of the nonzero value of $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}$ that would be zero in the ideal DAG structure are instead small but nonzero, due to estimation error in the sample covariance matrix. We use the matrix Δ to capture these small perturbations. Then, using the proof of Theorem 7 part i), we can immediately conclude that the matrix $\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}_n$ can be decomposed as the sum $N + \Delta$. Here, the off-diagonal entries of N specify a DAG, with ones on the diagonal and under the assumption on n, with probability greater than $1 - \mathcal{O}(1/n)$, $\|\Delta\|_\infty \leq 4\bar{\kappa}(1 + d_{\max})\lambda$ with zeros on the diagonal of Δ (here, $\|\cdot\|_\infty$ denotes the maximum entry of the input matrix in absolute value). Consequently, $\|\Delta\|_2 \leq 4m\bar{\kappa}(1 + d_{\max})\lambda$. Furthermore, by Lemma 10, we get $\sigma_{\min}(\hat{\Gamma}\hat{\Gamma}^{\mathrm{T}}\hat{\Sigma}_n) \geq \sigma_{\min}(\hat{\Gamma})^2\sigma_{\min}(\hat{\Sigma}) \geq \underline{\kappa}^4/4$. The reverse triangle inequality yields $\sigma_{\min}(N) \geq \underline{\kappa}^4/4 - 4m\bar{\kappa}^2(1+d_{\max})\lambda$. Consider any matrix $\bar{N}$ with $|\bar{N}_{ij} - N_{ij}| \leq |\Delta_{ij}|$. Using the reverse triangle inequality again, we get $\sigma_{\min}(\bar{N}) \geq \underline{\kappa}^4 - 8m\bar{\kappa}^2(1 + d_{\max})\lambda$ with probability greater than $1 - \mathcal{O}(1/n)$. By the assumption that the sample size satisfies $n/\log(n) \geq \mathcal{O}(m^2 \log m)$, $\bar{N}$ is invertible, and so we can use the first-order Taylor series expansion to obtain $-\log\det(N + \Delta) = -\log\det(N) - \mathrm{tr}(\bar{N}^{-1}\Delta)$. Since $\log\det(N) = 0$, we obtain the bound $-\log\det(N+\Delta) \leq -\mathrm{tr}(\bar{N}^{-1}\Delta) \leq \|\bar{N}^{-1}\|_2\|\Delta\|_\star$ with $\|\cdot\|_\star$ denoting the nuclear norm. Thus, $-\log\det(N + \Delta) \leq \|\bar{N}^{-1}\|_2\|\Delta\|_\star \leq \frac{m}{\sigma_{\min}(\bar{N})}\|\Delta\|_2 \leq \frac{4m^2\bar{\kappa}^2(1+d_{\max})\lambda}{\underline{\kappa}^4/4-8m\bar{\kappa}^2(1+d_{\max})\lambda}$. As $\lambda^2 = \mathcal{O}(\log m/n)$, by the assumption on the sample size, $f(\hat{\Gamma}) - f(\hat{\Gamma}^{\mathrm{opt}}) \leq \mathcal{O}(\sqrt{d_{\max}^2 m^4 \log m/n})$. ■

4.3 On the Optimization Landscape of Problem (3)

The optimization problem (3) is highly non-convex, due to the ℓ_0 penalty as well as the DAG constraint. As a result, its landscape may contain many local minimizers. The following corollary to Theorem 7 analyzes these local minimizers.

Corollary 11 *Consider the setup in Theorem 7. Let $\tilde{\Gamma}$ be any local minimizer to* (3)*. Then, $f(\tilde{\Gamma}) - f(\hat{\Gamma}^{\mathrm{opt}}) \to_P 0$ as $n \to \infty$. Furthermore, if $n/\log(n) \geq \mathcal{O}(m^2 \log m)$, with probability greater than $1 - 1/\mathcal{O}(n)$, we have $0 \leq f(\tilde{\Gamma}) - f(\hat{\Gamma}^{\mathrm{opt}}) \leq \mathcal{O}(\sqrt{d_{max}^2 m^4 \log m/n})$.*

Algorithm 3 Topological Ordering (Chen et al., 2019)

Input: $\hat{\Sigma} \in \mathbb{R}^{p \times p}$ (estimated) covariance of X
$C \leftarrow \emptyset$
for $t = 1, \ldots, m$ **do**
  $j^* \leftarrow \arg\min_{j \in V \setminus C} \hat{\Sigma}_{j,j} - \hat{\Sigma}_{j,C} \hat{\Sigma}_{C,C}^{-1} \hat{\Sigma}_{C,j} = \frac{1}{\{(\hat{\Sigma}_{C \cup \{j\}, C \cup \{j\}})^{-1}\}_{j,j}}$
  Append j^* to C
end for
Output: C

This corollary follows from Theorem 7 and the fact that coordinate-wise optimality is weaker than local optimality; therefore, any local minima of this problem satisfies the same result. This result implies that when the sample size is large, all local minima achieve objective values close to a global minimum; moreover, in the infinite sample regime, all local minimizers are global minimizers. Given the non-convexity of our objective function, this result offers significant insight into the optimization landscape of the problem.

5. Selecting a Suitable Ordering for Coordinate Descent Updates

A significant challenge for nonconvex optimization problems is that selecting a good starting point and/or an effective coordinate update order is crucial for achieving good solutions. For example, in our setting, the performance of Algorithm 1 can be sensitive to the input ordering O for coordinate descent updates. To address this challenge, we propose estimating a topological ordering of the underlying DAG $\mathcal{G}^\star$ and using it as the input ordering O. A valid topological ordering of $\mathcal{G}^\star$ is a permutation $(a_1, a_2, \ldots, a_m)$ of $(1, 2, \ldots, m)$ such that for every edge $i \to j$ in $\mathcal{G}^\star$, the index a_i precedes a_j in the ordering.

To estimate a good topological ordering, we use an algorithm proposed by Chen et al. (2019). This procedure, outlined in Algorithm 3, iteratively selects the node with the smallest conditional variance given the variables selected in the previous step. When the variances of the noise (i.e., variance of the coordinates of ϵ in (1)) are identical, they show that with high probability, their algorithm correctly recovers a valid topological ordering of $\mathcal{G}^\star$.

We improve upon the theoretical results of Chen et al. (2019) and show that we can still use Algorithm 3 to recover a valid topological ordering, even when the noise variances are not identical. Our result assumes that the noise variances are not too far apart—that is, the noise distribution is nearly homoscedastic.

Assumption 3 (Near homoscedastic noise) *Define $\zeta := \min_{(j,k) \in E} (B^\star_{jk})^2$. Let $\Omega^\star_{\max} := \max_j \Omega^\star_{j,j}$ and $\Omega^\star_{\min} := \min_j \Omega^\star_{j,j}$ denote the maximum and minimum error variance, respectively. We assume $\Omega^\star_{\max} < \Omega^\star_{\min}(1+\zeta)$; that is, the maximum and minimum error variances are not too far apart.*

Theorem 12 *Let X be variables defined in (1). Let $\lambda_{\min} > 0$ be the smallest eigenvalue of the population covariance matrix $\Sigma^\star$. Suppose Assumption 3 holds. If*

$$n > 128m^2 \log\left(\frac{2m^2+2m}{\epsilon}\right)\left(1+4\frac{\Omega^\star_{\max}}{\Omega^\star_{\min}}\right)^2\left(\max_{j\in V}\Sigma^\star_{j,j}\right)^2 \times\left(\frac{1}{\lambda_{\min}}+\frac{2\Omega^\star_{\max}\Omega^\star_{\min}(1+\zeta)}{(\Omega^\star_{\min}(1+\zeta)-\Omega^\star_{\max})\lambda^2_{\min}}\right)^2,$$

then Algorithm 3 recovers a valid topological ordering of $\mathcal{G}^\star$ *with probability at least* $1-\epsilon$.

The proof of Theorem 12, presented in Appendix B, adapts the argument from Chen et al. (2019). Theorem 12 shows that, under Assumption 3 and with a sufficiently large sample size, Algorithm 3 recovers the correct topological ordering with high probability. Unlike the result in Chen et al. (2019), which assumes equal noise variances (homoscedasticity), our analysis allows for heteroscedastic noise, provided the variances are not too dissimilar, as formalized in Assumption 3. This generalization is important in practice, where the assumption of exactly equal noise variances can be unrealistic.

Remark 13 (Statistical consistency.) *Suppose a valid topological ordering of the* m *variables is given. Then, under such ordering, the true weighted adjacency matrix* $\Gamma^\star$ *of the DAG is strictly upper-triangular, and the population covariance satisfies* $\Sigma^\star = \Gamma^{\star-\mathrm{T}}\Gamma^{\star-1}$. *This implies that* $\Gamma^\star = L^{\star-\mathrm{T}}$, *where* $L^\star$ *is the Cholesky factor of* $\Sigma^\star$, *i.e.,* $\Sigma^\star = L^\star L^{\star\mathrm{T}}$. *Thus, the problem of finding the DAG reduces to the known sparse Cholesky factor estimation problem. In fact, consistency (in Frobenius or operator norm) of the sparse Cholesky factor follows directly from existing results on regularized Cholesky-factor estimation; see, for example, Theorem 10 of Lam and Fan (2009).*

6. Synthetic and Real Experiments

In this section, we illustrate the utility of our method in Algorithm 1 on synthetic and real data and compare its performance with competing methods. We dub our method CD-ℓ_0 as it is a coordinate descent method using ℓ_0 penalized loss function. The competing methods we compare against include greedy equivalence search (GES) (Chickering, 2002), NOTEARS (Zheng et al., 2018), and the mixed-integer convex program (MICODAG) (Xu et al., 2025). We also compare our method with other coordinate descent algorithms (CCDr-MCP) (Aragam and Zhou, 2015; Aragam et al., 2019; Fu and Zhou, 2013), which use a minimax concave penalty instead of ℓ_0 norm and are implemented as an `R` package *sparsebn*. To obtain an input ordering O for CD-ℓ_0, we use the algorithm of Chen et al. (2019) (presented in Algorithm 3). Consequently, we include in our comparisons the method of Chen et al. (2019), which performs statistical testing to infer a network structure using this estimated ordering. All experiments were conducted on an Intel Xeon Gold 6230R CPU with eight cores and 8 GB of memory, using `Gurobi` 10.0.1 as the optimization solver.

As the input super-structure E_{super}, we supply an estimated moral graph, computed using the graphical lasso procedure (Friedman et al., 2008). In all experiments, we set the graphical lasso penalty to 0.01 (using a larger value for some data sets to avoid ill-conditioning) and then threshold the estimated precision matrix by retaining only those entries with absolute value ≥ 0.1. To make our comparisons fair, we appropriately modify the competing methods so that E_{super} can also be supplied as input. Note that we count the number of support after each update in Algorithm 1. Converting the graph into a string key at each iteration is inefficient. Therefore, in the implementation, we count the support only after each full loop, setting the threshold to C instead of Cm^2. Throughout this paper, C is set to 5. We experimented with different initializations of Γ^{init} beyond the default value and observed similar performance across the various choices. As a result, throughout this section, we use the default Γ^{init} = identity. We emphasize again that our theoretical convergence and optimality results in Section 4 hold for any initialization.

We use the metric d_{cpdag} to evaluate the estimation accuracy as the underlying DAG is generally identifiable up to the Markov equivalence class. The metric d_{cpdag} is the number of different entries between the unweighted adjacency matrices of the estimated completed partially directed acyclic graph (CPDAG) and the true CPDAG. A CPDAG has a directed edge from a node i to a node j if and only if this directed edge is present in every DAG in the associated Markov equivalence class, and it has an undirected edge between nodes i and j if the corresponding Markov equivalence class contains DAGs with both directed edges from i to j and from j to i.

The time limit for the integer programming method MICODAG is set to $50m$. If the algorithm does not terminate within the time limit, we report the solution time (in seconds) and the achieved relative optimality gap, computed as RGAP = (upper bound − lower bound)/lower bound. Here, the upper bound and lower bound refer to the objective value associated with the best feasible solution and best lower bound, obtained respectively by MICODAG. A zero value for RGAP indicates that an optimal solution has been found.

For each method, we report results under oracle tuning, i.e., selecting the parameter that yields the best performance. This provides a benchmark of the best-case potential of the methods.

Setup of synthetic experiments: For all the synthetic experiments, following the experimental setup in Xu et al. (2025), once we specify a DAG, we generate data according to the SEM (1), where the nonzero entries of $B^\star$ are drawn uniformly at random from the set $\{-0.8, -0.6, 0.6, 0.8\}$. Unless otherwise specified, the diagonal entries of $\Omega^\star$ are drawn uniformly at random from the set $\{0.8, 1, 1.2\}$. This generation of noise variance introduces some heteroscedasticity. Later, we vary this set to study how our method performs under different levels of heteroscedasticity.

6.1 Convergence of CD-ℓ_0 Solution to an Optimal Solution

Theorem 7 states that as the sample size tends to infinity, CD-ℓ_0 identifies an optimally scoring model. To observe how quickly the asymptotics take effect, we generate three synthetic DAGs with $m = 10$ nodes, where the total number of edges is chosen from the set $\{7, 12, 21\}$. We obtain 10 independently and identically distributed data sets according to the SEM described earlier with sample size $n = \{100, 200, 300, 400, 500, 800, 1000, 1600, 3200\}$. In Figure 2, we compute the normalized difference $(\text{obj}^{\text{method}} - \text{obj}^{\text{opt}})/\text{obj}^{\text{opt}}$ as a function of n for the three graphs, averaged across the ten independent trials. Here, $\text{obj}^{\text{method}}$ is the objective value obtained by the corresponding method (CD-ℓ_0 or GES), while obj^{opt} is the optimal objective of (3) obtained by the integer programming approach MICODAG. For moderately large sample sizes, CD-ℓ_0 attains the optimal objective value, whereas GES does

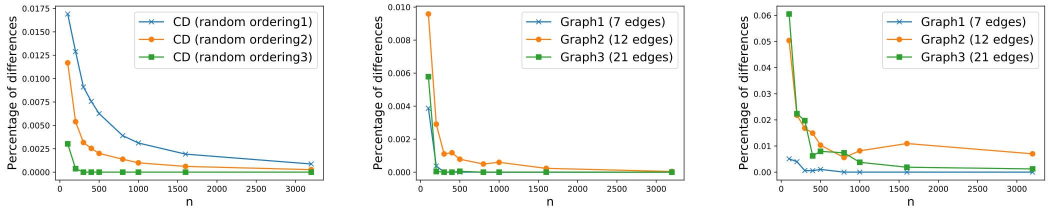


Figure 2: Convergence of CD-ℓ_0 to an optimal solution
Left: normalized difference of CD-ℓ_0 with 3 different random orderings; Middle: normalized difference for different graphs; Right: normalized difference of objectives of solutions obtained from MICODAG and GES. All results are computed and averaged over ten independent trials.

not. Meanwhile, CD-ℓ_0 obtains a near-optimal solution at about 0.1% of time, as compared to MICODAG. In the left panel of Figure 2, we run CD-ℓ_0 with three different random orderings. In all cases, the objective converges to zero as expected, though at varying rates. The worse-case rate is $\mathcal{O}(\sqrt{d_{\max}^2 m^4 \log m/n})$ according to Theorem 7.

6.2 Comparison to Benchmarks Under Near-homoscedastic Error

We next generate data sets from twelve publicly available networks sourced from Manzour et al. (2021) and the Bayesian Network Repository (bnlearn). These networks have different numbers of nodes, ranging from $m = 6$ to $m = 70$. We generate 10 independently and identically distributed data sets for each network according to the SEM described earlier with sample size $n = 500$. Since the diagonal entries of $\Omega^\star$ are close (with the largest deviation being 0.8 to 1.2), our algorithm performs very similarly to TD, since the topological ordering obtained by Algorithm 3 is likely to be correct in this near-homoscedastic case, and both methods perform well. For this reason, we present the results for the nearly homoscedastic case in Appendix C, and instead focus on a slightly higher degree of heteroscedasticity to highlight that our method is more robust to heteroscedasticity than TD. To that end, the diagonal entries of $\Omega^\star$ are drawn uniformly from $\{0.6, 1, 1.2\}$.

Table 1 compares the performance of our method CD-ℓ_0 with the competing ones. First, consider small graphs ($m \leq 20$) for which the integer programming approach MICODAG achieves an optimal or near-optimal solution with a small RGAP. As expected, in terms of the accuracy of the estimated model, MICODAG exhibits the best performance. For these small graphs, CD-ℓ_0 performs similarly to MICODAG but attains the solutions much faster. Next, consider moderately sized graphs ($m > 20$). In this case, MICODAG cannot solve these problem instances within the time limit and hence finds inaccurate models, whereas CD-ℓ_0 obtains much more accurate models much faster. Finally, CD-ℓ_0 outperforms CCDr-MCP, GES, TD, and NOTEARS in most problem instances. The improved performance of CD-ℓ_0 over CCDr-MCP highlights the advantage of using ℓ_0 penalization over a minimax concave penalty: ℓ_0 penalization ensures that DAGs in the same Markov equivalence class have the same score. In contrast, this property does not hold for other penalties. Our method outperforms NOTEARS in terms of the d_{cpdag} for most graphs. Both methods are designed for computational efficiency, but NOTEARS uses a least squares loss, which implicitly assumes homoscedastic (equal variance) noise (van de Geer and Bühlmann, 2013). In contrast, our approach directly optimizes the original likelihood function, allowing us to

	MICODAG			CCDr-MCP		GES		TD		NOTEARS		CD-ℓ_0	
Network(m)	Time	RGAP	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}
Dsep(6)	≤ 1	0	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.1($\pm$0.3)	≤ 1	2.0($\pm$0.0)	≤ 1	2.0($\pm$0)
Asia(8)	≤ 1	0	2.0($\pm$0)	≤ 1	2.3($\pm$0.9)	≤ 1	2.1($\pm$0.3)	≤ 1	2.1($\pm$0.3)	≤ 1	3.5($\pm$2.4)	≤ 1	2.1($\pm$0.3)
Bowling(9)	1.2	0	2.0($\pm$0)	≤ 1	3.1($\pm$1.9)	≤ 1	11.8($\pm$8.0)	≤ 1	2.2($\pm$0.6)	≤ 1	3.6($\pm$0.8)	≤ 1	2.0($\pm$0)
InsSmall(15)	≥ 750	.007	7.2($\pm$1.0)	≤ 1	27.0($\pm$4.1)	≤ 1	37.2($\pm$4.7)	≤ 1	3.9($\pm$4.8)	≤ 1	10.9($\pm$1.6)	≤ 1	4.0($\pm$4.6)
Rain(14)	56.0	0	2.0($\pm$0)	≤ 1	8.5($\pm$2.1)	≤ 1	15.5($\pm$2.2)	≤ 1	6.6($\pm$2.1)	≤ 1	5.4($\pm$2.1)	≤ 1	5.5($\pm$2.0)
Cloud(16)	9.7	0	12.7($\pm$5.5)	≤ 1	8.5($\pm$4.5)	≤ 1	10.6($\pm$8.2)	≤ 1	11.9($\pm$7.8)	≤ 1	17.6($\pm$3.7)	≤ 1	11.6($\pm$6.5)
Funnel(18)	8.0	0	2.0($\pm$0)	≤ 1	3.3($\pm$1.7)	≤ 1	2.6($\pm$1.3)	≤ 1	6.4($\pm$2.5)	≤ 1	2.9($\pm$1.4)	≤ 1	5.4($\pm$2.0)
Galaxy(20)	48.6	0	1.0($\pm$0)	≤ 1	9.1($\pm$4.3)	≤ 1	26.2($\pm$0.6)	≤ 1	11.7($\pm$6.7)	≤ 1	12.4($\pm$2.1)	≤ 1	9.7($\pm$4.9)
Insurance(27)	≥ 1350	.272	18.5($\pm$8.5)	≤ 1	32.5($\pm$6.0)	≤ 1	24.2($\pm$7.9)	≤ 1	16.8($\pm$2.2)	≤ 1	12.8($\pm$4.0)	≤ 1	18.3($\pm$1.9)
Factors(27)	≥ 1350	.242	60.4($\pm$7.3)	≤ 1	65.1($\pm$6.4)	≤ 1	68.5($\pm$9.3)	≤ 1	26.9($\pm$6.4)	≤ 1	41.7($\pm$5.1)	≤ 1	25.2($\pm$9.7)
Hfinder(56)	≥ 2800	.181	12.7($\pm$4.6)	≤ 1	14.9($\pm$3.8)	≤ 1	27.5($\pm$14.4)	1.2	38.3($\pm$7.7)	1.1	17.3($\pm$4.0)	≤ 1	45.1($\pm$8.9)
Hepar(70)	≥ 3500	.868	45.6($\pm$9.8)	≤ 1	51.6($\pm$9.5)	≤ 1	71.9($\pm$19.7)	2.6	54.2($\pm$9.8)	2.2	43.9($\pm$5.4)	≤ 1	38.5($\pm$5.9)

Table 1: Comparison of our method, CD-ℓ_0, with competing methods
Here, MICODAG, mixed-integer convex program (Xu et al., 2025); CCDr-MCP, minimax concave penalized estimator with coordinate descent (Aragam et al., 2019); GES, greedy equivalence search algorithm (Chickering, 2002); TD, top-down method (Chen et al., 2019); NOTEARS, (Zheng et al., 2018); d_{cpdag}, differences between the true and estimated completed partially directed acyclic graphs; RGAP, relative optimality gap. All results are computed over ten independent trials, where the average d_{cpdag} values are presented with their standard deviations.

	CCDr-MCP		GES		CD-ℓ_0	
Network(m)	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}
Pathfinder(109)	5.5	206.6($\pm$4.86)	≤ 1	253.8($\pm$19.52)	24.0	95.0($\pm$18.40)
Andes(223)	12.6	121.5($\pm$5.23)	1.9	152.8($\pm$22.26)	108.9	98.4($\pm$6.82)
Diabetes(413)	42.6	322.9($\pm$9.39)	10.7	458.7($\pm$28.73)	732.3	158.4($\pm$10.76)

Table 2: Comparison of our method, CD-ℓ_0, with competing methods for large graphs
See Table 1 for the description of the methods. All results are computed over ten independent trials, where the average d_{cpdag} values are presented with their standard deviations.

account for heteroscedasticity. This results in improved accuracy and robustness under varying noise conditions.

Large graphs: We next demonstrate the scalability of our coordinate descent algorithm for learning large DAGs with over 100 nodes. We consider networks from the Bayesian Network Repository and generate 10 independent data sets similar to those in the previous experiment. The method TD is excluded since it cannot scale to very large graphs. Table 2 presents the results where we see that our method CD-ℓ_0 can effectively scale to large graphs and obtain better or comparable performance to competing methods, as measured by the d_{cpdag} metric.

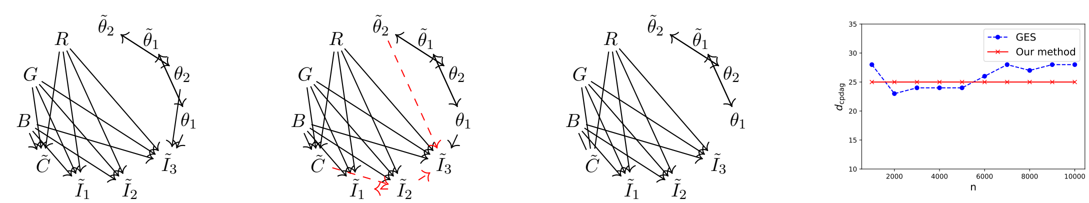
6.3 Severe Heteroscedastic Error and Limitations of Our Method

One limitation of our algorithm is that under severe heteroscedasticity, recovering an accurate topological ordering becomes difficult, which can degrade performance. To demonstrate this, we draw diagonal entries of $\Omega^\star$ uniformly at random from the set $\{0.2, 1, 5\}$, so that the noise distribution is highly heteroscedastic. In this setup, Assumption 3 is unlikely to hold, making the input topological ordering less reliable for both TD and our method. Despite this, the results in Table 3 show that both approaches remain competitive.

Our algorithm CD-ℓ_0 does not rely on a specific topological ordering and can take any ordering as input—not just the one produced by TD. In practice, if an accurate method for

Network(m)	GES	d_{cpdag} TD	CD-ℓ_0
Dsep(6)	4.9(±1.4)	5.4(±0.8)	4.7(±1.2)
Asia(8)	6.4(±2.8)	14.4(±1.6)	16.4(±1.0)
Bowling(9)	8.2(±6.8)	10.0(±0)	10.0(±0)
InsuranceSmall(15)	22.5(±5.1)	18.7(±2.5)	17.2(±1.6)
Rain(14)	26.6(±1.8)	10.5(±2.8)	9.9(±2.3)
Cloud(16)	14.9(±3.6)	19.4(±1.8)	19.0(±3.8)
Funnel(18)	10.0(±2.1)	9.4(±2.1)	7.6(±1.4)
Galaxy(20)	6.0(±2.4)	35.6(±3.9)	26.5(±1.6)
Insurance(27)	61.1(±7.7)	65.8(±3.6)	46.9(±4.1)
Factors(27)	74.1(±4.8)	47.5(±7.5)	40.0(±10.0)
Hfinder(56)	41.0(±5.1)	93.6(±2.8)	100.0(±3.7)
Hepar(70)	68.5(±11.9)	124.1(±8.7)	91.5(±12.5)

Table 3: Comparison of GES, Top-down (TD), and CD-ℓ_0 using data sets with very large heteroscedasticity.



(a) Ground truth (b) GES (c) CD-ℓ_0 (d) d_{cpdag} comparison

Figure 3: Learning causal models from causal chambers data in Gamella et al. (2025)
Here, a. ground-truth DAG described in Gamella et al. (2025), b-c. the estimated CPDAGs by GES and CD-ℓ_0 for sample size $n = 10000$, d. comparing the accuracy of the CPDAGs estimated by our method CD-ℓ_0 and GES with different sample sizes n; here the accuracy is computed relative to CPDAG of the ground-truth DAG and uses the metric d_{cpdag}.

obtaining the topological ordering in a heteroscedastic setting is available, it can be used as input to Algorithm 1.

6.4 Real Data from Causal Chambers

Recently, Gamella et al. (2025) constructed two devices, referred to as causal chambers, allowing us to quickly and inexpensively produce large data sets from non-trivial but well-understood real physical systems. The ground-truth DAG underlying this system is known and shown in Figure 3(a). We collect $n = 1000$ to $n = 10000$ observational samples of $m = 20$ variables at increments of 1000. To maintain clarity, we only plot a subset of the variables in Figure 3(a, b, c). However, the analysis includes all variables. With this data, we obtain estimates for the Markov equivalence class of the ground-truth DAG using GES and our method CD-ℓ_0 and measure the accuracy of the estimates using the d_{cpdag} metric.

Figures 3(b-c) show the estimated CPDAG for each approach when $n = 10000$. Neither method picks up the edges between the polarizer angles θ_1 and θ_2 and other variables. As mentioned in Gamella et al. (2025), this phenomenon is likely due to these effects being nonlinear. However, with a large sample size, CD-ℓ_0 produces graphs with fewer false positive edges. Figure 3(d) compares the accuracy of CD-ℓ_0 and GES in estimating the Markov equivalence class of the ground-truth DAG for different sample sizes. We observe that CD-ℓ_0 and GES perform similarly on average.

7. Discussion

In this paper, we propose the first coordinate descent procedure with proven optimality guarantees in the context of learning Bayesian networks. Numerical experiments demonstrate that our coordinate descent method is scalable and provides high-quality solutions.

We showed in Theorem 6 that our coordinate descent algorithm converges. It would be of interest to characterize the speed of convergence. In addition, the computational complexity of our algorithm may be improved by updating blocks of variables instead of one coordinate at a time. Further, an open question is whether, in the context of our guarantees in Theorem 7, the sample size requirement and the rates of convergence to an optimal solution can be improved. Finally, another open question is to provide statistical consistency guarantees for our algorithm. Xu et al. (2025) showed that, for sufficiently large n, any feasible Γ for (3) whose objective value differs from the optimum by at most $\mathcal{O}(\lambda^2)$ also belongs to the population Markov equivalence class. According to Theorem 7, the gap achieved by coordinate descent can be much larger. It would be interesting to investigate whether under some orderings O, the gap obtained by coordinate descent can be on the order λ^2 so that we can attain consistency. Indeed, the synthetic result in Figure 2 (left) suggests that some orderings have much faster convergence to optimality than others.

Acknowledgments

We thank the AE and the two referees for their constructive comments that improved this paper. We also thank Rohit Bhattacharya for his thoughtful feedback on an earlier version of this manuscript, which led to improvements in the paper. Simge Küçükyavuz and Tong Xu are supported, in part, by the Office of Naval Research Global [Grant N00014-22-1-2602]. Armeen Taeb is supported by NSF [Grant DMS-2413074] and by the Royalty Research Fund at the University of Washington. Ali Shojaie is supported by grant R01GM133848 from the National Institutes of Health.

Appendix A. Proof of Theorem 6

The proof of Theorem 6 relies on the following definitions and lemmas, and it closely follows the approach outlined in Hazimeh and Mazumder (2020). With a slight abuse of notation, we let $\ell(\Gamma) := \sum_{i=1}^{m} -2\log(\Gamma_{ii}) + \mathrm{tr}(\Gamma\Gamma^{\mathrm{T}}\hat{\Sigma}_n)$ to be the negative log-likelihood function associated with parameter $\Gamma \in \mathbb{R}^{m\times m}$.

Definition 14 *(Coordinate-wise (CW) minimum Hazimeh and Mazumder, 2020) A connectivity matrix $\Gamma^{\mathrm{CW}} \in \mathbb{R}^{m\times m}$ of a DAG is the CW minimum of problem* (3) *if for every $(u, v), u, v = 1, \ldots, m$, $\Gamma^{\mathrm{CW}}_{uv}$ is a minimizer of $g(\Gamma_{uv})$ with other coordinates of Γ^{CW} held fixed.*

Lemma 15 *Let $\{\Gamma^j\}_{j=1}^{\infty}$ be the sequence generated by Algorithm 1. Then the sequence of objective values $\{f(\Gamma^j)\}_{j=1}^{\infty}$ is decreasing and converges.*

Proof The Hessian matrix of function $\ell(\Gamma)$ with respect to Γ is $2\mathrm{diag}\left(1/\Gamma_{11}^2, \ldots, 1/\Gamma_{mm}^2\right) + 2\hat{\Sigma}$. Therefore, by Assumption 1, $\ell(\Gamma)$ is strongly convex and thus bounded below, and so is $f(\Gamma)$. If Γ^j is the result of a non-spacer step, then the inequality $f(\Gamma^j) \le f(\Gamma^{j-1})$ holds trivially. Similarly, we know that if Γ^j results from a spacer step, then, $\ell(\Gamma^j) \le \ell(\Gamma^{j-1})$. Since a spacer step updates only coordinates on the support, it cannot increase the support size of Γ^{j-1}, that is, $\|\Gamma^j - \mathrm{diag}(\Gamma^j)\|_{\ell_0} \le \|\Gamma^{j-1} - \mathrm{diag}(\Gamma^{j-1})\|_{\ell_0}$, thus $f(\Gamma^j) \le f(\Gamma^{j-1})$. Since $f(\Gamma^j)$ is non-increasing and bounded below, it must converge. ∎

Lemma 16 *The sequence $\{\Gamma^t\}_{t=1}^{\infty}$ generated by Algorithm 1 is bounded.*

Proof By Algorithm 1, $\Gamma^t \in G := \{\Gamma \in \mathbb{R}^{m\times m} \mid f(\Gamma) \le f(\Gamma^0)\}$. It suffices to show that the set G is bounded. From Proposition 11.11 in Bauschke and Combettes (2011), if the function f is coercive, then the set G is bounded. Since $f(\Gamma) \ge \ell(\Gamma)$ for every Γ, it suffices to show that the function ℓ is coercive. By Assumption 1, we have that the function ℓ is strongly convex. The lemma then follows from the classical result in convex analysis that strongly convex functions are coercive. ∎

The following lemma characterizes the limit points of Algorithm 1.

Lemma 17 *Let $\hat{E}$ be a support set that is generated infinitely often by the non-spacer steps of Algorithm 1, and let $\{\Gamma^l\}_{l\in L}$ be the estimates from the spacer steps when the support of the input matrix is $\hat{E}$. Then:*

1. *There exists a positive integer M such that for all $l \in L$ with $l \ge M$, $\mathrm{support}(\Gamma^l) = \hat{E}$.*
2. *There exists a subsequence of $\{\Gamma^l\}_{l\in L}$ that converges to a stationary solution Γ^{CW}, where, Γ^{CW} is the unique minimizer of $\min_{\mathrm{support}(\Gamma)\subseteq\hat{E}} \ell(\Gamma)$.*
3. *Every subsequence of $\{\Gamma^t\}_{t\ge 0}$ with support $\hat{E}$ converges to Γ^{CW}.*

Proof Part 1.) Since spacer steps optimize only over the coordinates in $\hat{E}$, no element outside $\hat{E}$ can be added to the support. Thus, for every $l \in L$ we have support$(\Gamma^l) \subseteq \hat{E}$. We next show that strict containment is not possible via contradiction. Suppose support$(\Gamma^l) \subsetneq \hat{E}$ occurs infinitely often, and consider some $l \in L$ where this occurs. By the spacer step of Algorithm 1, the previous iterate Γ^{l-1} has support $\hat{E}$, implying $\left\|\Gamma^{l-1}\right\|_0 - \left\|\Gamma^l\right\|_0 \geq 1$. Moreover, from the definition of the spacer step, we have $\ell(\Gamma^l) \leq \ell(\Gamma^{l-1})$. Therefore, we get $f(\Gamma^{l-1}) - f(\Gamma^l) = \ell(\Gamma^{l-1}) - \ell(\Gamma^l) + \lambda^2(\|\Gamma^{l-1}\|_0 - \|\Gamma^l\|_0) \geq \lambda^2$. Thus, when support$(\Gamma^l) \subsetneq \hat{E}$ occurs, f decreases by at least λ^2. Therefore, $\Gamma^l \subsetneq \hat{E}$ infinitely many times implies that $f(\Gamma)$ is not lower-bounded, which is a contradiction.

Part 2.) The proof follows the conventional procedure for establishing the convergence of cyclic coordinate descent (CD) (Bertsekas, 2016; Hazimeh and Mazumder, 2020). We obtain Γ^l by updating every coordinate in $\hat{E}$ of Γ^{l-1}. Denote the intermediate steps as $\Gamma^{l,1}, \ldots, \Gamma^{l,|\hat{E}|}$, where $\Gamma^{l,|\hat{E}|} = \Gamma^l$. We aim to show that the sequence $\{\Gamma^{l,|\hat{E}|}\}_{l \in L}$ converges to a point Γ^{CW}, and similarly, other sequences $\{\Gamma^{l,i}\}_{l \in L}, i = 1, \ldots, |\hat{E}| - 1$, also converge to Γ^{CW}. By Lemma 16, since $\{\Gamma^{l,|\hat{E}|}\}_{l \in L}$ is a bounded sequence, there exists a converging subsequence $\{\Gamma^{l',|\hat{E}|}\}_{l' \in L'}$ with a limit point Γ^{CW}. Without loss of generality, we choose the subsequence satisfying $l' > M$, $\forall l' \in L'$. From Part 1 of the lemma, $\{\Gamma^{l',1}\}_{l' \in L'}, \ldots, \{\Gamma^{l',|\hat{E}|-1}\}_{l' \in L'}$ all have the same support $\hat{E}$. For $\{\Gamma^{l',|\hat{E}|-1}\}_{l' \in L'}$, we have $f(\Gamma^{l',|\hat{E}|-1}) - f(\Gamma^{l',|\hat{E}|}) = \ell(\Gamma^{l'|\hat{E}|-1}) - \ell(\Gamma^{l',|\hat{E}|})$. If the change from $\Gamma^{l',|\hat{E}|-1}$ to $\Gamma^{l',|\hat{E}|}$ is on a diagonal entry, say Γ_{uu}, then,

$$
\begin{aligned}
&\ell\left(\Gamma^{l',|\hat{E}|-1}\right) - \ell\left(\Gamma^{l',|\hat{E}|}\right) \\
=& -2\log\left(\Gamma_{uu}^{l',|\hat{E}|-1}/\Gamma_{uu}^{l',|\hat{E}|}\right) + \left(\left(\Gamma_{uu}^{l',|\hat{E}|-1}\right)^2 - \left(\Gamma_{uu}^{l',|\hat{E}|}\right)^2\right)\hat{\Sigma}_{uu} + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)A_{uu} \\
=& -2\log\left(\Gamma_{uu}^{l',|\hat{E}|-1}/\Gamma_{uu}^{l',|\hat{E}|}\right) + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)^2\hat{\Sigma}_{uu} - 2\left(\Gamma_{uu}^{l',|\hat{E}|}\right)^2\hat{\Sigma}_{uu} \\
&+ 2\Gamma_{uu}^{l',|\hat{E}|-1}\Gamma_{uu}^{l',|\hat{E}|}\hat{\Sigma}_{uu} + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)A_{uu} \\
=& -2\log\left(\Gamma_{uu}^{l',|\hat{E}|-1}/\Gamma_{uu}^{l',|\hat{E}|}\right) + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)^2\hat{\Sigma}_{uu} \\
&+ 2\left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)\Gamma_{uu}^{l',|\hat{E}|}\hat{\Sigma}_{uu} + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)A_{uu}.
\end{aligned}
\tag{12}
$$

From the update rule in Proposition 3, we have

$$
(4\Gamma_{uu}^{l',|\hat{E}|}\hat{\Sigma}_{uu} + A_{uu})^2 = A_{uu}^2 + 16\hat{\Sigma}_{uu},
$$

which gives

$$
2\Gamma_{uu}^{l',|\hat{E}|}\hat{\Sigma}_{uu} + A_{uu} = \frac{2}{\Gamma_{uu}^{l',|\hat{E}|}}.
\tag{13}
$$

Combining (12) and (13), we have

$$
\begin{aligned}
&\ell\left(\Gamma^{l',|\hat{E}|-1}\right) - \ell\left(\Gamma^{l',|\hat{E}|}\right) \\
=&2\left(-\log\left(\Gamma_{uu}^{l',|\hat{E}|-1}/\Gamma_{uu}^{l',|\hat{E}|}\right) + \Gamma_{uu}^{l',|\hat{E}|-1}/\Gamma_{uu}^{l',|\hat{E}|} - 1\right) + \left(\Gamma_{uu}^{l',|\hat{E}|-1} - \Gamma_{uu}^{l',|\hat{E}|}\right)^2\hat{\Sigma}_{uu}.
\end{aligned}
\tag{14}
$$

Since $a-1 \geq \log(a)$ for $a \geq 0$, each of the two terms above is non-negative. From Lemma 15, as $l' \to \infty$, $f(\Gamma^{l',|\hat{E}|-1}) - f(\Gamma^{l',|\hat{E}|})$ or equivalently $\ell(\Gamma^{l',|\hat{E}|-1}) - \ell(\Gamma^{l',|\hat{E}|})$ converges to 0 as $l' \to \infty$. Combining this with the fact that $\ell(\Gamma^{l',|\hat{E}|-1}) - \ell(\Gamma^{l',|\hat{E}|}) \geq 0$ and that each term in (14) is non-negative, we conclude that $\Gamma^{l',|\hat{E}|-1}$ must converge to $\Gamma^{l',|\hat{E}|}$ as $l' \to \infty$. Since $\Gamma^{l',|\hat{E}|}$ converges to Γ^{CW}, $\Gamma^{l',|\hat{E}|-1}$ must also converge to Γ^{CW}. Repeating a similar argument, we conclude that $\Gamma^{l',j}$ converges to Γ^{CW} for all $j = 1, 2, \ldots, |\hat{E}|$.

If the change from $\Gamma^{l',|\hat{E}|-1}$ to $\Gamma^{l',|\hat{E}|}$ is on an off-diagonal entry, say Γ_{uv} with $u \neq v$, then, after some algebra,

$$f\left(\Gamma^{l',|\hat{E}|-1}\right) - f\left(\Gamma^{l',|\hat{E}|}\right) = \ell\left(\Gamma^{l',|\hat{E}|-1}\right) - \ell\left(\Gamma^{l',|\hat{E}|}\right) = \left(\Gamma_{uv}^{l',|\hat{E}|-1} - \Gamma_{uv}^{l',|\hat{E}|}\right)^2 \hat{\Sigma}_{uu}\,.$$

Again, appealing to Lemma 15 as before, we can conclude that $\Gamma^{l',|\hat{E}|-1}$ converges to Γ^{CW} as $l' \to \infty$. Similarly, $\Gamma^{l',j}$ converges to Γ^{CW} for every $j = 1, 2, \ldots, |\hat{E}| - 1$.

Consider $k, l \in L'$ with $k > l$ such that for the j-th coordinate in $\hat{E}$, $f(\Gamma^k) \leq f(\Gamma^{l,j}) \leq f(\tilde{\Gamma}^{l,j})$. Here, $\tilde{\Gamma}^{l,j}$ equals to $\Gamma^{l,j}$ except for the j-th nonzero coordinate in $\hat{E}$. As $k, l \to \infty$, we have, from the above analysis, that there exists a matrix Γ^{CW} such that $\Gamma^k \to \Gamma^{\mathrm{CW}}$ and $\Gamma^{l,j} \to \Gamma^{\mathrm{CW}}$. Thus, Γ^{CW} and $\lim_{l\to\infty} \tilde{\Gamma}^{l,j}$ differ by only one coordinate in the j-th position. We conclude that $f(\Gamma^{\mathrm{CW}}) \leq f(\lim_{l\to\infty} \tilde{\Gamma}^{l,j})$. In other words, Γ^{CW} is coordinate-wise minimum. Furthermore, since the optimization problem $\min_{\mathrm{support}(\Gamma)\subseteq\hat{E}} \ell(\Gamma)$ is strongly convex by Assumption 1, Γ^{CW} is the unique minimizer of this optimization problem.

Part 3.) Consider any subsequence $\{\Gamma^k\}_{k\in K}$ such that $\mathrm{support}(\Gamma^k) = \hat{E}$. We will show by contradiction that $\{\Gamma^k\}_{k\in K}$ must converge to Γ^{CW}. Suppose $\{\Gamma^k\}_{k\in K}$ has a limit point $\hat{\Gamma} \neq \Gamma^{\mathrm{CW}}$. Then there exist a subsequence $\{\Gamma^{k'}\}_{k'\in K'}$, with $K' \subseteq K$, that converges to $\hat{\Gamma}$. Therefore, $\lim_{k'\to\infty} f(\Gamma^{k'}) = \ell(\hat{\Gamma}) + \lambda^2|\hat{E}|$. From part 1 and part 2, we have that, for the subsequence $\{\Gamma^{l'}\}$, $\lim_{l'\to\infty} f(\Gamma^{l'}) = \ell(\Gamma^{\mathrm{CW}}) + \lambda^2|\hat{E}|$. By Lemma 15, we have $\lim_{k'\to\infty} f(\Gamma^{k'}) = \lim_{l'\to\infty} f(\Gamma^{l'})$. Thus, we conclude that $\ell(\hat{\Gamma}) = \ell(\Gamma^{\mathrm{CW}})$, which contradicts the fact that Γ^{CW} is the unique minimizer of $\min_{\mathrm{support}(\Gamma)\subseteq\hat{E}} \ell(\Gamma)$. Therefore, we conclude that any subsequence with support $\hat{E}$ converges to Γ^{CW} as $k \to \infty$. ■

The next lemma shows that if the limit point of $\{\Gamma^k\}$ has a certain support, then that support must appear infinitely often in the sequence.

Lemma 18 *Let Γ be a limit point of $\{\Gamma^k\}_{k=1}^{\infty}$ with* $\mathrm{support}(\Gamma) = E$. *We have* $\mathrm{support}(\Gamma^k) = E$ *for infinitely many k's.*

Proof We prove this result by contradiction. Assume that there are only finitely many k's such that $\mathrm{support}(\Gamma^k) = E$. Since there are finitely many possible support sets, there is a support $E' \neq E$ and a subsequence $\{\Gamma^{k'}\}$ of $\{\Gamma^k\}$ such that $\mathrm{support}(\Gamma^{k'}) = E'$ for all k', and $\lim_{k'\to\infty} \Gamma^{k'} = \Gamma$. However, by Lemma 17, the subsequence converges to a minimizer Γ^{CW} with $\mathrm{support}(\Gamma^{\mathrm{CW}}) = E'$ and thus $\Gamma^{\mathrm{CW}} \neq \Gamma$. This is a contradiction. ■

This lemma rules out the possibility that a coordinate remains nonzero throughout the sequence, gradually decreasing to zero in the limit.

We are now ready to complete the proof of Theorem 6.

Proof [Proof of Theorem 6] Let Γ be a limit point of $\{\Gamma^k\}$ with the largest support size and denote its support by $\hat{E}$. By Lemma 18, there is a subsequence $\{\Gamma^r\}_{r\in R}$ of $\{\Gamma^k\}$ such that $\text{support}(\Gamma^r) = \hat{E}, \forall r \in R$, and $\lim_{r\to\infty} \Gamma^r = \Gamma$. By Lemma 17, there exists an integer M such that for every $r \geq M$ and $r+1$ is a spacer step, we have $\text{support}(\Gamma^r) = \text{support}(\Gamma^{r+1})$. Without loss of generality, we choose the subsequence for which $r > M, \forall r \in R$. We will demonstrate by contradiction that any coordinate (u, v) in $\hat{E}$ cannot be dropped infinitely often in $\{\Gamma^k\}$. To this end, assume that $(u, v) \notin \{\text{support}(\Gamma^k)\}_{k>M}$ infinitely often. Let $\{\Gamma^{r'}\}_{r'\in R'}$, where $R' \subseteq R$, be the subsequence with $\text{support}(\Gamma^{r'+1}) = \hat{E} \setminus \{(u, v)\}, \forall r' \in R'$. Since $r' > M$ and the support has been changed, $r'+1$ is not a spacer step. Therefore, using Proposition 3, we have $f(\Gamma^{r'}) - f(\Gamma^{r'+1}) = \ell(\Gamma^{r'}) - \ell(\Gamma^{r'+1}) + \lambda^2 = \lambda^2 + \Gamma^{r'}_{uv}(\Gamma^{r'}_{uv}\hat{\Sigma}_{uv} + A_{uv}) = \lambda^2 - A^2_{uv}/4\hat{\Sigma}_{uu} > 0$, where we use the update rule that $\Gamma^{r'}_{uv} = -A_{uv}/(2\hat{\Sigma}_{uu})$. By Lemma 15, we have $\lim_{r'\to\infty} f(\Gamma^{r'}) - f(\Gamma^{r'+1}) = 0$. Thus, $\lambda^2 = A^2_{uv}/4\hat{\Sigma}_{uu}$, where $A_{uv} = \sum_{j\neq u} \Gamma^{r'}_{jv}\hat{\Sigma}_{ju} + \sum_{k\neq u} \Gamma^{r'}_{kv}\hat{\Sigma}_{uk}$. By Proposition 3, in step $r'+1$, we have $|\Gamma^{r'+1}_{uv}| = \lambda/\sqrt{\hat{\Sigma}_{uu}} > 0$, which contradicts the definition of $\{\Gamma^{r'}\}_{r'\in R'}$. Therefore, no coordinate in $\hat{E}$ can be dropped infinitely often. Moreover, no coordinate can be added to $\hat{E}$ infinitely often as $\hat{E}$ is the largest support. As a result, the support converges to $\hat{E}$. With stabilized support $\hat{E}$, by Lemma 17, we have that $\{\Gamma^k\}$ converges to the limit Γ^{CW} with support $\hat{E}$. From Algorithm 1 and Proposition 3, we have Γ_{uv} is a minimizer of $f(\Gamma_{uv})$ with respect to the coordinate (u, v) and others fixed. Therefore, Γ^{CW} is the CW minimum.

■

Appendix B. Proof of Theorem 12

To facilitate the proof, we first introduce several technical lemmas that extend the homoscedastic case of Chen et al. (2019)'s Theorem 2 to our setting with mild heteroscedasticity. These lemmas rely on standard terminology and structural properties of directed acyclic graphs, which we recall below.

In a directed graph $\mathcal{G}(B^\star)$ induced by $B^\star$, if there is an edge $k \to j$, then k is a parent of j, and j is a child of k. We write $\text{pa}(j)$ for the set of parents of j and $\text{ch}(j)$ for its children. If there is a directed path $k \to \cdots \to j$, then k is an ancestor of j, and j is a descendant of k. The sets of ancestors and descendants of j are denoted by $\text{an}(j)$ and $\text{de}(j)$, respectively, with $j \in \text{an}(j)$ and $j \in \text{de}(j)$ by convention. A set C is *ancestral* if $\text{an}(j) \subseteq C$ for all $j \in C$.

The first lemma clarifies that the sources in $\mathcal{G}(B^\star)$ are characterized by minimal variances.

Lemma 19 *Let X be the variables defined in* (1)*, we have*

- *If* $\text{pa}(j) = \emptyset$*, then* $\text{var}(X_j) = \Omega^\star_{jj}$.
- *If* $\text{pa}(j) \neq \emptyset$*, then* $\text{var}(X_j) \geq \Omega^\star_{jj} + \Omega^\star_{ll}\zeta$ *for some* $l \in V$.

Proof Recall that $X = (I - B^{\star\top})^{-1}\epsilon$. Let $\Pi = (\pi_{jk}) = (I - B^{\star\top})^{-1}$, where π_{jk} denotes the total causal effect from node k to node j in the DAG defined by $B^\star$. we have

$$\text{var}(X_j) = \sum_{k=1}^{m} \Omega^\star_{kk}\pi^2_{jk}.$$

If $\mathrm{pa}(j) = \emptyset$, then $\pi_{jk}^2 = 0$ for all $k \neq j$, and thus

$$\mathrm{var}(X_j) = \Omega_{jj}^{\star} \pi_{jj}^2 = \Omega_{jj}^{\star}.$$

If $\mathrm{pa}(j) \neq \emptyset$, then by acyclicity of $\mathcal{G}(B^{\star})$, there exists a node $l \in \mathrm{pa}(j)$ such that $\mathrm{de}(l) \cap \mathrm{pa}(j) = \{l\}$. Then

$$\pi_{jl}^2 = {B_{jl}^{\star}}^2 \geq \zeta,$$

and

$$\mathrm{var}(X_j) = \Omega_{jj}^{\star} + \sum_{k \neq j} \Omega_{kk}^{\star} \pi_{jk}^2 \geq \Omega_{jj}^{\star} + \Omega_{ll}^{\star} \pi_{jl}^2 \geq \Omega_{jj}^{\star} + \Omega_{ll}^{\star} \zeta.$$

■

The next lemma shows that by conditioning on a source, or more generally, an ancestral set, one recovers a structural equation model whose associated graph has the source node or the entire ancestral set removed. For a variable X_j and a vector $X_C = (X_k : k \in C)$, we define $X_{j,C} = X_j - \mathbb{E}(X_j | X_C)$.

Lemma 20 *Let C be an ancestral set in $\mathcal{G}(B^{\star})$. Then $(X_{j,C} : j \notin C)$ is associated with the submatrix $B^{\star}[-C] = (\beta_{jk})_{j,k \notin C}$.*

Proof Let $j \notin C$. Since C is ancestral, X_C is a function of $\epsilon_C = (\epsilon_k : k \in C)$ only and thus independent of ϵ_j. Hence, $\mathbb{E}(\epsilon_j | X_C) = \mathbb{E}(\epsilon_j) = 0$. Because it also holds that $X_{k,C} = 0$ for $k \in C$, we have that

$$X_{j,C} = \sum_{k \in \mathrm{pa}(j) \setminus C} \beta_{jk} X_{k,C} + \epsilon_j.$$

■

Before proving Theorem 12, we give a lemma that addresses the estimation error for inverse covariances.

Lemma 21 (Lemma 6 in Chen et al. 2019) *Suppose all $(q+1) \times (q+1)$ principal submatrices of the covariance matrix $\Sigma^{\star} = \mathbb{E}(XX^T)$ has minimum eigenvalue at least $\lambda_{\min} > 0$. If for $\epsilon, \eta > 0$ we have*

$$n > (q+1)^2 \left\{ \log\left(\frac{2m^2 + 2m}{\epsilon} \right) \right\} 128 \left(1 + 4 \frac{\Omega_{\max}^{\star}}{\Omega_{\min}^{\star}} \right)^2 \left(\max_{j \in V} \Sigma_{jj}^{\star} \right)^2 \left(\frac{\eta \lambda_{\min} + 1}{\eta \lambda_{\min}^2} \right)^2, \quad (15)$$

then

$$\max_{C \subseteq V, |C| \leq q+1} \| (\Sigma_{C,C}^{\star})^{-1} - (\hat{\Sigma}_{C,C})^{-1} \|_{\infty} \leq \eta$$

with probability at least $1 - \epsilon$.

Proof [Proof of Theorem 12] Our assumption on n is as in (15) with $\eta = \frac{1}{2\Omega^\star_{\max}} - \frac{1}{2\Omega^\star_{\min}(1+\zeta)}$. Lemma 21 thus implies that, with probability at least $1-\epsilon$, we have, for all subsets $C \subseteq V$,

$$\left\|(\hat{\Sigma}_{C,C})^{-1} - (\Sigma_{C,C})^{-1}\right\|_\infty \leq \frac{1}{2\Omega^\star_{\max}} - \frac{1}{2\Omega^\star_{\min}(1+\zeta)}. \tag{16}$$

Let j be a source in $\mathcal{G}(B^\star)$, and let k be a non-source. Note that the variance of j conditional on some set C is

$$\Omega^\star_{j|C} = \frac{1}{\left\{(\Sigma_{C\cup\{j\},C\cup\{j\}})^{-1}\right\}_{j,j}}.$$

For an ancestral set $C \subseteq V \setminus \{j,k\}$ such that $\mathrm{pa}(j) \subseteq C$ and $\mathrm{pa}(k) \not\subseteq C$,

$$\left\{(\Sigma_{C\cup\{j\},C\cup\{j\}})^{-1}\right\}_{j,j} - \left\{(\Sigma_{C\cup\{k\},C\cup\{k\}})^{-1}\right\}_{k,k} \geq \frac{1}{\Omega^\star_{jj}} - \frac{1}{\Omega^\star_{kk} + \Omega^\star_{\min}\zeta} \geq \frac{1}{\Omega^\star_{\max}} - \frac{1}{\Omega^\star_{\min}(1+\zeta)}. \tag{17}$$

Using (16), we obtain that

$$\left\{(\hat{\Sigma}_{C\cup\{j\},C\cup\{j\}})^{-1}\right\}_{j,j} - \left\{(\hat{\Sigma}_{C\cup\{k\},C\cup\{k\}})^{-1}\right\}_{k,k} > 0. \tag{18}$$

Thus, $\hat{\Omega}^2_{j|C} < \hat{\Omega}^2_{k|C}$, which implies that Algorithm 3 correctly selects a source node at each step. In the first step, $C = \emptyset$, which is trivially an ancestral set. By induction, each subsequent step then correctly adds a sink to C, so C remains ancestral and a correct ordering is recovered. ■

Appendix C. Results of near homoscedastic case

Table 4 presents results for a near-homoscedastic scenario, where the diagonal entries of $\Omega^\star$ are sampled uniformly from $\{0.8, 1, 1.2\}$. Because the variance differences are small, the TD method is able to identify an good topological ordering, leading to comparable performance across methods.

References

Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research*, 16(1):2273–2328, 2015.

Bryon Aragam, Jiaying Gu, and Qing Zhou. Learning large-scale Bayesian networks with the sparsebn package. *Journal of Statistical Software*, 91:1–38, 2019.

Heinz H. Bauschke and Patrick L. Combettes. Convex analysis and monotone operator theory in Hilbert spaces. In *CMS Books in Mathematics*, 2011.

Kayhan Behdin, Wenyu Chen, and Rahul Mazumder. Sparse Gaussian graphical models with discrete optimization: Computational and statistical perspectives. *arXiv preprint arXiv:2307.09366*, 2023.

	MICODAG			CCDr-MCP		GES		TD		NOTEARS		CD-ℓ_0	
Network(m)	Time	RGAP	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}	Time	d_{cpdag}
Dsep(6)	≤ 1	0	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.0($\pm$0)	≤ 1	2.0($\pm$0)
Asia(8)	≤ 1	0	2.0($\pm$0)	≤ 1	2.3($\pm$0.95)	≤ 1	2.1($\pm$0.32)	≤ 1	3.2($\pm$1.93)	≤ 1	2.8($\pm$1.75)	≤ 1	3.2($\pm$1.93)
Bowling(9)	≤ 1	0	2.0($\pm$0)	≤ 1	3.1($\pm$1.91)	≤ 1	2.4($\pm$1.26)	≤ 1	2.0($\pm$0)	≤ 1	3.8($\pm$0.63)	≤ 1	2.0($\pm$0)
InsSmall(15)	569	.007	12.2($\pm$2.74)	≤ 1	27.0($\pm$4.08)	≤ 1	20.2($\pm$5.69)	≤ 1	8.0($\pm$0)	≤ 1	6.7($\pm$1.49)	≤ 1	8.0($\pm$0)
Rain(14)	36.7	0	2.2($\pm$0.63)	≤ 1	8.5($\pm$2.12)	≤ 1	4.2($\pm$3.33)	≤ 1	2.0($\pm$0)	≤ 1	3.4($\pm$2.50)	≤ 1	2.0($\pm$0)
Cloud(16)	30.0	0	5.0($\pm$0)	≤ 1	8.5($\pm$4.45)	≤ 1	5.0($\pm$0)	≤ 1	3.2($\pm$1.03)	≤ 1	19.1($\pm$2.73)	≤ 1	5.2($\pm$2.62)
Funnel(18)	14.4	0	2.1($\pm$0.32)	≤ 1	3.3($\pm$1.70)	≤ 1	8.9($\pm$10.29)	≤ 1	2.0($\pm$0)	≤ 1	3.2($\pm$1.55)	≤ 1	2.0($\pm$0)
Galaxy(20)	125.4	0	1.0($\pm$0)	≤ 1	9.1($\pm$4.33)	≤ 1	2.1($\pm$3.48)	≤ 1	1.2($\pm$0.63)	≤ 1	17.0($\pm$2.91)	≤ 1	1.2($\pm$0.63)
Insurance(27)	1379.5	.220	15.8($\pm$5.12)	≤ 1	32.5($\pm$6.04)	≤ 1	28.8($\pm$4.61)	0.124	11.7($\pm$2.67)	≤ 1	12.7($\pm$5.93)	0.111	10.3($\pm$1.49)
Factors(27)	1490.9	.217	65.1($\pm$6.69)	≤ 1	65.1($\pm$6.37)	≤ 1	67.0($\pm$8.26)	≤ 1	18.7($\pm$3.92)	≤ 1	18.4($\pm$4.27)	≤ 1	20.9($\pm$4.28)
Hfinder(56)	3940.4	.156	12.9($\pm$4.09)	≤ 1	14.9($\pm$3.84)	≤ 1	29.6($\pm$11.65)	1.50	14.1($\pm$7.34)	1.66	18.2($\pm$3.88)	≤ 1	14.7($\pm$9.19)
Hepar(70)	4132.8	.732	32.6($\pm$6.54)	≤ 1	51.6($\pm$9.50)	≤ 1	76.6($\pm$29.45)	3.60	18.5($\pm$7.37)	3.21	42.6($\pm$5.68)	1.23	11.6($\pm$6.33)

Table 4: Comparison of our method, CD-ℓ_0, with competing methods
Here, MICODAG, mixed-integer convex program (Xu et al., 2025); CCDr-MCP, minimax concave penalized estimator with coordinate descent (Aragam et al., 2019); GES, greedy equivalence search algorithm (Chickering, 2002); TD, top-down method (Chen et al., 2019); NOTEARS, (Zheng et al., 2018); d_{cpdag}, differences between the true and estimated completed partially directed acyclic graphs; RGAP, relative optimality gap. All results are computed over ten independent trials where the average d_{cpdag} values are presented with their standard deviations.

Dimitri Bertsekas. *Nonlinear Programming*, volume 4. Athena Scientific, 2016.

Wenyu Chen, Mathias Drton, and Y Samuel Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4):973–980, 2019.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

Byron Ellis and Wing Hung Wong. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302 – 332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Fei Fu and Qing Zhou. Learning sparse causal Gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

Juan L. Gamella, Jonas Peters, and Peter Bühlmann. The causal chambers: Real physical systems as a testbed for AI methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025.

Hussein Hazimeh and Rahul Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, 68(5):1517–1537, 2020.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8(22):613–636, 2007.

Simge Küçükyavuz, Ali Shojaie, Hasan Manzour, Linchuan Wei, and Hao-Hsiang Wu. Consistent second-order conic integer programming for learning Bayesian networks. *Journal of Machine Learning Research*, 24(322):1–38, 2023.

Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of Statistics*, 37(6B):4254, 2009.

Hasan Manzour, Simge Küçükyavuz, Hao-Hsiang Wu, and Ali Shojaie. Integer programming for learning directed acyclic graphs from continuous data. *INFORMS Journal on Optimization*, 3(1):46–73, 2021.

Nicolai Meinshausen and Peter Buhlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

Preetam Nandy, Alain Hauser, and Marloes Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *Annals of Statistics*, 46(6A):3151–3183, 2018.

Rajen Dinesh Shah and J. Peters. The hardness of conditional independence testing and the generalised covariance measure. *Annals of Statistics*, 2018.

Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In *Conference on Uncertainty in Artificial Intelligence*, page 445–452, 2006.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 1993. ISBN 978-1-4612-7650-0.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.

Caroline Uhler, Garvesh Raskutti, Peter Buhlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *Annals of Statistics*, 41:436–463, 2012.

Sara van de Geer and Peter Bühlmann. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *Annals of Statistics*, 41(2):536 – 567, 2013.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Conference on Uncertainty in Artificial Intelligence*, pages 255–270, 1990.

Tong Xu, Armeen Taeb, Simge Küçükyavuz, and Ali Shojaie. Integer programming for learning directed acyclic graphs from non-identifiable Gaussian models. *Biometrika*, 112 (3):asaf032, 04 2025.

Qiaoling Ye, Arash A Amini, and Qing Zhou. Optimizing regularized Cholesky score for order-based learning of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3555–3572, 2020.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric P Xing. DAGs with NO TEARS: continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31:9492–9503, 2018.