Learning Gaussian DAGs from Network Data

Hangjian Li lihangjian123@ucla.edu
Oscar Hernan Madrid Padilla oscar.madrid@stat.ucla.edu
Qing Zhou zhou@stat.ucla.edu
Department of Statistics
University of California, Los Angeles
Los Angeles, CA 90095, USA

Editor:

Abstract

Structural learning of directed acyclic graphs (DAGs) or Bayesian networks has been studied extensively under the assumption that data are independent. We propose a new Gaussian DAG model for dependent data which assumes the observations are correlated according to an undirected network. Under this model, we develop a method to estimate the DAG structure given a topological ordering of the nodes. The proposed method jointly estimates the Bayesian network and the correlations among observations by optimizing a scoring function based on penalized likelihood. We show that under some mild conditions, the proposed method produces consistent estimators after one iteration. Extensive numerical experiments also demonstrate that by jointly estimating the DAG structure and the sample correlation, our method achieves much higher accuracy in structure learning. When the node ordering is unknown, through experiments on synthetic and real data, we show that our algorithm can be used to estimate the correlations between samples, with which we can de-correlate the dependent data to significantly improve the performance of classical DAG learning methods.

Keywords: Bayesian networks, matrix normal, Lasso, network data

1. Introduction

Bayesian networks (BNs) with structure given by a directed acyclic graph (DAG) are a popular class of graphical models in statistical learning and causal inference. Extensive research has been done to develop new methods and theories to estimate DAG structures and its parameters from data. In this study we focus on the Gaussian DAG model defined as follows. Let $\mathcal{G}^* = (V, E)$ be a DAG that represents the structure of a BN for p random variables $X_1, \ldots, X_p$. The vertex set $V = \{1, \ldots, p\}$ represents the set of random variables and the edges set $E = \{(j, i) \in V \times V : j \to i\}$ represents the directed edges in $\mathcal{G}^*$. Let $\Pi_i = \{j \in V : (j, i) \in E\}$ denote the parent set of vertex i. A data matrix $X \in \mathbb{R}^{n \times p}$ is generated by the following Gaussian linear structural equations induced by $\mathcal{G}^*$:

$$X_j = \sum_{k \in \Pi_j} \beta^*_{kj} X_k + \varepsilon_j, \quad \varepsilon_j = (\varepsilon_{1j}, \ldots, \varepsilon_{nj}) \sim \mathcal{N}_n \left(0, \omega^{*2}_j I_n\right), \tag{1}$$

for $j = 1, \ldots, p$, where X_j is the jth column in X, ω^{*2}_j the error variance, and $B^* = (\beta^*_{kj})_{p \times p}$ is the weighted adjacency matrix (WAM) of $\mathcal{G}^*$ such that $\beta^*_{kj} \neq 0$ if and only if $(k, j) \in E$,

and $\beta^*_{jj} = 0$. The errors $\{\varepsilon_j\}$ are independent, and ε_j is independent of X_k for $k \in \Pi_j$. The goal is to estimate the structure of $\mathcal{G}^*$ from X, which is equivalent to estimating the support of B^*.

A key assumption under (1) is that the rows of X are jointly independent as the covariance matrix of each ε_j is diagonal. Under such i.i.d. assumption, many structure learning algorithms for DAGs have been developed, which can be largely categorized into three groups: score-based, constraint-based, and hybrid of the two. Score-based methods search for the optimal DAG by maximizing a scoring function such as minimum description length (Roos, 2017), BIC (E. Schwarz, 1978), and Bayesian scores (Heckerman et al., 1995; Cooper and Herskovits, 1992) with various search strategies, such as order-based search (Scanagatta et al., 2016; Schmidt et al., 2007; Ye et al., 2020), greedy search (Ramsey et al., 2017; Chickering, 2003), and coordinate descent (Fu and Zhou, 2013; Aragam and Zhou, 2015; Gu et al., 2019). Constraint-based methods, such as the PC algorithm in Spirtes et al. (2000), perform conditional independence tests among variables to construct a skeleton and then proceed to orient some of the edges. There are also hybrid methods such as in Tsamardinos et al. (2006) and Gasse et al. (2014) that combine the above two approaches.

In real applications, however, it is common for observations to be dependent as in network data, which violates the i.i.d assumption for the aforementioned methods. For example, when modeling the characteristics of an individual in a social network, the observed characteristics from different individuals can be dependent because they belong to the same social group such as friends, family and colleagues who often share similar features. Another example appears when modeling a gene regulatory network from individuals that are potentially linked genetically. When estimating brain functional networks, we often have a matrix of fMRI measurements for each individual, $X \in \mathbb{R}^{T\times\nu}$, across T time points and ν brain regions of interests. The existence of correlations across both time points and brain regions often renders the estimates from standard graphical modeling methods inaccurate (Kundu and Risk, 2020). Motivated by these applications, we are interested in developing a Gaussian DAG model that can take into account the dependence between observations. Based on this model, we will develop a learning algorithm that can simultaneously infer the DAG structure and the sample dependencies. Moreover, since many real-world networks are sparse, we also want our method to be able to learn a sparse DAG and scale to large number of vertices. A sparsity constraint on the estimated DAG can also effectively prevent over-fitting and greatly improve the computational efficiency. Lastly, we would like to have theoretical guarantees on the consistency and finite-sample accuracy of the estimators. With these requirements in mind, we seek to

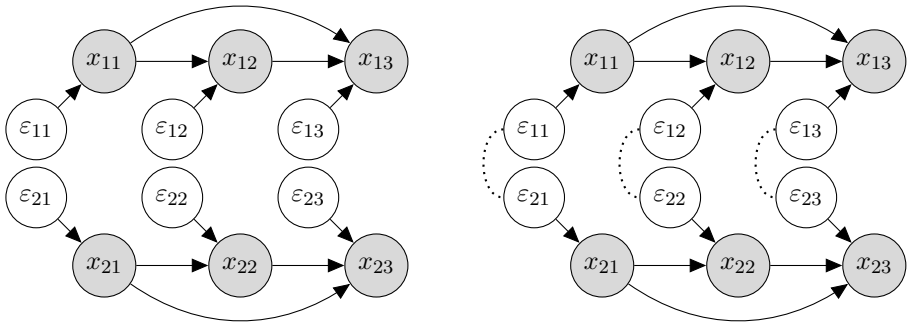
1. Develop a novel Bayesian network model for network data;
2. Develop a method that can jointly estimate a sparse DAG and the sample dependencies under the model;
3. Establish finite-sample error bound and consistency of our estimators;
4. Achieve good empirical performance on both synthetic and real data sets.

Because X is defined by the Gaussian noise vectors ε_j according to the structural equations in (1), dependence among the rows of X may be introduced by modeling the covariance

structure among the variables $\varepsilon_{1j}, \ldots, \varepsilon_{nj}$ in ε_j. Based on this observation, we will use an undirected graph G^* to define the sparsity pattern in the precision matrix of ε_j. When G^* is an empty graph, the variables in ε_j are independent as in the classical Gaussian DAG model. However, when G^* is not empty, X follows a more complex matrix normal distribution, and the variance is defined by the product of two covariance matrices, one for the DAG $\mathcal{G}^*$ and the other for the undirected graph G^*. As a result, estimating the structure of the DAG $\mathcal{G}^*$ as well as other model parameters under the sparsity constraints in both graphs is a challenging task. We will start off by assuming that a topological ordering π^* of $\mathcal{G}^*$ is given so that the search space for DAGs can be largely reduced. However, due to the presence of the second graph for network data, the usual likelihood-based objective function used in traditional score-based methods is non-convex. The constraint-based methods do not naturally extend to network data either due to the dependence among the individuals in X, which complicates the conditional independence tests. In order to find a suitable objective function and develop an optimization algorithm, we exploit the biconvex nature of a regularized likelihood score function and develop an effective blockwise coordinate descent algorithm with a nice convergence property. If the topological ordering of the DAG is unknown, it is impossible to identify a unique DAG from data due to the Markov equivalence of DAGs (Chickering, 2003). Moreover, due to the lack of independence, it is very difficult to estimate the equivalence class defined by $\mathcal{G}^*$. In this case, we take advantage of an invariance property of the matrix normal distribution. Under some sparsity constraint on $\mathcal{G}^*$, we show that even with a random ordering, we can still get a good estimate of the covariance of ε_j, which can be used to decorrelate X so that existing DAG learning algorithms can be applied to estimate an equivalence class of $\mathcal{G}^*$.

The remainder of the paper is structured as follows. In Section 2 we introduce a novel Gaussian DAG model for network data and discuss its connections with some existing models. We propose a structural learning algorithm for the model and go through its details in Section 3. Section 4 is devoted to our main theoretical results, as well as their implications under various high-dimensional asymptotic settings. Section 5 reports numerical results of our method with detailed comparisons with some competing methods on simulated data. Section 6 presents an application of our method on a real single-cell RNA sequencing data set. All proofs are deferred to the Appendix.

Notations For the convenience of the reader, we now summarize some notations to be used throughout the paper. We write $\mathcal{G}^*$ and G^* for the true DAG and the true undirected graph, respectively. Let $\Omega^* := \mathrm{diag}(\omega_j^{*2})$ be a $p \times p$ diagonal matrix of error variances, B^* denote the true WAM of $\mathcal{G}^*$, and $s := \sup_j \|\beta_j^*\|_0$ denote the maximum number of parents of any node in $\mathcal{G}^*$. Furthermore, X_j denotes the jth column of X for $j = 1, \ldots, p$, and x_i denotes the ith row of X for $i = 1, \ldots, n$. Given two sequences f_n and g_n, we write $f_n \lesssim g_n$ if $f_n = O(g_n)$, and $f_n \asymp g_n$ if $f_n \lesssim g_n$ and $g_n \lesssim f_n$. Denote by $[p]$ the index set $\{1, \ldots, p\}$. For $x \in \mathbb{R}^n$, we denote by $\|x\|_q$ its ℓ_q norm for $q \in [0, \infty]$. For $A \in \mathbb{R}^{n \times m}$, $\|A\|_2 = \sup_v \{\|Av\|_2 : \|v\|_2 \leq 1, v \in \mathbb{R}^m\}$ is the operator norm of A, $\|A\|_f$ is the Frobenius norm of A, $\|A\|_\infty = \max_{i,j} |a_{ij}|$ is the element-wise maximum norm of A, and $|||A|||_\infty = \max_{i \in [n]} \sum_{j=1}^m |a_{ij}|$ is the maximum row-wise ℓ_1 norm of A. Denote by $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$, respectively, the smallest and the largest singular values of a matrix A. Let $|S|$ be the size of a set S.



(a) Markovian DAG model (b) Semi-Markovian DAG model

Figure 1: Graphical representations of the models in (1) and (2).

2. A Novel DAG Model for Dependent Data

We model sample dependency through an undirected graph G^* on n vertices, with each vertex representing an observation $x_i, i \in [n]$, and the edges representing the conditional dependence relations among them. More explicitly, let $A(G^*)$ be the adjacency matrix of G^* so that

$$A(G^*)_{ij} = 0 \Rightarrow x_i \perp\!\!\!\perp x_j | x_{\setminus\{i,j\}}, \quad \forall\, i \neq j.$$

Suppose we observe not only the dependent samples $\{x_i\}_{i=1}^n$ but also the graph (network) G^*. We generalize the structural equation model (SEM) in (1) to

$$X_j = \sum_{k \in \Pi_j} \beta_{kj}^* X_k + \varepsilon_j, \quad \varepsilon_j = (\varepsilon_{1j}, \ldots, \varepsilon_{nj}) \sim \mathcal{N}_n\left(0, \omega_j^{*2} \Sigma^*\right), \tag{2}$$

where $\Sigma^* \in \mathbb{R}^{n \times n}$ is positive definite. The support of the precision matrix $\Theta^* = (\Sigma^*)^{-1}$ is restricted by $\mathrm{supp}(\Theta^*) \subseteq A(G^*)$. We fix $\omega_1^* = 1$ so that the model is identifiable. Note that when $\Sigma^* = I_n$, the SEM (2) reduces to (1). Hence, the classical Gaussian DAG model in (1) is a special case of our proposed model (2). Under the more general model (2), we are facing a more challenging structural learning problem: Given dependent data X generated from a DAG $\mathcal{G}^*$ and the undirected graph G^* that encodes the sample dependencies, we want to estimate the DAG coefficients B^*, the noise variance $\Omega^* = \mathrm{diag}(\omega_j^{*2})$, and the precision matrix Θ^* of the samples. Before introducing our method, let us look at some useful properties of model (2) first.

2.1 Semi-Markovian Model

The distinction between (1) and (2) becomes more clear when we regard (2) as a semi-Markovian causal model (Tian et al., 2006). Following its causal reading (Pearl, 1995), we can represent each variable z_i in a DAG $\mathcal{G}$ on vertices $\{z_1, \ldots, z_p\}$ using a deterministic function:

$$z_i = f_i(\Pi_i, u_i), \quad i \in [p], \tag{3}$$

where Π_i is the set of parents of node z_i in G and u_i are noises, sometimes also referred to as background variables. The model (3) is *Markovian* if the noise variables u_i are jointly

independent, and it is *semi-Markovian* if they are dependent. Now for a data matrix X with $n = 2$ and $p = 3$, consider the DAG models defined, respectively, by (1) and (2) over all six random variables $x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}$. Under SEM (1) we model $x_1 = (x_{11}, x_{12}, x_{13})$ and $x_2 = (x_{21}, x_{22}, x_{23})$ using the same SEM and assume they are independent, as shown in Figure 1a.[1] In contrast, the model proposed in (2) allows observations to be dependent by relaxing the independence assumption between ε_{1k} and ε_{2k}, $k = 1, 2, 3$. If we use dashed edges to link correlated background variables, then we arrive at a semi-Markovian DAG model as shown in Figure 1b. In general, the variables $x_{i1}, \ldots, x_{ip}$ in each individual under the semi-Markovian model satisfy the same conditional independence constraints defined by a DAG, while the background variables $\varepsilon_{1j}, \ldots, \varepsilon_{nj}$ across the n individuals are dependent. When estimating the DAG structure with such data, the correlations among individuals will reduce the effective sample size. Therefore, we need to take the distribution of the correlated ε_i into account.

2.2 Matrix Normal Distribution

Our model (2) defines a matrix normal distribution for X. To see this, note that $\varepsilon = (\varepsilon_{ij})_{n\times p}$ in (2) follows a matrix normal distribution:

$$\varepsilon \sim \mathcal{N}_{n,p}(0, \Sigma^*, \Omega^*) \Leftrightarrow \mathrm{vec}(\varepsilon) \sim \mathcal{N}_{np}(0, \Omega^* \otimes \Sigma^*),$$

where $\mathrm{vec}(\cdot)$ is the vectorization operator and $\otimes$ is the Kronecker product. Then, the random matrix X satisfies

$$X \sim \mathcal{N}_{n,p}(0, \Sigma^*, \Psi^*), \tag{4}$$

where $\Psi^* = (I - B^*)^{-\top} \Omega^* (I - B^*)^{-1}$. From the properties of a matrix normal distribution, we can easily prove the following lemma which will come in handy when estimating the row covariance matrix Σ^* from different orderings of nodes. Given a permutation π of the set $[p]$, define P_π as the permutation matrix such that $hP_\pi = (h_{\pi^{-1}(1)}, \ldots, h_{\pi^{-1}(p)})$ for any row vector $h = (h_1, \ldots, h_p)$.

Lemma 1 *If X follows the model (2), then for any permutation π of $[p]$ we have*

$$XP_\pi \sim \mathcal{N}_{n,p}(0, \Sigma^*, P_\pi^\top \Psi^* P_\pi).$$

Although matrix normal distributions have been studied extensively in the past, the structural learning problem we consider here is quite unique. First of all, previous studies on matrix normal model usually assume we observe m copies of X and the MLE exists when $m \geq \max\{p/n, n/p\} + 1$ (Dutilleul, 1999). In our case, we only observe one copy of X and thus the MLE does not exist without additional sparsity constraints. Allen and Tibshirani (2010) proposed to use ℓ_1 regularization to estimate the covariance matrices when $m = 1$, but the estimation relies on the assumption that the model is transposable, meaning that the two components (Σ, Ψ) of the covariance are symmetric and can be estimated in a symmetric fashion. In model (2), however, the two covariance components have different structural constraints and cannot be estimated in the same way. Lastly, practitioners

1. Independent background variables are often omitted in the graph, but we include them here to better illustrate the differences between the two models.

are often interested in estimating large Bayesian networks with hundreds or more nodes under certain sparsity assumptions on the WAM B. For example, for methods that minimize a score function to estimate the covariances, adding a sparsity regularization term on $\Psi = (I - B)^{-\top}\Omega(I - B)^{-1}$ to the score function does not necessarily lead to a sparse estimate of B. In this paper, we propose a new DAG estimation method under the assumption that both the underlying undirected network among individuals and the Bayesian network are sparse. We are not interested in estimating Ψ but a sparse factorization of Ψ represented by the WAM B. This would require imposing sparsity constraints on B itself instead of on Ψ. This is different from the recent work by Tsiligkaridis et al. (2013), Allen and Tibshirani (2010), and Zhou (2014) on the Kronecker graphical lasso.

2.3 Score-equivalence

The likelihood function of the proposed model (2) also satisfies the desired score-equivalence property. To see this, let $\beta_j = (\beta_{1j}, \ldots, \beta_{pj})^\top$ be the jth column of the WAM B. Define an $n \times n$ sample covariance matrix of $\varepsilon_1/\omega_1, \ldots, \varepsilon_p/\omega_p$ from X as

$$S(\Omega, B) = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{\omega_j^2}(X_j - X\beta_j)(X_j - X\beta_j)^\top. \tag{5}$$

Then the negative log-likelihood $L(B, \Omega, \Theta \mid X)$ from (2) is given by

$$2L(B, \Omega, \Theta \mid X) = n\log\det\Omega - p\log\det\Theta + p\,\mathrm{tr}(\Theta S(\Omega, B)). \tag{6}$$

Due to the dependence among observations, it is unclear whether the well-known score-equivalence property for Gaussian DAGs (Chickering, 2003) still holds for our model. Let $(\widehat{B}(\mathcal{G}), \widehat{\Omega}(\mathcal{G}), \widehat{\Theta}(\mathcal{G}))$ denote the MLE of (B, Ω, Θ) given a DAG $\mathcal{G}$ and the support restriction on Θ. Then, the following theorem confirms the score-equivalence property for our DAG model.

Theorem 2 *(Score equivalence) Suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are two Markov equivalent DAGs on the same set of p nodes. If the MLEs $(\widehat{B}(\mathcal{G}_m), \widehat{\Omega}(\mathcal{G}_m), \widehat{\Theta}(\mathcal{G}_m))$, $m = 1, 2$, exist for the matrix $X = (x_{ij})_{n\times p}$, then*

$$L(\widehat{B}(\mathcal{G}_1), \widehat{\Omega}(\mathcal{G}_1), \widehat{\Theta}(\mathcal{G}_1) \mid X) = L(\widehat{B}(\mathcal{G}_2), \widehat{\Omega}(\mathcal{G}_2), \widehat{\Theta}(\mathcal{G}_2) \mid X).$$

This property justifies the evaluation of estimated DAGs using common model selection criterion such as AIC and BIC. For examples, we show in Section 5 that one can use BIC scores to select the optimal penalty level for our proposed DAG estimation algorithm.

3. Methods

We have discussed the properties of our novel DAG model for dependent data and the unique challenges faced by the structural learning task. In this section, we develop a new method to estimate the parameters in model (2). Our estimator is defined by the minimizer of a score function that derives from a penalized log-likelihood. In order to explain our method, let us start from the penalized negative log-likelihood function:

$$f(B, \Omega, \Theta) := 2L(B, \Omega, \Theta \mid X) + \rho_1(B) + \rho_2(\Theta), \quad B \in \mathcal{D}, \tag{7}$$

where $\mathcal{D}$ is the space of WAMs for DAGs and ρ_1 and ρ_2 are some penalty functions. This loss function is difficult to minimize due to the non-convexity of L and the exponentially large search space of DAGs. One way to reduce the search space is to assume a given topological ordering. A *topological ordering* of a DAG $\mathcal{G}$ with p vertices is a permutation π of indices $(1, \ldots, p)$ such that for every edge $(i, j) \in E(\mathcal{G})$, $\pi^{-1}(i) < \pi^{-1}(j)$. Recall that a WAM B is defined as $(\beta_{kj})_{p\times p}$ such that $\beta_{kj} \neq 0$ if and only if $(k, j) \in E(\mathcal{G})$; therefore, given a topological ordering π, we can define a set $\mathcal{D}(\pi)$ of WAMs compatible to π such that all $B \in \mathcal{D}(\pi)$ are strictly upper triangular after permuting its rows and columns according to π. Given a topological ordering π, the loss function (7) becomes

$$f(B, \Omega, \Theta) = -p \log\det\Theta + \sum_{j=1}^{p} \frac{1}{\omega_j^2} \|LX_j - LX\beta_j\|_2^2 + \rho_1(B) + \rho_2(\Theta), \quad B \in \mathcal{D}(\pi), \tag{8}$$

where L is the Cholesky factor of Θ (i.e. $\Theta = L^\top L$). If $\rho_1(\cdot)$ and $\rho_2(\cdot)$ are convex loss functions and the noise covariance matrix $\Omega = \mathrm{diag}(\omega_j^2)$ is known, (8) will be a bi-convex function in (B, Θ), which can be minimized using iterative methods such as coordinate descent. Tseng (2001) showed that the coordinate descent algorithm in bi-convex problems converges to a stationary point. Inspired by this observation, we propose the following two-step algorithm:

Step 1: Pre-estimate Ω^* to get $\widehat{\Omega} = \mathrm{diag}(\hat{\omega}_j^2)$.

Step 2: Estimate $\widehat{B}$ and $\widehat{\Theta}$ by minimizing a biconvex score function derived from the penalized negative log-likelihood conditioning on $\hat{\omega}_j$.

Many existing noise estimation methods for high-dimensional linear models can be used to estimate $\widehat{\Omega}$ in Step 1 such as scaled lasso/MCP (Sun and Zhang, 2012), natural lasso (Yu and Bien, 2019), and refitted cross-validation (Fan et al., 2012). We will present the natural estimator of Ω and discuss a few other alternatives in Section 3.2. Importantly, the statistical properties of the chosen estimator $\widehat{\Omega}$ in Step 1 will affect the properties of the $\widehat{\Theta}$ and $\widehat{B}$ we get in Step 2, and thus we must choose the estimator carefully. We leave the detailed discussion of the theoretical properties of $\widehat{\Omega}$ and their implications to Section 4. Suppose $\widehat{\Omega}$ is given, we propose the following estimator for Step 2:

$$\begin{aligned}\left(\widehat{\Theta}, \widehat{B}(\pi)\right) = \underset{\Theta\succ 0, B\in\mathcal{D}(\pi)}{\arg\min} \Bigg\{ & -p \log\det\Theta + \sum_{j=1}^{p} \frac{1}{\hat{\omega}_j^2} \|LX_j - LX\beta_j\|_2^2 \\ & + \frac{\lambda_1}{\hat{\omega}_j^2}\|\beta_j\|_1 + \lambda_2\|\Theta\|_1 \Bigg\}.\end{aligned} \tag{9}$$

The ℓ_1 regularization on $\beta_j/\hat{\omega}_j^2$ not only helps promote sparsity in the estimated DAG but also prevents the model from over-fitting variables that have small variances. The ℓ_1 regularization on Θ ensures that the estimator is unique and can improve the accuracy of $\widehat{\Theta}$ by controlling the error carried from the previous step. We will discuss how to control the estimation errors in more detail in Section 4.

In Section 3.1, we assume a topological ordering π^* of the true DAG $\mathcal{G}^*$ is known. In this case, we will order the columns of X according to π^* so that for each j, only the first $j-1$ entries in β_j can be nonzero. When minimizing (9), we fix $\beta_{jk} = 0$ for $k \geq j$ and the resulting $\widehat{B}$ is guaranteed to be upper-triangular. If π^* is unknown, we show in Section 3.3 how the score function in (9) is still useful for estimating Θ^* and describe a method of de-correlation so that standard DAG learning methods can be applied on the de-correlated data.

3.1 Block Coordinate Descent

We denote an estimate of the true precision matrix Θ^* at iteration t by $\widehat{\Theta}^{(t)}$. We also write $\widehat{L}^{(t)}$ and L^* for the Cholesky factors of the $\widehat{\Theta}^{(t)}$ and Θ^*, respectively. Since (9) is biconvex, it can be solved by iteratively minimizing over Θ and B, i.e., using block coordinate descent. Consider the tth iteration of block coordinate descent. Fixing $\widehat{\Theta}^{(t)}$, the optimization problem in (9) becomes the standard Lasso problem (Tibshirani, 1996) for each j:

$$\hat{\beta}_j^{(t+1)} = \arg\min_{\beta_j} \frac{1}{2n} \|\widehat{L}^{(t)} X_j - \widehat{L}^{(t)} X \beta_j\|_2^2 + \lambda_n \|\beta_j\|_1, \quad \lambda_n = \lambda_1/(2n), \tag{10}$$

where $\widehat{\Theta}^{(t)} = \widehat{L}^{(t)\top} \widehat{L}^{(t)}$ is the Cholesky decomposition. Since the columns of X are ordered according to π, we can set $\hat{\beta}_{ij}^{(t+1)} = 0$ for $i = j, j+1, \ldots, p$ and reduce the dimension of feasible β_j in (10) to $j-1$. In particular, $\hat{\beta}_1^{(t+1)}$ is always a zero vector. Fixing $\widehat{B}^{(t+1)}$, solving for $\widehat{\Theta}^{(t+1)}$ is equivalent to a graphical Lasso problem with fixed support (Ravikumar et al., 2011)

$$\widehat{\Theta}^{(t+1)} = \arg\min_{\Theta \succ 0,\ \mathrm{supp}(\Theta) \subseteq A(G^*)} -\log\det\Theta + \mathrm{tr}(\widehat{S}^{(t+1)}\Theta) + \lambda_p \|\Theta\|_1, \tag{11}$$

where $\widehat{S}^{(t+1)} = S(\widehat{\Omega}, \widehat{B}^{(t+1)})$ and $\lambda_p = \lambda_2/p$. The details of the method are given in Algorithm 1.

Algorithm 1: Block coordinate descent (BCD) algorithm

Input : X, $\Theta^{(0)}$, $\widehat{\Omega}$, ρ, $A(G^*)$, T

while $\max\left\{\|\widehat{\Theta}^{(t+1)} - \widehat{\Theta}^{(t)}\|_f, \|\widehat{B}^{(t+1)} - \widehat{B}^{(t)}\|_f\right\} > \rho$ *and* $t < T$ **do**

  for $j = 1, \ldots, p$ **do**

    $\hat{\beta}_j^{(t+1)} \leftarrow$ Lasso regression (10)

  end

  $\widehat{\Theta}^{(t+1)} \leftarrow$ graphical Lasso with support restriction (11)

  $t \leftarrow t+1$

end

Output : $\widehat{B} \leftarrow \widehat{B}^{(t)}, \widehat{\Theta} \leftarrow \widehat{\Theta}^{(t)}$

As shown in Proposition 3, Algorithm 1 will converge to a stationary point of the objective function (9). The stationary point here is defined as a point where all directional directives are nonnegative (Tseng, 2001).

Proposition 3 *Let* $\{(\widehat{B}^{(t)}, \widehat{\Theta}^{(t)}) : t = 1, 2, \ldots\}$ *be a sequence generated by the block coordinate descent Algorithm 1 for any* $\lambda_1, \lambda_2 > 0$. *Then for almost all* $X \in \mathbb{R}^{n \times p}$, *every cluster point of* $\{(\widehat{B}^{(t)}, \widehat{\Theta}^{(t)})\}$ *is a stationary point of the objective function in* (9).

3.2 A Natural Estimator of Ω

We restrict our attention mostly to sparse undirected graphs G consisting of N connected components, which implies that the row precision matrix Θ is block-diagonal:

$$\Theta = \begin{pmatrix} \Theta_1 & & & \\ & \Theta_2 & & \\ & & \ddots & \\ & & & \Theta_N \end{pmatrix}. \tag{12}$$

The support of Θ inside each diagonal block Θ_i could be dense. This type of network is often seen in applications where individuals in the network form clusters: nodes in the same cluster are densely connected and those from different clusters tend to be more independent from each other. The underlying network G will be sparse if the individuals are from a large number of small clusters. In other words, the sparsity of G depends primarily on the number of diagonal blocks in Θ. More general network structures also are considered in the numerical experiments in Section 5.

Given the block-diagonal structure of Θ in (12), there are a few ways to estimate Ω. We use the *natural estimator* introduced by Yu and Bien (2019). We estimate $\hat{\omega}_j^2$ using independent samples in X according to the block structure of Θ^*. Let $B \subseteq [n]$ be a row index set and A^B denote the submatrix formed by selecting rows from a matrix $A_{n \times m}$ with row index $i \in B$. We draw one sample from each block and form a smaller $N \times p$ design matrix X^B. It is not difficult to see that $X_j^B = X^B \beta_j^* + \varepsilon_j^B$. Next define the natural estimator of ω_j^{*2} for $j \in [p]$ as in Yu and Bien (2019):

$$\hat{\omega}_j^2 = \min_{\beta_j} \left\{ \frac{1}{N} \|X_j^B - X^B \beta_j\|_2^2 + 2\lambda_N \|\beta_j\|_1 \right\}, \tag{13}$$

where $\lambda_N > 0$ is a tuning parameter. In Section 4, we discuss the estimation error rate of $\widehat{\Omega}$. Alternative methods, such as scaled lasso (Sun and Zhang, 2012) and the Stein's estimator (Bayati et al., 2013), can also be used to estimate ω_j^2.

3.3 Estimating DAGs with Unknown Ordering

Given any permutation π of $[p]$, there exists a DAG $\mathcal{G}_\pi$ such that (i) π is a topological sort of $\mathcal{G}_\pi$ and (ii) the joint distribution $\mathbb{P}$ of the p random variables factorizes according to $\mathcal{G}_\pi$. Under the assumption that the true DAG $\mathcal{G}^*$ is sparse, i.e., the number of nonzero entries in β_j^* is at most s for all j, for any random ordering π' we choose, the corresponding DAG $\mathcal{G}_{\pi'}$ is also likely to be sparse where the number of parents for each node is less than some positive constant s'. Following this intuition, we can randomly pick a permutation π' for the nodes and apply Algorithm 1 on $X_{\pi'} := XP_{\pi'}$, where $(XP_{\pi'})_{ij} = X_{i\pi'(j)}$. If the sparsity s' is small compared to the sample size n, the estimate $\hat{\beta}'_{ij}$ we get from solving the Lasso problem (10) will be consistent as well (we discuss the error bound on $\hat{\beta}_{ij}$ in details

in Section 4). Moreover, since the covariance Θ^* is invariant to permutations by Lemma 1, the resulting estimate $\widehat{\Theta}$ under the random ordering π' will still be a good estimate of Θ^*. With the Cholesky factor $\widehat{L}$ of $\widehat{\Theta}$, we de-correlate the rows of X and treat

$$\widehat{X} = \widehat{L}X, \tag{14}$$

as the new data. Because the row correlations in $\widehat{X}$ vanish, we can apply existing structure learning methods which require independent observations to learn the underlying DAG. We find that this de-correlation step is able to substantially improve the accuracy of structure learning by well-known state-of-the-art methods, such as the greedy equivalence search (GES) (Chickering, 2003) and the PC algorithm (Spirtes et al., 2000). See Section 5 for more details.

4. Main Theoretical Results

In this section, we present our main theoretical results for Algorithm 1 assuming a true ordering is given. Section 4.1 is devoted to the error bounds of $\widehat{\Omega}$ using the natural estimator. We state our main theorem, Theorem 6, in Section 4.2, along with some important corollaries. Finally, in Section 4.3, we compare the error rates of our estimators with those in related problems. Before we start, let us introduce some additional notations used in this section.

Notations Let the errors of $\widehat{L}$ and $\widehat{\Theta}$ be defined as $\widehat{\Delta}_{chol} := \widehat{L} - L^*$ and $\widehat{\Delta}_{prec} := \widehat{\Theta} - \Theta^*$. Let $\widehat{\Delta}_j := \hat{\beta}_j - \beta_j^* \in \mathbb{R}^p$ denote the estimation error of the jth column of B^*. Let

$$\bar{\beta} = \sup_{1\le i,j\le p} |\beta_{ij}^*|, \quad \bar{\omega} = \sup_{1\le j\le p} \omega_j^*, \quad \bar{\psi}^2 = \sup_{1\le j\le p} \Psi_{jj}^*,$$

where $\Psi^* = (I - B^*)^{-\top}\Omega^*(I - B^*)^{-1}$. In the proofs, we also use $X_{i\cdot}$ and $X_{\cdot j}$ to denote the ith row and jth column of X, respectively. Let $\widetilde{X} = L^*X \sim \mathcal{N}(0, \Psi^* \otimes I_n)$ and $\tilde{\varepsilon} = L^*\varepsilon$. Then the rows of $\widetilde{X}$, i.e. $\tilde{x}_i, i \in [n]$, are i.i.d from $\mathcal{N}(0, \Psi^*)$. Let m denote the maximum degree of the undirected graph G^*, which is allowed to grow with n. Following the setup in Ravikumar et al. (2011), the set of non-zero entries in the precision matrix is denoted as $\text{supp}(\Theta^*) := \{(i,j) \mid \Theta_{ij}^* \neq 0\}$. Let us use the shorthand S and S^c to denote the support and its complement in the set $[n] \times [n]$, respectively. Define the following constants:

$$\begin{aligned} \kappa_{\Sigma^*} &:= |||\Sigma^*|||_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |\Sigma_{ij}^*|, \\ \Gamma_{SS}^* &:= \left[\Theta^{*-1} \otimes \Theta^{*-1}\right]_{SS} \in \mathbb{R}^{(|S|+n)\times(|S|+n)}, \\ \kappa_{\Gamma^*} &:= \left|\left|\left|(\Gamma_{SS}^*)^{-1}\right|\right|\right|_\infty. \end{aligned} \tag{15}$$

4.1 Consistency of $\widehat{\Omega}$ under Block-diagonal Θ^*

Recall from Section 3 that $\widehat{\Omega}$ is pre-estimated at Step 1 in our two-step learning procedure. As we will discuss in more detail in Section 4.2, the accuracy of $\widehat{\Theta}$ obtained in Step 2 using Algorithm 1 depends on the accuracy of $\widehat{\Omega}$. Existing methods for estimating the error

variance in linear models, such as the scaled Lasso (Sun and Zhang, 2012), square-root lasso (Belloni et al., 2011), and natural lasso (Yu and Bien, 2019), often assume independence among samples, which is not necessarily true under our network setting. However, if we assume the network of the samples is block diagonal and the samples form many small clusters, we would be able to collect independent samples from different clusters. This intuition suggests that existing methods are readily applicable in our setting to get consistent estimates of Ω^*, as long as there are enough independent clusters in the undirected network G^*.

Formally, suppose Θ^* has a block-diagonal structure defined in (12). Let N be the number of blocks. If we use the *natural estimator* from Yu and Bien (2019) (described in Section 3) to get $\hat{\omega}_j$, then we have the following error bound:

Lemma 4 *Let X be generated from (2) and assume Θ^* is block-diagonal with N blocks. Recall that $s = \sup_j \|\beta_j^*\|_0$. Let $\widehat{\Omega}$ be the natural estimator defined in (13) with*

$$\lambda_N = 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{N}} + \sqrt{\frac{2\log 2 + 4\log p}{N}}\right),$$

then with probability at least $1 + 1/p^2 - 3/p$,

$$\sup_{1\le j\le p}\left|\hat{\omega}_j^2 - \omega_j^{*2}\right| \le \lambda_N s\bar{\beta} + 4\bar{\omega}\sqrt{\frac{\log 2 + \log p}{N}}.$$

Lemma 4 shows that the maximum error of $\hat{\omega}_j$ is upper bounded by $C \cdot s\sqrt{\log p/N}$ where $C > 0$ is constant, and thus is consistent as long as $s\sqrt{\log p/N} \to 0$. When p is fixed and N increases, there are more diagonal blocks in Θ^*, indicating more independent samples, and thus, $\hat{\omega}_j^2$ will be more accurate. When $N = n$, Θ^* becomes a diagonal matrix and the rows of X become i.i.d. Another useful quantity for our subsequent discussion is the estimation error of $1/\hat{\omega}_j^2$. Let $r(\widehat{\Omega})$ be defined as:

$$r(\widehat{\Omega}) := \sup_{1\le j\le p}\left|\frac{1}{\hat{\omega}_j^2} - \frac{1}{\omega_j^{*2}}\right|. \tag{16}$$

We can easily show that the following lemma holds:

Lemma 5 *Suppose $\frac{1}{2}\inf_{1\le j\le p}\omega_j^{*2} - \sup_{1\le j\le p}\left|\hat{\omega}_j^2 - \omega_j^{*2}\right| > b > 0$. Then*

$$r(\widehat{\Omega}) \le \frac{1}{b^4}\sup_{1\le j\le p}\left|\hat{\omega}_j^2 - \omega_j^{*2}\right|.$$

When $s(\log p/N)^{1/2}$ is small enough, Lemma 4 implies that b can be taken as a positive constant with high probability.

4.2 Error Bounds and Consistency of $\widehat{B}^{(1)}$ and $\widehat{\Theta}^{(1)}$

Applying Algorithm 1 with a pre-estimated $\widehat{\Omega}$ as input gives us $\widehat{B}^{(t)}$ and $\widehat{\Theta}^{(t)}$. In this section, we study the error bounds and consistency of $\widehat{B}^{(t)}$ and $\widehat{\Theta}^{(t)}$. Although Algorithm 1 is computational efficient and has a desirable convergence behavior as described in Proposition 3, there are technical difficulties in establishing the consistency of $\widehat{B}^{(\infty)}$ and $\widehat{\Theta}^{(\infty)}$ after convergence, due to the dependence between $(\widehat{B}^{(t)}, \widehat{\Theta}^{(t)})$ across iterations. However, we show that as long as we have a suitable initial estimate satisfying Assumption 1, Algorithm 1 can produce consistent estimators after one iteration, i.e., $(\widehat{B}^{(1)}, \widehat{\Theta}^{(1)})$ is consistent.

Assumption 1 *There exists a constant $0 < M \leq \sigma^2_{\min}(L^*)$, such that the initial estimate $\widehat{\Theta}^{(0)}$ satisfies*

$$\|\widehat{\Theta}^{(0)} - \Theta^*\|_2 \leq M.$$

Assumption 1 states that the initial estimate $\widehat{\Theta}^{(0)}$ is inside an operator norm ball centered at the true parameter Θ^* with a radius smaller than a constant M. The constant M is less than or equal to the smallest eigenvalue of Θ^*. Assumption 1 may not be easy to verify in practice, but since it only requires $\widehat{\Theta}^{(0)}$ to be within an ℓ_2-ball of constant radius around Θ^*, it is not difficult for Assumption 1 to be met if Θ^* is sparse and normalized. Under Assumption 1 we can establish finite-sample error bounds for $(\widehat{B}^{(1)}, \widehat{\Theta}^{(1)})$. Recall that m denotes the maximum degree of the undirected graph G^* and $s = \sup_{1 \leq j \leq p} \|\beta^*_j\|_0$. Define

$$\bar{R}(s, p, n) := \max\left\{6\bar{\omega} r(\widehat{\Omega}), \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p \log(\max\{n,p\})^2}{n}}\right\}, \tag{17}$$

which depends on $r(\widehat{\Omega})$, the error of $\widehat{\Omega}$ defined in (16).

Theorem 6 *Consider a sample matrix X from model (2). Let $\widehat{\Theta}^{(1)}, \widehat{B}^{(1)}$ be the estimates after one iteration of the Algorithm 1, given initial estimator $\widehat{\Theta}^{(0)}$ satisfying Assumption 1. Suppose $b > 0$ as defined in Lemma 5. Pick the regularization parameters in (10) and (11) such that*

$$\lambda_n \geq 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{n}} + \sqrt{\frac{2\log 2 + 4\log p}{n}}\right),$$
$$\lambda_p \geq 40\sqrt{2}\sqrt{\frac{\tau \log n + \log 4}{p}} + \bar{R}(s, p, n),$$

where $\tau > 2$ and $r(\widehat{\Omega})$ is defined in (16). Let $\bar{\kappa} = \sigma_{\min}(\Psi^*)$. Then for some positive constant c_1, we have

$$\sup_j \|\hat{\beta}^{(1)}_j - \beta^*_j\|_2 \leq \frac{\sqrt{s}}{c_1 \bar{\kappa}}\lambda_n,$$

with probability at least $(1-2/p)^2 - \left\{1/(\exp\{n/32\}-1) + 1/n^{\tau-2} + 5n^2/\max\{n,p\}^4\right\}$. *If in addition* n, p *satisfy*

$$3200\log(4n^\tau)\max\{160, 24mC\} \le p,$$
$$\max\left\{r(\widehat{\Omega}), \frac{4s\bar{\omega}^3}{b}\sqrt{\frac{\log p \log^2\max\{n,p\}}{n}}\right\} \le 1/(24C), \tag{18}$$

where $C = \max\left\{\kappa_{\Sigma^*}\kappa_{\Gamma^*}, \kappa_{\Sigma^*}^3\kappa_{\Gamma^*}^2\right\}$, *we also have*

$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 \le 4\kappa_{\Gamma^*} m\lambda_p,$$

with the same probability.

We leave the detailed proof for Theorem 6 to the Appendix. The quantities κ_{Γ^*} and κ_{Σ^*} defined in (15) measure, respectively, the scale of the entries in Σ^* and the inverse Hessian matrix Γ_{SS}^{*-1} of the graphical Lasso log-likelihood function (11), and they may scale with n and p in Theorem 6. To simplify the following asymptotic results, we assume they are bounded by a constant as $n, p \to \infty$; see Ravikumar et al. (2011) for a related discussion. In addition, assume $\bar{\kappa}, \bar{\psi}, \bar{\omega}$ stay bounded as well. Then, under the conditions in Theorem 6, we have for fixed positive constants c_2, c_3, c_4 that

$$\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 \le c_2 s\frac{\log p}{n},$$
$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 \le c_3 m\left(\sqrt{\frac{\tau\log n}{p}} + \max\left\{r(\widehat{\Omega}), c_4 s\sqrt{\frac{\log p\log^2\max\{n,p\}}{n}}\right\}\right), \tag{19}$$

with high probability. For simplicity, we assume that Θ^* consists of N blocks as in (12) hereafter. If $\widehat{\Omega}$ satisfies the convergence rate specified in Lemmas 4 and 5, i.e., $r(\widehat{\Omega}) \lesssim s\sqrt{\log p/N}$, then the sample constraints in (18) are satisfied as long as

$$m\log n \lesssim p, \quad s^2\log p \lesssim N, \quad s^2\log^3\max\{n,p\} \lesssim n. \tag{20}$$

As a result, we have the following two asymptotic results. The first one considers the scaling $p \gg n$ under which DAG estimation is high-dimensional. The second one considers the case $n \gg p$ so that the estimation of Θ^* is a high-dimensional problem.

Corollary 7 *Suppose the sample size and the number of blocks satisfy*

$$p \gg N\log^2 p \gtrsim n \gtrsim N \gg \log p \to \infty.$$

Assume $\bar{\beta}, \bar{\omega}, \bar{\psi}, \kappa_{\Gamma^*}, \kappa_{\Sigma^*} < \infty$ *as* $n, p \to \infty$ *and* $r(\widehat{\Omega}) \lesssim s\sqrt{\log p/N}$. *Then under the same assumptions as Theorem 6, we have*

$$\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 = O_p\left(s\frac{\log p}{n}\right),$$
$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 = O_p\left(ms\sqrt{\frac{\log^3 p}{n}}\right).$$

Corollary 8 *Suppose the sample size and block numbers satisfy*

$$n \gg s^2 p \log p \log n \gtrsim N \gtrsim s^2 p \to \infty.$$

Assume $\bar{\beta}, \bar{\omega}, \bar{\psi}, \kappa_{\Gamma^*}, \kappa_{\Sigma^*} < \infty$ *as* $n, p \to \infty$ *and* $r(\widehat{\Omega}) \lesssim s\sqrt{\log p/N}$. *Then under the same assumptions as Theorem 6, we have*

$$\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 = O_p\left(s\frac{\log p}{n}\right),$$

$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 = O_p\left(m\sqrt{\frac{\log n}{p}}\right).$$

Remark 9 *Although we derived the consistency of* $\widehat{\Theta}^{(1)}$ *and* $\widehat{B}^{(1)}$ *in the above two corollaries under the setting where* Θ^* *is block-diagonal, these consistency properties still hold even when* Θ^* *is not block-diagonal. The main purpose of the block-diagonal setting is to provide an example where we can conveniently control the error of* $\widehat{\Omega}$. *But in practice* Θ^* *does not have to be block-diagonal. In particular, we did not assume any block-diagonal structure of* Θ^* *for the error bounds in Theorem 6. It can be seen that the error bound of* $\widehat{B}^{(1)}$ *does not depend on the error of* $\widehat{\Omega}$ *at all. Hence, the accuracy of* $\widehat{\Omega}$ *has no impact on the accuracy of* $\widehat{B}^{(1)}$. *The error bound of* $\widehat{\Theta}^{(1)}$ *in* (19) *is determined by the trade-off among three terms, one of which is the error rate* $r(\widehat{\Omega})$ *as in* (16). *This is supported by our numerical results as well. In Section 5, we demonstrate, with both simulated and real networks where* Θ^* *is not block-diagonal, that our proposed BCD method can still accurately estimate* Θ^* *and* B^* *whenever a relatively accurate* $\widehat{\Omega}$ *is provided.*

4.3 Comparison to Other Results

If the data matrix X consists of i.i.d. samples generated from the Gaussian linear SEM (1), assuming the topological sort of the vertices is known, the DAG estimation problem in (9) is reduced to solving the standard Lasso regression in (10) with $\widehat{L}^{(t)} = I_n$, and thus independent of the initial $\widehat{\Theta}^{(0)}$ estimator. Under the *restricted eigenvalue condition* and letting $\lambda_n \asymp \sqrt{\log p/n}$, it is known the Lasso estimator has the following optimal rate for ℓ_2 error (van de Geer and Bühlmann, 2009):

$$\sup_j \|\hat{\beta}_j - \beta_j^*\|_2^2 = O_p\left(s\frac{\log p}{n}\right).$$

When the data are dependent, Theorem 6 shows that the estimator from Algorithm 1 can achieve the same optimal rate if we make the extra assumptions above. In particular, what we need is a reasonably good initial $\widehat{\Theta}^{(0)}$ estimate such that $\|\widehat{\Theta}^{(0)} - \Theta^*\|_2 \leq M$ for a small positive constant M.

On the other hand, if the underlying DAG is an empty graph and $\Omega^* = I_p$, the problem of estimating Θ^* can be solved using graphical Lasso in (11) because the data (columns in X) are i.i.d. The sample variance $\widehat{S}$ would also be an unbiased estimator of Σ^*. In this case, Ravikumar et al. (2011) showed that

$$\|\widehat{\Theta} - \Theta^*\|_2 = O_p\left(m\sqrt{\frac{\log n}{p}}\right).$$

This results does not require knowing $\text{supp}(\Theta^*)$ but assumes a *mutual incoherence* condition on the Hessian of the log likelihood function. In our case, $\widehat{S}^{(1)}$ is biased due to the accumulated errors from the previous Lasso estimation as well as $\widehat{\Omega}$. As a result, there is an extra bias term $\bar{R}(s,n,p)$ in $\|\widehat{S}^{(1)} - \Sigma^*\|_\infty$ (see Lemma 21 in Appendix A.3) compared to the i.i.d. setting:

$$\|\widehat{S}^{(1)} - \Sigma^*\|_\infty = \bar{\delta}_f(p, n^\tau) + \bar{R}(s,n,p),$$

where $\bar{\delta}_f(p, n^\tau) \asymp \sqrt{\log n/p}$ is the classical graphical Lasso error rate, and

$$\bar{R}(n,p,s) \asymp \max\left\{r(\widehat{\Omega}), s\sqrt{\log p \log\max\{n,p\}/n}\right\}$$

depends on the estimation errors of $\widehat{B}^{(1)}$ and $\widehat{\Omega}$. When $n \gg p$ and $r(\widehat{\Omega})$ is dominated by $\sqrt{\log(n)/p}$, we get the same rate for the ℓ_2 consistency of $\widehat{\Theta}$ (Corollary 8) under slightly more strict constraint on the sample size (20). If $n \ll p$, then the ℓ_2 error rate is determined by $\max\{r(\widehat{\Omega}), s(\log^3 p/n)^{1/2}\}$. Suppose Θ^* is block-diagonal. If the number of blocks N is much smaller than n, then the ℓ_2 rate will be dominated by $r(\widehat{\Omega}) \asymp s\sqrt{\log p/N}$, which could be slower than the optimal graphical Lasso rate. But that is expected due to the error introduced in the DAG estimates $\widehat{B}^{(1)}$ and $\widehat{\Omega}$.

5. Numerical Experiments

Under the assumption that observations generated from a DAG model are dependent, we will evaluate the performance of the block coordinate descent (BCD) algorithm, i.e., Algorithm 1, in recovering the DAG compared to traditional methods that treat data as independent. We expect that the BCD method would give more accurate structural estimation than the baselines by taking the dependence information into account. When a topological ordering of the true DAG is known, we can identify a DAG from data using BCD. When the ordering is unknown, the BCD algorithm may still give an accurate estimate of the row correlations that are invariant to node-wise permutations according to Lemma 1. The estimated row correlation matrix can then be used to de-correlate the data so that traditional DAG learning algorithms would be applicable. We will demonstrate this idea of de-correlation with numerical results as well.

5.1 Simulated Networks

We first perform experiments on simulated networks for both ordered and unordered cases. To apply the BCD algorithm, we need to set values for λ_1 and λ_2 in (9). Since the support of Θ^* is restricted to G^*, we simply fixed λ_2 to a small value ($\lambda_2 = 0.01$) in all the experiments. For each data set, we computed a solution path from the largest $\lambda_{1\max}$, for which we get an empty DAG, to $\lambda_{1\min} = \lambda_{1\max}/100$. The optimal λ_1 was then chosen by minimizing the BIC score over the DAGs on the solution path.

We generated random DAGs with p nodes and fixed the total number of edges s_0 in each DAG to $2p$. The entries in the weighted adjacency matrix B^* of each DAG were drawn uniformly from $[-1,-0.1] \cup [0.1,1]$, and ω_j^*'s were sampled uniformly from $[0.1,2]$. In our simulations of Θ^*, we first considered networks with a clustering structure, i.e., Θ^* was

block-diagonal as in (12). We fixed the size of the clusters to 20 or 30, and within each cluster, the individuals were correlated according to the following four covariance structures.

- Toeplitz: $\Sigma^*_{ij} = 0.3^{|i-j|/5}$.
- Equal correlation: $\Sigma^*_{ij} = 0.7$ if $i \neq j$, and $\Sigma^*_{ii} = 1$.
- Star-shaped: $\Theta^*_{1j} = \Theta^*_{i1} = a, i, j \geq 2, a \in (0,1)$, and $\Theta^*_{ii} = 1$.
- Autoregressive (AR): $\Theta^*_{ij} = 0.7^{|i-j|}$ if $|i-j| \leq \lceil b/4 \rceil$; $\Theta^*_{ij} = 0$ otherwise, where b is the cluster size.

Toeplitz covariance structure implies that the observations are correlated as in a Markov chain. Equal correlation structure represents the cases when all observations are fully connected in a cluster. Star-shaped and AR structures capture intermediate dependence levels. Besides these block-diagonal covariances, we also considered a more general covariance structure defined through *stochastic block models* (SBM), in which G^* consists of several clusters and nodes within a cluster have a higher probability to be connected than those from different clusters. More explicitly, we generated Θ^* as follows:

1. Let $\mathcal{B}_1, \ldots, \mathcal{B}_L$ be L clusters with varying sizes that form a partition of $\{1, \ldots, n\}$, where the number of clusters L ranges from 5 to 10 in our experiments. Define a probability matrix $P \in \mathbb{R}^{n \times n}$ where $P_{ij} = 0.5$ if $i, j \in \mathcal{B}_l, l \in \{1, \ldots, L\}$; otherwise, $P_{ij} = 0.1$.

2. Construct the adjacency matrix A of G^*:

$$A_{ij} \sim \text{Bern}(P_{ij}).$$

3. Sample $\Theta'_{ij} \sim \text{Unif}[-5, 5]$ if $A_{ij} = 1$. Otherwise, $\Theta'_{ij} = 0$. To ensure a positive-definite Θ^*, we then perform the following transformations to get Θ^*:

$$\begin{aligned} \widetilde{\Theta} &= (\Theta' + \Theta'^{\top})/2 \\ \Theta^* &= \widetilde{\Theta} - \Big(\sigma_{\min}(\widetilde{\Theta}) - 0.01\Big) \cdot I_n \end{aligned} \tag{21}$$

Under the stochastic block model, two nodes from different clusters in G^* are connected with probability 0.1, so Θ^* is not block-diagonal in general. As explained in Section 4.2, our proposed BCD algorithm does not require Θ^* to be block-diagonal in practice to produce accurate estimates of B^* and Θ^*. Our numerical experiments will confirm this theory and demonstrate the robustness of the BCD method.

We compared the BCD algorithm with its competitors under both high-dimensional ($p > n$) and low-dimensional ($p < n$) settings with respect to DAG learning. For each (n, p) and each type of covariances, we simulated 10 random DAGs and then generated one data set following equation (2) for each DAG. Thus, we had 10 results for each of the $2 \times 5 = 10$ simulation settings. In the end, we averaged the results over the 10 simulations under each setting for comparison.

5.1.1 Learning with given ordering

This subsection provides additional results for the simulation studies in Section 5.2.1 in the paper. Assuming the nodes in the DAG are sorted according to a given topological ordering, we compared our BCD algorithm against a baseline setting which fixes $\Theta^* = I_n$. In other words, the baseline algorithm ignores the dependencies among observations when estimating the DAG with BCD. The block sizes in Θ^* were set to 20 in all cases except SBM whose block sizes ranged from 5 to 25. Among other estimates, both algorithms return an estimated weighted adjacency matrix $\widehat{B}$ for the optimal λ_1 selected by BIC. For the BCD algorithm, we use $\widehat{B}$ and $\widehat{\Theta}$ for $\widehat{B}^{(\infty)}$ and $\widehat{\Theta}^{(\infty)}$ after convergence (see Algorithm 1). Note that, since $\widehat{\Theta}^{(0)}$ is initialized to I_n by default in the BCD algorithm, the estimated $\widehat{B}$ from the baseline algorithm is the same as the estimate $\widehat{B}^{(1)}$ from BCD after one iteration.

We also included the Kronecker graphical Lasso (KGLasso) algorithm (Allen and Tibshirani, 2010; Tsiligkaridis et al., 2013) mentioned in Section 2 in our comparison, which estimates both $\widehat{\Psi}$ and $\widehat{\Theta}$ via graphical Lasso in an alternating fashion. When estimating Θ^*, KGLasso also makes use of its block-diagonal structure. After KGLasso converges, we perform Cholesky factorization on $\widehat{\Psi} = (I - \widehat{B})^{-\top}\widehat{\Omega}(I - \widehat{B})^{-1}$ according to the given ordering to obtain $\widehat{B}$ and $\widehat{\Omega}$. A distinction between BCD and KGLasso is that KGLasso imposes a sparsity regularization on Ψ instead of B, so the comparison between these two will highlight the importance of imposing sparsity directly on the Cholesky factor.

Given the estimate $\widehat{B}$ from a method, we hard-thresholded the entries in $\widehat{B}$ at a threshold value $\bar{\tau}$ to obtain an estimated DAG. To compare the three methods, we chose $\bar{\tau}$ such that they predicted roughly the same number of edges (E). Then we calculated the number of true positives (TP), false positives (FP) and false negatives (FN, missing edges), and two overall accuracy metrics: Jaccard index (TP / (FP + s_0)) and structural Hamming distances (SHD = FP+FN). Note that, there were no reserved edges (i.e., estimated edges whose orientation is incorrect) because the ordering of the nodes was given. Detailed comparisons are summarized in Table 1 and Table 2. In general, the BCD algorithm outperformed the competitors by having more true positives and less false positives in every case. Because the KGLasso method does not impose sparsity directly on the DAG structure, it suffered from having too many false negatives after thresholding when $p > n$. When $p < n$, the correlations between observations had a more significant impact on the estimation accuracy for DAGs. As a result, BCD and KGLasso which take this correlation into account performed better than the baseline. In particular, BCD substantially reduced the number of missing edges (FNs) and FDR, compared to the baseline. Both BCD and KGLasso yielded accurate estimates of $\widehat{\Theta}$ when $n < p$. When $n > p$, as the sample size p for estimating $\Theta^* \in \mathbb{R}^{n\times n}$ decreased relative to the dimension n, $\widehat{\Theta}$ became less accurate. The difference in the accuracy of $\widehat{\Theta} = \widehat{\Theta}^{(\infty)}$ and $\widehat{\Theta}^{(1)}$ was not significant.

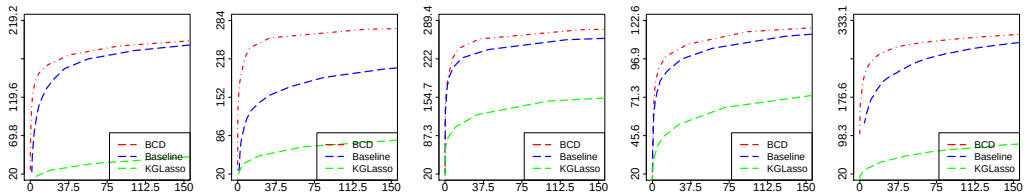
Figure 2 shows the ROC curves of all three methods over a sequence of $\bar{\tau}$ under the 10 settings. The $\bar{\tau}$ sequence contains 30 equally spaced values in $[0, 0.5]$. The BCD algorithm uniformly outperformed the others in terms of the area under the curve (AUC) with substantial margins when $n < p$. When $n > p$, the BCD still did better than the other two most of the time but its lead over KGLasso and baseline was not as significant in some cases. This was largely due to insufficient regularization on $\widehat{\Theta}$. Fixing $\lambda_2 = 0.01$ in this case implies $\lambda_p = 0.01/p = 0.0001$ in the graphical Lasso step (11) of BCD, resulting in

Θ-Network	Method	(n, p, s_0)	E	FN	TP	FDR	JI	SHD	err($\widehat{\Theta}$) (err($\widehat{\Theta}^{(1)}$))
equi-cor	BCD	(150, 300, 600)	686.2	214.0	386.0	0.355	0.443	514.2	0.00034 (0.00032)
	Baseline	(150, 300, 600)	642.4	240.3	359.7	0.383	0.410	523.0	—
	KGLasso	(150, 300, 600)	756.2	504.9	95.1	0.822	0.080	1166.0	0.00019
toeplitz	BCD	(200, 400, 800)	535.0	306.4	493.6	0.077	0.586	347.8	0.00143 (0.00833)
	Baseline	(200, 400, 800)	549.5	425.2	374.8	0.317	0.384	599.9	—
	KGLasso	(200, 400, 800)	550.6	634.3	165.7	0.698	0.139	1019.2	0.01617
star	BCD	(200, 400, 800)	543.1	301.1	498.9	0.081	0.591	345.3	0.00006 (0.00051)
	Baseline	(200, 400, 800)	515.7	338.3	461.7	0.103	0.541	392.3	—
	KGLasso	(200, 400, 800)	495.8	504.3	295.7	0.403	0.295	704.4	0.00233
AR(5)	BCD	(100, 200, 400)	253.1	193.8	206.2	0.184	0.461	240.7	0.00247 (0.00219)
	Baseline	(100, 200, 400)	245.8	208.9	191.1	0.217	0.420	263.6	—
	KGLasso	(100, 200, 400)	270.4	271.1	128.9	0.521	0.237	412.6	0.02587
SBM	BCD	(100, 300, 600)	343.3	313.6	286.4	0.159	0.435	370.5	0.51266 (0.52297)
	Baseline	(100, 300, 600)	344.6	338.4	261.6	0.233	0.383	421.4	—
	KGLasso	(100, 300, 600)	301.3	510.7	89.3	0.696	0.110	722.7	0.46201

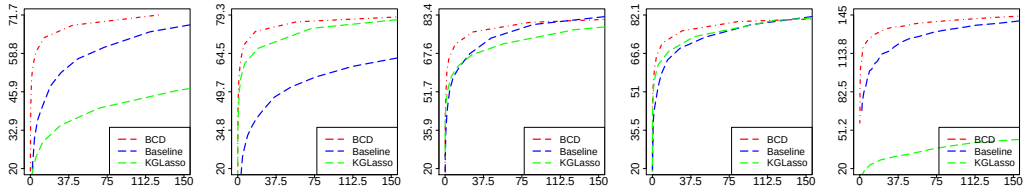
Table 1: Results for ordered DAGs on simulated data when $n < p$. The last column shows the ℓ_2-estimation errors of $\widehat{\Theta}$ and $\widehat{\Theta}^{(1)}$ normalized by the true support size. The numbers in the brackets are errors after one iteration of BCD. Each number corresponds to the average over 10 simulations.

Θ-Network	Method	(n, p, s_0)	E	FN	TP	FDR	JI	SHD	err($\widehat{\Theta}$) (err($\widehat{\Theta}^{(1)}$))
equi-cor	BCD	(200, 100, 200)	143.0	75.1	124.9	0.119	0.570	93.2	0.00211 (0.00199)
	Baseline	(200, 100, 200)	140.0	103.8	96.2	0.305	0.394	147.6	—
	KGLasso	(200, 100, 200)	149.0	129.5	70.5	0.501	0.255	208.0	0.00207
toeplitz	BCD	(200, 100, 200)	167.1	56.7	143.3	0.137	0.639	80.5	0.51735 (0.68703)
	Baseline	(200, 100, 200)	166.7	104.8	95.2	0.425	0.351	176.3	—
	KGLasso	(200, 100, 200)	158.6	70.6	129.4	0.176	0.564	99.8	0.85023
star	BCD	(200, 100, 200)	186.9	50.6	149.4	0.199	0.629	88.1	0.54769 (0.39316)
	Baseline	(200, 100, 200)	171.7	69.2	130.8	0.236	0.543	110.1	—
	KGLasso	(200, 100, 200)	183.0	66.9	133.1	0.271	0.532	116.8	0.18472
AR(5)	BCD	(200, 100, 200)	185.9	52.2	147.8	0.200	0.622	90.3	0.01644 (0.01184)
	Baseline	(200, 100, 200)	180.1	66.2	133.8	0.253	0.545	112.5	—
	KGLasso	(200, 100, 200)	177.0	62.7	137.3	0.215	0.574	102.4	0.01038
SBM	BCD	(300, 100, 200)	140.1	72.2	127.8	0.086	0.602	84.5	0.31189 (0.31448)
	Baseline	(300, 100, 200)	139.9	85.7	114.3	0.181	0.507	111.3	—
	KGLasso	(300, 100, 200)	128.8	161.1	38.9	0.689	0.134	251.0	0.32263

Table 2: Results for ordered DAGs on simulated data when $n > p$.



(a) Equal Corr (b) Toeplitz (c) Star (d) AR (e) SBM



(f) Equal Corr (g) Toepltiz (h) Star (i) AR (j) SBM

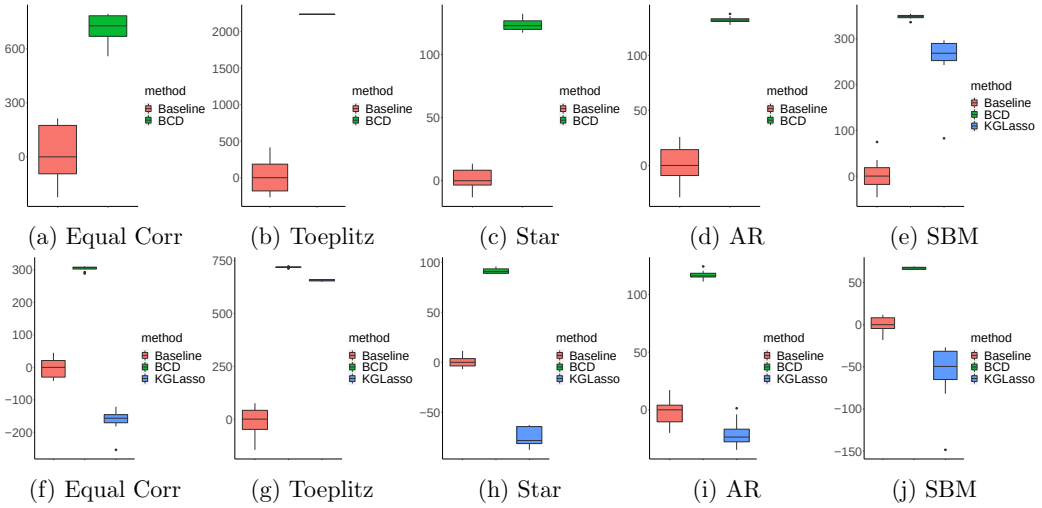
Figure 2: ROC curves of BCD, baseline, and KGLasso on simulated and sorted DAGs: x-axis reports the number of false positive edges and y-axis true positive edges. Top row: $n < p$. Bottom row: $n > p$. Each data point in the ROC curves corresponds to the average over 10 simulations.

severe overestimates of the magnitude of the entries in Θ^*. After we increased λ_p to 0.1 which is still quite small, the BCD indeed outperformed the other two methods by much larger margins. KGLasso also performed much better when $n > p$ as shown in Table 2 and Figure 2. This is expected because when n is large compared to p, the dependence among individuals will have a larger impact on the accuracy of the estimation of DAGs. Since KGLasso is designed to iteratively estimate Θ^* and Ψ^*, the more accurate estimates of Θ^* as reported in Table 2 compensated for the relatively inaccurate $\widehat{B}$.

We also compared test data log-likelihood among the three methods. Specifically, under each setting, we generated a test sample matrix X_{test} from the true distribution for each of the 10 repeated simulations and computed $-L(\widehat{B}, \widehat{\Theta}, \widehat{\Omega} \mid X_{\text{test}})$ using the estimates from the three methods following equation (6). Figure 3 shows the boxplots of the test data log-likelihood, normalized by $\sqrt{np}$ after subtracting the median of the baseline method: $\mathcal{L}_{\text{plot}} = \left(\mathcal{L}_0 - \text{median}(\mathcal{L}_0^{\text{baseline}})\right)/\sqrt{np}$, where $\mathcal{L}_0$ is the original test data log-likelihood. The top row shows the test log-likelihood when $n < p$, where we did not include the data for KGLasso in four cases because its test data log-likelihood values were too small to fit in the same plot with the other two methods. The bottom row shows the results for $n > p$. For both cases, we see that the test data log-likelihood of the BCD method (in green) is consistently higher than that of the other methods.

5.1.2 Learning with de-correlation

When the natural ordering is unknown, we focus on estimating the row-wise covariance Σ^*. Given $\widehat{\Sigma}$ we can de-correlate the data by Equation (14) and apply existing structural learning methods. In this study, we compared the performances of three structure learning methods before and after de-correlation: GES (Chickering, 2003) and sparsebn (Aragam et al.,



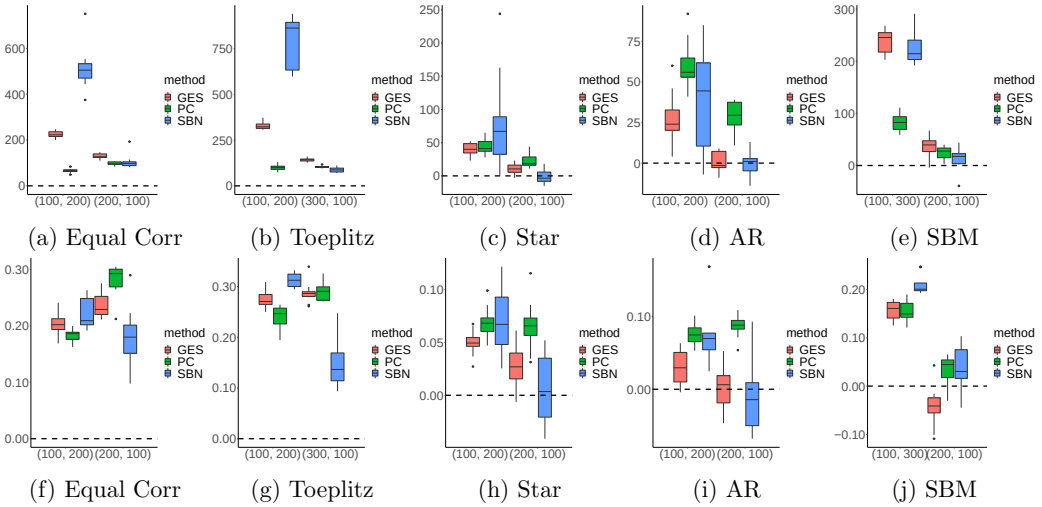
(a) Equal Corr (b) Toeplitz (c) Star (d) AR (e) SBM

(f) Equal Corr (g) Toeplitz (h) Star (i) AR (j) SBM

Figure 3: Normalized test data log-likelihood of BCD and baseline methods on simulated sorted DAGs. Top row: $n < p$. Bottom row: $n > p$. Each boxplot contains 10 data points from the 10 repeated experiments.

2019) which are score-based methods implemented respectively in the R packages `rcausal` (Ramsey et al., 2017) and `sparsebn`, and PC (Spirtes et al., 2000) which is a constraint-based method implemented in `pcalg` (Kalisch et al., 2012). All three methods rely on the independent data assumption, so we expect the de-correlation step to improve their performances significantly. Different from the previous comparison, the ordering of the nodes is unknown so GES and PC return an estimated CPDAG (completed acyclic partially directed graph) instead of a DAG. Thus, in the following comparisons, we converted both the estimated DAG from sparsebn and the true DAG to CPDAGs, so that all the reported metrics are computed with respect to CPDAGs.

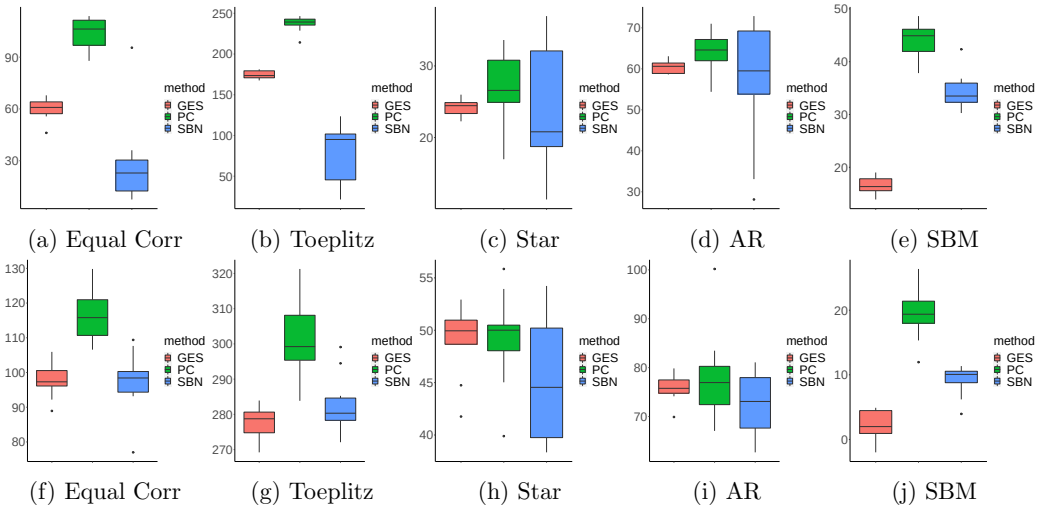
As before, we divided the cases into $n < p$ and $n > p$. The block size for the four block-diagonal Θ was fixed to 30. The estimated Cholesky factor $\widehat{L}$ of $\widehat{\Theta}$ used for de-correlating X in (14) was calculated by our BCD algorithm with tuning parameter λ_1 selected by BIC. Figure 4 shows the decrease in SHD and increase in Jaccard index via de-correlation of GES, PC and sparsebn on 10 random DAGs, generated under each row-covariance structure and each sample size. For almost all types of covariances and (n, p) settings we considered, there is significant improvement of all three methods in estimating the CPDAG structures after de-correlation. Additional tables with detailed results can be found in the Supplementary Material. Before decorrelation, GES and sparsebn, both score-based methods, tend to significantly overestimate the number of edges, resulting in high false positives, so does PC in some of the cases. After decorrelation, both GES and sparsebn had significant improvements and outperformed PC, as long as $\widehat{\Theta}$ was accurately estimated. The test data log-likelihood (normalized by $\sqrt{np}$) of all three algorithms also increased significantly after decorrelation as shown in Figure 5.



(a) Equal Corr (b) Toeplitz (c) Star (d) AR (e) SBM

(f) Equal Corr (g) Toeplitz (h) Star (i) AR (j) SBM

Figure 4: Decrease in SHD (top row) and increase in Jaccard index (bottom row) via decorrelation on simulated unsorted DAGs, with x-axis reporting the value of (n, p). In each panle, the three boxplots on the left and the three on the right correspond to the cases of $n < p$ and $n > p$, respectively. Each boxplot contains 10 data points from 10 simulations.



(a) Equal Corr (b) Toeplitz (c) Star (d) AR (e) SBM

(f) Equal Corr (g) Toeplitz (h) Star (i) AR (j) SBM

Figure 5: Increase in the normalized test data log-likelihood after decorrelation on simulated unsorted DAGs. Top row: $n < p$. Bottom row: $n > p$.

5.2 Experiments on Real Network Structures

In this section, we look at the performance of the BCD algorithm on real network structures. We took four real DAGs from the bnlearn repository (Scutari, 2010): `Andes`, `Hailfinder`, `Barley`, `Hepar2`, and two real undirected networks from `tnet` (Opsahl, 2009): `facebook` (Opsahl and Panzarasa, 2009) and `celegans_n306` (Watts and Strogatz, 1998). Only the structures (supports) of these real networks were used and the parameters of the edges were simulated as follows. Given a DAG structure, we sampled the coefficients β_j^* uniformly from $[-1, -0.1] \cup [0.1, 1]$. Given the support of Θ^*, we generated Θ'_{ij} uniformly from $[-5, 5]$. Then, we applied the transformations in (21) to get Θ^*. In order to increase the size of the underlying DAG and show the scalability of the algorithm, we duplicated the DAGs above to form larger networks. In Section 5.2.1 and 5.2.2, we again consider undirected networks consisting of several disconnected subgraphs, corresponding to a block-diagonal structure in Θ^*. Each of the subgraphs was sub-sampled from the original real network. In Section 5.2.3 we present experiments on more general Θ^* without a block-diagonal structure. The ω_j^* were uniformly sampled from $[0.1, 2]$ as before. With these parameters, we generated observational samples X following the structural equation (2).

5.2.1 Learning with given ordering

Similar to the previous section, we first looked at the results on ordered DAGs. We considered four different combinations of network structures as shown in Table 3, with both $n > p$ and $n < p$. Because KGLasso does not scale well with n, we did not include it in our comparisons. BCD continued to outperform the baseline method by modeling the sample correlation. This improvement was more prominent when $n > p$, where BCD significantly reduced the number of false positive edges, achieving higher JI and lower SHD compared to the baseline as well as to itself in the $n < p$ case. Figure 6 compares the test data log-likelihood of the two methods across 10 simulations, and BCD scored significantly higher test data log-likelihood in all the cases. The ROC curves of the two methods is provided in a figure in the Supplementary Material. Both figures indicate the BCD method indeed gives better DAG estimates than the baseline method.

5.2.2 Learning with de-correlation

When the ordering of the DAG nodes is not given, we compared the effect of decorrelation as in Section 5.1.2. All network parameters were generated in the same way as before but we randomly shuffled the columns of X. The decrease in the structural Hamming distance and increase in Jaccard index from decorrelation over 10 simulations are summarized as boxplots in Figure 7. PC performed uniformly better after decorrelation compared to before. GES and sparsebn also improved after decorrelation in most cases. The changes in the test data log-likelihood are shown in Figure 8, which are positive for almost all date sets, except two outliers (removed from plots) of sparsebn in the second and fourth panels in the top row.

5.2.3 Learning under general covariance structure

In the following experiments, we generated the support of Θ^* without the block-diagonal constraint by directly sampling the real undirected networks `facebook` and `celegans_n306`.

DAG	Θ-Network	Method	(n, p, s_0)	E	FN	TP	FDR	JI	SHD	err($\widehat{\Theta}$) (err($\widehat{\Theta}^{(1)}$))
		BCD	(100 446, 676)	440.9	314.7	361.3	0.176	0.478	394.3	0.03458 (0.03564)
		Baseline	(100 446, 676)	436.6	334.2	341.8	0.211	0.444	429.0	—
Andes	facebook									
(2)		BCD	(500 446, 676)	500.0	197.3	478.7	0.043	0.686	218.6	0.07816 (0.07758)
		Baseline	(500 446, 676)	499.1	206.4	469.6	0.058	0.666	235.9	—
		BCD	(100, 224, 264)	154.5	124.2	139.8	0.092	0.502	138.9	0.03922 (0.02167)
		Baseline	(100, 224, 264)	154.8	126.8	137.2	0.110	0.487	144.4	—
Hailfinder	celegan_n306									
(4)		BCD	(500, 224, 264)	168.4	97.1	166.9	0.009	0.629	98.6	0.02010 (0.02138)
		Baseline	(500, 224, 264)	168.0	98.0	166.0	0.012	0.624	100.0	—
		BCD	(100, 192, 336)	211.0	150.5	185.5	0.119	0.513	176.0	0.01453 (0.00100)
		Baseline	(100, 192, 336)	211.9	156.2	179.8	0.150	0.489	188.3	—
Barley	facebook									
(4)		BCD	(500, 192, 336)	260.5	91.3	244.7	0.061	0.696	107.1	0.23144 (0.28172)
		Baseline	(500, 192, 336)	260.2	97.6	238.4	0.083	0.666	119.4	—
		BCD	(100, 280, 492)	366.7	234.5	257.5	0.295	0.428	343.7	0.05101 (0.03104)
		Baseline	(100, 280, 492)	373.9	238.0	254.0	0.314	0.416	357.9	—
Hepar2	celegan_n306									
(4)		BCD	(500, 280, 492)	417.5	122.2	369.8	0.114	0.685	169.9	0.00759 (0.00755)
		Baseline	(500, 280, 492)	417.1	126.0	366.0	0.122	0.674	177.1	—

Table 3: Results for ordered DAGs on real network data. Block size is 20 for $n < p$ and 50 for $n > p$. The number under each DAG reports the number of times it is duplicated to form a large DAG. All numbers represent the average over 10 simulations.

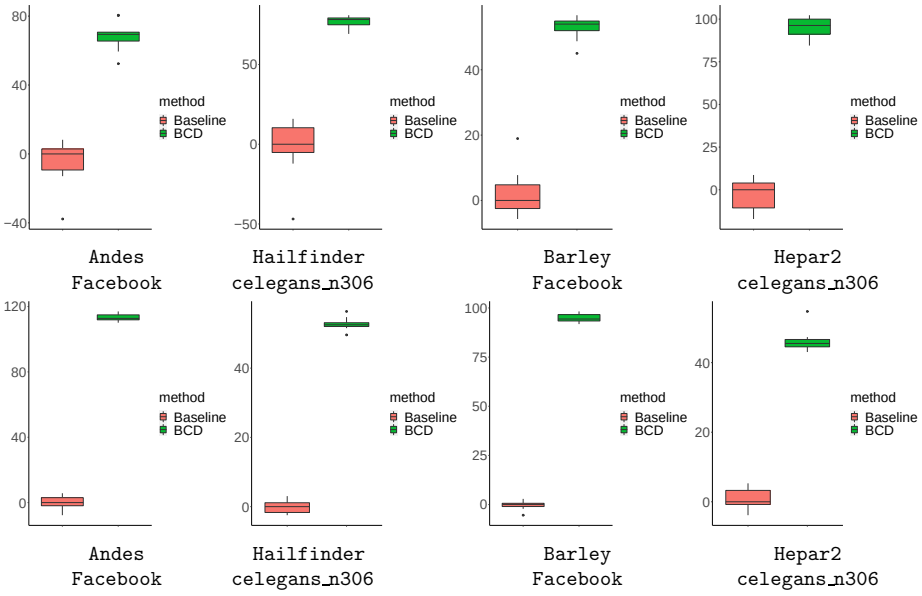


Figure 6: Test data log-likelihood normalized by $\sqrt{np}$ on real sorted DAGs. Top row: $n < p$. Bottom row: $n > p$.

In other words, the underlying undirected network may have only one connected subgraph where all individuals are dependent. This setup poses major challenge particularly for the estimation of Θ because its support becomes much larger, and as a result, we will need to

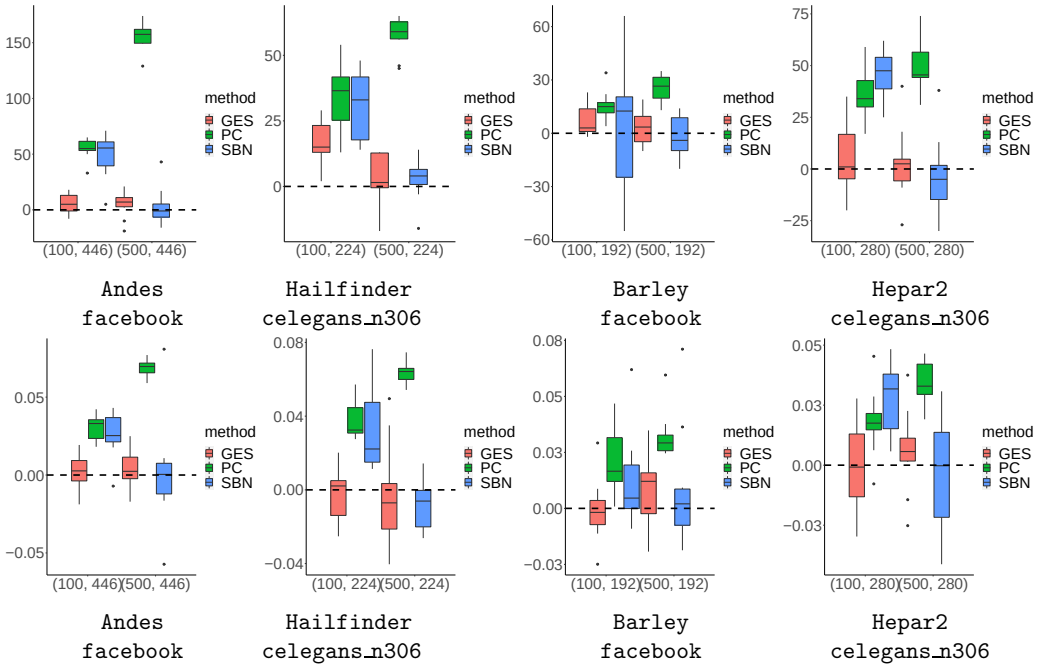


Figure 7: Experiments on real unsorted DAGs. Decrease in SHD (top row) and increase in JI (bottom row) via de-correlation for real networks, where the x-axis reports the value of (n, p). In each panel, the three boxplots on the left and the three on the right correspond to cases of $n < p$ and $n > p$, respectively.

impose stronger regularization in the graphical Lasso step when $n > p$. For simplicity, we focus on the setting $p > n$ so that we can still fix $\lambda_2 = 0.01$. Proceeding as before, we generated B^* from real DAGs with duplications: `Andes`, `Hailfinder`, `Barley`, and `Hepar2`.

DAG	Θ-Network	Method	(n, p, s_0)	E	FN	TP	FDR	JI	SHD	err($\widehat{\Theta}$) (err($\widehat{\Theta}^{(1)}$))
Andes	facebook	BCD	(100 446, 676)	424.1	319.9	356.1	0.155	0.478	387.9	0.01531 (0.00288)
(2)		Baseline	(100 446, 676)	422.2	326.2	349.8	0.165	0.467	398.6	—
Hailfinder	celegans_n306	BCD	(100,224,264)	122.0	163.0	101.0	0.172	0.354	184.0	0.08678 (0.08309)
(4)		Baseline	(100,224,264)	122.0	161.0	103.0	0.156	0.364	180.0	—
Barley	facebook	BCD	(100,192,336)	252.6	147.3	188.7	0.248	0.471	211.2	0.08021 (0.08555)
(4)		Baseline	(100,192,336)	249.4	155.1	180.9	0.268	0.448	223.6	—
Hepar2	celegans_n306	BCD	(100, 420, 738)	492.3	386.7	351.3	0.285	0.399	527.7	0.03725 (0.03614)
(6)		Baseline	(100, 420, 738)	490.8	397.8	340.2	0.303	0.384	548.4	—

Table 4: Results for ordered DAGs on real network data without block structure.

First we assume that the DAG ordering is known and compare the BCD method against the baseline method. In three out of the four cases we considered, BCD gave better estimates of the DAG structure in terms of Jaccard index and structural Hamming distance as shown in Table 4. Next, without assuming a known DAG ordering, we compare the performance of GES, PC, and sparsebn before and after de-correlation. The first row in Figure 9 shows

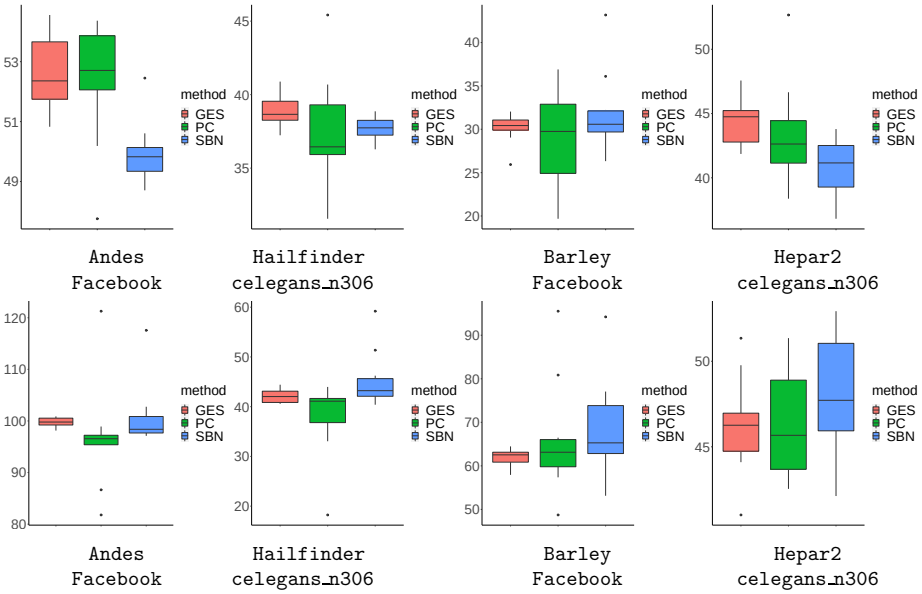


Figure 8: Increases in test data log-likelihood on real unsorted DAGs. Top row: $n < p$. Bottom row: $n > p$. Some outliers in the top panels are not shown for a better view of the boxplots.

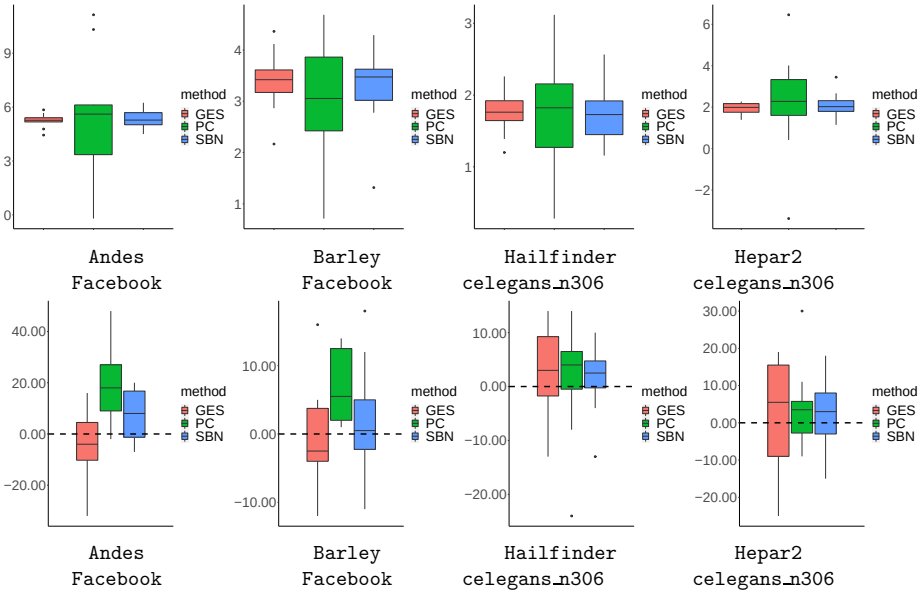


Figure 9: Results on real unsorted DAGs with general Θ. Top row: increase in normalized test data log-likelihood after de-correlation. Bottom row: Decrease in SHD after de-correlation.

the increase in the normalized test data log-likelihood after de-correlation, and the increases are positive across all 10 simulations for each of the four scenarios. The second row shows

the distribution of the decrease in SHD across 10 simulations after de-correlation. In most cases, all three methods gave much more accurate estimates after de-correlation. We defer the additional tables and figures containing more detailed results to the Supplementary Material. The above results confirm that our methods can indeed improve the accuracy in DAG estimation even Θ^* is not block-diagonal, as suggested by the theoretical results in Section 4.

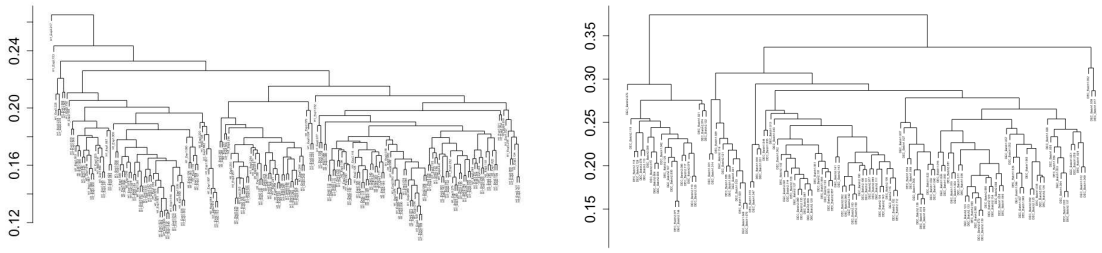
6. Application on RNA-seq Data

Gene regulatory networks (GRNs) enable biologists to examine the causal relations in gene expression during different biological processes, and are usually estimated from gene expression data. Recent advances in single-cell RNA sequencing technology has made it possible to trace cellular lineages during differentiation and to identify new cell types by measuring gene expression of thousands of individual cells. A key question arises now is whether we can discover the GRN that controls cellular differentiation and drives transitions from one cell type to another using this type of data. Such GRNs can be interpreted as causal networks among genes, where nodes correspond to different genes and a directed edge encodes a direct causal effect of one gene on another.

The RNA-seq data set used in this section can be found in NCBI's Gene Expression Omnibus and is accessible through GEO series accession number GSE75748. The data set contains gene expression measurements of around 20,000 genes from $n = 1018$ cells. Before conducting the experiment, we processed the data according to Li and Li (2018) by imputing missing values and applying log transformation. In this study, we focus on estimating a GRN among $p = 51$ target genes selected by Chu et al. (2016), while the rest of the genes, which we call background genes, are used to estimate an undirected network for all 1018 cells.

6.1 Pre-estimate the Undirected Network

An essential input to Algorithm 1 is $A(G^*)$, the adjacency matrix of a known undirected network of observations (cells in this case). The 1018 cells in our data come from 7 distinct cell types (H1, H9, HFF, TB, NPC, DEC, and EC), and it is reasonable to assume that the similarity between cells of different types is minimal. Therefore, we posit that the network of the 1018 cell consists of at least 7 connected components, i.e. $N \geq 7$, where N denotes the number of diagonal blocks in Θ^* as in Section 4. Since it is unlikely that all cells of the same type are strongly associated with one another, we further divided each type of cells into smaller clusters by applying classical clustering algorithm on the background genes. More specifically, we randomly selected 8000 genes from the background genes and applied hierarchical clustering on each type of cells. In this experiment, we used hierarchical clustering with complete-linkage and a distance metric between two cells defined as $1 - \rho$, where ρ is the correlation between their observed gene expression levels. We verified our choice of clustering algorithm by applying it on the entire data set, and it clearly grouped the cells into 7 groups, coinciding with the 7 cell types. At the end of the hierarchical clustering step, we needed to pick cutoff levels in order to finish clustering. Because the levels of dependence among cells are quite different across cell types as shown in Figure 10, we picked cutoff points separately for each cell type. Generally, we chose the cutoff



(a) H1 (b) DEC

Figure 10: Cluster dendrograms of H1 and DEC cells from hierarchical clustering. The y-axis represents $1 - \rho$ and the leaf nodes are individual cells.

thresholds such that the largest cluster is smaller than $p = 51$, so λ_2 can be set to 0.01. By shifting the cutoff levels, we also obtained different number N of blocks in Θ^*. In the end, this clustering process returns an adjacency matrix A of the estimated network defined by the N clusters. In our experiments, we compared results from three choices of $N \in \{383, 519, 698\}$. The cluster size varied from 1 (singleton clusters) to 43 across the three cases.

6.2 Model Evaluation

In this experiment, the input to the BCD algorithm is a data matrix $X_{1018\times 51}$ and $\text{supp}(\Theta^*) \subseteq A$ estimated by the above hierarchical clustering. The matrix of error variances $\widehat{\Omega}$ was obtained following the method described in Section 3.2. The output is a solution path of $(\widehat{B}, \widehat{\Theta})$ for a range of λ_1's. We computed the corresponding MLEs $(\widehat{B}^{MLE}, \widehat{\Omega}^{MLE})$ given the support of each $\widehat{B}$ on the solution path. We picked the $(\widehat{B}^{MLE}, \widehat{\Omega}^{MLE})$ with the smallest BIC from the solution path as in Section 5 and used the corresponding $\widehat{\Theta}$ to de-correlate X. Table 5 shows the results of GES, PC, and sparsebn before and after de-correlation. In each case, we computed the BIC of the estimated GRN and we used the Likelihood Ratio (LR) test to determine whether the increase in log-likelihood from de-correlation is significant or not. The LR test statistic is defined as follows:

$$\text{LR} = 2(\log p(X \mid \widehat{\Theta}, \widehat{B}^{MLE}_{decor}, \widehat{\Omega}^{MLE}_{decor}) - \log p(X \mid I_n, \widehat{B}^{MLE}_{baseline}, \widehat{\Omega}^{MLE}_{baseline})),$$

where $(\widehat{B}^{MLE}_{baseline}, \widehat{\Omega}^{MLE}_{baseline})$ and $(\widehat{B}^{MLE}_{decor}, \widehat{\Omega}^{MLE}_{decor})$ denote the MLEs given the estimated graph structures from GES, PC, and sparsebn before and after de-correlation, respectively. If the baseline model (before de-correlation) is true, then the LR statistic follows approximately a χ^2 distribution with degrees of freedom

$$df = \frac{|\,\text{supp}(\widehat{\Theta})| - n}{2} + |\,\text{supp}(\widehat{B}^{MLE}_{decor})| - |\,\text{supp}(\widehat{B}^{MLE}_{baseline})|.$$

In most cases, we saw significant improvements, in terms of both the BIC and the χ^2 statistic, in all three DAG estimation methods by de-correlating X using the estimated $\widehat{\Theta}$ from the

model_type	N	BIC(baseline)	BIC(decor)	ll(baseline)	ll(decor)	LR χ^2	df	p-value
GES		-35036.60	-41799.39	17593.30	22667.70	10148.79	3386	0
PC	383	-14102.81	-32274.01	7102.91	17901.01	21596.19	3425	0
sparsebn		-28553.26	-37894.85	14336.13	20697.42	12722.59	3381	0
GES		-35036.60	-33312.95	17593.30	17597.47	8.34	1732	1
PC	519	-14102.81	-18738.62	7102.91	10297.31	6388.81	1753	0
sparsebn		-28553.26	-28797.87	14336.13	15324.43	1976.61	1732	0
GES		-35036.60	-34921.63	17593.30	17879.82	573.03	688	1
PC	698	-14102.81	-16959.60	7102.91	8879.30	3552.79	696	0
sparsebn		-28553.26	-29751.59	14336.13	15273.80	1875.33	677	0

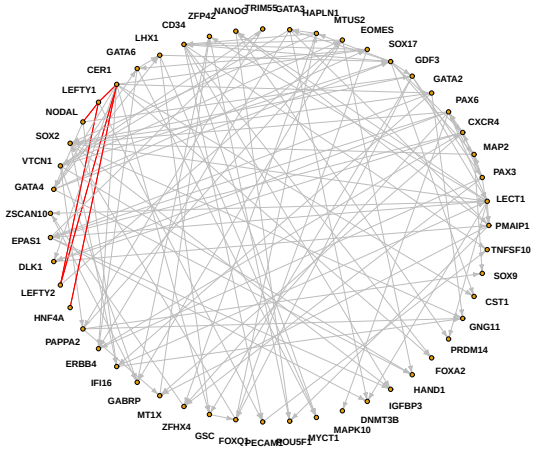
Table 5: BIC scores and log-likelihood(ll) values from GES, PC, and sparsebn before and after de-correlation. N denotes the number of clusters among cells.

BCD algorithm. This confirms the dependence among individual cells and implies that our proposed network model fits this real-world data better. Figure 11 shows the estimated CPDAGs after de-correlation for the case $N = 383$, which corresponds to the minimum BIC for all three methods in our experiments. It is interesting to note that a directed edge NANOG→POU5F1, between the two master regulators in embryonic stem cells, appears in all three estimated CPDAGs, consistent with previously reported gene regulatory networks (Chen et al., 2008; Zhou et al., 2007).

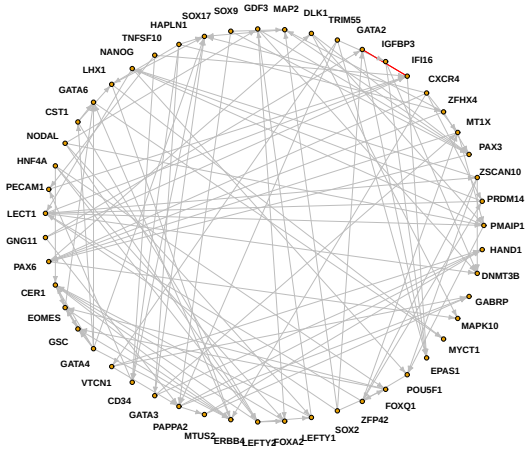
7. Discussion

In this paper our main goal is to generalize the existing Gaussian DAG model to dependent data. We proposed to model the covariance between observations by assuming a non-diagonal covariance structure of the noise vectors. This generalization is related to the semi-Markovian assumption in causal DAG models. Our main contributions include the development of a consistent structural learning method for the DAG and the sample network under sparsity assumptions and finite-sample guarantees for the estimators.

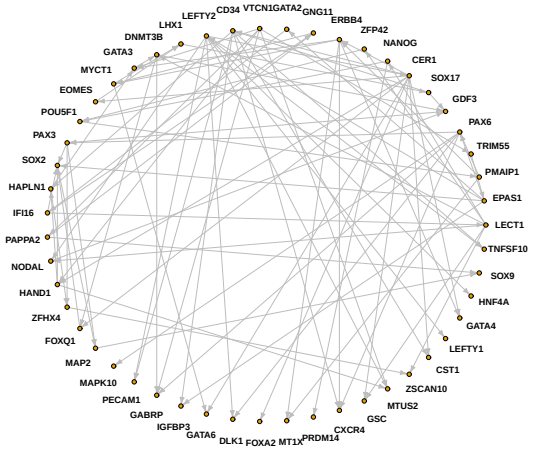
Our proposed BCD algorithm is built upon existing Lasso regression and graphical Lasso covariance estimation methods. When a topological ordering of the true DAG is known, it estimates the covariance between the observations Σ and the WAM of the DAG B in an iterative way. The method is fast and often converges in a few iterations. Our theoretical analysis shows that the estimates after one iteration are ℓ_2-consistent under various asymptotic frameworks including both $n \ll p$ and $n \gg p$, assuming a proper initialization of the precision matrix $\widehat{\Theta}^{(0)}$. The estimate of the DAG WAM $\widehat{B}^{(1)}$ achieves the optimal rate as Lasso estimators. The estimate of the precision matrix $\widehat{\Theta}^{(1)}$ achieves the same optimal rate as the graphical Lasso method when $n \gg p$ and there are sufficiently many independent subgroups within the data. Otherwise, it has a slightly worse rate due to the bias of the sample covariance matrix. When the DAG ordering is unknown, we showed the covariance Σ is invariant under permutations of the DAG nodes. Therefore, if the true DAG is sparse, our BCD algorithm can still give a good estimate of Θ which can be used to decorrelate the data. In addition to the theoretical analysis, we conducted extensive experiments on both synthetic and real data to compare our method with existing methods.



(a) GES ($E = 131, U = 5$)



(b) PC ($E = 119, U = 1$)



(c) sparsebn ($E = 90, U = 0$)

Figure 11: Estimated gene regulatory networks (CPDAGs) after de-correlation, with E edges and U undirected edges colored in red.

When a true ordering of the DAG was given, the BCD algorithm significantly improved the structural estimation accuracy compared to the baseline method which ignored the sample dependency. When the ordering was unknown, classical DAG learning methods, such as GES, PC, and sparsebn, can all be greatly improved with respect to structural learning of CPDAGs by using our proposed de-correlation method based on the BCD algorithm. In all cases, our estimation methods under the proposed network Gaussian DAG model yielded significantly higher test data log-likelihood compared to other competing methods, indicating better predictive modeling performance.

There are several unexplored directions from our research. First, the current error bounds and consistency results are based upon a known topological ordering of the true DAG. In practice, however, it can be hard to obtain the ordering in advance. It would be interesting to see if we can combine the method of estimating partial orders such as Niinimäki et al. (2016) with our method and extend the theoretical results. Second, part of the reason that the current model relies on a known DAG ordering is the lack of experimental data. From purely observational data, it is impossible to orient some of the edges and find a topological ordering of the true DAG. In the next step, we would like to extend our method to handle both observational and experimental data sets. Finally, there are recent methods that use continuous optimization for DAG learning without imposing the acyclicity constraint, such as NOTEARS (Zheng et al., 2018). It is a promising future direction to incorporate such ideas into DAG learning on network data.

Acknowledgments

We would like to acknowledge support for this project from NSF grant DMS-1952929.

Appendix A. Technical Details of Section 4

A.1 Some Auxiliary Results

Here we introduce four lemmas that we use to establish the error bounds of $\widehat{\Theta}^{(1)}$ and $\widehat{B}^{(1)}$. Let us start by deriving an upper bound on the ℓ_2 deviation of $\widehat{L}^{(0)}$ from L^* under Assumption 1.

Lemma 10 *Suppose Assumption 1 holds and let $L^*, \widehat{L}^{(t)}$ be the Cholesky factors of Θ^* and $\widehat{\Theta}^{(t)}$, respectively. Let $\widehat{\Delta}^{(t)}_{chol} = \widehat{L}^{(t)} - L^*$. Then,*

$$\|\widehat{\Delta}^{(0)}_{chol}\|_2 \leq \frac{M}{2\sigma_{\min}(L^*)},$$

where $\sigma_{\min}(L^*)$ is the smallest singular value of L^*, and M is from Assumption 1.

To generalize the basic bound on $\|\hat{\beta}^{lasso} - \beta^*_j\|_2$ from Buhlmann and van de Geer (2011) to dependent data, we need to control the ℓ_∞-norm of an empirical process component $2X^\top \widehat{\Theta}^{(0)} \epsilon_j / n$. Let us start with the case when the data are independent. Define the following events, where $\widetilde{X} = L^* X$ represents the independent data as explained in Section

4:

$$\mathscr{E} := \bigcap_{k=1}^{p} \left\{ \|\widetilde{X}_k\|_2 \leq 6\bar{\psi}\sqrt{n} \right\}. \tag{22}$$

Then the following lemma follows from $\widetilde{X}$ being sub-Gaussian.

Lemma 11 *Let $\alpha > 2$ be an integer. If $n > 2\sqrt{2\alpha}\log p$, then the event $\mathscr{E}$ defined in (22) holds with probability at least $1 - 1/p^{\alpha-1}$.*

Next, define the following events that depend on λ_n and are standard in the classical Lasso problem,

$$\begin{aligned} \mathscr{T}_j &:= \left\{ 2\|\widetilde{X}^\top \tilde{\varepsilon}_j\|_\infty / n \leq \lambda_n \right\}, \quad j = 1, \ldots, p \\ \mathscr{T} &:= \bigcap_{j=1}^{p} \mathscr{T}_j. \end{aligned} \tag{23}$$

Lemma 12 *Let $\widetilde{X}_k$ consist of n i.i.d sub-Gaussian random variables with parameter $\bar{\psi}^2$ for $k = 1, \ldots, p$. If*

$$\lambda_n = 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{n}} + \sqrt{\frac{2\log 2 + 4\log p}{n}} \right),$$

then the probability of $\mathscr{T}$ satisfies

$$\mathbb{P}(\mathscr{T}) \geq \left(1 - \frac{1}{p}\right)^2.$$

Lemma 12 implies that if $\lambda_n \asymp \sqrt{\frac{\log p}{n}}$, then the error terms will be uniformly under control with high probability, especially when both n and p are large.

Lemma 13 (Maximal inequality) *Let $x_i = (x_{i1}, \ldots, x_{ip})$ be a random vector where each element x_{ij} is sub-Gaussian with parameter $\bar{\psi}^2$, then*

$$\mathbb{P}\left(\|x_i\|_\infty \geq 2\bar{\psi}\sqrt{\log p} \right) \leq 2/p.$$

From model (4), it is clear that each row in $\widetilde{X}$ is sub-Gaussian with parameter $\bar{\psi}^2$. By Lemma 13, we have $\|\tilde{x}_i\|_\infty \lesssim \sqrt{\log p}$ w.h.p.

A.2 Error Bound of $\widehat{B}^{(1)}$

The estimation error bound for the classical Lasso problem where samples are i.i.d. was established by choosing a penalty coefficient that dominates the measurement error term. Specifically, as shown in Lemma 12, this can be achieved with high probability by setting $\lambda_n \asymp \sqrt{\frac{\log p}{n}}$. In order to prove the consistency of $\widehat{B}^{(1)}$ from Algorithm 1, we need to control a similar error term which depends on $\widehat{\Theta}^{(0)}$. Notably, such error can be controlled under the same rate as λ_n, see Theorem 14.

Theorem 14 (Control the empirical process) *Let λ_n be the same as in Lemma 12. Suppose the initial estimator $\widehat{\Theta}^{(0)}$ satisfies Assumption 1. Then*

$$\mathbb{P}\left(\sup_{j\in[p]} 2\|X^\top \widehat{\Theta}^{(0)}\varepsilon_j\|_\infty/n \le \lambda_n\right) \ge \left(1-\frac{1}{p}\right)\left(1-\frac{2}{p}\right). \tag{24}$$

Next, we show that the random matrix $\widehat{L}^{(0)}X$ satisfies the Restricted Eigenvalue (RE) condition (Wainwright, 2019) w.h.p. Towards that end, we define the event $\mathscr{K}$ as in Theorem 7.16 from Wainwright (2019) given as

$$\mathscr{K} := \left\{\|\widetilde{X}\beta\|_2^2/n \ge \tilde{c}_1\|\sqrt{\Psi^*}\beta\|_2^2 - \tilde{c}_2\rho^2(\Psi^*)\frac{\log p}{n}\|\beta\|_1^2\right\}, \tag{25}$$

where $\rho^2(\Psi^*)$ is the maximum diagonal entry of Ψ^* and $\widetilde{X} = L^*X$ is the de-correlated data.

Lemma 15 (Restricted eigenvalue condition) *Consider a random matrix $X \in \mathbb{R}^{n\times p}$, which is drawn from a $\mathcal{N}_{n\times p}(0, \Sigma^*, \Psi^*)$ distribution. Let $\widehat{\Theta}^{(0)}$ be the initial estimate of $\Theta^* = \Sigma^{*-1}$ satisfying Assumption 1, $\widehat{L}^{(0)}$ be the Cholesky factor of $\widehat{\Theta}^{(0)}$, and $\rho^2(\Psi^*)$ be the maximum diagonal entry of Ψ^*. Then under event $\mathscr{K}$ defined in* (25)*, there are universal positive constants $c_1 < 1 < c_2$ such that*

$$\frac{\|\widehat{L}^{(0)}X\beta\|_2^2}{n} \ge c_1\|\sqrt{\Psi^*}\beta\|_2^2 - c_2\rho^2(\Psi^*)\frac{\log p}{n}\|\beta\|_1^2, \tag{26}$$

for all $\beta \in \mathbb{R}^p$.

The probability of event $\mathscr{K}$ can be found in Theorem 7.16 from Wainwright (2019) . This event is a restriction on the design matrix $\widetilde{X}$ and it holds with high probability for a variety of matrix ensembles. With Theorem 14 and Lemma 15, it is possible to prove an *oracle inequality* for the dependent Lasso problem, which yields a family of upper bounds on the estimation error.

Theorem 16 (Lasso oracle inequality) *Consider the Lasso problem in* (10) *for $t = 0$. Suppose the inequality* (26) *and the event in* (24) *hold. Let $\bar{\kappa} = \sigma_{\min}(\Psi^*)$. For $j \in [p]$ and any $\beta_j^* \in \mathbb{R}^p$, if*

$$\lambda_n \ge 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{n}} + \sqrt{\frac{2\log 2 + 4\log p}{n}}\right),$$

then any optimal solution $\hat{\beta}_j^{(1)}$ *satisfies:*

$$\|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 \leq \frac{768\lambda_n^2}{c_1^2\bar{\kappa}^2}|S| + \frac{64\lambda_n}{4c_1\bar{\kappa}}\|\beta_{j,S^c}^*\|_1 + \frac{128c_2}{c_1}\frac{\rho^2(\Psi^*)}{\bar{\kappa}}\frac{\log p}{n}\|\beta_{j,S^c}^*\|_1^2, \tag{27}$$

for any subset S *with cardinality* $|S| \leq \frac{c_1}{64c_2}\frac{\bar{\kappa}}{\rho^2(\Psi^*)}\frac{n}{\log p}$. *Let* $\widehat{L}^{(0)}$ *be the Cholesky factor of* $\widehat{\Theta}^{(0)}$. *Then,*

$$\|\widehat{L}^{(0)}X\left(\hat{\beta}_j^{(1)} - \beta_j^*\right)\|_2^2/n \leq 6\lambda_n\|\beta_j^*\|_1. \tag{28}$$

Theorem 16 implies $\sup_{j\in[p]}\|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 \leq \frac{768\lambda_n^2}{c_1^2\bar{\kappa}^2}s \asymp s\frac{\log p}{n}$, where s is the maximum in-degree of the true DAG.

A.3 Error Bounds of $\widehat{\Theta}^{(1)}$

Recall that s denotes the maximum number of nonzero entries in β_j^* for $j \in [p]$. In order to control $\|\widehat{\Theta}^{(1)} - \Theta^*\|_2$, we need to rely on certain type of error bound on $\widehat{S}^{(1)} - \Sigma^*$, where $\widehat{S}^{(1)}$ is the sample covariance defined in (11) when $t = 0$. Therefore, we adopt the definition of tail condition on the sample covariance from Ravikumar et al. (2011).

Definition 17 *(Tail conditions) We say the* $n\times p$ *random matrix* X *from model* (2) *satisfies tail condition* $\mathcal{T}(f, v_*)$ *if there exists a constant* $v_* \in (0,\infty]$ *and a function* $f : \mathbb{N}\times(0,\infty) \to (0,\infty)$ *such that for any* $(i,j) \in [n]\times[n]$,

$$\mathbb{P}\left[|\widehat{S}_{ij}^{(1)} - \Sigma_{ij}^*| \geq \delta\right] \leq 1/f(p,\delta) \quad \forall\delta \in (0, 1/v_*].$$

We require $f(p,\delta)$ to be monotonically increasing in p, so for a fixed $\delta > 0$, define the inverse function

$$\bar{p}_f(\delta; r) := \arg\max\{p \mid f(p,\delta) \leq r\}.$$

Similarly, f should be increasing in δ for each fixed p, so we define an inverse function in the second argument:

$$\bar{\delta}_f(p; r) := \arg\max\{\delta \mid f(p,\delta) \leq r\}. \tag{29}$$

Under the setting of a Gaussian DAG model, we can derive a sub-Gaussian tail bound.

Lemma 18 *Let* X *be a sample from our Gaussian DAG model* (2). *The sample covariance matrix*

$$\widehat{\Sigma} = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{\omega_j^{2*}}\left(X_j - X\beta_j^*\right)\left(X_j - X\beta_j^*\right)^\top, \tag{30}$$

satisfies the tail bound

$$\mathbb{P}\left(\sup_{i,j}|\widehat{\Sigma}_{ij} - \Sigma_{ij}^*| > \delta\right) \leq 4\exp\left\{-\frac{p\delta^2}{3200}\right\},$$

for all $\delta \in (0, 40)$.

Corollary 19 *If* $f(p,\delta) = 4\exp\left\{\frac{p\delta^2}{3200}\right\}$, *then the inverse function* $\bar{\delta}_f(p;n^\tau)$ *takes the following form,*

$$\bar{\delta}_f(p;n^\tau) = 40\sqrt{2}\sqrt{\frac{\tau\log n + \log 4}{p}}.$$

Based on the tail bound in Corollary 19, we can control the sampling noise $\widehat{\Sigma} - \Sigma^*$ as in Lemma 20.

Lemma 20 (Lemma 8 in Ravikumar et al. 2011) *Define event*

$$\mathcal{A} = \left\{\|\widehat{\Sigma} - \Sigma^*\|_\infty \leq \bar{\delta}_f(p;n^\tau)\right\}, \tag{31}$$

where $\bar{\delta}_f(p;n^\tau) = 40\sqrt{2}\sqrt{\frac{\tau\log n+\log 4}{p}}$. *For any* $\tau > 2$ *and* (n,p) *such that* $\bar{\delta}_f(p;n^\tau) \leq 1/40$, *we have*

$$\mathbb{P}\left[\mathcal{A}^c\right] \leq \frac{1}{n^{\tau-2}} \to 0.$$

Recall $r(\widehat{\Omega})$ defined in (16) and the constant b defined in Lemma 5.

Lemma 21 *Suppose* $b > 0$ *and*

$$\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2 \leq c\cdot s\frac{\log p}{n}, \tag{32}$$

for a fixed positive constant c. *Let* $\widehat{W}^{(1)} := \widehat{S}^{(1)} - \Sigma^*$. *Then, we have*

$$\|\widehat{W}^{(1)}\|_\infty \leq 40\sqrt{2}\sqrt{\frac{\tau\log n + \log 4}{p}} + \max\left\{6\bar{\omega}r(\widehat{\Omega}), \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p\log^2\max\{n,p\}}{n}}\right\}$$

with probability at least $1 - 1/n^{\tau-2} - 5n^2/\max\{n,p\}^4$.

Theorem 22 *Assume* $\widehat{B}^{(1)}$ *satisfies* (32) *and* $b > 0$. *Let*

$$\bar{R}(s,p,n) = \max\left\{6\bar{\omega}r(\widehat{\Omega}), \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p\log^2\max\{n,p\}}{n}}\right\},$$

$$\bar{\delta}_f(p;n^\tau) = 40\sqrt{2}\sqrt{\frac{\tau\log n + \log 4}{p}}.$$

Consider the graphical Lasso estimate $\widehat{\Theta}^{(1)}$ *from Algorithm 1 with* $\lambda_p = \bar{\delta}_f(p;n^\tau) + \bar{R}$ *for* $\tau > 2$. *Assume*

$$\bar{p}_f\left(1/\max\left\{160, 24mC\right\}, n^\tau\right) \leq p \quad \textit{and} \quad \bar{R} \leq \frac{1}{24C}, \tag{33}$$

where $C = \max\left\{\kappa_{\Sigma^*}\kappa_{\Gamma^*}, \kappa_{\Sigma^*}^3\kappa_{\Gamma^*}^2\right\}$. *Then, with probability at least* $1-1/n^{\tau-2}-5n^2/\max\{n,p\}^4$, *we have*

$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_\infty \leq 4\kappa_{\Gamma^*}\left(\bar{\delta}_f(p;n^\tau) + \bar{R}\right),$$
$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 \leq 4\kappa_{\Gamma^*}(m+1)\left(\bar{\delta}_f(p;n^\tau) + \bar{R}\right).$$

where m *is the maximum degree of the undirected network* G^*.

Appendix B. Proofs of Main Results

We collect the proofs of our main theoretical results here, including the proofs of Theorem 2 in Section 2, Proposition 3 in Section 3, and Theorem 6, Corollary 7 and Corollary 8 in Section 4.

B.1 Proof of Proposition 3

Proof First, notice that, with probability one, X has full column rank. Also, because the Cholesky factor $\widehat{L}^{(t)}$ is always positive definite for each iteration, $\widehat{L}^{(t)}X$ is in *general position* a.s. Note that the first three terms of (9) are differentiable (regular) and the whole function is continuous. Furthermore, solving (9) with respect to each variable gives a unique coordinate-wise minimum. Therefore, by Theorem 4.1 (c) in Tseng (2001), the block coordinate descent converges to a stationary point. ■

B.2 Proof of Theorem 2

Proof By Proposition 3 from Chickering (2003), there exists a finite sequence of covered edge reversals in $\mathcal{G}_1$ such that at each step $\mathcal{G}_1 \backsimeq \mathcal{G}_2$ and eventually $\mathcal{G}_1 = \mathcal{G}_2$. Hence it suffices to show the result for those $\mathcal{G}_1$ and $\mathcal{G}_2$ that differ only by one edge reversal.

Before we show the main result, let us first prove that given the same initial Θ_0, the BCD algorithm will generate the same limiting estimate $(\widehat{\Theta}, \widehat{B}, \widehat{\Omega})$. This limit point is a stationary and partial optimal point for (6) by proposition 3. To do this, it suffices to show that the sample covariance matrices calculated from (5) (appear in the trace term $\mathrm{tr}(S\Theta)$) for the two networks are equal: $S_{\mathcal{G}_1} = S_{\mathcal{G}_2}$.

Suppose X_i and X_j are two nodes in the DAGs and the edges between them have opposite directions in $\mathcal{G}_1$ and $\mathcal{G}_2$. We assume the nodes follow a topological order, and let Z denote the common parents of X_i and X_j. The sample covariance matrix $S(\widehat{B}, \widehat{\Omega})$ is defined for a DAG $\mathcal{G}$ as the following:

$$S_{\mathcal{G}} = \sum_{k=1}^{p} \frac{1}{(\widehat{\omega}_k^{\mathcal{G}})^2} \varepsilon_k^{\mathcal{G}} \varepsilon_k^{\mathcal{G}\top},$$

where ε_k is the residual after projecting X_k onto its parents (given by a DAG). Also, $\widehat{\omega}_k^2 = \|\varepsilon_k^{\mathcal{G}}\|_2^2/n$. In our case, we simply need to show that

$$\frac{1}{(\widehat{\omega}_i^{\mathcal{G}_1})^2} \varepsilon_i^{\mathcal{G}_1} \varepsilon_i^{\mathcal{G}_1\top} + \frac{1}{(\widehat{\omega}_j^{\mathcal{G}_1})^2} \varepsilon_j^{\mathcal{G}_1} \varepsilon_j^{\mathcal{G}_1\top} = \frac{1}{(\widehat{\omega}_i^{\mathcal{G}_2})^2} \varepsilon_i^{\mathcal{G}_2} \varepsilon_i^{\mathcal{G}_2\top} + \frac{1}{(\widehat{\omega}_j^{\mathcal{G}_2})^2} \varepsilon_j^{\mathcal{G}_2} \varepsilon_j^{\mathcal{G}_2\top}, \tag{34}$$

because all other terms in the summation are equal for both DAGs.
Let $X_i^{\perp}, X_j^{\perp}$ be the respective residuals in the projection of X_i and X_j to the span(Z), and $\tilde{X}_i^{\perp}, \tilde{X}_j^{\perp}$ be the normalized $X_i^{\perp}$ and $X_j^{\perp}$. Then if we let $X_i \to X_j$ in $\mathcal{G}_1$ and $X_j \to X_i$ in $\mathcal{G}_2$, we have

$$\text{LHS of (34)} = \tilde{X}_i^{\perp} \tilde{X}_i^{\perp\top} + \left(\frac{X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp} \rangle \tilde{X}_i^{\perp}}{\|X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp} \rangle \tilde{X}_i^{\perp}\|} \right) \cdot \left(\frac{X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp} \rangle \tilde{X}_i^{\perp}}{\|X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp} \rangle \tilde{X}_i^{\perp}\|} \right)^{\top}. \tag{35}$$

Denote $\langle \tilde{X}_i^{\perp}, \tilde{X}_j^{\perp} \rangle = \cos\theta$, and notice that

$$
\begin{aligned}
(35) &= \tilde{X}_i^{\perp}\tilde{X}_i^{\perp\top} + \frac{\left(X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp}\rangle \tilde{X}_i^{\perp}\right)\cdot\left(X_j^{\perp} - \langle X_j^{\perp}, \tilde{X}_i^{\perp}\rangle \tilde{X}_i^{\perp}\right)^{\top}}{\|X_j^{\perp}\|^2 \sin^2\theta} \\
&= \tilde{X}_i^{\perp}\tilde{X}_i^{\perp\top} + \frac{\tilde{X}_j^{\perp}\tilde{X}_j^{\perp\top}}{\sin^2\theta} + \frac{\tilde{X}_i^{\perp}\tilde{X}_i^{\perp\top}\cos^2\theta}{\sin^2\theta} + \text{A shared term} \\
&= \frac{\tilde{X}_j^{\perp}\tilde{X}_j^{\perp\top} + \tilde{X}_i^{\perp}\tilde{X}_i^{\perp\top}}{\sin^2\theta} + \text{A shared term.} \qquad (36)
\end{aligned}
$$

Since (36) is symmetric in i and j, we have that LHS of (34) = RHS of (34). In other words, given the same initial $\widehat{\Theta}_0$, the iterative algorithm will generate the same sequence of $(\widehat{B}, \widehat{\Omega}, \widehat{\Theta})$, thus the same limiting point.

Note that the MLE estimates for $\mathcal{G}_1$ and $\mathcal{G}_2$ are also limiting points given some initial Θ. Let us suppose the MLE exists and $(\widehat{\Theta}_1, \widehat{B}_1), (\widehat{\Theta}_2, \widehat{B}_2)$ are the MLEs for $\mathcal{G}_1, \mathcal{G}_2$, respectively, then according to the results above we have

$$
L_{\mathcal{G}_1}(\widehat{\Theta}_1, \widehat{B}_1(\mathcal{G}_1)) = L_{\mathcal{G}_2}(\widehat{\Theta}_1, \widehat{B}_1(\mathcal{G}_2)) \leq L_{\mathcal{G}_2}(\widehat{\Theta}_2, \widehat{B}_2(\mathcal{G}_2)) = L_{\mathcal{G}_1}(\widehat{\Theta}_2, \widehat{B}_2(\mathcal{G}_1)).
$$

Therefore,

$$
L_{\mathcal{G}_1}(\widehat{\Theta}_1, \widehat{B}_1(\mathcal{G}_1)) = L_{\mathcal{G}_2}(\widehat{\Theta}_2, \widehat{B}_2(\mathcal{G}_2)).
$$

Thus, all MLEs for $\mathcal{G}_1$ yield the same likelihood value which is equal to the likelihood value of any MLE for $\mathcal{G}_2$. ■

B.3 Proof of Theorem 6

Proof We first prove the consistency in $\widehat{B}^{(1)}$. Under Assumption 1, Theorem 14 shows that for the given λ_n, the empirical process term of the noises can be uniformly bounded with high probability. Therefore, in order to obtain the conclusion in Theorem 16, we only need the inequality (26) in Lemma 15 to hold. Since the event $\mathscr{K}$ in (25) holds with high probability by Theorem 7.16 in Wainwright (2019), (26) holds by Lemma 15. Next, we show $\widehat{\Theta}^{(1)}$ is consistent by invoking Theorem 22. For the chosen λ_p and under the constraint on (n, p) specified in (18), the sample size requirement in (33) is satisfied. Therefore, the results follow from Theorem 22. Combining Theorem 16 and 22, we get the desired results. ■

B.4 Proof of Corollary 7

Proof The rate of $\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2$ follows directly from the choice of $\lambda_n \asymp \sqrt{\frac{\log p}{n}}$. Since $r(\widehat{\Omega}) = O_p\left(s\sqrt{\frac{\log p}{N}}\right)$ and $p \gg n$,

$$\begin{aligned}
\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 &= O_p\left(m\left(\sqrt{\frac{\log n}{p}} + s\max\left\{\sqrt{\frac{\log p}{N}}, \sqrt{\frac{\log^3 p}{n}}\right\}\right)\right) \\
&= O_p\left(ms\max\left\{\sqrt{\frac{\log p}{N}}, \sqrt{\frac{\log^3 p}{n}}\right\}\right) \quad (n \gtrsim N) \\
&= O_p\left(ms\sqrt{\frac{\log^3 p}{n}}\right) \quad (N\log^2 p \gtrsim n).
\end{aligned}$$

■

B.5 Proof of Corollary 8

Proof The rate of $\sup_j \|\hat{\beta}_j^{(1)} - \beta_j^*\|_2^2$ can be derived in the same way as in the proof of Corollary 7. Since $r(\widehat{\Omega}) = O_p\left(s\sqrt{\frac{\log p}{N}}\right)$ and $n \gg p$,

$$\begin{aligned}
\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 &= O_p\left(m\left(\sqrt{\frac{\log n}{p}} + s\max\left\{\sqrt{\frac{\log p}{N}}, \sqrt{\frac{\log p\log^2 n}{n}}\right\}\right)\right) \\
&= O_p\left(m\left(\sqrt{\frac{\log n}{p}} + s\max\left\{\sqrt{\frac{\log p}{N}}, \sqrt{\frac{\log p\log^2 n}{n}}\right\}\right)\right)
\end{aligned}$$

$N \gtrsim s^2 p \implies \sqrt{\frac{\log n}{p}} \gtrsim \sqrt{\frac{s^2\log p}{N}}$ and $n \gg s^2 p\log p\log n \implies \sqrt{\frac{\log n}{p}} \gtrsim \sqrt{\frac{s^2\log p\log^2 n}{n}}$. Thus,

$$\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 = O_p\left(m\sqrt{\frac{\log n}{p}}\right).$$

■

Appendix C. Proofs of Intermediate Results for Theorem 6

We include the proofs for all the intermediates results that lead to Theorem 6 in this section.

C.1 Proof of Theorem 14

Proof For any $j = 1, \ldots, p$,

$$\begin{aligned}\frac{2}{n}\|X^\top \widehat{\Theta}^{(0)}\varepsilon_j\|_\infty &= \frac{2}{n}\|X^\top(\Theta^* + \widehat{\Delta}^{(0)}_{prec})\varepsilon_j\|_\infty \leq \frac{2}{n}\|\widetilde{X}^\top\tilde{\varepsilon}_j\|_\infty + \frac{2}{n}\|X^\top\widehat{\Delta}^{(0)}_{prec}\varepsilon_j\|_\infty \\ &\leq \frac{2}{n}\|\widetilde{X}^\top\tilde{\varepsilon}_j\|_\infty + \frac{2}{n}\|\widetilde{X}^\top L^{*-\top}\widehat{\Delta}^{(0)}_{prec}L^{*-1}\tilde{\varepsilon}_j\|_\infty.\end{aligned}$$

Let $\widehat{K}^{(0)} = L^{*-\top}\widehat{\Delta}^{(0)}_{prec}L^{*-1}$. Then following Assumption 1,

$$\|\widehat{K}^{(0)}\|_2 \leq \|\widehat{\Delta}^{(0)}_{prec}\|_2/\sigma^2_{\min}(L^*) \leq M/\sigma^2_{\min}(L^*).$$

Under event $\mathscr{E}$ defined in (22), for $j \in [p]$,

$$\|\widehat{K}^{(0)}\widetilde{X}_j\|_2 \leq 6\bar{\psi}M\sqrt{n}/\sigma^2_{\min}(L^*) \leq 6\bar{\psi}\sqrt{n}.$$

For $j \in [p]$, define the event $\overline{\mathscr{T}}_j$ as the following

$$\overline{\mathscr{T}}_j := \left\{2\|\widetilde{X}^\top\widehat{K}^{(0)}\tilde{\varepsilon}_j\|_\infty/n < \lambda_n/2\right\}.$$

Similar to the proof of Lemma 12, we can show

$$\mathbb{P}\left(\bigcup_{j=1}^p \overline{\mathscr{T}}_j^c \mid \mathscr{E}\right) \leq \frac{1}{p},$$

and

$$\mathbb{P}\left(\overline{\mathscr{T}}\bigcap\mathscr{T}\right) \geq \mathbb{P}\left(\overline{\mathscr{T}}\bigcap\mathscr{T} \mid \mathscr{E}\right)\mathbb{P}(\mathscr{E}) \geq (1-\frac{1}{p})(1-\frac{2}{p}) \geq (1-\frac{2}{p})^2,$$

where $\mathscr{T}$ is defined in Lemma 12. ■

C.2 Proof of Lemma 15

Proof We observe that

$$\begin{aligned}\|\widehat{L}^{(0)}X\theta\|_2/\sqrt{n} &\geq \|L^*X\theta\|_2\sqrt{n} - \|\widehat{\Delta}^{(0)}_{chol}X\theta\|_2/\sqrt{n} \\ &= \|L^*X\theta\|_2/\sqrt{n} - \|\widehat{\Delta}^{(0)}_{chol}L^{*-1}L^*X\theta\|_2/\sqrt{n} \\ &\geq \left(1 - \|\widehat{\Delta}^{(0)}_{chol}\|_2/\sigma_{\min}(L^*)\right)\|L^*X\theta\|_2/\sqrt{n} \\ &\geq \left(1 - \frac{M}{2\sigma^2_{\min}(L^*)}\right)\|\widetilde{X}\theta\|_2/\sqrt{n} \quad \text{(By Assumption 1 and Lemma 10)} \\ &\geq \frac{1}{2\sqrt{n}}\|\widetilde{X}\theta\|_2,\end{aligned}$$

when n is sufficiently large. Since event $\mathscr{K}$ defined in (25) holds, by Theorem 7.16 in Wainwright (2019), we have

$$\begin{aligned}\frac{\|\widehat{L}^{(0)}X\theta\|_2^2}{n} &\geq \frac{1}{4}\left(\tilde{c}_1\|\sqrt{\Psi^*}\theta\|_2^2 - \tilde{c}_2\rho^2(\Psi^*)\frac{\log p}{n}\|\theta\|_1^2\right)\\ &\geq c_1\|\sqrt{\Psi^*}\theta\|_2^2 - c_2\rho^2(\Psi^*)\frac{\log p}{n}\|\theta\|_1^2.\end{aligned}$$

■

C.3 Proof of Theorem 16

Proof Consider the penalized negative likelihood function from (10):

$$\mathcal{L}(\beta_j, \lambda_n) = \frac{1}{2n}\|\widehat{L}^{(0)}X_j - \widehat{L}^{(0)}X\beta_j\|_2^2 + \lambda_n\|\beta_j\|_1.$$

For simplicity, we drop the superscript (t) in $\hat{\beta}^{(1)}$ and $\widehat{L}^{(0)}$. Let ρ stand for $\rho(\Psi^*)$, $\beta_j^* \in \mathbb{R}^p$, and $\widehat{\Delta}_j = \hat{\beta}_j - \beta_j^*$. We start from the *basic inequality* (Wainwright, 2019):

$$\mathcal{L}(\hat{\beta}_j, \lambda_n) \leq \mathcal{L}(\beta_j^*, \lambda_n) = \frac{1}{2n}\|\widehat{L}\varepsilon_j\|_2^2 + \lambda_n\|\beta_j^*\|_1.$$

After rearranging some terms,

$$0 \leq \frac{1}{2n}\|\widehat{L}X\widehat{\Delta}_j\|_2^2 \leq \frac{\varepsilon_j^\top\widehat{\Theta}X\widehat{\Delta}_j}{n} + \lambda_n\left(\|\beta_j^*\|_1 - \|\hat{\beta}_j\|_1\right). \tag{37}$$

Next, for any subset $S \subseteq [p]$, we have

$$\|\beta_j^*\|_1 - \|\hat{\beta}_j\|_1 = \|\beta_{j,S}^*\|_1 + \|\beta_{j,S^c}^*\|_1 - \|\beta_{j,S}^* + \widehat{\Delta}_{j,S}\|_1 - \|\widehat{\Delta}_{j,S^c} + \beta_{j,S^c}^*\|_1. \tag{38}$$

Combined (37) with (38), and apply triangle and Hölder's inequalities,

$$\begin{aligned}0 \leq \frac{1}{2n}\|\widehat{L}X\widehat{\Delta}_j\|_2^2 &\leq \frac{1}{n}\varepsilon_j^\top\widehat{\Theta}X\widehat{\Delta}_j + \lambda_n\left(\|\widehat{\Delta}_{j,S}\|_1 - \|\widehat{\Delta}_{j,S^c}\|_1 + 2\|\beta_{j,S^c}^*\|_1\right)\\ &\leq \|X^\top\widehat{\Theta}\varepsilon_j\|_\infty/n\|\widehat{\Delta}_j\|_1 + \lambda_n\left(\|\widehat{\Delta}_{j,S}\|_1 - \|\widehat{\Delta}_{j,S^c}\|_1 + 2\|\beta_{j,S^c}^*\|_1\right)\\ &\leq \frac{\lambda_n}{2}\left(\|\widehat{\Delta}_j\|_1 + 2\|\widehat{\Delta}_{j,S}\|_1 - 2\|\widehat{\Delta}_{j,S^c}\|_1 + 4\|\beta_{j,S^c}^*\|_1\right)\\ &\leq \frac{\lambda_n}{2}\left[3\|\widehat{\Delta}_{j,S}\|_1 - \|\widehat{\Delta}_{j,S^c}\|_1 + 4\|\beta_{j,S^c}^*\|_1\right], \qquad (39)\\ \|\widehat{\Delta}_j\|_1 &\leq 4\left(\|\widehat{\Delta}_{j,S}\|_1 + \|\beta_{j,S^c}^*\|_1\right).\end{aligned}$$

This inequality implies (apply Cauchy-Schwarz inequality)

$$\|\widehat{\Delta}_j\|_1^2 \leq \left(4\|\widehat{\Delta}_{j,S}\|_1 + 4\|\beta_{j,S^c}^*\|_1\right)^2 \leq 32\left(|S|\ \|\widehat{\Delta}_j\|_2^2 + \|\beta_{j,S^c}^*\|_1^2\right). \tag{40}$$

Next, from (26) and (40), we know,

$$\begin{aligned}\frac{\|\widehat{L}X\widehat{\Delta}_j\|_2^2}{n} &\geq \left(c_1\bar{\kappa} - 32c_2\rho^2|S|\frac{\log p}{n}\right)\|\widehat{\Delta}_j\|_2^2 - 32c_2\rho^2\frac{\log p}{n}\|\beta^*_{j,S^c}\|_1^2 \\ &\geq \frac{c_1\bar{\kappa}}{2}\|\widehat{\Delta}_j\|_2^2 - 32c_2\rho^2\frac{\log p}{n}\|\beta^*_{j,S^c}\|_1^2, \end{aligned} \tag{41}$$

where the last inequality comes from the condition $|S| \leq \frac{c_1}{64c_2}\frac{\bar{\kappa}}{\rho^2(\Psi^*)}\frac{n}{\log p}$. Now let's analyze the following two cases regarding (41):

Case 1 *Suppose that* $\frac{c_1\bar{\kappa}}{4}\|\widehat{\Delta}_j\|_2^2 \geq 32c_2\rho^2\frac{\log p}{n}\|\beta^*_{j,S^c}\|_1^2$*, then from* (39) *we can get*

$$\frac{c_1\bar{\kappa}}{4}\|\widehat{\Delta}_j\|_2^2 \leq \lambda_n\left(3\sqrt{|S|}\|\widehat{\Delta}_j\|_2 + 4\|\beta^*_{j,S^c}\|_1\right).$$

Solving for the zeros of this quadratic form in $\|\widehat{\Delta}_j\|_2$ *yields*

$$\|\widehat{\Delta}_j\|_2^2 \leq \frac{48\lambda_n^2}{c_1^2\bar{\kappa}^2}|S| + \frac{16\lambda_n\|\beta^*_{j,S^c}\|_1}{c_1\bar{\kappa}}.$$

Case 2 *Otherwise, we have* $\frac{c_1\bar{\kappa}}{4}\|\widehat{\Delta}_j\|_2^2 \leq 32c_2\rho^2\frac{\log p}{n}\|\beta^*_{j,S^c}\|_1^2$.

After combining the two cases, we obtain the claimed bound in (27). To get the prediction bound in (28), we first show $\|\widehat{\Delta}_j\|_1 \leq 4\|\beta^*_j\|_1$. From basic inequality, we have

$$0 \leq \frac{1}{2n}\|\widehat{L}X\widehat{\Delta}_j\|_2^2 \leq \frac{\varepsilon_j^\top\widehat{\Theta}X\widehat{\Delta}_j}{n} + \lambda_n\left(\|\beta^*_j\|_1 - \|\hat{\beta}_j\|_1\right).$$

By Hölder's inequality and Theorem 14,

$$\left|\frac{\varepsilon_j^\top\widehat{\Theta}X\widehat{\Delta}_j}{n}\right| \leq \left\|\frac{X^\top\widehat{\Theta}\varepsilon_j}{n}\right\|_\infty\|\widehat{\Delta}_j\|_1 \leq \frac{\lambda_n}{2}\left(\|\beta^*_j\|_1 + \|\hat{\beta}_j\|_1\right).$$

Combine the two inequalities above, we get

$$0 \leq \frac{3\lambda_n}{2}\|\beta^*_j\|_1 - \frac{\lambda_n}{2}\|\hat{\beta}_j\|_1,$$

which implies $\|\hat{\beta}_j\|_1 \leq 3\|\beta^*_j\|_1$. Consequently, we have

$$\|\widehat{\Delta}_j\|_1 \leq \|\hat{\beta}_j\|_1 + \|\beta^*_j\|_1 \leq 4\|\beta^*_j\|_1.$$

Return to the basic inequality, we have

$$\begin{aligned}\frac{\|\widehat{L}X\widehat{\Delta}_j\|_2^2}{2n} &\leq \frac{\lambda_n}{2}\|\widehat{\Delta}_j\|_1 + \lambda_n\left(\|\beta^*_j\|_1 - \|\beta^*_j + \widehat{\Delta}_j\|_1\right) \\ &\leq \frac{3}{2}\lambda_n\|\widehat{\Delta}_j\|_1 \leq 6\lambda_n\|\beta^*_j\|_1.\end{aligned}$$

■

C.4 Proof of Lemma 18

Proof The proof follows a similar approach as the proof for Lemma 1 in Ravikumar et al. (2011). ■

C.5 Proof of Corollary 19

Proof A little calculation using Lemma 18 and Definition 17 shows that the corresponding inverse functions for data from the Gaussian DAG model (2) are:

$$\bar{\delta}_f(p;r) = 40\sqrt{\frac{2\log(4r)}{p}}, \quad \text{and} \quad \bar{p}_f(\delta;r) = \frac{3200\log(4r)}{\delta^2}.$$

Setting $r = n^\tau$ yields the desired result. ■

C.6 Proof of Lemma 21

Proof Let $\widehat{S}_{ij}^{(t)}$ denote the (i,j) entry of the sample variance matrix $\widehat{S}^{(t)}$ defined in (11). Let $X_{i\cdot}$ and $X_{\cdot j}$ denote the ith row and jth column of X, respectively. Let $\varepsilon_{ik}^* := X_{ik} - X_{i\cdot}\beta_k^*$ where β_k^* is the kth column of B^*, $\rho_k^* = 1/\omega_k^{*2}$, $\hat{\rho}_k = 1/\hat{\omega}_k^2$, $\widehat{\Delta}_k^{(t)} := \hat{\beta}_k^{(t)} - \beta_k^*$, and $\hat{\delta}_k = \hat{\rho}_k - \rho_k^*$. Then,

$$\begin{aligned}
\widehat{S}_{ij}^{(1)} &= \frac{1}{p}\sum_{k=1}^{p} \hat{\rho}_k \left(X_{ik} - X_{i\cdot}\hat{\beta}_k^{(1)}\right)\left(X_{jk} - X_{j\cdot}\hat{\beta}_k^{(1)}\right) \\
&= \frac{1}{p}\sum_{k=1}^{p} \hat{\rho}_k \left(\varepsilon_{ik}^* - X_{i\cdot}\widehat{\Delta}_k^{(1)}\right)\left(\varepsilon_{jk}^* - X_{j\cdot}\widehat{\Delta}_k^{(1)}\right) \\
&= \widehat{\Sigma}_{ij} + \frac{1}{p}\sum_{k=1}^{p} \rho_k^* \left(-\varepsilon_{ik}^* X_{j\cdot}\widehat{\Delta}_k^{(1)} - \varepsilon_{jk}^* X_{i\cdot}\widehat{\Delta}_k^{(1)} + X_{i\cdot}\widehat{\Delta}_k^{(1)} X_{j\cdot}\widehat{\Delta}_k^{(1)}\right) \\
&\quad + \frac{1}{p}\sum_{k=1}^{p} \hat{\delta}_k \left(\varepsilon_{ik}^* - X_{i\cdot}\widehat{\Delta}_k^{(1)}\right)\left(\varepsilon_{jk}^* - X_{j\cdot}\widehat{\Delta}_k^{(1)}\right) \\
&= \widehat{\Sigma}_{ij} + \frac{1}{p}\sum_{k=1}^{p} \hat{\rho}_k \left(-\varepsilon_{ik}^* X_{j\cdot}\widehat{\Delta}_k^{(1)} - \varepsilon_{jk}^* X_{i\cdot}\widehat{\Delta}_k^{(1)} + X_{i\cdot}\widehat{\Delta}_k^{(1)} X_{j\cdot}\widehat{\Delta}_k^{(1)}\right) + \frac{1}{p}\sum_{k=1}^{p} \hat{\delta}_k \varepsilon_{ik}^* \varepsilon_{jk}^*.
\end{aligned}$$

If we let $R_{ij} = \frac{1}{p}\sum_{k=1}^{p} \hat{\rho}_k \left(-\varepsilon_{ik}^* X_{j\cdot}\widehat{\Delta}_k^{(1)} - \varepsilon_{jk}^* X_{i\cdot}\widehat{\Delta}_k^{(1)} + X_{i\cdot}\widehat{\Delta}_k^{(1)} X_{j\cdot}\widehat{\Delta}_k^{(1)}\right) + \frac{1}{p}\sum_{k=1}^{p} \hat{\delta}_k \varepsilon_{ik}^* \varepsilon_{jk}^*$, we can upper bound $|R_{ij}|$ by dividing it into three terms and controlling each term separately.

Part 1.

We observe that

$$\begin{aligned}\left|\frac{1}{p}\sum_{k=1}^{p}\hat{\rho}_k\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right| &= \left|\frac{1}{p}\sum_{k=1}^{p}\left(\rho_k^{*}+\hat{\delta}_k\right)\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right| \\ &\leq \left|\frac{1}{p}\sum_{k=1}^{p}\rho_k^{*}\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right| + \left|\frac{1}{p}\sum_{k=1}^{p}\hat{\delta}_k\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right|.\end{aligned}$$

If $\widehat{\Delta}_k^{(1)}$ and $\hat{\delta}_k$ satisfy (32), both ρ_k^{*} and $\hat{\delta}_k$ can be bounded by positive constants ($r(\widehat{\Omega}) \ll 1$). Define the following events:

$$\begin{aligned}\mathcal{B}_1 &= \bigcup_{i=1}^{n}\left\{\|\varepsilon_{i\cdot}\|_\infty \geq 6\bar{\omega}\sqrt{\log\max\{n,p\}}\right\}, \\ \mathcal{B}_2 &= \bigcup_{k=1}^{n}\left\{\|\varepsilon_{k\cdot}^{*}\|_2 \geq 6\bar{\omega}\sqrt{p}\right\}, \\ \mathcal{B}_3 &= \bigcup_{k=1}^{n}\left\{\|X_{k\cdot}\|_\infty \geq 6\bar{\psi}\sqrt{\log\max\{n,p\}}\right\}.\end{aligned}$$

Under event $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$,

$$\begin{aligned}\left|\frac{1}{p}\sum_{k=1}^{p}\hat{\rho}_k\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right| &\leq \frac{2}{b}\sup_k|\varepsilon_{ik}^{*}X_{j\cdot}\widehat{\Delta}_k^{(1)}| \\ &\leq \frac{12\bar{\omega}}{b}\sqrt{\log\max\{n,p\}}\sup_k\|X_{j\cdot}\|_\infty\|\widehat{\Delta}_k^{(1)}\|_1 \quad \text{(By Hölder's Inequality and } \mathcal{B}_1\text{)} \\ &\leq \frac{12\bar{\omega}s\sqrt{\log p\log\max\{n,p\}}}{b\sqrt{n}}\|X_{j\cdot}\|_\infty \quad \text{(From } \|\widehat{\Delta}_k^{(1)}\|_1 \leq 4\sqrt{2s}\|\widehat{\Delta}_k^{(1)}\|_2\text{)} \\ &\leq \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p\log^2\max\{n,p\}}{n}} \quad \text{(By event } \mathcal{B}_3\text{)}.\end{aligned}$$

The second last inequality comes from $\|\widehat{\Delta}^{(1)}\|_1 \leq 4\sqrt{2s}\|\widehat{\Delta}^{(1)}\|_2$ in the proof of Theorem 16.

Part 2
Notice that

$$\begin{aligned}\left|\frac{1}{p}\sum_{k=1}^{p}\hat{\rho}_kX_{i\cdot}\widehat{\Delta}_k^{(1)}X_{j\cdot}\widehat{\Delta}_k^{(1)}\right| &\leq \frac{2}{b}\sup_k|X_{j\cdot}\widehat{\Delta}_k^{(1)}||X_{i\cdot}\widehat{\Delta}_k^{(1)}| \\ &\leq \frac{2s^2\log p}{bn}\|X_{j\cdot}\|_\infty\|X_{i\cdot}\|_\infty \\ &\leq \frac{12\bar{\psi}s^2}{b}\sqrt{\frac{\log^2 p\log^2\max\{n,p\}}{n^2}}.\end{aligned}$$

Part 3

By Lemma 5, $\|\hat{\delta}\|_\infty = \sup_k |\hat{\rho}_k - \rho_k^*| = \sup_k |1/\hat{\omega}_k^2 - 1/\omega_k^{*2}| = r(\widehat{\Omega})$. Combining with $\mathcal{B}_2$, we have

$$\left|\frac{1}{p}\sum_{k=1}^{p}\hat{\delta}_k\varepsilon_{ik}^*\varepsilon_{jk}^*\right| \leq \frac{1}{p}\|\hat{\delta}\|_\infty \sum_{k=1}^{p}|\varepsilon_{ik}^*\varepsilon_{jk}^*| \leq \frac{1}{p}\|\delta\|_\infty \|\varepsilon_{i\cdot}^*\|_2\|\varepsilon_{j\cdot}^*\|_2 \leq 6\bar{\omega}\|\hat{\delta}\|_\infty = 6\bar{\omega}r(\widehat{\Omega}).$$

Combine all three parts, we have

$$|R_{ij}| \leq \max\left\{6\bar{\omega}r(\widehat{\Omega}), \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p \log^2 \max\{n,p\}}{n}}\right\}.$$

Using Lemma 13 and Lemma 11, we can derive the upper bound for the probabilities of $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$:

$$\begin{aligned}
&\mathbb{P}(\mathcal{B}_1) \leq 2/\max\{n,p\}^4,\\
&\mathbb{P}(\mathcal{B}_2) \leq 1/\max\{n,p\}^4 \quad \text{if } n > 2\sqrt{10}\log\max\{n,p\},\\
&\mathbb{P}(\mathcal{B}_2) \leq 2/\max\{n,p\}^4,\\
&\mathbb{P}\left(\bigcup_{l=1}^{3}\mathcal{B}_i\right) \leq 5/\max\{n,p\}^4.
\end{aligned}$$

Applying union bound,

$$\|R\|_\infty \leq \max\left\{6\bar{\omega}r(\widehat{\Omega}), \frac{72\bar{\omega}\bar{\psi}s}{b}\sqrt{\frac{\log p \log^2 \max\{n,p\}}{n}}\right\},$$

with probability at least $1 - \frac{5n^2}{\max\{n,p\}^4}$. Take event $\mathcal{A}$ from Lemma 20 into account and apply union bound one more time, we arrive at the desired conclusion. ■

C.7 Proof of Theorem 22

Proof Let $\bar{R}(s,p,n)$ and $\bar{\delta}_f(p;n^\tau)$ be defined as stated, then the monotonicity of the inverse tail function (29) and condition (33) on (n,p) implies that $\bar{\delta}_f(p;n^\tau) \leq 1/40$. Lemma 20 and Lemma 21 imply that the event $\mathcal{A}$ defined in (31) and the events $\mathcal{B}_1$,$\mathcal{B}_2$,$\mathcal{B}_3$ defined in the proof of Lemma 21 hold with high probability. Conditioning on these events, we have

$$\|\widehat{W}^{(1)}\|_\infty \leq \bar{\delta}_f(p;n^\tau) + \bar{R}(s,p,n).$$

Choose $\lambda_p = \bar{\delta}_f(p;n^\tau) + \bar{R}$. By Lemma 21 and condition (33) we have that

$$2\kappa_{\Gamma^*}\left(\|\widehat{W}^{(1)}\|_\infty + \lambda_p\right) \leq 4\kappa_{\Gamma^*}\left(\bar{\delta}_f(p;n^\tau) + \bar{R}\right) \leq \min\left\{\frac{1}{3\kappa_{\Sigma^*}m}, \frac{1}{3\kappa_{\Sigma^*}^3 2\kappa_{\Gamma^*}m}\right\}.$$

Applying Lemma 6 in Ravikumar et al. (2011) we obtain

$$\begin{aligned}
&\|\widehat{\Theta}^{(1)} - \Theta^*\|_\infty \leq 4\kappa_{\Gamma^*}\left(\bar{\delta}_f(p;n^\tau) + \bar{R}\right),\\
&\|\widehat{\Theta}^{(1)} - \Theta^*\|_2 \leq \|A\|_2\|\widehat{\Theta}^{(1)} - \Theta^*\|_\infty \leq (m+1)\|\widehat{\Theta}^{(1)} - \Theta^*\|_\infty.
\end{aligned}$$

■

Appendix D. Proofs of Other Auxiliary Results

This section includes the proofs for Lemma 4 and 5 as well as the four lemmas introduced in Section A.1.

D.1 Proof of Lemma 4

Proof From Lemma 1 in Yu and Bien (2019) we know that if $\lambda_N \geq N^{-1}\|X^{(B)\top}\varepsilon_j^{(B)}\|_\infty$, then

$$\left|\hat{\omega}_j^2 - N^{-1}\|\varepsilon_j^{(B)}\|_2^2\right| \leq 2\lambda_N\|\beta_j^*\|_1 \leq \lambda_N s\bar{\beta}.$$

Pick $\lambda_N = 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{N}} + \sqrt{\frac{2\log 2 + 6\log p}{N}}\right)$, then applying the Chernoff bound for sub-Gaussian random variables (e.g., see proof of Lemma 12) we can show that

$$\lambda_N \geq \sup_j N^{-1}\|X^{(B)\top}\varepsilon_j^{(B)}\|_\infty.$$

holds with probability at least $(1-\frac{1}{p})^2$. This proves the first inequality. To prove the second inequality, notice that

$$\begin{aligned}\sup_j\left|\hat{\omega}_j^2 - \omega_j^{*2}\right| &= \sup_j\left|\hat{\omega}_j^2 - \frac{\|\varepsilon_j^{(B)}\|_2^2}{N} + \frac{\|\varepsilon_j^{(B)}\|_2^2}{N} - \omega_j^{*2}\right| \\ &\leq \sup_j\left|\hat{\omega}_j^2 - \frac{\|\varepsilon_j^{(B)}\|_2^2}{N}\right| + \sup_j\left|\frac{\|\varepsilon_j^{(B)}\|_2^2}{N} - \omega_j^{*2}\right|.\end{aligned}$$

From χ^2 concentration inequality, (e.g. Wainwright 2019 Example 2.11)

$$\begin{aligned}\left|\omega_j^{*2} - \frac{1}{N}\|\varepsilon_j^{(B)}\|_2^2\right| &\geq 2\sqrt{2}\bar{\omega}\sqrt{\frac{\log 2 + 3\log p}{N}} \quad \text{with probability at most } 1/p^3, \\ \sup_j\left|\omega_j^{*2} - \frac{1}{N}\|\varepsilon_j^{(B)}\|_2^2\right| &\geq 2\sqrt{2}\bar{\omega}\sqrt{\frac{\log 2 + 3\log p}{N}} \quad \text{with probability at most } 1/p^2.\end{aligned}$$

Combining all the inequalities, we can show that the second inequality holds with probability at least $(1-1/p)^2 - 1/p$. ■

D.2 Proof of Lemma 5

Proof Simply notice that

$$\sup_{1\leq j\leq p}\left|\frac{1}{\hat{\omega}_j^2} - \frac{1}{\omega_j^{*2}}\right| = \sup_{1\leq j\leq p}\left|\frac{1}{\hat{\omega}_j^2\omega_j^{*2}}\right| \sup_{1\leq j\leq p}\left|\omega_j^{*2} - \hat{\omega}_j^2\right| \leq \frac{1}{b^4}\sup_{1\leq j\leq p}|\omega_j^{*2} - \hat{\omega}_j^2|.$$

■

D.3 Proof of Lemma 10

Proof The claim follows since

$$\begin{aligned}
\|\widehat{\Delta}_{prec}\|_2 &= \|\widehat{\Theta} - \Theta^*\|_2 \\
&= \| \left(L^* + \widehat{\Delta}_{chol}\right)^\top \left(L^* + \widehat{\Delta}_{chol}\right) - L^{*\top} L^* \|_2 \\
&= \|L^{*\top} \widehat{\Delta}_{chol} + \widehat{\Delta}_{chol}^\top L^* + \widehat{\Delta}_{chol}^\top \widehat{\Delta}_{chol}\|_2 \\
&\geq \max_{x \in \mathbb{S}^{n-1}} x^\top \left(L^{*\top} \widehat{\Delta}_{chol} + \widehat{\Delta}_{chol}^\top L^* + \widehat{\Delta}_{chol}^\top \widehat{\Delta}_{chol}\right) x \\
&\geq \max_{x \in \mathbb{S}^{n-1}} x^\top \left(L^{*\top} \widehat{\Delta}_{chol} + \widehat{\Delta}_{chol}^\top L^*\right) x \quad (\text{as } \widehat{\Delta}_{chol}^\top \widehat{\Delta}_{chol} \succcurlyeq 0.) \\
&= \|L^{*\top} \widehat{\Delta}_{chol} + \widehat{\Delta}_{chol}^\top L^*\|_2 \\
&\geq 2\sigma_{\min}\left(L^*\right) \|\widehat{\Delta}_{chol}\|_2.
\end{aligned}$$

■

D.4 Proof of Lemma 11

Proof Notice that $\widetilde{X}_k \in \mathbb{R}^n$ is a sub-Gaussian random vector with variance smaller than $\bar{\psi}$. By Theorem 1.19 in Rigollet (2015), we have that

$$\mathbb{P}\left(\|\widetilde{X}_k\|_2 > 4\bar{\psi}\sqrt{n} + 2\bar{\psi}\sqrt{2\log(1/\delta)}\right) \leq \delta.$$

Setting $\delta = 1/p^\alpha$ and using union bound we obtain the desired conclusion. ■

D.5 Proof of Lemma 12

Proof In other words, with probability at least $1 - 1/p$,

$$\|\widetilde{X}_k\|_2 \leq 4\bar{\psi}\sqrt{n} + 2\bar{\psi}\sqrt{2\log p} \leq 6\bar{\psi}\sqrt{n}, \tag{42}$$

for all k. Under event the $\mathscr{E}$ defined in (22), $\|\widetilde{X}_{[j-1]}^\top \tilde{\varepsilon}_j\|_\infty / n$ corresponds to the absolute maximum of $j-1$ zero-mean Gaussian variables, each with variance at most $36\bar{\psi}^2\bar{\omega}^2/n$. Next, we calculate the probability of the event $\mathscr{T} \cap \mathscr{E}$, where $\delta = 1/p^2$. We also let

$$\begin{aligned}
t &= \sqrt{\frac{2\log 2 + 4\log p}{n}}, \\
\lambda_n &= 12\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{n}} + t\right).
\end{aligned}$$

Because both $\widetilde{X}$ and $\tilde{\varepsilon}$ are random, we use the equivalence: $p(y) = \mathbb{E}_{p(x)}\left[p(y \mid x)\right]$ to apply the properties of fixed-design Lasso: Let $X_{[j-1]}$ denote the first $j-1$ columns in X,

$$\begin{aligned}
1 - \mathbb{P}(\mathscr{T}_j \mid \mathscr{E}) &= \mathbb{E}_{X_{[j-1]}}\mathbb{P}\left\{2\|\widetilde{X}_{[j-1]}^\top \tilde{\varepsilon}_j\|_\infty/n > \lambda_n \mid X_{[j-1]}, \mathscr{E}\right\} \\
&= \mathbb{E}_{X_{[j-1]}}\mathbb{P}\left\{2\|\widetilde{X}_{[j-1]}^\top \tilde{\varepsilon}_j\|_\infty/n > 6\bar{\psi}\bar{\omega}\left(\sqrt{\frac{2\log p}{n}} + t\right) \mid X_{[j-1]}, \mathscr{E}\right\} \\
&\leq 2\exp\left\{-\frac{nt^2}{2}\right\} = 1/p^2.
\end{aligned}$$

where in the last inequality we apply the Chernoff standard Gaussian tail bound. Hence,

$$1 - \mathbb{P}\left(\mathscr{T} \mid \mathscr{E}\right) = 1 - \mathbb{P}\left(\bigcap_{j=1}^{p} \mathscr{T}_j \mid \mathscr{E}\right) = \mathbb{P}\left(\bigcup_{j=1}^{p} \mathscr{T}_j^c \mid \mathscr{E}\right) \leq \frac{1}{p}. \tag{43}$$

Finally, by Lemma 11 with $\alpha = 2$ we get

$$\mathbb{P}(\mathscr{T}) \geq \mathbb{P}(\mathscr{E})\mathbb{P}\left(\mathscr{T} \mid \mathscr{E}\right) \geq \left(1 - \frac{1}{p}\right)^2. \tag{44}$$

■

D.6 Proof of Lemma 13

Proof By the sub-Gaussian maximal inequality (e.g., Theorem 1.14 in Rigollet 2015), we know that if $X_1, \ldots X_N$ are random variables such that $X_i \sim$ sub-Gaussian with parameter σ^2, then for any $t > 0$,

$$\mathbb{P}\left(\max_{1\leq i\leq N} |X_i| \geq t\right) \leq 2N\exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Letting $t = \sqrt{4\bar{\psi}^2\log p}$ and taking $\sigma^2 = \bar{\psi}^2$, we arrive at the desired result. ■

References

Genevera I Allen and Robert Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, 4(2):764, 2010.

Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse gaussian Bayesian networks. *Journal of Machine Learning Research*, 16:2273–2328, 2015. URL `http://jmlr.org/papers/v16/aragam15a.html`.

Bryon Aragam, Jiaying Gu, and Qing Zhou. Learning large-scale Bayesian networks with the sparsebn package. *Journal of Statistical Software, Articles*, 91(11):1–38, 2019. ISSN 1548-7660. doi: 10.18637/jss.v091.i11. URL `https://www.jstatsoft.org/v091/i11`.

Mohsen Bayati, Murat A Erdogdu, and Andrea Montanari. Estimating lasso risk and noise level. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/2b8a61594b1f4c4db0902a8a395ced93-Paper.pdf

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/23076172.

Peter Buhlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 3642201911.

Xi Chen, Han Xu, Ping Yuan, Fang Fang, Mikael Huss, Vinsensius B Vega, Eleanor Wong, Yuriy L Orlov, Weiwei Zhang, Jianming Jiang, et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell*, 133(6): 1106–1117, 2008.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, March 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717. URL https://doi.org/10.1162/153244303321897717.

Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T. Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A. Thomson. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biology*, 17(1):173, 2016. doi: 10.1186/s13059-016-1033-x. URL https://doi.org/10.1186/s13059-016-1033-x.

Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, Oct 1992. ISSN 1573-0565. doi: 10.1007/BF00994110. URL https://doi.org/10.1007/BF00994110.

Pierre Dutilleul. The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, 64(2):105–123, 1999. doi: 10.1080/00949659908811970. URL http://dx.doi.org/10.1080/00949659908811970.

Gideon E. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6, 03 1978. doi: 10.1214/aos/1176344136.

Jianqing Fan, Shaojun Guo, and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(1):37–65, 2012. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/41430928.

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.

Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for Bayesian network structure learning with application to multi-label learning. *Expert Syst. Appl.*, 41(15):6755–6772, November 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2014.04.032. URL https://doi.org/10.1016/j.eswa.2014.04.032.

Jiaying Gu, Fei Fu, and Qing Zhou. Penalized estimation of directed acyclic graphs from discrete data. *Statistics and Computing*, 29(1):161–176, 2019.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20 (3):197–243, Sep 1995. ISSN 1573-0565. doi: 10.1023/A:1022623210503. URL https://doi.org/10.1023/A:1022623210503.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. URL http://www.jstatsoft.org/v47/i11/.

Suprateek Kundu and Benjamin B. Risk. Scalable Bayesian matrix normal graphical models for brain functional networks. *Biometrics*, n/a(n/a), 2020. doi: https://doi.org/10.1111/biom.13319. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.13319.

Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature Communications*, 9(1):997, 2018. doi: 10.1038/s41467-018-03405-7. URL https://doi.org/10.1038/s41467-018-03405-7.

Teppo Niinimäki, Pekka Parviainen, and Mikko Koivisto. Structure discovery in Bayesian networks by sampling partial orders. *The Journal of Machine Learning Research*, 17(1): 2002–2048, 2016.

Tore Opsahl. *Structure and Evolution of Weighted Networks*. University of London (Queen Mary College), London, UK, 2009. URL http://toreopsahl.com/publications/thesis/.

Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31 (2):155 – 163, 2009. ISSN 0378-8733. doi: https://doi.org/10.1016/j.socnet.2009.02.002. URL http://www.sciencedirect.com/science/article/pii/S0378873309000070.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 12 1995. ISSN 0006-3444. doi: 10.1093/biomet/82.4.669. URL https://doi.org/10.1093/biomet/82.4.669.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3 (2):121–129, Mar 2017. ISSN 2364-4168. doi: 10.1007/s41060-016-0032-z. URL https://doi.org/10.1007/s41060-016-0032-z.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5(none):935 – 980, 2011. doi: 10.1214/11-EJS631. URL https://doi.org/10.1214/11-EJS631.

Philippe Rigollet. High dimensional statistics lecture notes. https://ocw.mit.edu/courses/mathematics/18-s997-high-dimensional-statistics-spring-2015/l 2015. [Online; accessed 10-December-2020].

Teemu Roos. *Minimum Description Length Principle*, pages 823–827. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi: 10.1007/978-1-4899-7687-1_894. URL https://doi.org/10.1007/978-1-4899-7687-1_894.

Mauro Scanagatta, Giorgio Corani, Cassio P de Campos, and Marco Zaffalon. Learning treewidth-bounded Bayesian networks with thousands of variables. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/e2a2dcc36a08a345332c751b2f2e476c-Paper.pdf

Mark Schmidt, Alexandru Niculescu-Mizil, and Kevin Murphy. Learning graphical model structure using l1-regularization paths. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1278–1283. AAAI Press, 2007. ISBN 978-1-57735-323-2. URL http://dl.acm.org/citation.cfm?id=1619797.1619850.

Marco Scutari. Learning Bayesian networks with the bnlearn r package. *Journal of Statistical Software, Articles*, 35(3):1–22, 2010. ISSN 1548-7660. doi: 10.18637/jss.v035.i03. URL https://www.jstatsoft.org/v035/i03.

Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, Prediction, and Search*. MIT press, 2000.

Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012. ISSN 00063444, 14643510. URL http://www.jstor.org/stable/41720740.

Jin Tian, Changsung Kang, and Judea Pearl. A characterization of interventional distributions in semi-markovian causal models. In *Proceedings of The National Conference on Artificial Intelligence*, volume 21, page 1239. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.*, 65(1): 31–78, October 2006. ISSN 0885-6125. doi: 10.1007/s10994-006-6889-7. URL https://doi.org/10.1007/s10994-006-6889-7.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3): 475–494, Jun 2001. ISSN 1573-2878. doi: 10.1023/A:1017501703105. URL https://doi.org/10.1023/A:1017501703105.

Theodoros Tsiligkaridis, Alfred O Hero III, and Shuheng Zhou. On convergence of kronecker graphical lasso algorithms. *IEEE transactions on signal processing*, 61(7):1743–1755, 2013.

Sara A. van de Geer and Peter Bühlmann. On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3(none):1360 – 1392, 2009. doi: 10.1214/09-EJS506. URL https://doi.org/10.1214/09-EJS506.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

Qiaoling Ye, Arash Amini, and Qing Zhou. Optimizing regularized cholesky score for order-based learning of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2020. doi: 10.1109/TPAMI.2020.2990820.

Guo Yu and Jacob Bien. Estimating the error variance in a high-dimensional linear model. *Biometrika*, 106(3):533–546, 05 2019. ISSN 0006-3444. doi: 10.1093/biomet/asz017. URL https://doi.org/10.1093/biomet/asz017.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

Qing Zhou, Hiram Chipperfield, Douglas A. Melton, and Wing Hung Wong. A gene regulatory network in mouse embryonic stem cells. *Proceedings of the National Academy of Sciences*, 104(42):16438–16443, 2007. ISSN 0027-8424. doi: 10.1073/pnas.0701014104. URL https://www.pnas.org/content/104/42/16438.

Shuheng Zhou. Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014.