Learning linear structural equation models in polynomial time and sample complexity

8 Appendix

A Detailed Proofs

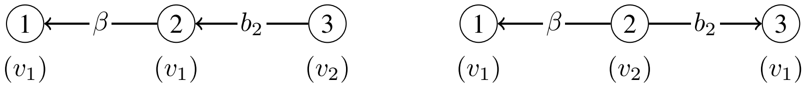
Proof of Proposition 1. When $\sigma_i^2 = \sigma^2$ for all $i \in [p]$, then (4) reduces to:

$$\sum_{l \in \phi_{\mathsf{G}[m,\tau]}(j)} \frac{B_{l,j}^2}{\sigma_l^2} > 0,$$

which holds trivially by causal minimality since $B_{l,j}^2 > 0$ for $(l, j) \in \mathsf{E}$. This proves part (i).

Now under (ii), ${}^{1}/{\sigma_i^2} - {}^{1}/{\sigma_j^2} < 1\,, \forall i, j \in [p]$. Also, ${B_{l,j}^2}/{\sigma_l^2} \geq 1$ for all $(l, j) \in \mathsf{E}$. Thus (4) is satisfied. □

Proof of Lemma 1. Consider the following two SEMs over three nodes, where the noise variances are shown within braces below each node, and the edge weights are shown on the edges.



Both the SEMs make the following conditional independence assertion: $X_1 \perp\!\!\!\perp X_3 \mid X_2$, and are therefore Markov and causal minimal to $\mathcal{P}(X)$. Set $b_2 = \sqrt{1 - \frac{v_1}{v_2}}$. Then using the formulas derived in Proposition 2 it can be verified that the full precision matrix and the precision matrix obtained after removing vertex 1 ($\mathbf{\Omega}_{(-1)}$), for both the SEMs is:

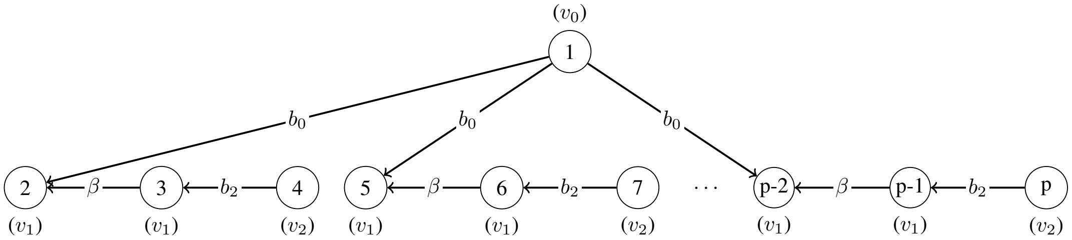
$$\mathbf{\Omega} = \frac{1}{v_1} \times \begin{bmatrix} 1 & -\beta & 0 \\ -\beta & 1 + \beta^2 & -b_2 \\ 0 & -b_2 & 1 \end{bmatrix} \qquad \mathbf{\Omega}_{(-1)} = \frac{1}{v_1} \times \begin{bmatrix} 1 & -b_2 \\ -b_2 & 1 \end{bmatrix} \tag{9}$$

The SEM on the left does not satisfy Assumption 1 because vertex 3 is a non-terminal vertex but $3 \in \operatorname{argmin}(\mathbf{diag}(\mathbf{\Omega}))$. The SEM on the right does not satisfy Assumption 1 because after the vertex 1 is removed we have that vertex 2 is a non-terminal vertex but satisfies $2 \in \operatorname{argmin}(\mathbf{diag}(\mathbf{\Omega}_{(-1)}))$.

Now we construct the subset $\widetilde{\mathcal{G}}_{p,d}$ with $p = 3k$ for $k = 1, 2, \ldots$, as follows. We randomly set the DAG structure over nodes $(3i-1)$, $(3i)$ and $(3i+1)$ to one of the two configurations shown in the above figure. Therefore we have, $|\widetilde{\mathcal{G}}_{p,d}| = 2^{(p-1)/3}$. We generate matrices $\mathbf{B}(\beta)$ and $\mathbf{D}(v_1, v_2)$ as prescribed. The precision matrix block over the nodes $(3i-1)$, $(3i)$, and $(3i+1)$, for $i \in [{(p-1)}/{3}]$, is given by (9), and all the other entries of the precision matrix are zeros. This proves our claim.

While the above constructions constructs a family of disconnected DAGs, with $d = 1$, it is easy to come up with subsets of DAGs that are connected and still satisfy the statement of the lemma. One such construction is shown below where $d = {(p-1)}/{3}$. The entries of the first row (and also the first column) of the precision matrix, for $i \in [{(p-1)}/{3}]$, are as follows:

$$\Omega_{1,1} = \frac{1}{v_0} + \frac{(p-1)b_0^2}{3v_1},\ \Omega_{1,3i-1} = -\frac{b_0}{v_1},\ \Omega_{1,3i} = \frac{b_0 \beta}{v_1}.$$



As shown before, each triplet of nodes $(3i-1) \leftarrow (3i) \leftarrow (3i+1)$, for $i \in [{}^{(p-1)}\!/_3]$, can be oriented as $(3i-1) \leftarrow (3i) \rightarrow (3i+1)$ without changing the block of the precision matrix over the nodes $(3i-1)$, $(3i)$ and $(3i+1)$, and the entries $\mathbf{\Omega}_{1,*}$ or $\mathbf{\Omega}_{*,1}$. □

Proof of Proposition 2. From (2) we have that $(\mathbf{I} - \mathbf{B})X = N$, and since $(\mathbf{I} - \mathbf{B})$ is invertible, $X = (\mathbf{I} - \mathbf{B})^{-1}N$. Therefore:

$$\mathbf{\Sigma} = \mathbb{E}\left[XX^T\right] = \mathbb{E}\left[(\mathbf{I} - \mathbf{B})^{-1}NN^T(\mathbf{I} - \mathbf{B})^{-T}\right] = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{D}(\mathbf{I} - \mathbf{B})^{-T}.$$

From which it follows that $\mathbf{\Omega} = (\mathbf{I} - \mathbf{B})^T\mathbf{D}^{-1}(\mathbf{I} - \mathbf{B})$, where $\mathbf{D}^{-1} = \mathbf{Diag}({}^1\!/_{\sigma_1^2}, \ldots, {}^1\!/_{\sigma_p^2})$. From this the result for the entries of the precision matrix follows by sparsity pattern of $\mathbf{B}$. □

Proof of Proposition 3. From (5) we have that for a terminal vertex i, $\Omega_{i,i} = {}^1\!/_{\sigma_i^2}$, while for a non-terminal vertex j, $\Omega_{j,j} = {}^1\!/_{\sigma_j^2} + \sum_{l \in \phi(j)} {}^{B_{l,j}^2}\!/_{\sigma_l^2}$. Therefore, by Assumption 1 we have that for all non-terminal vertices j and terminal vertices i, $\Omega_{j,j} > \Omega_{i,i}$.

Now since every DAG has at least one terminal vertex, if $i \in \operatorname{argmin}(\mathbf{diag}(\mathbf{\Omega}))$, then once again by Assumption 1, we have that i must be a terminal vertex. □

Proof of Lemma 2. First note that since i is a terminal vertex, the autoregression matrix over $X_{-\mathsf{i}}$ is simply $\mathbf{B}_{-\mathsf{i},-\mathsf{i}}$. Therefore, denoting $\mathbf{D}' \overset{\text{def}}{=} \mathbf{Diag}(\sigma_1^2, \ldots, \sigma_{i-1}^2, \sigma_{i+1}^2, \sigma_p^2)$ and by Proposition 2 we have:

$$\begin{aligned}\mathbf{\Omega}_{(-\mathsf{i})} &= (\mathbf{I} - \mathbf{B}_{-\mathsf{i},-\mathsf{i}})^T(\mathbf{D}')^{-1}(\mathbf{I} - \mathbf{B}_{-\mathsf{i},-\mathsf{i}}) = \sum_{j \in -\mathsf{i}} \frac{1}{\sigma_j^2}((\mathbf{e}_j)_{-\mathsf{i}} - \mathbf{B}_{j,-\mathsf{i}}^T)((\mathbf{e}_j)_{-\mathsf{i}}^T - \mathbf{B}_{-\mathsf{i},j}) \\ &= \sum_{j \in [p]} \frac{1}{\sigma_j^2}\left((\mathbf{e}_j - \mathbf{B}_{j,*}^T)(\mathbf{e}_j^T, -\mathbf{B}_{j,*})\right)_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\sigma_i^2}\left((\mathbf{e}_i - \mathbf{B}_{i,*}^T)(\mathbf{e}_i^T - \mathbf{B}_{i,*})\right)_{-\mathsf{i},-\mathsf{i}} \\ &= \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\sigma_i^2}\left(\mathbf{B}_{i,-\mathsf{i}}^T\mathbf{B}_{i,-\mathsf{i}}\right) = \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \Omega_{i,i}\frac{\mathbf{\Omega}_{i,-\mathsf{i}}^T}{\Omega_{i,i}}\frac{\mathbf{\Omega}_{i,-\mathsf{i}}}{\Omega_{i,i}} = \mathbf{\Omega}_{-\mathsf{i},-\mathsf{i}} - \frac{1}{\Omega_{i,i}}\mathbf{\Omega}_{-\mathsf{i},i}\mathbf{\Omega}_{i,-\mathsf{i}},\end{aligned}$$

where in the last line we used the fact that for a terminal vertex $\Omega_{i,i} = {}^1\!/_{\sigma_i^2}$ (Proposition 3), and $\mathbf{B}_{i,-\mathsf{i}} = -{}^{\mathbf{\Omega}_{i,-\mathsf{i}}}\!/_{\Omega_{i,i}}$ (Proposition 4). □

Proof of Lemma 3. First consider the case when $j \notin \pi_{\mathsf{G}}(i)$. Then, for any $k \in [p] \setminus \{i, j\}$, $i \notin (\phi_{\mathsf{G}}(j) \cap \phi_{\mathsf{G}}(k))$. Therefore, by Proposition 2, $(\Omega_{(-\mathsf{i})})_{j,k} = \Omega_{j,k}$, and by symmetry of the precision matrix $(\Omega_{(-\mathsf{i})})_{k,j} = \Omega_{k,j}$. Thus, we have that for any (j, k) if at least one of $\{j, k\}$ is not in $\pi_{\mathsf{G}}(i)$, then $(\Omega_{(-\mathsf{i})})_{j,k} = \Omega_{j,k}$, which proves our first claim. Thus, the only remaining case to consider is when both $j, k \in \pi_{\mathsf{G}}(i)$. The are two ways is which the set $\mathcal{S}((\mathbf{\Omega}_{(-\mathsf{i})})_{j,*})$ can be larger than the set $\mathcal{S}(\mathbf{\Omega}_{j,*})$, i.e., the support set of the j-th node can increase after deleting the terminal node i. The first being when $j, k \in \pi_{\mathsf{G}}(i)$ and either $(j, k) \in \mathsf{E}$ or $(k, j) \in \mathsf{E}$ but $\Omega_{j,k} = 0$, in which case we have:

$$\sum_{l \in \phi(j) \cap \phi(k)} {}^{(B_{l,j}B_{l,k})}\!/_{\sigma_l^2} = {}^{B_{j,k}}\!/_{\sigma_j^2} + {}^{B_{k,j}}\!/_{\sigma_k^2}.$$

Then, after removing the terminal node i, we have

$$(\Omega_{(-\mathsf{i})})_{j,k} = -{}^{B_{j,k}}\!/_{\sigma_j^2} - {}^{B_{k,j}}\!/_{\sigma_k^2} + \sum_{l \in (\phi(j) \cap \phi(k) \setminus \{i\})} {}^{(B_{l,j}B_{l,k})}\!/_{\sigma_l^2} \neq 0.$$

The other case is when $j, k \in \pi_{\mathsf{G}}(i)$, $(j, k) \notin \mathsf{E}$, $(k, j) \notin \mathsf{E}$ but $\Omega_{j,k} = 0$, in which case we have:

$$\sum_{l \in \phi(j) \cap \phi(k)} {}^{(B_{l,j}B_{l,k})}\!/_{\sigma_l^2} = 0.$$

Therefore, after removing the terminal node we have:

$$(\Omega_{(-\mathsf{i})})_{j,k} = \sum_{l \in (\phi(j) \cap \phi(k) \setminus \{i\})} {}^{(B_{l,j}B_{l,k})}\!/_{\sigma_l^2} \neq 0.$$

Thus, $\mathcal{S}((\mathbf{\Omega}_{(-\mathsf{i})})_{j,*}) \subseteq (\mathcal{S}(\mathbf{\Omega}_{j,*}) \setminus \{i\}) \cup \pi_{\mathsf{G}}(i)$. □

Proof of Theorem 1. Let i_t be the terminal vertex identified in iteration t, $\mathcal{I}_t \overset{\text{def}}{=} \{i_1, \ldots, i_t\}$ and $\mathcal{R}_t \overset{\text{def}}{=} [p] \setminus \mathcal{I}_t$. Let $\mathbf{\Omega}_{(i)}$ be the precision matrix after iteration i. The correctness of the algorithm follows from the following loop invariants:

(i) By Lemma 2 we have that, $(\mathbf{\Omega}_{(t)})_{\mathcal{R}_t,\mathcal{R}_t}$ is the correct precision matrix over $X_{\mathcal{R}_t}$.

(ii) The algorithm identifies a correct terminal vertex in iteration t, since $(\mathbf{\Omega}_{(t-1)})_{\mathcal{R}_{t-1},\mathcal{R}_{t-1}}$ is the correct precision matrix over $X_{\mathcal{R}_{t-1}}$, the SEM over $X_{\mathcal{R}_{t-1}}$ satisfies Assumption 1 by definition, and $\forall i \in \mathcal{I}_{t-1}$, $\Omega_{i,i} = \infty$.

(iii) By proposition 3 we have that at the end of round t, the sub-matrix $\mathbf{B}_{\mathcal{I}_t,*}$ has been correctly set and that $\forall i \in \mathcal{I}_t$, $\pi_{\mathsf{G}}(i) = \mathcal{S}(\mathbf{B}_{i,*})$.

To see that the algorithm returns a unique autoregression matrix $\widehat{\mathbf{B}}$, consider the following. If at iteration t there is a unique minimizer of $\mathbf{diag}(\mathbf{\Omega}_{(t-1)})$, which implies a single terminal vertex, then the algorithm selects it and the incoming edge weights of the node is uniquely determined. While, in iteration t if there are multiple terminal vertices, leading to multiple minimizers of $\mathbf{diag}(\mathbf{\Omega}_{(t-1)})$, then the order in which they are eliminated does not matter. Or in other words, once a vertex becomes a terminal vertex, for instance after deletion of its children, its edge weights do not change. To see this, assume that there are two terminal vertices, i and j after iteration $t-1$. Then i and j are not in each other's parent sets. Therefore, if node i is eliminated in iteration t, then by Lemma 3 we have that $(\Omega_{(t)})_{j,k} = (\Omega_{(t-1)})_{j,k}$, $\forall k \in \pi_{\mathsf{G}}(j)$. Hence, we have that $\mathbf{B}$ is the unique autoregression matrix returned by the algorithm. □

Proof of Lemma 5. Let $\mathbf{\Omega}_{(-\mathsf{i})} = (\boldsymbol{\omega}_j)_{j \in -\mathsf{i}}$ be the true precision matrix over $X_{-\mathsf{i}}$ and let $\widehat{\mathbf{\Omega}}' = (\boldsymbol{\omega}'_j)_{j \in [p]}$ be the matrix returned by the function UPDATE. The estimator $\widehat{\mathbf{\Omega}}_{(-\mathsf{i})} = (\widehat{\boldsymbol{\omega}}_j)_{j \in -\mathsf{i}}$ of $\mathbf{\Omega}_{(-\mathsf{i})}$ can be obtained by solving (7) using $\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}}$. By Lemma 4, and the facts that $|\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}} - \mathbf{\Sigma}_{-\mathsf{i},-\mathsf{i}}|_\infty \leq |\mathbf{\Sigma}^n - \mathbf{\Sigma}|_\infty$ and $\|\mathbf{\Omega}_{(i)}\|_1 \leq M$, we have that $|\mathbf{\Omega}_{(-\mathsf{i})} - \widehat{\mathbf{\Omega}}_{(-\mathsf{i})}| \leq 4M\lambda_n$. Since i is a terminal vertex, by Proposition 4 we have $\pi_{\mathsf{G}}(i) = \mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}$. Further, since $\mathcal{S}(\mathbf{\Omega}_{j,*}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_{j,*})$, $\forall j \in [p]$, we have by Assumption 2 (ii) that, $\pi_{\mathsf{G}}(i) \subseteq \widehat{\pi}(i) = \mathcal{S}(\widehat{\mathbf{\Omega}}_{i,*}) \setminus \{i\} \subseteq \widehat{\mathsf{S}}$. By Lemma 3 and Assumption 2 (ii) we have that $\forall j \in \widehat{\mathsf{S}}_j$, $\mathcal{S}(\boldsymbol{\omega}_j) \subseteq \mathcal{S}\left(\mathbf{\Omega}_{j,*} \setminus \{i\}\right) \cup \pi_{\mathsf{G}}(i) \subseteq \mathcal{S}\left(\widehat{\mathbf{\Omega}}_{j,*} \setminus \{i\}\right) \cup \widehat{\pi}(i) \overset{\text{def}}{=} \widehat{\mathsf{S}}_j$. Or in other words we have $\left(\mathbf{\Omega}_{(i)}\right)_{j,\widehat{\mathsf{S}}^c_j} = \left(\mathbf{\Omega}_{(i)}\right)_{\widehat{\mathsf{S}}^c_j,j} = \mathbf{0}$. Now for $j \in -\mathsf{i}$ we set $(\boldsymbol{\omega}'_j)_{\widehat{\mathsf{S}}_j} = \bar{\boldsymbol{\omega}}_j$ and $(\boldsymbol{\omega}'_j)_{\widehat{\mathsf{S}}^c_j} = \mathbf{0}$, where $\bar{\boldsymbol{\omega}}_j$ is obtained by solving:

$$
\begin{aligned}
\underset{\boldsymbol{\omega} \in \mathbb{R}^{|\widehat{\mathsf{S}}_j|}}{\text{argmin}} \quad & \|\boldsymbol{\omega}\|_1, \\
\text{sub. to} \quad & \left|\mathbf{\Sigma}^n_{k,\widehat{\mathsf{S}}_j} \boldsymbol{\omega}\right| \leq \lambda_n, \forall k \notin \{i, j\}, \\
& \left|\mathbf{\Sigma}^n_{j,\widehat{\mathsf{S}}_j} \boldsymbol{\omega} - 1\right| \leq \lambda_n.
\end{aligned}
$$

Since $\bar{\boldsymbol{\omega}}_j$ is a solution to the above linear program, we have that $|\mathbf{\Sigma}^n_{-\mathsf{i},-\mathsf{i}} \boldsymbol{\omega}'_j - \mathbf{e}_j| \leq \lambda_n$ and $\|\boldsymbol{\omega}'_j\|_1 \leq \|\widehat{\boldsymbol{\omega}}_j\|_1$. Therefore, $|\mathbf{\Omega}_{(-\mathsf{i})} - \widehat{\mathbf{\Omega}}'_{-\mathsf{i},-\mathsf{i}}| \leq 4M\lambda_n$. Moreover, by Assumption 2 (ii), and the fact that $\widehat{\mathbf{\Omega}}'_{i,*} = \widehat{\mathbf{\Omega}}'_{*,i} = \mathbf{0}$, we get: $\mathcal{S}(\mathbf{\Omega}_{(-\mathsf{i})}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}')$. □

Proof of Theorem 2. Let i_t denote the terminal vertex identified in iteration t and let $\mathcal{I}_t \overset{\text{def}}{=} \{i_1, \ldots, i_t\}$. Let $\mathcal{R}_t \overset{\text{def}}{=} [p] \setminus \mathcal{I}_t$ denote the vertices remaining after iteration t. Let $\widehat{\mathbf{\Omega}}_{(t)}$ denote the precision matrix at the end of iteration t, $\widehat{\mathbf{\Omega}}_{(\mathcal{R}_t)} \overset{\text{def}}{=} (\widehat{\mathbf{\Omega}}_{(t)})_{\mathcal{R}_t,\mathcal{R}_t}$, and $\mathbf{\Omega}^*_{(\mathcal{R}_t)}$ be the true precision matrix over $X_{\mathcal{R}_t}$. Since $\|\mathbf{\Omega}^*\|_1 \leq M$, where M is defined in (8), we have that $\lambda_n \geq M|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty \geq \|\mathbf{\Omega}^*\|_1 |\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty$. Therefore, by Lemma 4 and Assumption 2 (ii), we have that $\left|\widehat{\mathbf{\Omega}}_{(\mathcal{R}_0)} - \mathbf{\Omega}^*_{(\mathcal{R}_0)}\right|_\infty = |\widehat{\mathbf{\Omega}} - \mathbf{\Omega}^*|_\infty \leq 4M\lambda_n$, and $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_0)}) \subseteq \mathcal{S}(\widehat{\mathbf{\Omega}}_0)$. Therefore, by Assumption 2 we have that the Algorithm 1 identifies the correct terminal vertex in iteration 1. Therefore, by Lemma 5 we have that $\left|\mathbf{\Omega}^*_{(\mathcal{R}_{t_1})} - \widehat{\mathbf{\Omega}}_{(\mathcal{R}_{t_1})}\right| \leq 4M\lambda_n$ and $\mathcal{S}(\mathbf{\Omega}^*_{(\mathcal{R}_{t_1})}) \subseteq \widehat{\mathbf{\Omega}}_{(t_1)}$.

Let $\mathbf{E} = (\varepsilon_{i,j})$, where $\varepsilon_{i,j} = \Omega^*_{i,j} - \widehat{\Omega}_{i,j}$. To simplify notation in this paragraph, we will denote the i_1 vertex by simply i.

Then, for any $j \neq i$, we have that

$$\begin{aligned}
|\widehat{B}_{i,j} - B^*_{i,j}| &= \left| \frac{\widehat{\Omega}_{i,j}}{\widehat{\Omega}_{i,i}} - \frac{\Omega^*_{i,j}}{\Omega^*_{i,i}} \right| = \left| \frac{\Omega^*_{ii}(\Omega^*_{i,j} - \varepsilon_{i,j}) - (\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,j}}{(\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,i}} \right| \\
&= \left| \frac{\Omega^*_{i,i}\varepsilon_{i,j} - \Omega^*_{i,j}\varepsilon_{i,i}}{(\Omega^*_{i,i} - \varepsilon_{i,i})\Omega^*_{i,i}} \right| = \left| \frac{\varepsilon_{i,i} - \sigma_i^2 \Omega^*_{i,j}\varepsilon_{i,i}}{{}^1/\sigma_i^2 - \varepsilon_{i,i}} \right| \\
&= \left| \frac{\varepsilon_{i,i} - B^*_{i,j}\varepsilon_{i,i}}{{}^1/\sigma_i^2 - \varepsilon_{i,i}} \right| \\
&\leq \frac{4M\lambda_n(1 + |B^*_{i,j}|)}{|{}^1/\sigma_i^2 - \varepsilon_{i,i}|} \leq 4cM(1 + |B^*_{i,j}|)\sigma_i^2 \lambda_n,
\end{aligned}$$

where the second and third lines follow from the fact that i is a terminal vertex and therefore, $\Omega^*_{i,i} = {}^1/\sigma_i^2$ and $\Omega_{i,j} = {}^{-B_{i,j}}/\sigma_i^2$. Therefore, we have that $|\mathbf{B}^*_{i_1,*} - \widehat{\mathbf{B}}_{i_1,*}|_\infty = 4cM(1 + B_{\max})\sigma^2_{\max}\lambda_n$.

Next, assume that the algorithm correctly identifies terminal vertices upto round t. Then $|\widehat{\boldsymbol{\Omega}}_{(\mathcal{R}_t)} - \boldsymbol{\Omega}^*_{(\mathcal{R}_t)}|_\infty \leq 4M\lambda_n$, $\mathcal{S}(\boldsymbol{\Omega}^*_{(\mathcal{R}_t)}) \subseteq \mathcal{S}(\widehat{\boldsymbol{\Omega}}_{(t)})$, and $|\mathbf{B}^*_{\mathcal{I}_t,\mathcal{I}_t} - \widehat{\mathbf{B}}_{\mathcal{I}_t,\mathcal{I}_t}| \leq 4cM(1 + B_{\max})\sigma^2_{\max}\lambda_n$. Therefore, once again by Assumption 2, it follows that the algorithm identifies the correct terminal vertex in round $t+1$, $|\widehat{\boldsymbol{\Omega}}_{(\mathcal{R}_{t+1})} - \boldsymbol{\Omega}^*_{(\mathcal{R}_{t+1})}|_\infty \leq 4M\lambda_n$, $\mathcal{S}(\boldsymbol{\Omega}^*_{(\mathcal{R}_{t+1})}) \subseteq \mathcal{S}(\widehat{\boldsymbol{\Omega}}_{(t+1)})$, and $|\mathbf{B}^*_{\mathcal{I}_{t+1},\mathcal{I}_{t+1}} - \widehat{\mathbf{B}}_{\mathcal{I}_{t+1},\mathcal{I}_{t+1}}| \leq 4cM(1 + B_{\max})\sigma^2_{\max}\lambda_n$. Hence, the final claim follows by induction. The claim that $\mathcal{S}(\mathbf{B}^*) \subseteq \mathcal{S}(\widehat{\mathbf{B}})$ follows from the fact that $\mathcal{S}(\boldsymbol{\Omega}^*) \subseteq \mathcal{S}(\widehat{\boldsymbol{\Omega}})$. Finally, since $\mathcal{S}(\mathbf{B}^*) \subseteq \mathcal{S}(\widehat{\mathbf{B}})$ implies that $\mathcal{T}_{\widehat{\mathsf{G}}} \subseteq \mathcal{T}_{\mathsf{G}^*}$. □

Proof of Theorem 3. Given that the data was generated by the SEM $(\mathsf{G}^*, \mathbf{B}^*, \{\sigma_i^2\})$, each X_i can be written as follows:

$$X_i = \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j} N_j,$$

for some $w_{i,j} \neq 0$.

Sub-Gaussian case. N_i is sub-Gaussian with parameter $\sigma_i \nu$, X_i is sub-Gaussian with parameter $\nu\sqrt{\sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2}$ and $\Sigma^*_{i,i} = \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2$. Therefore, it follows that ${}^{X_i}/\sqrt{\Sigma^*_{i,i}}$ is sub-Gaussian with parameter ν. From Lemma 1 of [RWRY11] and Theorem 2 we have that the regularization parameter λ_n need to satisfy the following bound in order to guarantee that $|\widehat{\mathbf{B}} - \mathbf{B}^*|_\infty \leq \varepsilon$:

$$MC_1 \sqrt{\frac{2}{n} \log\left(\frac{2p}{\sqrt{\delta}}\right)} \leq \lambda_n \leq \frac{\varepsilon}{c4M(1 + B_{\max})\sigma^2_{\max}}. \tag{10}$$

The above holds in the regime where the number of samples scales as given in the statement of the Theorem.

Bounded moment case. In this case we have:

$$\left(\sqrt{\boldsymbol{\Sigma}^*_{i,i}}\right)^{4m} = \left(\sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^2 \sigma_i^2 \right)^{2m} \geq \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} (w_{i,j}\sigma_i)^{4m} \tag{11}$$

Now, by Rosenthal's inequality we have:

$$\begin{aligned}
\mathbb{E}\left[(X_i)^{4m}\right] &\leq C_m \left\{ \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \mathbb{E}\left[N_j^{4m}\right] + \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \mathrm{Var}\left[N_i\right]^{2m} \right\} \\
&\leq C_m \left\{ \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \sigma_i^{4m} K_m + \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} w_{i,j}^{4m} \sigma_i^{4m} \right\} \\
&= C_m(K_m + 1) \sum_{j \in \mathsf{A}_{\mathsf{G}^*}(i)} (w_{i,j}\sigma_i)^{4m} \qquad (12)
\end{aligned}$$

Combining (11) and (12) we have

$$\mathbb{E}\left[\left(\frac{X_i}{\sqrt{\Sigma^*_{i,i}}}\right)^{4m}\right] \leq C_m(K_m + 1). \tag{13}$$

From the above and invoking Lemma 2 of [RWRY11] we get:

$$|\mathbf{\Sigma}^n - \mathbf{\Sigma}^*|_\infty < C_2 \left(\frac{p^2}{n^m \delta}\right)^{1/2m}, \tag{14}$$

with probability at least $1 - \delta$. From Theorem 2 and (14) we have that the regularization parameter λ should satisfy the following for $|\widehat{\mathbf{B}} - \mathbf{B}^*|_\infty \leq \varepsilon$ to hold:

$$MC_2 \left(\frac{p^2}{n^m \delta}\right)^{1/2m} \leq \lambda_n \leq \frac{\varepsilon}{c4M(1 + B_{\max})\sigma^2_{\max}}. \tag{15}$$

The above holds in the regime where the number of samples scales as given in the statement of the Theorem. □

Proposition 6. *Let $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ be an SEM over X with $\mathsf{G} \in \mathcal{G}_{p,d}$ and precision matrix $\mathbf{\Omega}$. Let ρ be the maximum degree of a node in G. Then, $|\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}| \leq \rho^2 \leq d$, $\forall i \in [p]$.*

Proof of Proposition 6. For any node i, we will define the following set: $\mathsf{S}_\mathsf{G}(i) = \{j \in -\mathsf{i} \mid (i,j) \notin \mathsf{E} \wedge (j,i) \notin \mathsf{E} \wedge |\Omega_{i,j}| \neq 0\}$. Then, from Proposition 2, we have: if $j \in \mathsf{S}_\mathsf{G}(i)$ then $\Omega_{i,j} = \sum_{l \in \phi(i) \cap \phi(j)} {}^{(B_{l,i}B_{l,j})}/_{\sigma_l^2} \neq 0$. In other words, if $j \in \mathsf{S}_\mathsf{G}(i)$ then i and j have at least one common child, i.e., $\phi_\mathsf{G}(i) \cap \phi_\mathsf{G}(j) \neq \varnothing$. Node i can have at most ρ children, and each child $k \in \phi_\mathsf{G}(i)$ can have at most $\rho - 1$ parents other than i making them all members of $\mathsf{S}(i)$. Thus, $\mathsf{S}(i) \leq \rho(\rho - 1)$. Therefore, we have that $\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\} \subseteq \mathsf{N}_\mathsf{G}(i) \cup \mathsf{S}_\mathsf{G}(i)$. Then, using the inclusion-exclusion principle we have that:

$$|\mathcal{S}(\mathbf{\Omega}_{i,*}) \setminus \{i\}| \leq |\mathsf{N}_\mathsf{G}(i)| + |\mathsf{S}_\mathsf{G}(i)| - |\mathsf{N}_\mathsf{G}(i) \cap \mathsf{S}_\mathsf{G}(i)| = |\mathsf{N}_\mathsf{G}(i)| + |\mathsf{S}_\mathsf{G}(i)| \leq \rho + \rho(\rho - 1) = \rho^2.$$

The SEM which achieves the above upper bound is precisely the one constructed in the proof, i.e., there exists a node i with exactly ρ children, each child in turn has $\rho - 1$ "other parents" which are all members of $\mathsf{S}_\mathsf{G}(i)$. □

Proposition 7. *Given an SEM $(\mathsf{G}, \mathbf{B}, \{\sigma_i^2\})$ with precision matrix $\mathbf{\Omega}$, if $\sigma_i^2 = \mathcal{O}(1)$ for all $i \in [p]$, and $B_{i,j} = \mathcal{O}(1)$ for all $(i,j) \in \mathsf{E}$, then the quantity M as defined in* (8) *is $\mathcal{O}(d)$.*

Proof of Proposition 7. Let $\sigma^2_{\min} = \min\{\sigma_i^2\}$. Let $\phi_{ij} \overset{\text{def}}{=} \phi(i) \cap \phi(j)$ and let $C_i \overset{\text{def}}{=} \{j \neq i \mid \phi_{ij} \neq \varnothing\}$. Define:

$$f_i(\mathbf{B}) = \frac{1}{\sigma^2_{\min}} \sum_{j \in \mathsf{N}(i)} |B_{i,j} + B_{j,i}| + \frac{1}{\sigma^2_{\min}} \sum_{j \in C_i} \Big| \sum_{l \in \phi_{ij}} B_{l,i} B_{l,j} \Big| + \frac{1}{\sigma^2_{\min}} \sum_{l \in \phi(i)} B^2_{l,i} + \frac{1}{\sigma^2_{\min}} \tag{16}$$

Then by (5) and by definition of M in (8), $M \leq \max_{i=1}^p f_i(\mathbf{B})$. Now, $f_i(\mathbf{B})$ is maximized when $\mathsf{MB}(i) = d$. There are two cases to consider: Case (i), $\phi(i) = \sqrt{d}$, $\pi(i) = \varnothing$, $C_i = d - \sqrt{d}$ and $|\phi_{ij}| = 1$ for all $j \in C_i$. In this case, the first and third term of (16) are $\mathcal{O}\left(\sqrt{d}\right)$ while the second term is $\mathcal{O}\left(d - \sqrt{d}\right)$, and therefore $M = \mathcal{O}(d)$. Case (ii), $\pi(i) = d$ or $\phi(i) = d$. In this case $C_i = \varnothing$ and therefore, the first and third term in (16) dominate and $M = \mathcal{O}(d)$. Therefore, in the worst case $M = \mathcal{O}(d)$. □

B Comparison with state-of-the-art methods

B.1 Synthetic experiments

We compared the performance of our method against three other state-of-the-art methods for learning SEMs, viz. MMHC (max-min hill climbing) [TBA06], GES (greedy equivalence search) [Chi03], and the PC algorithm [SGS00] on Erdős-Rényi random SEMs with sub-Gaussian noise. First, we generated random graphs by sampling Erdős-Rényi undirected graphs with edge probability q, and then generating a random causal ordering of the vertices $[p]$, and finally orienting the undirected edges according to the causal order. The edge weights were generated from the uniform distribution over $[-1, -0.5] \cup [0.5, 1]$. To generate sub-Gaussian noise, the i-th noise variable was set as $N_i = \sigma_i R$, where R is a Rademacher random variable. We generated 30 random DAGs for each value of $p \in \{50, 100, 150, 200\}$ (and corresponding $q \in \{0.001, 0.005, 0.0033, 0.0025\}$) and computed the average accuracy, recall and execution time for each method across the 30 graphs. The number of samples n was set to $100 \lfloor k^2 \log n \rfloor$, where k was the size of the maximum Markov blanket of a sampled DAG, and the regularization parameter for our method was set to $0.5\sqrt{\log p/n}$. The PC and MMHC algorithm take a parameter α, the target nominal type-I error rate of the conditional independence tests, and was set to 0.05. We performed two sets of experiments: in the first set of experiments we enforced the identifiability condition (Assumption 1) on the sampled SEMs, and in the second set of experiments we did not enforce the identifiability condition.

Method	Accuracy	Recall	Seconds	Edges	Accuracy	Recall	Seconds	Edges
		p = 50				p = 100		
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.12 ± 0.01**	12.07	**1.00 ± 0.00**	**1.00 ± 0.00**	0.92 ± 0.01	25.30
MMHC	0.53 ± 0.03	0.55 ± 0.03	0.25 ± 0.01	12.50	0.53 ± 0.02	0.57 ± 0.02	0.95 ± 0.02	24.67
GES	0.24 ± 0.02	0.32 ± 0.02	0.32 ± 0.01	13.07	0.20 ± 0.01	0.34 ± 0.02	0.53 ± 0.01	22.67
PC	0.56 ± 0.01	1.00 ± 0.00	0.18 ± 0.00	12.53	0.52 ± 0.01	0.99 ± 0.01	**0.41 ± 0.01**	22.97
		p = 100				p = 200		
Ours	**1.00 ± 0.00**	**1.00 ± 0.00**	3.16 ± 0.02	37.10	**1.00 ± 0.00**	**1.00 ± 0.00**	9.22 ± 0.03	47.77
MMHC	0.46 ± 0.02	0.53 ± 0.02	2.17 ± 0.03	37.33	0.49 ± 0.01	0.59 ± 0.01	3.83 ± 0.04	50.30
GES	0.18 ± 0.01	0.35 ± 0.02	**0.75 ± 0.01**	37.30	0.16 ± 0.01	0.34 ± 0.01	**1.07 ± 0.02**	50.53
PC	0.51 ± 0.01	0.98 ± 0.00	0.88 ± 0.02	37.23	0.49 ± 0.01	0.98 ± 0.00	1.36 ± 0.01	49.33

Table 1: Performance of our method vis-à-vis other state-of-the-art methods on Erdős-Rényi random DAGs that *satisfy the identifiability condition* given in Assumption 1.

Identifiable case. To enforce identifiability of generated SEMs we simply set all the noise variances to $\sigma_i^2 = 0.8$. Note that this is a sufficient condition for indentifiability (Proposition 1). Table 1 shows the mean accuracy, recall, execution time in seconds, and the average number of edges for each of the method. Our algorithm recovers the structure perfectly as is evident from the accuracy and recall scores while being comparable in speed to the other methods. Among the other methods, the PC algorithm has a recall score that is close to one indicating that it recovers the skeleton correctly most of the time. However, its poor accuracy, hovering at a mere 50%, indicates that it fails to orient most of the edges even though the true SEM is identifiable. Note the number of edges recovered by all the methods are very close to each other indicating that the hyper-parameters of the methods were set correctly. MMHC and GES, which are heuristic algorithms, perform very poorly.

Method	Accuracy	Recall	Seconds	Edges	Accuracy	Recall	Seconds	Edges
		p = 50				p = 100		
Ours	**0.97 ± 0.01**	0.97 ± 0.01	**0.12 ± 0.01**	12.30	**0.95 ± 0.01**	0.96 ± 0.01	0.93 ± 0.01	24.97
MMHC	0.53 ± 0.03	0.56 ± 0.03	0.25 ± 0.01	12.37	0.54 ± 0.02	0.59 ± 0.02	0.96 ± 0.02	25.53
GES	0.27 ± 0.02	0.36 ± 0.03	0.31 ± 0.01	12.03	0.20 ± 0.01	0.34 ± 0.02	0.54 ± 0.01	25.37
PC	0.55 ± 0.01	**1.00 ± 0.00**	0.19 ± 0.00	13.60	0.54 ± 0.01	**0.99 ± 0.01**	**0.41 ± 0.01**	24.90
		p = 100				p = 200		
Ours	**0.96 ± 0.01**	0.96 ± 0.01	3.24 ± 0.02	36.40	**0.96 ± 0.01**	0.96 ± 0.00	9.44 ± 0.04	47.80
MMHC	0.49 ± 0.01	0.56 ± 0.02	2.12 ± 0.02	36.23	0.46 ± 0.01	0.56 ± 0.01	3.74 ± 0.03	47.23
GES	0.18 ± 0.01	0.33 ± 0.01	**0.74 ± 0.01**	36.93	0.14 ± 0.01	0.31 ± 0.01	**1.04 ± 0.02**	48.63
PC	0.53 ± 0.01	**0.99 ± 0.00**	0.81 ± 0.01	38.77	0.50 ± 0.01	**0.98 ± 0.00**	1.38 ± 0.01	50.30

Table 2: Performance of our method vis-à-vis other state-of-the-art methods on Erdős-Rényi random DAGs.

Non-identifiable case. In this case, we sampled the noise variances σ_i^2 from the uniform distribution over $[0.5, 1]$. It is easy to show that in this regime, where both the noise variances and the absolute edge weights are drawn from the uniform distribution over $[0.5, 1]$, the sampled SEMs do not satisfy the identifiability condition (Assumption 1) globally. Table 2 shows the mean accuracy, recall, execution time and average number of edges for the four methods, across 30 randomly sampled SEMs. As expected, our method is no-longer able to recover the graph perfectly. However, our method still achieves *close-to-perfect* structure recovery as is evidenced by its accuracy and recall scores, which are close to one. Also note that, while the PC algorithm has slightly better recall than our method, its accuracy is very poor. Therefore, our method is to be preferred over the PC algorithm Other methods, on the other hand, achieve performance similar to that of the indentifiable case.

B.2 Real-world experiments

Dataset	Disease	# Samples	# Variables	# Sampled Variables
GSE13294	Colon cancer	155	54,675	124.0
GSE1476	Colon cancer	150	59,381	120.0
GSE17951	Prostate cancer	154	54,675	123.0
GSE18105	Colon cancer	111	54,675	88.0
GSE18638	Colon cancer	98	235,826	78.0
GSE1898	Liver cancer	182	21,794	145.0
GSE22219	Breast cancer	216	24,332	172.0

Table 3: Gene expression data sets.

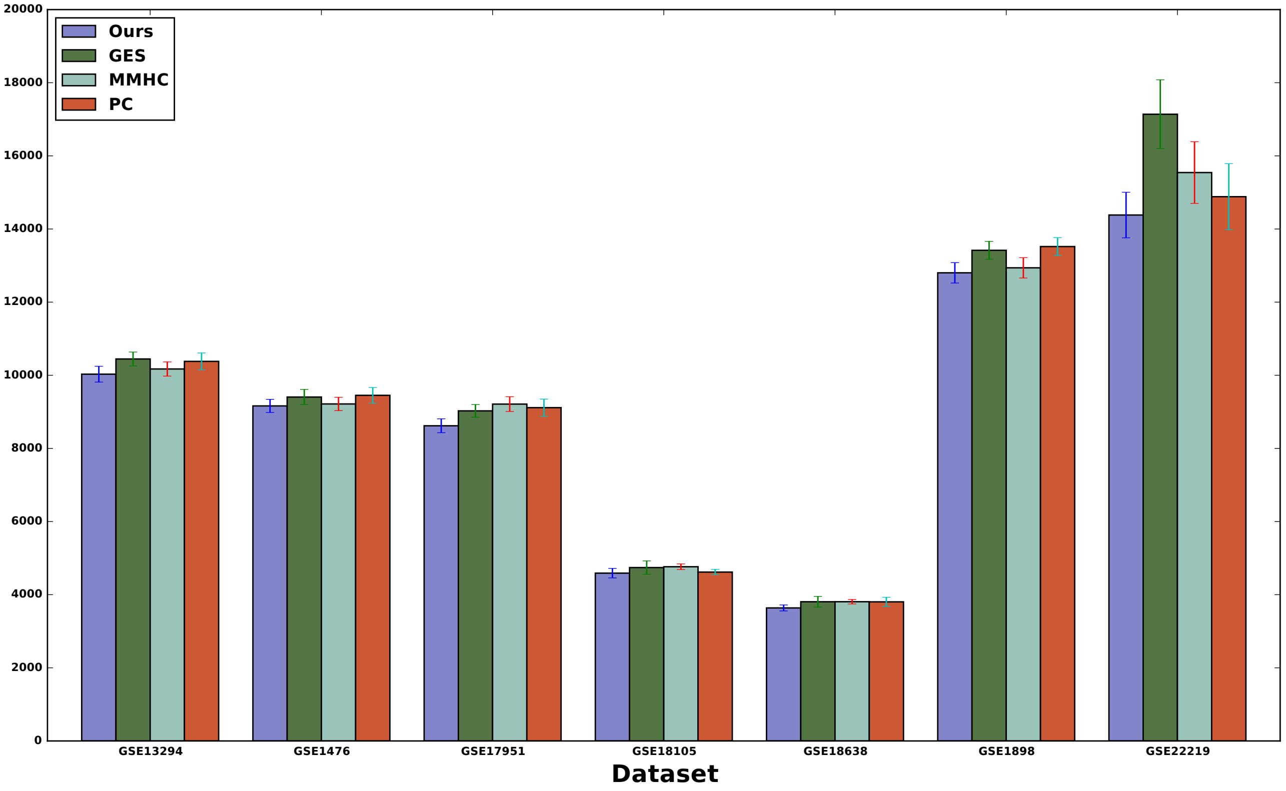


Figure 2: The mean negative log likelihood of each method, on the test set, computed across 10 bootstrap runs.

Finally, we compared the performance of our algorithm with the three state-of-the-art methods on 7 real-world gene expression data sets. The various attributes of the data sets, which are publicly available at the Gene Expression Omnibus (`http://www.ncbi.nlm.nih.gov/geo/`), are shown in Table 3. In order to avoid a high-dimensional regime we selected the $\lfloor 0.8n \rfloor$ highest variance genes for analysis, as was done in [PB14]. We computed the average test negative-log-likelihood of each method on the 7 data sets across 10 bootstrap runs. In each bootstrap run, we created a training set by sampling n samples, with replacement, from the original data set and held out the remaining samples (those that were not picked in the sampling) as the test set. For our method, the regularization parameter was set to $0.01\sqrt{\log p/n}$, while for PC and MMHC the parameter α was set to 0.05. GES takes no parameters. We used the implementation of the MMHC

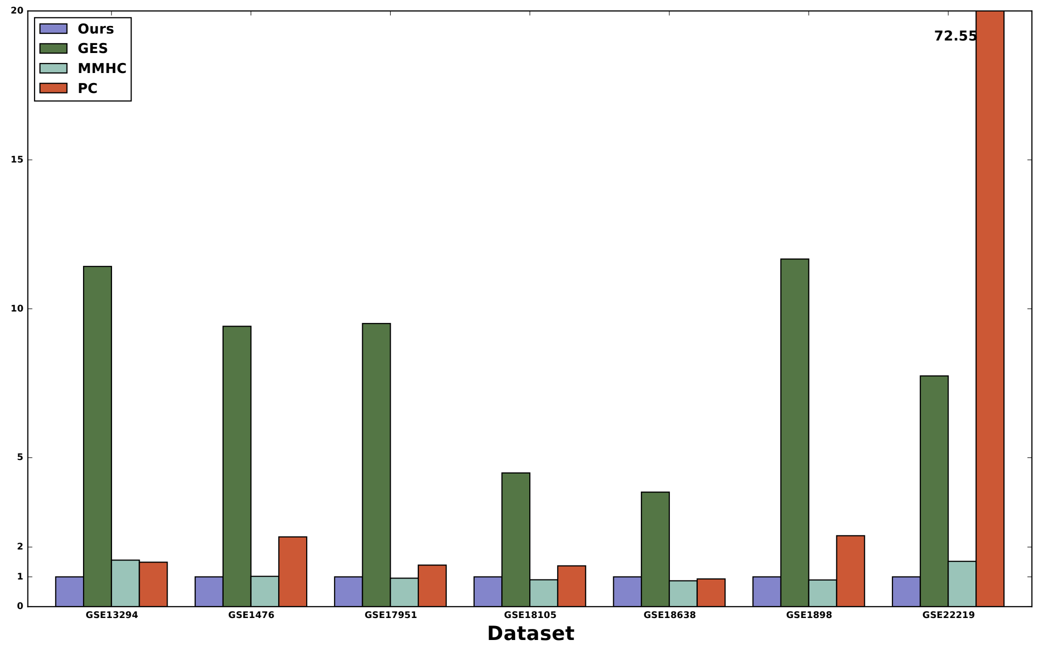


Figure 3: The mean speed-up of our method vs. other state-of-the-art methods.

algorithm provided by the *bnlearn* R package, while the *pcalg* R package provided the implementations of the GES and PC algorithm. We implemented our method, along with the CLIME algorithm for inverse covariance estimation, in Python.

Figure 2 shows the mean test negative-log-likelihood, along with standard errors, of each method on the 7 gene expression data sets. Our method achieves the lowest test negative-log-likelihood on all seven data sets. This is noteworthy since MMHC and GES explicitly try to find the highest scoring structure while our method does not try to maximize any score. Further, unlike PC, MMHC, and GES, which return a PDAG, our method always returns a DAG.

Figure 3 shows the speed-up of our method with respect to the other three methods. On the largest and third largest data set (GSE22219) our method is close to 2 times faster than MMHC, 72 times faster than PC and around 10 times faster than GES. On five out of seven data sets our method achieves speed up of around 10 as compared to GES.

C Computational Complexity

In the population setting, i.e., given the true precision matrix, our algorithm can be implemented by storing the diagonal of the precision matrix separately and sorting it once which takes $\mathcal{O}\left(p \log p\right)$ time. In each iteration, updating the precision matrix in line 5 takes $\mathcal{O}\left(d\right)$ time since $\mathbf{\Omega}_{*,i}$ and $\mathbf{\Omega}_{i,*}$ are d-sparse. Updating the diagonal takes $\mathcal{O}\left(d \log p\right)$ time, while searching for the minimum diagonal element takes $\mathcal{O}\left(\log p\right)$ time. Therefore, Algorithm 1 computes the $\widehat{\mathbf{B}}$ matrix in $\mathcal{O}\left(p(d + d \log p)\right)$ time. In the population setting, the computational complexity of [LB13]'s algorithm is $\mathcal{O}\left(p 2^{2(w+1)(w+d)}\right)$, where w is the tree-width of the DAG structure of the true SEM and $d = \max\{|\mathsf{N}(i)|\}$. Note that the population version of our algorithm can still be used in the finite sample setting if the precision matrix is estimated accurately enough.

In the finite sample setting, the computational complexity of our algorithm is dominated by the steps for estimating and updating the precision matrix — the latter depends on how well the sparsity pattern of the precision matrix is estimated. First, we analyze the computational complexity of our algorithm assuming exact support recovery, then we analyze the worst-case performance of our algorithm without assuming sparsity of the estimated precision matrix. Estimating the precision matrix can be done by solving p linear programs in $2p$-dimension and with $4p$ constraints. The smoothed complexity of this step is $\mathcal{O}\left(p^3 \log(p/\sigma)\right)$ when using interior point LP solvers [DST11], where σ^2 is variance of the Gaussian perturbations [2]. Next observe that $|\mathbf{\Omega}^* - \widehat{\mathbf{\Omega}}|_\infty \leq |\mathbf{B}^* - \widehat{\mathbf{B}}|_\infty \leq \varepsilon$. By thresholding $\widehat{\mathbf{\Omega}}$ at the level ε, each time the precision matrix is updated, we can ensure exact support recovery in each iteration. Thus, in the UPDATE function $\widehat{\pi}(i) = \pi_{\mathsf{G}^*}(i)$

[2] The worst-case complexity of interior point methods for solving LPs is $\mathcal{O}\left(p^3 L\right)$ where L " is a parameter measuring the precision

and $|\widehat{\mathsf{S}}_j| \leq d \leq p$. Therefore, the UPDATE function takes $\mathcal{O}\left(d^4 \log(d/\sigma)\right)$ operations, leading to an overall complexity of $\widetilde{\mathcal{O}}\left(p^3 + pd^4\right)$. In the worst case, i.e., without any thresholding, $\widehat{\mathbf{\Omega}}$ can be dense. Therefore, the UPDATE function might re-estimate the full precision matrix over $p - t$ variables in iteration t, which takes $\mathcal{O}\left((p-t)^4 \log((p-t)/\sigma)\right)$ operations, leading to an overall complexity of $\widetilde{\mathcal{O}}\left(p^5\right)$. Thus, in the finite sample setting the complexity of our algorithm is between $\widetilde{\mathcal{O}}\left(p^3 + pd^4\right)$ and $\widetilde{\mathcal{O}}\left(p^5\right)$. Note that [LB13]'s analysis of the computational complexity of their algorithm assumes perfect support recovery of the precision matrix. In this regime, the computational complexity of their method is $\mathcal{O}\left(p2^{2(w+1)(w+d)} + p^3\right)$, including the step to estimate the precision matrix using graphical Lasso [FHT08], where w is the tree-width of the true DAG. However, without thresholding the output of graphical Lasso can be dense leading to a worst-case computational complexity that is exponential in p.

D Discussion

Our Techniques. Our algorithm for learning linear SEMs differs conceptually from previous test-based, score-based or inverse-covariance-estimation-based methods. We are therefore able to get rid of many of the shortcomings of existing methods like requirement of strict non-Gaussianity of noise [SIS$^+$11], homoscedasticity [PB14], and faithfulness [LB13, KP07]. We do so my obtaining and exploiting various properties of terminal vertices in linear SEMs. We obtain our sample complexity results by using various properties of sub-Gaussian and bounded-moment variables and using concentration results for the empirical covariance matrix under the aforementioned noise conditions. Lastly, we improve the computational complexity of our algorithm by exploiting the sparsity structure of the precision matrix to obtain solutions of "larger" LPs (size $\mathcal{O}\left(p\right)$) by solving much "smaller" LPs (size $\mathcal{O}\left(d^2\right)$).

Inverse covariance estimation. A popular approach for inverse covariance estimation, under high-dimensional settings, is the ℓ_1-penalized Gaussian maximum likelihood estimate (MLE) studied by [YL07], [BGd08], and [FHT08], among others. The ℓ_1-penalized Gaussian MLE estimate of the inverse covariance matrix has attractive theoretical guarantees as shown by [RWRY11]. However, the elementwise ℓ_∞ guarantees for the inverse covariance estimate obtained by [RWRY11] require an edge-based mutual incoherence condition that is quite restrictive. Many algorithms have been developed in the recent past for solving the ℓ_1-penalized Gaussian MLE problem [HSD$^+$13, HBDR12, RRG$^+$12, JJR12]. While, technically, these algorithms can be used in conjunction with our algorithm for learning SEMs, in this paper we use the method called CLIME, developed by [CLL11]. The primary motivation behind using CLIME is that the theoretical guarantees obtained by [CLL11] does not require the edge-based mutual incoherence condition. Further, CLIME is computationally attractive because it computes $\widehat{\mathbf{\Omega}}$ columnwise by solving p independent linear programs. Even though the CLIME estimator $\widehat{\mathbf{\Omega}}$ is not guaranteed to be positive-definite (it is positive-definite with high probability) it is suitable for our purpose.

needed to perform the arithmetic operations exactly" and grows as $\Omega\left(p\right)$ [ST03]. However, interior-point methods work much more efficiently in practice and have an average complexity of $\mathcal{O}\left(p^3 \log p\right)$ (see [ST03] and the references therein).