A. Related Concepts

In this section, we briefly review a few related mathematical concepts in the paper, which can be found in (Bang-Jensen and Gutin, 2008; Jiang et al., 2011). We use the same notation from the main paper.

A.1. Concepts in Linear Algebra

Definition A.1 (Symmetric and Skew-Symmetric)**.** *A real matrix $Y \in \mathbb{R}^{d\times d}$ is called symmetric if and only if $[Y]_{ij} = [Y]_{ji}$ for all $i, j \in \{1, \cdots, n\}$. Similarly, Y is called skew-symmetric if and only if $[Y]_{ij} = -[Y]_{ji}$ for all $i, j \in \{1, \cdots, n\}$.*

A.2. Concepts in Calculus and Graph Calculus

Definition A.2 (Hilbert Space)**.** *A Hilbert space is a complete vector space with an inner product defined on the space.*

Definition A.3 (Complete Graph)**.** *A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.*

Definition A.4 (Cliques)**.** *For an undirected graph $\widehat{\mathcal{G}} = (V, E)$, the set of $k-$th cliques $K_k(\widehat{\mathcal{G}})$ is defined by*

$$\{i_1, \cdots, i_k\} \in K_k(\widehat{\mathcal{G}})$$

if and only if all pairs of vertices in $\{i_1, \cdots, i_k\}$ are in E. Therefore, when $\widehat{\mathcal{G}}$ is a complete graph, the $k-$th cliques of $\widehat{\mathcal{G}}$ is equivalent to $\left\{\binom{V}{k}\right\}$.

Definition A.5 (Alternating Function)**.** *For an undirected graph $\widehat{\mathcal{G}} = (V, E)$, an alternating function on $k-$th cliques is: $f : V \times \cdots \times V \to \mathbb{R}$ satisfying*

$$f(i_\sigma(1), \cdots, i_\sigma(k)) = sgn(\sigma) f(i_1, \cdots, i_k)$$

for all $\{i_1, \cdots, i_k\} \in K_k$ and σ is any permutation on $\{1, \cdots, k\}$. Here $sgn(\sigma)$ denotes the sign of σ which is 1 when the parity of the number of inversions in $(i_\sigma(1), \cdots, i_\sigma(k))$ is even, and $sgn(\sigma) = -1$ if the parity of the number of inversions is odd.

Definition A.6 (L^2 Functions)**.** *For an undirected graph $\widehat{\mathcal{G}} = (V, E)$, the Hilbert space of all potential functions $f : V \to \mathbb{R}$ is denoted as $L^2(V)$, with the inner product taken to be the standard inner product: for $f, g \in L^2(V)$,*

$$\langle f, g\rangle := \sum_{i=1}^{d} f(i)g(i).$$

For the $k-$th cliques, we denote the Hilbert space of all alternating functions on K_k as $L^2_\wedge(K_k)$, with the inner product defined as: for $\Theta, \Phi \in L^2_\wedge(K_k)$,

$$\langle \Theta, \Phi\rangle := \sum_{\{i_1,\cdots,i_k\}\in K_k} \Theta(i_1, \cdots, i_k)\Phi(i_1, \cdots, i_k).$$

Definition A.7 (Curl-Free, Divergence-Free)**.** *An edge function $f \in L^2_\wedge(E)$ is called curl-free if and only if*

$$\mathrm{curl}(f)(i, j, k) = 0, \quad \forall\{i, j, k\} \in T,$$

or, equivalently, $f \in ker(\mathrm{curl})$. Similarly, $f \in L^2_\wedge(E)$ is called divergence-free if and only if

$$\mathrm{div}(f)(i) = -\mathrm{grad}^*(f)(i) = 0, \quad \forall i \in V,$$

or, equivalently, $f \in ker(\mathrm{div}) = ker(\mathrm{grad}^*)$.

Definition A.8 (Harmonic)**.** *An edge function $f \in L^2_\wedge(E)$ is called harmonic if and only if*

$$\triangle_1(f)(i, j) = 0, \quad \forall\{i, j\} \in E,$$

or, equivalently, $f \in ker(\triangle_1)$.

A.3. Concepts in Directed Graphs

Definition A.9 (Connectivity Matrix)**.** *For a directed graph $\mathcal{G} = (V, E)$ with d vertices, its connectivity matrix $C(\mathcal{G})$ is a $d \times d$ matrix such that $[C(\mathcal{G})]_{ij} = 1$ if there exists a directed path from vertex i to vertex j, and $[C(\mathcal{G})]_{ij} = 0$ otherwise.*

B. Proof of Lemmas and Theorems

Here we provide the detailed proof of some critical lemmas and theorems in the main paper.

B.1. Proof of Lemma 3.4

Lemma 3.4 *Consider a complete undirected graph $\widehat{\mathcal{G}}(V, E)$ and a **curl-free** function $Y \in L^2_\wedge(E)$, then $\mathrm{ReLU}(Y) \in \mathbb{R}^{d\times d}$ is the weighted adjacency matrix of a DAG. Moreover, given any skew-symmetric matrix $W \in \mathbb{R}^{d\times d}$, $W \circ \mathrm{ReLU}(Y)$ is also a DAG, where $\circ$ is the Hadamard product.*

Proof. We prove the lemma by contradiction. Assuming that there is a cycle in $\mathcal{G}_{\mathrm{ReLU}(Y)}$ (the graph with weighted adjacency matrix $\mathrm{ReLU}(Y)$) on an (ordered) set of nodes $(c_1, c_2, \cdots, c_k, c_1)$ and denoting $c_{k+1} := c_1$ just for notation simplicity, the curl-free property of Y yields

$$\sum_{i=1}^{k} Y(c_i, c_{i+1}) = \sum_{i=2}^{k-1} \mathrm{curl}(Y)(c_1, c_i, c_{i+1}) = 0.$$

There exists at least 1 pair of (c_i, c_{i+1}) such that $Y(c_i, c_{i+1}) \le 0$ and hence $(c_i, c_{i+1}) \notin E_{\mathrm{ReLU}(Y)}$, which contradicts with the assumption that $(c_1, c_2, \cdots, c_k, c_1)$ forms a cycle. □

B.2. Proof of Theorem 3.7

Theorem 3.7 *Let $A \in \mathbb{R}^{d\times d}$ be the weighted adjacency matrix of a DAG with d nodes, denote $\widehat{\mathcal{G}}(V, E)$ as the complete undirected graph on these d nodes, then there exists a skew-symmetric matrix $W \in \mathbb{R}^{d\times d}$ and a potential function $p \in L^2(V)$ such that $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$, i.e., $\mathbb{D} \subset \{\mathcal{G}_{W\circ\mathrm{ReLU}(\mathrm{grad}(p))}\}$. Here p is associated with the topological order of the DAG, such that $p(j) > p(i)$ if there is a directed path from vertex i to j.*

Proof. We first show that there exists a $p \in L^2(V)$ such that

$$(\mathrm{grad}(p))(i, j) > 0, \quad \text{when } A(i, j) \neq 0. \tag{12}$$

Since $\mathcal{G}_A$ is a DAG, there exists at least one topological (partial) order for its vertices (Bang-Jensen and Gutin, 2008). Taking an topological (partial) order $\prec = (c_1, c_2, \cdots, c_d)$ of all the vertices in $\mathcal{G}_A$, p defined as $p(c_i) = i$ satisfies condition (12). We now construct the weight matrix W. Since A represents a DAG, for any two vertices i and j, at least one or both of $A(i, j) = 0$ and $A(j, i) = 0$ must hold true. We define an skew-symmetric matrix W as:

$$[W]_{ij} = \begin{cases} 0, \text{ if } p(i) = p(j) \text{ or } A(i, j) = A(j, i) = 0; \\ \frac{A(i,j)}{p(j)-p(i)}, \text{if } A(i, j) \neq 0 \text{ and } A(j, i) = 0; \\ \frac{A(j,i)}{p(j)-p(i)}, \text{if } A(i, j) = 0 \text{ and } A(j, i) \neq 0. \end{cases} \tag{13}$$

Then $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$, and we have proved the conclusion. Moreover, combining Theorem 3.5 and Theorem 3.7, we note that

$$\mathbb{D} = \{\mathcal{G}_{W\circ\mathrm{ReLU}(\mathrm{grad}(p))}\},$$

which is our main theoretical result. □

B.3. Proof of Theorem 4.3

Theorem 4.3 *Let $A \in \mathbb{R}^{d\times d}$ be the weighted adjacency matrix of a DAG with d nodes, then*

$$p = -\triangle_0^{\dagger} \,\mathrm{div}\left(\frac{1}{2}(C(A) - C(A)^T)\right), \tag{14}$$

preserves the topological order in A such that $p(j) > p(i)$ if there is a directed path from vertex i to j. Moreover, we have $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$ with the skew-symmetric matrix W defined as in (13).

Proof. Taking any two vertices i, j with a directed path from i to j, we show that $p(j) > p(i)$. We assume that $i, j \neq d$ without loss of generality, since the proof can be trivially extended to the cases of $i = d$ or $j = d$. Since $C(A)$ is the connectivity matrix of A and A is the weighted adjacency matrix of a DAG, we have the following facts hold:

$$[C(A)]_{ii} = [C(A)]_{jj} = [C(A)]_{ji} = 0,\ [C(A)]_{ij} = 1. \tag{15}$$

Moreover, for any other vertex k, if there exists a directed path from j to k, there is also a directed path from i to k. Therefore, $[C(A)]_{jk} = 1 \Rightarrow [C(A)]_{ik} = 1$, i.e., $[C(A)]_{ik} \geq [C(A)]_{jk}$. On the other hand, if there exists a directed path from k to i, there is also a directed path from k to j. Therefore $[C(A)]_{ki} = 1 \Rightarrow [C(A)]_{kj} = 1$ and $[C(A)]_{kj} \geq [C(A)]_{ki}$.

From the definition of p we note that

$$-\triangle_0 p = \mathrm{div}\left(\frac{1}{2}(C(A) - C(A)^T)\right).$$

The i-th and j-th rows of the above system write:

$$-dp(i) + \sum_{k=1}^{d} p(k) = \frac{1}{2}\left(\sum_{k=1}^{d}[C(A)]_{ik} - \sum_{k=1}^{d}[C(A)]_{ki}\right),$$
$$-dp(j) + \sum_{k=1}^{d} p(k) = \frac{1}{2}\left(\sum_{k=1}^{d}[C(A)]_{jk} - \sum_{k=1}^{d}[C(A)]_{kj}\right).$$

Subtracting the above two equations from each other and applying the facts in (15) yield

$$\begin{aligned} d(p(j) - p(i)) =& \frac{1}{2}\sum_{k\neq i,j}([C(A)]_{ik} + [C(A)]_{kj} - [C(A)]_{jk} \\ & - [C(A)]_{ki}) + [C(A)]_{ij} - [C(A)]_{ji} \geq 1. \end{aligned}$$

Therefore $p(j) > p(i)$, and $A = W \circ \mathrm{ReLU}(\mathrm{grad}(p))$ can be similarly proved as in Theorem 3.7. □

C. Complexities

The computational and space complexities depend on the optimization method used. Let K be the time complexity to solve one objective. For example, for L-BFGS $K = O(mn)$, where n is the number of variables and m is the number of steps stored in memory. Our method takes $O(K)$ time, compared to $O(LK)$ of NOTEARS where L is the number of iteration in the augmented Lagrangian method. We share the same space complexity as NOTEARS.

D. Examples of Graph Projection

In this section we provide the detailed procedure of graph projection described in Theorems 4.2 and 4.3, for four representative graphs as shown in Figure 2. In all examples we consider graphs with 4 vertices which are denoted as A, B, C and D in Figure 2. In the following calculations we assume A as the first vertex and D as the last vertex. In

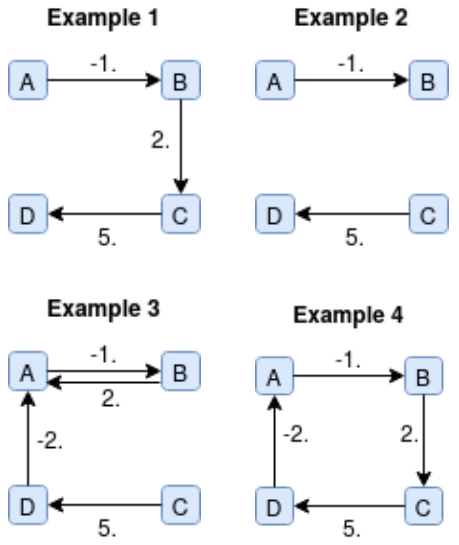

Figure 2. Representative graphs as examples to demonstrate the graph projection procedure.

each example, for a given graph $\mathcal{G}_{A^{pre}}$, we first calculate its approximated gradient flow component via

$$\tilde{p} = -\triangle_0^{\dagger} \operatorname{div}\left(\frac{1}{2}(C(A^{pre}) - C(A^{pre})^T)\right),$$

then the weights $\tilde{W}$ are computed from (10). We note that the matrix for graph Laplacian $\triangle_0$ given by (8) writes:

$$[\triangle_0] = \begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & -3 & -1 & 0 \\ -1 & -1 & -3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $\tilde{p}(4) = 0$ is fixed, we only need to invert the submatrix of $[\triangle_0]$ formed by ignoring its 4-th row and 4-th column, and this submatrix is invertible. Hence the calculation described in Theorem 4.2 is well-posed.

Example 1, projection for a connected acyclic graph (a tree): We first consider a fully connected acyclic graph as shown in the first plot of Figure 2, with the weighted adjacency matrix:

$$A^{pre} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The connectivity matrix of A^{pre} writes

$$C(A^{pre}) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Therefore, the projection result $\tilde{p}$ from (9) and the weights $\tilde{W}$ from (10) are obtained:

$$\tilde{p} = \begin{bmatrix} -0.75 \\ -0.5 \\ -0.25 \\ 0 \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & -4 & 0 & 0 \\ 4 & 0 & 8 & 0 \\ 0 & -8 & 0 & 20 \\ 0 & 0 & -20 & 0 \end{bmatrix}.$$

We then have the acyclic approximation of A^{pre} as

$$\tilde{A} = \tilde{W} \circ \mathrm{ReLU}(\mathrm{grad}(\tilde{p})) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix} = A^{pre}.$$

Therefore, the projected potential function $\tilde{p}$ fully preserves the vertices ordering in this acyclic graph, which is consistent with Theorem 4.3.

Example 2, projection for a disconnected acyclic graph (a forest): Consider an acyclic graph consisting of two trees as shown in the second plot of Figure 2, with the weighted adjacency matrix:

$$A^{pre} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The connectivity matrix of A^{pre} writes

$$C(A^{pre}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Therefore, the projection result $\tilde{p}$ from (9) and the weight $\tilde{W}$ from (10) are

$$\tilde{p} = \begin{bmatrix} -0.25 \\ 0 \\ -0.25 \\ 0 \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & -4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 \\ 0 & 0 & -20 & 0 \end{bmatrix}.$$

We then have the acyclic approximation of A^{pre} as

$$\tilde{A} = \tilde{W} \circ \mathrm{ReLU}(\mathrm{grad}(\tilde{p})) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix} = A^{pre}.$$

The results indicate that the projected potential function $\tilde{p}$ fully preserves the (partial) ordering in each tree, and the projection procedure in Theorem 4.3 maps the acyclic graph to itself.

Example 3, projection for a cyclic graph with cycle length 2: We now consider a cyclic graph with a cycle

between the first and the second vertex, as shown in the third plot of Figure 2, with the weighted adjacency matrix:

$$A^{pre} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ -2 & 0 & 0 & 0 \end{bmatrix}.$$

The connectivity matrix of A^{pre} writes

$$C(A^{pre}) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Therefore, the projection result $\tilde{p}$ and the weight $\tilde{W}$ are

$$\tilde{p} = \begin{bmatrix} 0.375 \\ 0.375 \\ -0.25 \\ 0 \end{bmatrix}, \quad \tilde{W} = \begin{bmatrix} 0 & 0 & 0 & \frac{16}{3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 \\ -\frac{16}{3} & 0 & -20 & 0 \end{bmatrix}.$$

We then have the acyclic approximation of A^{pre} as

$$\tilde{A} = \tilde{W} \circ \mathrm{ReLU}(\mathrm{grad}(\tilde{p})) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 \\ -2 & 0 & 0 & 0 \end{bmatrix} \neq A^{pre}.$$

It can be seen that when there is a local cycle (between nodes A and B in this example), the projection procedure in Theorem 4.3 simply removes all edges involved in this cycle and keeps the ordering of vertices from all other edges.

Example 4, projection for a cyclic graph with cycle length 4: We now consider a cyclic graph as shown in the last plot of Figure 2, with the weighted adjacency matrix:

$$A^{pre} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 5 \\ -2 & 0 & 0 & 0 \end{bmatrix}.$$

The connectivity matrix of A^{pre} writes

$$C(A^{pre}) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Therefore, $C(A^{pre}) - C(A^{pre})^T) = 0$, and the projection results $\tilde{p} = (0, 0, 0, 0)^T$. We then have the acyclic approximation of A^{pre} as

$$\tilde{W} \circ \mathrm{ReLU}(\mathrm{grad}(\tilde{p})) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This example illustrates that when there is a cycle with length greater than 2, the projection procedure in Theorem 4.3 removes all edges between any two nodes in this cycle.

E. Detailed Algorithm and Experiment Settings

E.1. Settings on Synthetic Dataset

In the following we briefly describe the empirical process of generating synthetic datasets. The code will be publicly released at https://github.com/fishmoon1234/DAG-NoCurl.

Linear synthetic datasets: In the linear SEM tests, for each $d \in \{10, 30, 50, 100\}$ and each graph type-noise type combination, 100 trials were performed with 1000 samples in each dataset. For each trial, a ground truth DAG $\mathcal{G}_{A^0}$ is randomly sampled following either the Erdős–Rényi (ER) or the scale-free (SF) scheme. When (i, j) is a directed edge of the ground truth DAG $\mathcal{G}_{A^0}$, the weight of this edge A^0_{ij} is sampled from $\mathcal{U}([-2, -0.5] \cup [0.5, 2])$. Each sample $X^i \in \mathbb{R}^d$, $i = 1, \cdots, 1000$, is generated following:

$$X^i_j = (a^0_j)^T \pi^0(X^i_j) + Z^i_j$$

where X^i_j is the ith sample of jth variable X_j, $a^0_j \in \mathbb{R}^d$ is the jth column of the ground truth weighted adjacency matrix $A^0 = [a^0_1 | \cdots | a^0_d]$, $\pi^0(X^i_j)$ is a random vector of size d containing the variable values corresponding to the parents of jth variable X_j per A^0 in the ith sample, i.e., its k-th component $[\pi^0(X^i_j)]_k = X^i_k$ if X_k is a parent of X_j in A^0 otherwise $[\pi^0(X^i_j)]_k = 0$, Z^i_j is either a Gaussian noise $Z^i_j \sim \mathcal{N}(0, 1)$ or a Gumbel noise $Z^i_j \sim \text{Gumbel}(0, 1)$.

Nonlinear synthetic datasets: In the nonlinear SEM tests, 5 trials were performed for each case with 5000 samples in each dataset. For each trial the ground truth DAG $\mathcal{G}_{A^0}$ and the weighted adjacency matrix A^0 are generated following the same way as in the linear SEM tests. Three types of datasets were considered:

- *Nonlinear Case 1*: For each $d \in \{10, 20, 50, 100\}$, each sample $X^i \in \mathbb{R}^d$, $i = 1, \cdots, 5000$, is generated following

  $$X^i_j = \cos((a^0_j)^T \pi^0(X^i_j) + 1) + Z^i_j$$

  where A^0 is the weighted adjacency matrix of a graph sampled following the Erdős–Rényi (ER) scheme with $3d$ expected edges (denoted as **ER3**), and the noise $Z^i_j \sim \mathcal{N}(0, 1)$.

- *Nonlinear Case 2*: For each $d \in \{10, 20, 50, 100\}$, each sample $X^i \in \mathbb{R}^d$, $i = 1, \cdots, 5000$, is generated following

  $$\begin{aligned} X^i_j =& 2\sin((a^0_j)^T \pi^0(X^i_j) + 0.5) \\ &+ ((a^0_j)^T \pi^0(X^i_j) + 0.5) + Z^i_j \end{aligned}$$

where A^0 is the weighted adjacency matrix of a graph sampled following the Erdős–Rényi (ER) scheme with $3d$ expected edges (denoted as **ER3**), and the noise $Z_j^i \sim \mathcal{N}(0,1)$.

- *Nonlinear Case 3*: For each $d \in \{10, 20, 50, 100\}$, each sample $X^i \in \mathbb{R}^d$, $i = 1, \cdots, 5000$, is generated following

$$X_j^i = (a_j^0)^T \cos(\pi^0(X_j^i) + \mathbf{1}) + Z_j^i$$

where A^0 is the weighted adjacency matrix of a graph sampled following the Scale-free (SF) scheme with $3d$ expected edges (denoted as **SF3**), and the noise $Z_j^i \sim \mathcal{N}(0,1)$.

Here we note that Nonlinear Case 1 and 2 were adopted from (Yu et al., 2019). In Nonlinear Case 3, each sample were generated following almost the same scheme as in Nonlinear Case 1, but the ground truth graph was generated with the SF model. Comparing with the ER graphs which have a degree distribution following a Poisson distribution, SF graphs have a degree distribution following a power law and therefore few nodes have a high degree (Lachapelle et al., 2019).

E.2. Settings for Each Algorithm

In this section we describe the settings and parameters employed in each algorithm.

E.2.1. Linear SEM

DAG-NoCurl: In linear SEM we use the least-squares loss

$$F_{SEM}(A, \mathbf{X}) = \frac{1}{2n}||\mathbf{X} - A^T\mathbf{X}||_F^2 \tag{16}$$

regardless of the noise type, with the polynomial acyclicity penalty from (Yu et al., 2019)

$$h(A) = \text{tr}[(I + A \circ A/d)^d] - d. \tag{17}$$

We consider the penalty parameter λ in DAG-NoCurl as a tunable hyperparameter, with the range of $\{1, 10, 10^2, 10^3, 10^4\}$. We use the runtime and the score difference from the ground truth $\Delta F = F_{SEM}(A, \mathbf{X}) - F_{SEM}(A^0, \mathbf{X})$ as the measure to choose the best hyperparameters. For the detailed analysis and discussion, please refer to the Section F on hyperparameter study of this supplemental material. To solve for the unconstrained smooth minimization problems, although a number of efficient numerical algorithms are available, we employ the L-BFGS (Liu and Nocedal, 1989) algorithm with the stopping tolerance "ftol" (the relative score difference between the last two iterations) set as 10^{-8}. The implementation is in Python based on the original NOTEARS package from (Zheng et al., 2018). Unless otherwise stated, we use the threshold 0.3 on A^{pre} and $\tilde{A}$, as suggested in (Zheng et al., 2018).

NOTEARS: For baseline method NOTEARS, we use the NOTEARS package in Python from (Zheng et al., 2018) with the least-squares loss (16) and the polynomial acyclicity penalty (17). For the augmented Lagrangian method in NOTEARS, we use default parameters from the package, and the default stopping criteria $h(A) \leq 10^{-8}$.

GOBNILP: For the exact minimizer of the original optimization problem, we use the publicly available package Globally Optimal Bayesian Network learning using Integer Linear Programming (GOBNILP) (Cussens et al., 2016)[1]. It uses integer linear programming written in C program and SCIP optimization solvers to learn BN from complete discrete data or from local scores. We use GaussianL0 score with $k = 0.0$ and did not set a maximal parental set size ("palim = None"). We did not change any other parameter setting.

FGS: For baseline method fast greedy equivalent search (FGS), we use py-causal package from Carnegie Mellon University (Ramsey et al., 2017)[2]. This method is written in highly optimized Java code with a Python interface. We use the default parameter settings and did not tune any parameter. Instead of returning a DAG, a CPDAG is returned by FGS which contains undirected edges. Therefore, in our evaluations for FGS, we favorably treat undirected edges from FGS as true positives, as long as the ground truth graph has a directed edge in place of the undirected edge.

CAM: For baseline method causal additive models (CAM) (Bühlmann et al., 2014), we use Causal Discovery toolbox in Python[3]. Only two input parameters, "variablesel" and "pruning", were tuned, which enables preliminary neighborhood selection and pruning, respectively. We found that with the preliminary neighborhood selection applied the time consumption of CAM is reduced significantly, and the pruning step helps reducing the resultant SHD and therefore improves the accuracy. These observations are consistent with the experiments reported in (Bühlmann et al., 2014). Therefore, all results reported here are with these two parameters turned on.

MMPC: For baseline method Max-Min Parents and Children (MMPC) (Tsamardinos et al., 2006a), we also use Causal Discovery toolbox in Python[4], with the default parameter settings.

[1] https://www.cs.york.ac.uk/aig/sw/gobnilp/

[2] https://github.com/bd2kccd/py-causal

[3] https://github.com/FenTechSolutions/CausalDiscoveryToolbox

[4] https://github.com/FenTechSolutions/CausalDiscoveryToolbox

Algorithm 2 NoCurl algorithm combining with DAG-GNN

1. Step 1: Solve for an initial prediction (A^{pre}, θ^{pre}) with

$$(A^{pre}, \theta^{pre}) = \underset{A,\theta}{\text{argmin}} \{-L_{\text{ELBO}}(A, \mathbf{X}) + \lambda(\text{tr}[(I + A \circ A/d)^d] - d)\}$$

and threshold A^{pre}.
2. Step 2: Based on A^{pre}, obtain an approximate solution of p^* as $\tilde{p}$ with

$$\tilde{p} = -\triangle_0^{\dagger} \,\text{div}\left(\frac{1}{2}(C(A^{pre}) - C(A^{pre})^T)\right),$$

then solve for $\tilde{W}$ with fixed $\tilde{p}$ via

$$(\tilde{W}, \tilde{\theta}) = \underset{W,\theta}{\text{argmin}} \; - L_{\text{ELBO}}(W \circ \text{ReLU}(\text{grad}(\tilde{p})), \mathbf{X})$$

In this step, the initial prediction for parameters θ^{pre} from Step 1 is used as the initial guess of θ.
3. Obtain the final approximation solution $\tilde{A} = \tilde{W} \circ \text{ReLU}(\text{grad}(\tilde{p}))$ and threshold $\tilde{A}$.

Eq-TD & Eq-BU: we use the available code form github[5] and the same named functions as listed. We did not tune any hyperparameters.

E.2.2. Nonlinear SEM

DAG-GNN with NoCurl: In nonlinear SEM we combine NoCurl with DAG-GNN (Yu et al., 2019). In DAG-GNN, a deep generative model is employed to learn the DAG by maximizing the evidence lower bound (ELBO):

$$L_{\text{ELBO}}(A, \mathbf{X}) = \frac{1}{n}\sum_{k=1}^{n} L^k_{\text{ELBO}}(A, X^k)$$
$$\text{where } L^k_{\text{ELBO}}(A, X^k) = -D_{\text{KL}}\Big(q(Y|X^k; A) \,||\, p(Y)\Big) + \mathbb{E}_{q(Y|X^k;A)}\Big[\log p(X^k|Y; A)\Big].$$

Following the settings in (Yu et al., 2019), $Y \in \mathbb{R}^d$ is a latent variable and $p(Y)$ is the prior modeled with the standard multivariate normal $p(Y) = \mathcal{N}(0, I)$. $q(Y|X; A)$ is the variational posterior to approximate the actual posterior $p(Y|X)$, and D_{KL} denotes the KL-divergence between the variational posterior and the actual one. $q(Y|X; A)$ is modeled with a factored Gaussian with mean $M_Y \in \mathbb{R}^d$ and standard deviation $S_Y \in \mathbb{R}^d$, based on a multilayer perception (MLP):

$$[M_Y | \log S_Y] = (I - A^T)\text{MLP}(X, M^1, M^2)$$

where $M^1 \in \mathbb{R}^{1\times n_{hid}}$ and $M^2 \in \mathbb{R}^{n_{hid}\times 1}$ are parameters and n_{hid} is the number of neurons in the hidden layer. Similarly, $p(X|Y; A)$ is also modeled with a factored Gaussian with mean $M_X \in \mathbb{R}^d$ and standard deviation $S_X \in \mathbb{R}^d$, based on a multilayer perception (MLP):

$$[M_X | \log S_X] = \text{MLP}((I - A^T)^{-1}Y, M^3, M^4)$$

where $M^3 \in \mathbb{R}^{1\times n_{hid}}$ and $M^4 \in \mathbb{R}^{n_{hid}\times 1}$ are parameters. In DAG-GNN, the weighted adjacency matrix A is optimized together with all the parameters $\theta = (M^1, M^2, M^3, M^4)$ with the following learning problem:

$$(A^*, \theta^*) = \underset{A,\theta}{\text{argmin}} \; - L_{\text{ELBO}}(A, \mathbf{X}), \quad \text{subject to } h(A) = \text{tr}[(I + A \circ A/d)^d] - d = 0. \tag{18}$$

With the goal of boosting the efficiency of DAG-GNN without losing accuracy, in NoCurl we use the same score function from DAG-GNN:

$$F_{\text{ELBO}}(A, \mathbf{X}) = -L_{\text{ELBO}}(A, \mathbf{X})$$

and their implementation based on PyTorch (Paszke et al., 2017). The detailed steps are described in Algorithm 2. Specifically, we use DAG-GNN's default number of neurons in the hidden layer $n_{hid} = 64$. Each unconstrained optimization problem is solved using the Adam (Kingma and Ba, 2015), with the default learning rate$= 3e-3$ from DAG-GNN. To guarantee sufficient updates for the parameters θ, we use epoch number$= 400$ in Step 1 and epoch number$= 600$ while solving for $\tilde{W}$ in Step 2. Due to the computation load of neural models, we use one fixed $\lambda = 10$ as the hyperparameter in NoCurl since it is the fastest method while being reasonably accurate per our hyperparameter study on linear SEM datasets (see the hyperparameter study in Section F of this supplementary material).

DAG-GNN: We use the available code from github [6] to run DAG-GNN. We use the default hidden size 64 for all layers and did not tune any other hyperparameters (all default values).

GraN-DAG: We use the available code from github [7]. Following the suggestion in (Lachapelle et al., 2019), we turned on both the preliminary neighborhood selection (`pns`) and the pruning option (`cam-pruning`), for which we have observed a big improvement in SHD. For the rest of hyperparameters, we use default values with options `pns` and `cam-pruning`.

[5] https://github.com/WY-Chen/EqVarDAG

[6] https://github.com/fishmoon1234/DAG-GNN

[7] https://github.com/kurowasan/GraN-DAG

NOTEARS-MLP: We use the available code from github [8] to run NOTEARS-MLP. We tune the hidden size to 32 for all layers (increased from default size 10, for which we observe a big improvement in SHD) to improve the accuracy. All other hyperparameters are kept as their default values: the augmented Lagrangian method terminates when $h(A) = \text{tr}[(I + A \circ A/d)^d] - d \leq 10^{-8}$ and $\lambda 1$ and $\lambda 2$ are set as 0.01.

For **CAM** and **MMPC**, we use the same settings as discussed in the previous subsection.

GSGES: We use the available code from github [9]. We use the default settings for evaluations.

E.3. Other Experiment Details

A clear definition of the specific measure or statistics used to report results: To evaluate the accuracy of results from each algorithm, we mainly use the structure hamming distance (SHD) as a metric, which is the sum of extra, missing, and reverse edges in learned graphs. We report the computational time (in seconds) of each algorithm, as a main metric of their computational efficiency. When it is available, we also report the score difference from the ground truth (denoted as ΔF), the number of extra edges (denoted as #Extra E), the number of missing edges (denoted as #Missing E) and the number of reverse edges (denoted as #Reverse E). All metrics are the lower the better.

A description of results with central tendency (e.g. mean) & variation (e.g. error bars): We report mean and standard error of the mean for each metric, with a format as "mean $\pm$ standard error".

The average runtime for each result, or estimated energy cost: We use CPU and report the run time (in seconds) for each algorithm. We run all the algorithms up to 72 hours for each trial.

A description of the computing infrastructure used: We use a local Linux-based computing cluster, and all the codes are written in Python and/or PyTorch.

F. Hyperparameter Study

In this section we continue the discussion on hyperparameter study results in Section 5.1 of the main text and conduct a hyperparameter study for linear SEMs, with one fixed λ or two fixed λ's in Step 1 of the proposed algorithm. In particular, in the one fixed λ cases (denoted as the $\lambda = \cdot$ cases), we obtain the estimate A^{pre} in Step 1 by solving for only one unconstrained optimization problem:

$$A^{pre} = \underset{A}{\text{argmin}}\ F(A, \mathbf{X}) + \lambda h(A),$$

where $A \in \mathbb{R}^{d\times d}$ is initialized as $A_{ij} = 0$, $\forall i, j \in \{1, \cdots, d\}$. In the two fixed λ's cases (denoted as the $\lambda = (\lambda_1, \lambda_2)$ cases), we obtain the estimate A^{pre} in Step 1 by solving for two optimization problems sequentially. We firstly solve:

$$A^{pre,0} = \underset{A}{\text{argmin}}\ F(A, \mathbf{X}) + \lambda_1 h(A),$$

with initial guess $A_{ij} = 0$, $\forall i, j \in \{1, \cdots, d\}$, then use $A^{pre,0}$ as the initial guess to solve

$$A^{pre} = \underset{A}{\text{argmin}}\ F(A, \mathbf{X}) + \lambda_2 h(A)$$

for the estimate matrix A^{pre}. Here we explore the hyperparameter λ on ER3-Gaussian and ER6-Gaussian cases, to investigate the performances of NoCurl on both relatively sparse graphs (ER3) and relatively dense graphs (ER6). The results for ER3-Gaussian are provided in Table 4 and the results for ER6-Gaussian is in Table 5. For all cases we report the structure hamming distance (SHD), the score difference from the ground truth (denoted as ΔF), the run time (in seconds), the number of extra edges (denoted as #Extra E), the number of missing edges (denoted as #Missing E) and the number of reverse edges (denoted as #Reverse E), while we choose the hyperparameter mainly based on the considerations of both a low run time and a good resultant score from the predicted graph (low ΔF).

For cases with one fixed λ, we investigate the hyperparameter $\lambda \in [10^0, 10^4]$. From Tables 4 and 5 it can be observed that in both ER3-Gaussian and ER6-Gaussian cases, comparing with the other values of λ's, tests with $\lambda = 10$ and $\lambda = 10^2$ generally require short run time and their predicted graphs have relatively good scores according to their resultant loss values ΔF. $\lambda = 10$ is faster and more accurate in SHD in ER3 than $\lambda = 10^2$ for all d, but $\lambda = 10^2$ has better (i.e., lower) ΔF loss values. In denser graphs (ER6), $\lambda = 10^2$ becomes significantly better in both ΔF and SHD. As a result, we use $\lambda = 10^2$ as the default hyperparameter value for one fixed λ, which becomes **NoCurl-1**, experiments in the following.

For cases with with two fixed λ's, we test the cases with $\lambda_1, \lambda_2 \in [10^0, 10^4]$, and list some combinations with results in Tables 4 and 5. Specifically, in most cases $\lambda = (10, 10^3)$ and $\lambda = (10, 10^4)$ are the two combinations with the best score values ΔF. Among these two combinations, we found that $\lambda = (10, 10^4)$ results in slightly lower ΔF loss and SHD, but $\lambda = (10, 10^3)$ requires a lower run time, especially when d is large. Here we choose $\lambda = (10, 10^3)$ as the default parameters in two fixed λ's experiments, which becomes **NoCurl-2**.

[8] https://github.com/xunzheng/notears

[9] https://github.com/Biwei-Huang/Generalized-Score-Functions-for-Causal-Discovery

From Tables 4 and 5, we also observe that, to achieve the optimal loss and accuracy, larger and denser graphs generally require a larger value of penalty parameter λ. As a future direction, we are investigating the strategy of choosing λ automatically.

G. Ablation Study

In this section we continue the discussion on ablation study results in Section 5.2 of the main text and perform an ablation study, to investigate the effects of each step in our proposed algorithm. In particular, results from the following five settings are listed in Tables 4 and 5:

- **rand init cases**: We solve for $(\tilde{W}, \tilde{p})$ from the optimization problem

$$(\tilde{W}, \tilde{p}) = \underset{W \in S, p \in \mathbb{R}^d}{\operatorname{argmin}} \; F(W \circ \operatorname{ReLU}(\operatorname{grad}(p)), \mathbf{X}) \tag{19}$$

  directly, with random initialization of (W, p). The results are the average from 7 different random initializations $W_{ij} \sim \mathcal{U}([0, 1])$, $p_i \sim \mathcal{U}([0, 1])$ for each set of data. With this test we aim to investigate the importance of both Step 1 and Step 2.

- **rand p cases**: We omit Step 1 and initialize p^{init} with random initializations, then solve for an estimate of W from

$$W^{pre} = \underset{W \in S}{\operatorname{argmin}} \; F(W \circ \operatorname{ReLU}(\operatorname{grad}(p^{init})), \mathbf{X})$$

  and finally jointly optimize $(\tilde{W}, \tilde{p})$ from the optimization problem in (19). The results are also the average from 7 different random initializations of p following $p_i \sim \mathcal{U}([0, 1])$ for each set of data. With this test we aim to investigate the importance of Step 1.

- **$\lambda = 10^2$s and $\lambda = (10, 10^3)$s cases, which are NoCurl-1s and Nocurl-2s with "s"**: We test if Step 2 of the algorithm is important. In particular, we solve Step 1 and then use an incremental thresholding method to obtain a DAG from the potential cyclic graph A^{pre} of Step 1. In these cases, we repeatedly increase the threshold of the structure until a DAG is obtained. We use the thresholds starting from 0.3 (anything below produces much worse results) and with increments of 0.05 until $h(A) < 10^{-8}$.

- **$\lambda = 10^2-$ and $\lambda = (10, 10^3)-$ cases, which are NoCurl-1- and Nocurl-2- with "-"**: Instead of solving for $\tilde{W}$ from the optimization problem

$$\tilde{W} = \underset{W \in S}{\operatorname{argmin}} \; F(W \circ \operatorname{ReLU}(\operatorname{grad}(\tilde{p})), \mathbf{X}), \tag{20}$$

  we estimate W directly from A^{pre} with the formulation (13) above. When A^{pre} is a DAG, the formulation (13) will fully recover A^{pre}. Otherwise, when there is a cycle in $\mathcal{G}_{A^{pre}}$, this formulation will remove all edges between any two nodes in this cycle. With this study we aim to check the importance of the second part of Step 2, i.e., solving for $\tilde{W}$ from (20).

- **$\lambda = 10^2+$ and $\lambda = (10, 10^3)+$ cases, which are NoCurl-1+ and Nocurl-2+ with "+"**: After Step 1 and Step 2 of our algorithm, We add one additional post-processing step to jointly optimize $(\tilde{W}, \tilde{p})$ from the optimization problem in (19), so as to guarantee that the solution is a stationary point of the optimization problem (19). This study aims to investigate how far our approximated solution is from a stationary point.

As one can see from Tables 4 and 5, NoCurl with random initializations ("rand init") performs subpar, indicating the importance of Step 1 of our algorithm. Among the two random initialization cases, the "rand p" cases have a even worse accuracy, especially on the number of reserved edges, which indicates that a good estimate of the topological ordering in p plays a critical role in the algorithm. Results from threshold s cases show that they are not as good as the full algorithm, indicating that Step 2 is also critical to the performance of our method. Moreover, we list all threshold s cases from other empirical settings in Table 4 to 5 in Section I of the supplemental material, to show that poor results are consistent across different settings. In addition, by comparing the $\lambda = 10^2$ case with $\lambda = 10^2-$ case and the $\lambda = (10, 10^3)$ case with $\lambda = (10, 10^3)-$ case, we found that although the $\lambda = 10^2-$ and $\lambda = (10, 10^3)-$ cases are less likely to predict a wrong extra edge, but their predicted graphs tend to miss a relatively large number of edges and therefore have a large SHD. When there is a cycle in $\mathcal{G}_{A^{pre}}$, the formulation (13) will remove all edges between any two nodes in this cycle. On the other hand, the numbers of missing edges from the $\lambda = 10^2$ and $\lambda = (10, 10^3)$ cases are much lower, which indicates that the algorithm has successfully recovered some of the lost edges when solving for $\tilde{W}$ from (20). Lastly, by comparing the $\lambda = 10^2$ cases with $\lambda = 10^2+$ cases and the $\lambda = (10, 10^3)$ cases with $\lambda = (10, 10^3)+$ cases, we observe that adding extra optimization steps after Step 2 does not result much improvements on accuracy or ΔF. This result indicates that the estimated solution $(\tilde{W}, \tilde{p})$ from our algorithm is often very close to a stationary point of (19).

H. Optimization Objective Results

In this section we continue the discussion on optimization objective results in Section 5.3 of the main text, by displaying the additional results for optimization objective

results $\Delta F = F(\tilde{A}, \mathbf{X}) - F(A^0, \mathbf{X})$ for different graph-type and noise-type combinations in Table 3. As one may see, the two fixed λ case can achieve close objective values to NOTEARS, while in the denser graph case (ER6) the $\lambda = (10, 10^3)$ case even outperforms NOTEARS when $d = 30$ and $d = 50$. This result is encouraging but also surprising since the problem is often more difficult as the graph becomes larger and denser, and our algorithm only provides an approximated solution. We suspect one major reason could be the optimization difficulty in larger and dense graphs, which could easily be stuck at one of many more stationary points. We leave it to future work to investigate these problems further.

I. Detailed Results for Structure Recovery

In this section we provide the detailed numerical results of linear synthetic datasets for different algorithms, as a continuation of the discussion in Section 5.4 of the main text and as the supplementary results of the structure discovery in terms of SHD and the run time plotted in Figure 1 of the main text. The full results for ER3-Gaussian, ER4-Gaussian, ER6-Gaussian and SF4-Gumbel cases are provided in Tables 6, 7, 8 and 9, respectively. Besides SHD, we further list ΔF, the number of extra edges, missing edges and reverse edges as additional algorithm evaluation metric. From these tables we can see that the most accurate structure discovery results in terms of SHD are either from NOTEARS or NoCurl, while the other three algorithms (FGS, CAM and MMPC) rapidly deteriorates as the number of edges increase. Among the total 16 cases with different combinations of $d \in \{10, 30, 50, 100\}$ and graph/noise-types, NoCurl-2 outperforms NOTEARS (as well as all other algorithms) with a lower SHD in most (12 out of 16) cases. We further observe that the low SHD from NoCurl comes from the fact that this algorithm tends to miss much fewer numbers of edges comparing with other algorithms especially in large and dense graphs, possibly because Step 2 in NoCurl has successfully recovered some lost edges, as we have observed and discussed in the Ablation Study section G above. When comparing the computational time, NoCurl is faster than NOTEARS by one or two orders of magnitude.

J. Detailed Results for Nonlinear Synthetic Datasets

In this section we provide additional results and discussions for the experiments on nonlinear SEM datasets in Section 5.5 of the main text, with the details of dataset generation and algorithm settings provided in Section E.2.2 above. Table 10 shows the full results of the various methods in nonlinear synthetic datasets. We note that some $d = 100$ results for NOTEARS-MLP, GraN-DAG, and GSGES are missing since these algorithms could not finish within 72 hours on at least one trial.[10] Moreover, we also observe that the run time for MMPC vary drastically, potentially due to the conditioned variable set size. When its size is large, exhaustive search becomes prohibitively expensive. GSGES is the most accurate non-neural-network methods, and even outperforms one neural method, Gran-DAG, in Nonlinear 3 cases. However, its running time get prohibitive with large dimensions.

As one can see, in the Nonlinear Case 1 datasets, GraN-DAG and CAM perform the best among all the methods but become the worst in Nonlinear Case 2 and 3 datasets among finished methods. NOTEARS-MLP performs the best in Nonlinear Case 3 but does not do as well in Nonlinear Case 1 and 2 and have trouble handling larger graphs. DAG-GNN and NoCurl does the best in Nonlinear Case 2 cases in comparison and are able to beat NOTEARS-MLP in Nonlinear Case 1 when d is larger. It shows there is no universal best nonlinear DAG learner. Moreover, NoCurl with DAG-GNN as the base model performs as well as DAG-GNN and takes about 3 to 4 times faster, which indicate that NoCurl has successfully boost the efficiency of DAG-GNN without deteriorate its accuracy. NoCurl is also more than one order of magnitude of faster than Gran-DAG and NOTEARS-MLP in many testing cases.

As discussed in the main text, it should be interesting to extend NoCurl to use NOTEARS-MLP and Gran-DAG as base models by considering gradient-based adjacency matrix.

K. Learnt Protein Network

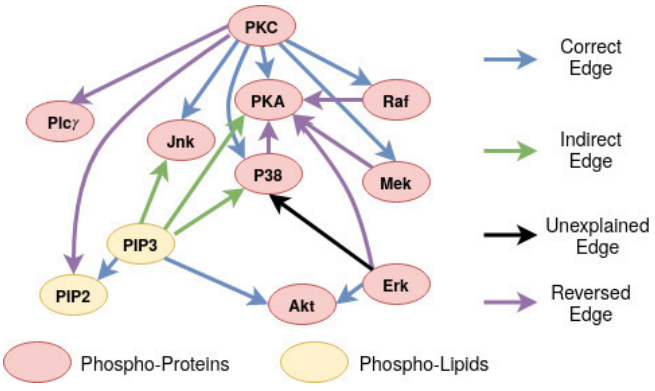

Figure 3. Learned protein signaling network.

We now consider a real-world bioinformatics dataset (Sachs

[10] Although GraN-DAG without the pruning phase finishes within 72 hours in Nonlinear Case 1 and Nonlinear Case 3, the results are with a very large SHD (762 ± 345 in Nonlinear Case 1 and 1606 ± 421 in Nonlinear Case 3), since the pruning step generally has the effect of greatly reducing the SHD, as reported in (Lachapelle et al., 2019).

Table 3. Comparison of different algorithms on score differences from the ground truth, $\triangle F = F(\tilde{A}, \mathbf{X}) - F(A^0, \mathbf{X})$. For each algorithm we show results as mean $\pm$ standard error over 100 trials.

	$\lambda = 10^2$	$\lambda = (10, 10^3)$	NOTEARS	GOBNILP
ER3-Gaussian, $d = 10$	0.09 ± 0.20	0.06 ± 0.20	0.03 ± 0.12	-0.03 ± 0.00
ER4-Gaussian, $d = 10$	0.14 ± 0.03	0.13 ± 0.34	0.08 ± 0.21	-0.03 ± 0.01
ER6-Gaussian, $d = 10$	0.54 ± 0.22	0.36 ± 0.75	0.22 ± 0.40	-0.03 ± 0.01
SF4-Gumbel, $d = 10$	-0.59 ± 0.01	-0.59 ± 0.14	-0.71 ± 0.08	-1.73 ± 0.07
ER3-Gaussian, $d = 30$	0.33 ± 0.19	0.07 ± 0.04	-0.06 ± 0.02	N/A
ER4-Gaussian, $d = 30$	0.31 ± 0.05	0.40 ± 0.19	0.25 ± 0.11	N/A
ER6-Gaussian, $d = 30$	1.78 ± 0.38	0.97 ± 0.16	1.02 ± 0.18	N/A
SF4-Gumbel, $d = 30$	-3.30 ± 0.04	-3.31 ± 0.02	-3.55 ± 0.02	N/A
ER3-Gaussian, $d = 50$	0.05 ± 0.05	-0.10 ± 0.05	-0.25 ± 0.04	N/A
ER4-Gaussian, $d = 50$	0.40 ± 0.13	0.42 ± 0.21	0.19 ± 0.09	N/A
ER6-Gaussian, $d = 50$	2.31 ± 0.41	1.77 ± 0.38	1.97 ± 0.26	N/A
SF4-Gumbel, $d = 50$	-6.74 ± 0.03	-6.74 ± 0.03	-7.08 ± 0.02	N/A
ER3-Gaussian, $d = 100$	-0.82 ± 0.16	-1.44 ± 0.95	-1.65 ± 0.78	N/A
ER4-Gaussian, $d = 100$	-0.28 ± 0.26	-0.32 ± 2.76	-0.64 ± 1.35	N/A
ER6-Gaussian, $d = 100$	4.30 ± 0.99	2.61 ± 9.32	2.49 ± 3.67	N/A
SF4-Gumbel, $d = 100$	-17.29 ± 0.05	-17.19 ± 0.58	-17.53 ± 0.49	N/A

et al., 2005) for the discovery of a protein signaling network based on expression levels of proteins and phospholipids. This is a widely used dataset for research on graphical models, with experimental annotations accepted by the biological research community. In Table 11, we compare our results and 6 baseline methods against the ground truth offered in (Sachs et al., 2005). On this dataset, DAG-NoCurl successfully learns the existence of 14 out of 20 ground-truth edges, and predicts the directions of 8 edges correctly (the learnt graph is plotted in Appendix Figure 3).

Table 4. Hyperparameter and Ablation Study: results (mean ± standard error over 100 trials) for ER3-Gaussian Cases from DAG-NoCurl, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	$\lambda = 1$	0.08 ± 0.00	0.98 ± 0.23	4.12 ± 0.30	1.89 ± 0.17	1.50 ± 0.14	0.73 ± 0.09
10	$\lambda = 10$	**0.07 ± 0.00**	0.22 ± 0.07	1.50 ± 0.22	0.72 ± 0.13	0.31 ± 0.08	0.47 ± 0.06
10	$\lambda = 10^2$	0.11 ± 0.00	0.09 ± 0.02	2.18 ± 0.28	1.24 ± 0.18	0.26 ± 0.06	0.68 ± 0.08
10	$\lambda = 10^3$	0.38 ± 0.01	0.16 ± 0.03	3.14 ± 0.37	1.90 ± 0.25	0.39 ± 0.08	0.85 ± 0.09
10	$\lambda = 10^4$	0.85 ± 0.02	0.28 ± 0.04	4.11 ± 0.42	2.43 ± 0.28	0.54 ± 0.09	1.14 ± 0.10
10	$\lambda = (10, 10^2)$	0.23 ± 0.01	0.08 ± 0.02	1.24 ± 0.20	0.65 ± 0.13	0.19 ± 0.05	0.40 ± 0.06
10	$\lambda = (10, 10^3)$	0.47 ± 0.01	**0.06 ± 0.02**	**1.08 ± 0.18**	0.54 ± 0.12	0.09 ± 0.03	0.45 ± 0.06
10	$\lambda = (10, 10^4)$	0.85 ± 0.02	0.08 ± 0.02	1.43 ± 0.23	0.83 ± 0.17	**0.08 ± 0.03**	0.52 ± 0.07
10	$\lambda = (10^2, 10^3)$	0.44 ± 0.01	0.07 ± 0.02	2.09 ± 0.27	1.20 ± 0.18	0.25 ± 0.06	0.64 ± 0.08
10	rand init	0.19 ± 0.01	5.97 ± 1.63	9.73 ± 0.54	4.59 ± 0.31	2.78 ± 0.26	2.35 ± 0.12
10	rand p	0.32 ± 0.03	11.59 ± 2.46	14.88 ± 0.57	6.48 ± 0.33	5.06 ± 0.30	3.34 ± 0.15
10	$\lambda = 10^2$s	0.09 ± 0.00	1.45 ± 0.02	3.11 ± 0.31	1.60 ± 0.18	0.95 ± 0.11	0.56 ± 0.08
10	$\lambda = (10, 10^3)$s	0.43 ± 0.01	0.53 ± 0.02	1.27 ± 0.20	0.66 ± 0.14	0.20 ± 0.05	0.41 ± 0.06
10	$\lambda = 10^2$-	0.11 ± 0.00	19.95 ± 0.43	3.62 ± 0.27	0.64 ± 0.11	2.93 ± 0.19	0.05 ± 0.03
10	$\lambda = (10, 10^3)$-	0.32 ± 0.01	18.30 ± 0.42	2.09 ± 0.17	**0.30 ± 0.08**	1.79 ± 0.13	**0.00 ± 0.00**
10	$\lambda = 10^2$+	0.29 ± 0.01	0.10 ± 0.02	2.15 ± 0.27	1.22 ± 0.18	0.25 ± 0.06	0.68 ± 0.08
10	$\lambda = (10, 10^3)$+	0.71 ± 0.01	**0.06 ± 0.02**	**1.08 ± 0.18**	0.54 ± 0.12	0.09 ± 0.03	0.45 ± 0.06
30	$\lambda = 1$	0.54 ± 0.03	2.55 ± 0.37	13.64 ± 0.77	8.34 ± 0.55	2.83 ± 0.19	2.47 ± 0.15
30	$\lambda = 10$	0.59 ± 0.03	0.50 ± 0.18	6.46 ± 0.50	4.14 ± 0.38	0.62 ± 0.09	1.70 ± 0.12
30	$\lambda = 10^2$	1.19 ± 0.04	0.33 ± 0.19	7.18 ± 0.61	5.05 ± 0.49	0.40 ± 0.07	1.73 ± 0.12
30	$\lambda = 10^3$	1.29 ± 0.03	0.23 ± 0.05	10.13 ± 0.72	7.33 ± 0.57	0.41 ± 0.07	2.39 ± 0.15
30	$\lambda = 10^4$	4.88 ± 0.19	0.23 ± 0.04	11.67 ± 0.78	8.36 ± 0.60	0.49 ± 0.09	2.82 ± 0.16
30	$\lambda = (10, 10^2)$	0.64 ± 0.02	0.15 ± 0.05	5.41 ± 0.49	3.56 ± 0.37	0.42 ± 0.08	1.43 ± 0.12
30	$\lambda = (10, 10^3)$	2.38 ± 0.06	0.07 ± 0.04	5.20 ± 0.49	3.63 ± 0.39	0.27 ± 0.05	1.30 ± 0.10
30	$\lambda = (10, 10^4)$	4.56 ± 0.10	**0.00 ± 0.02**	**4.92 ± 0.45**	3.38 ± 0.38	**0.14 ± 0.04**	1.40 ± 0.09
30	$\lambda = (10^2, 10^3)$	2.50 ± 0.06	0.05 ± 0.03	6.61 ± 0.63	4.77 ± 0.51	0.24 ± 0.05	1.60 ± 0.12
30	rand init	2.39 ± 0.14	8.83 ± 3.09	30.96 ± 1.37	20.35 ± 1.01	6.05 ± 0.43	4.56 ± 0.20
30	rand p	3.40 ± 0.23	48.63 ± 8.46	59.57 ± 1.51	34.03 ± 1.04	16.71 ± 0.59	8.83 ± 0.23
30	$\lambda = 10^2$s	**0.29 ± 0.01**	17.27 ± 0.20	13.39 ± 0.66	8.50 ± 0.53	3.89 ± 0.20	1.00 ± 0.10
30	$\lambda = (10, 10^3)$s	1.54 ± 0.04	10.98 ± 0.19	8.18 ± 0.61	5.26 ± 0.47	2.12 ± 0.16	0.80 ± 0.08
30	$\lambda = 10^2$-	0.34 ± 0.01	104.65 ± 1.09	12.09 ± 0.47	1.94 ± 0.22	10.00 ± 0.32	0.15 ± 0.04
30	$\lambda = (10, 10^3)$-	1.22 ± 0.03	95.39 ± 1.04	8.01 ± 0.38	**1.34 ± 0.17**	6.60 ± 0.28	**0.07 ± 0.03**
30	$\lambda = 10^2$+	1.77 ± 0.02	0.32 ± 0.19	7.10 ± 0.61	5.00 ± 0.49	0.38 ± 0.07	1.72 ± 0.12
30	$\lambda = (10, 10^3)$+	3.25 ± 0.04	0.07 ± 0.04	5.21 ± 0.49	3.64 ± 0.39	0.28 ± 0.05	1.29 ± 0.10
50	$\lambda = 1$	2.73 ± 0.14	4.56 ± 0.93	25.16 ± 1.23	16.19 ± 0.97	4.33 ± 0.25	4.64 ± 0.21
50	$\lambda = 10$	2.00 ± 0.10	0.38 ± 0.09	13.14 ± 0.98	9.16 ± 0.79	0.85 ± 0.10	3.13 ± 0.19
50	$\lambda = 10^2$	2.32 ± 0.09	0.05 ± 0.05	13.51 ± 1.00	9.78 ± 0.82	0.65 ± 0.10	3.08 ± 0.18
50	$\lambda = 10^3$	4.16 ± 0.14	0.03 ± 0.05	16.01 ± 1.18	11.60 ± 0.96	0.83 ± 0.12	3.58 ± 0.18
50	$\lambda = 10^4$	8.40 ± 0.26	0.03 ± 0.05	18.95 ± 1.08	13.83 ± 0.88	0.67 ± 0.10	4.45 ± 0.19
50	$\lambda = (10, 10^2)$	4.36 ± 0.19	-0.01 ± 0.05	9.95 ± 0.79	7.11 ± 0.66	0.60 ± 0.07	2.24 ± 0.14
50	$\lambda = (10, 10^3)$	6.48 ± 0.16	-0.10 ± 0.05	8.92 ± 0.70	6.35 ± 0.57	0.41 ± 0.08	2.16 ± 0.14
50	$\lambda = (10, 10^4)$	8.73 ± 0.20	**-0.16 ± 0.06**	9.21 ± 0.69	6.65 ± 0.60	**0.21 ± 0.05**	2.35 ± 0.14
50	$\lambda = (10^2, 10^3)$	5.20 ± 0.16	-0.01 ± 0.06	11.98 ± 0.96	8.69 ± 0.78	0.49 ± 0.09	2.80 ± 0.17
50	rand init	8.48 ± 0.56	12.03 ± 0.65	72.31 ± 2.55	52.15 ± 2.08	14.26 ± 0.71	5.91 ± 0.27
50	rand p	11.77 ± 0.60	78.54 ± 9.59	112.86 ± 2.38	68.74 ± 1.72	29.64 ± 0.85	14.48 ± 0.30
50	$\lambda = 10^2$s	**1.37 ± 0.06**	48.15 ± 1.02	25.53 ± 1.08	16.85 ± 0.81	7.02 ± 0.32	1.66 ± 0.12
50	$\lambda = (10, 10^3)$s	4.30 ± 0.10	24.66 ± 0.40	15.77 ± 0.71	10.77 ± 0.57	3.80 ± 0.19	1.20 ± 0.10
50	$\lambda = 10^2$-	2.04 ± 0.08	272.99 ± 4.79	20.45 ± 0.69	2.97 ± 0.28	17.31 ± 0.48	0.17 ± 0.04
50	$\lambda = (10, 10^3)$-	4.65 ± 0.11	252.60 ± 4.65	13.53 ± 0.47	**2.19 ± 0.19**	11.25 ± 0.37	**0.09 ± 0.03**
50	$\lambda = 10^2$+	3.13 ± 0.06	0.05 ± 0.05	13.63 ± 1.01	9.87 ± 0.82	0.66 ± 0.10	3.10 ± 0.18
50	$\lambda = (10, 10^3)$+	9.36 ± 0.10	-0.11 ± 0.05	**8.88 ± 0.70**	6.33 ± 0.57	0.39 ± 0.08	2.16 ± 0.15
100	$\lambda = 1$	8.05 ± 0.40	6.36 ± 0.62	54.27 ± 1.87	36.22 ± 1.49	8.69 ± 0.31	9.36 ± 0.28
100	$\lambda = 10$	16.02 ± 0.75	0.50 ± 0.27	29.14 ± 1.44	20.79 ± 1.17	1.91 ± 0.16	6.44 ± 0.26
100	$\lambda = 10^2$	18.20 ± 0.81	-0.82 ± 0.16	31.99 ± 1.66	24.09 ± 1.32	1.30 ± 0.16	6.60 ± 0.30
100	$\lambda = 10^3$	32.59 ± 0.90	-1.05 ± 0.12	37.99 ± 1.74	28.91 ± 1.46	1.25 ± 0.15	7.83 ± 0.29
100	$\lambda = 10^4$	56.89 ± 1.48	-1.04 ± 0.16	43.16 ± 1.72	32.92 ± 1.44	0.98 ± 0.12	9.26 ± 0.29
100	$\lambda = (10, 10^2)$	12.48 ± 0.54	-0.92 ± 0.12	23.48 ± 1.37	17.36 ± 1.14	0.99 ± 0.10	5.13 ± 0.23
100	$\lambda = (10, 10^3)$	26.02 ± 0.76	-1.44 ± 0.09	**19.16 ± 1.10**	14.30 ± 0.89	0.62 ± 0.09	4.24 ± 0.23
100	$\lambda = (10, 10^4)$	62.12 ± 1.69	**-1.52 ± 0.10**	19.66 ± 1.24	14.70 ± 1.03	**0.38 ± 0.09**	4.58 ± 0.23
100	$\lambda = (10^2, 10^3)$	26.69 ± 0.76	-1.20 ± 0.15	27.81 ± 1.49	21.08 ± 1.21	0.98 ± 0.14	5.75 ± 0.27
100	rand init	35.48 ± 1.67	25.53 ± 1.82	171.88 ± 4.79	129.30 ± 3.80	32.63 ± 1.45	9.94 ± 0.41
100	rand p	49.62 ± 1.77	151.17 ± 19.44	247.30 ± 4.09	159.63 ± 3.25	59.22 ± 1.16	28.45 ± 0.41
100	$\lambda = 10^2$s	**6.55 ± 0.30**	100.83 ± 0.84	57.03 ± 1.49	38.63 ± 1.20	14.85 ± 0.38	3.55 ± 0.20
100	$\lambda = (10, 10^3)$s	21.11 ± 0.56	66.03 ± 0.66	35.75 ± 1.17	25.47 ± 0.95	7.96 ± 0.29	2.32 ± 0.15
100	$\lambda = 10^2$-	7.65 ± 0.36	480.18 ± 4.61	45.27 ± 1.00	7.37 ± 0.41	37.48 ± 0.68	0.42 ± 0.06
100	$\lambda = (10, 10^3)$-	19.40 ± 0.51	440.55 ± 4.42	29.21 ± 0.74	**4.52 ± 0.29**	24.53 ± 0.57	**0.16 ± 0.04**
100	$\lambda = 10^2$+	44.8 ± 0.44	-0.84 ± 0.16	31.88 ± 1.66	24.00 ± 1.32	1.29 ± 0.16	6.59 ± 0.30
100	$\lambda = (10, 10^3)$+	57.9 ± 0.50	-1.43 ± 0.10	19.30 ± 1.14	14.42 ± 0.92	0.60 ± 0.09	4.28 ± 0.23

Table 5. Hyperparameter and Ablation Study: results (mean ± standard error over 100 trials) for ER6-Gaussian Cases from DAG-NoCurl, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	$\lambda = 1$	0.33 ± 0.02	2.49 ± 0.27	7.08 ± 0.43	2.57 ± 0.23	3.27 ± 0.22	1.24 ± 0.09
10	$\lambda = 10$	0.37 ± 0.02	0.83 ± 0.14	3.46 ± 0.33	1.18 ± 0.14	1.36 ± 0.16	0.92 ± 0.09
10	$\lambda = 10^2$	0.34 ± 0.02	0.54 ± 0.22	3.54 ± 0.37	1.37 ± 0.18	1.30 ± 0.17	0.87 ± 0.08
10	$\lambda = 10^3$	0.56 ± 0.02	0.48 ± 0.05	5.74 ± 0.46	2.58 ± 0.26	1.64 ± 0.18	1.52 ± 0.11
10	$\lambda = 10^4$	1.16 ± 0.03	0.75 ± 0.08	6.98 ± 0.51	3.15 ± 0.28	2.07 ± 0.20	1.76 ± 0.12
10	$\lambda = (10, 10^2)$	0.62 ± 0.03	0.76 ± 0.21	3.10 ± 0.32	0.99 ± 0.13	1.24 ± 0.16	0.87 ± 0.09
10	$\lambda = (10, 10^3)$	0.75 ± 0.03	**0.36 ± 0.07**	3.07 ± 0.30	0.97 ± 0.13	1.02 ± 0.14	1.08 ± 0.09
10	$\lambda = (10, 10^4)$	1.25 ± 0.04	0.49 ± 0.09	**2.97 ± 0.32**	0.99 ± 0.13	**0.88 ± 0.14**	1.10 ± 0.10
10	$\lambda = (10^2, 10^3)$	0.72 ± 0.03	0.37 ± 0.17	3.29 ± 0.35	1.31 ± 0.18	1.01 ± 0.15	0.97 ± 0.09
10	rand init	0.37 ± 0.03	78.00 ± 40.34	17.09 ± 0.62	5.63 ± 0.26	8.20 ± 0.53	3.25 ± 0.15
10	rand p	0.66 ± 0.06	128.79 ± 47.97	24.82 ± 0.47	6.10 ± 0.22	13.84 ± 0.41	4.88 ± 0.20
10	$\lambda = 10^2$s	0.37 ± 0.03	8.23 ± 0.15	4.79 ± 0.37	1.78 ± 0.18	2.32 ± 0.23	0.69 ± 0.08
10	$\lambda = (10, 10^3)$s	0.92 ± 0.03	5.55 ± 0.32	3.27 ± 0.35	1.10 ± 0.16	1.19 ± 0.17	0.98 ± 0.09
10	$\lambda = 10^2$-	**0.21 ± 0.01**	339.60 ± 10.48	7.28 ± 0.42	0.75 ± 0.11	6.52 ± 0.35	**0.01 ± 0.01**
10	$\lambda = (10, 10^3)$-	0.69 ± 0.03	332.83 ± 10.62	5.19 ± 0.39	**0.53 ± 0.09**	4.64 ± 0.33	0.02 ± 0.01
10	$\lambda = 10^2$+	0.53 ± 0.02	0.54 ± 0.22	3.54 ± 0.37	1.37 ± 0.18	1.30 ± 0.17	0.87 ± 0.08
10	$\lambda = (10, 10^3)$+	1.45 ± 0.04	**0.36 ± 0.07**	3.07 ± 0.30	0.97 ± 0.13	1.02 ± 0.14	1.08 ± 0.09
30	$\lambda = 1$	2.63 ± 0.17	16.64 ± 3.57	46.90 ± 1.76	32.35 ± 1.34	8.46 ± 0.46	6.09 ± 0.22
30	$\lambda = 10$	3.26 ± 0.19	4.22 ± 0.68	27.34 ± 1.91	19.59 ± 1.49	3.53 ± 0.36	4.22 ± 0.20
30	$\lambda = 10^2$	3.34 ± 0.21	1.78 ± 0.38	21.44 ± 1.56	15.98 ± 1.16	2.41 ± 0.33	3.05 ± 0.19
30	$\lambda = 10^3$	2.71 ± 0.13	1.42 ± 0.32	24.23 ± 1.81	18.38 ± 1.38	2.61 ± 0.34	3.24 ± 0.18
30	$\lambda = 10^4$	6.56 ± 0.20	1.25 ± 0.31	30.03 ± 1.73	23.19 ± 1.35	2.78 ± 0.28	4.06 ± 0.20
30	$\lambda = (10, 10^2)$	5.03 ± 0.29	2.09 ± 0.57	19.99 ± 1.50	14.77 ± 1.18	2.21 ± 0.27	3.01 ± 0.19
30	$\lambda = (10, 10^3)$	7.68 ± 0.39	**0.97 ± 0.16**	**17.37 ± 1.18**	12.93 ± 0.92	**1.71 ± 0.19**	2.73 ± 0.16
30	$\lambda = (10, 10^4)$	8.73 ± 0.29	0.98 ± 0.20	17.39 ± 1.44	12.79 ± 1.11	**1.71 ± 0.26**	2.89 ± 0.18
30	$\lambda = (10^2, 10^3)$	3.96 ± 0.21	1.23 ± 0.34	20.19 ± 1.71	15.29 ± 1.32	2.08 ± 0.30	2.82 ± 0.19
30	rand init	6.04 ± 0.26	41.38 ± 21.75	84.88 ± 2.65	59.53 ± 1.94	18.03 ± 0.85	7.32 ± 0.21
30	rand p	7.74 ± 0.38	871.38 ± 198.66	130.37 ± 1.66	70.36 ± 1.26	49.65 ± 1.00	10.36 ± 0.28
30	$\lambda = 10^2$s	**1.58 ± 0.10**	824.91 ± 15.39	37.36 ± 1.34	24.83 ± 0.94	11.04 ± 0.47	1.49 ± 0.14
30	$\lambda = (10, 10^3)$s	4.69 ± 0.23	667.06 ± 17.54	29.88 ± 1.55	20.25 ± 1.12	8.01 ± 0.46	1.62 ± 0.13
30	$\lambda = 10^2$-	1.69 ± 0.11	4908.93 ± 96.30	32.82 ± 1.07	**5.11 ± 0.40**	27.51 ± 0.77	0.20 ± 0.05
30	$\lambda = (10, 10^3)$-	4.67 ± 0.23	4610.52 ± 95.17	26.08 ± 1.07	5.42 ± 0.47	20.51 ± 0.70	**0.15 ± 0.04**
30	$\lambda = 10^2$+	5.32 ± 0.35	1.78 ± 0.38	21.44 ± 1.56	15.98 ± 1.16	2.41 ± 0.33	3.05 ± 0.19
30	$\lambda = (10, 10^3)$+	10.38 ± 0.23	1.02 ± 0.22	17.81 ± 1.29	13.37 ± 1.04	1.74 ± 0.18	2.70 ± 0.16
50	$\lambda = 1$	14.74 ± 0.85	54.63 ± 16.95	95.58 ± 3.36	70.11 ± 2.52	15.54 ± 0.91	9.93 ± 0.28
50	$\lambda = 10$	15.22 ± 0.90	6.71 ± 0.81	55.14 ± 2.88	42.10 ± 2.40	5.57 ± 0.43	7.47 ± 0.23
50	$\lambda = 10^2$	12.05 ± 0.77	2.31 ± 0.41	40.32 ± 2.40	32.10 ± 2.00	2.83 ± 0.24	5.39 ± 0.26
50	$\lambda = 10^3$	10.13 ± 0.56	1.98 ± 0.73	40.61 ± 2.96	32.65 ± 2.34	2.93 ± 0.49	5.03 ± 0.27
50	$\lambda = 10^4$	19.66 ± 0.62	**1.01 ± 0.38**	43.44 ± 2.79	34.93 ± 2.26	2.95 ± 0.44	5.56 ± 0.22
50	$\lambda = (10, 10^2)$	29.23 ± 1.75	3.10 ± 0.50	35.92 ± 2.16	28.07 ± 1.88	2.75 ± 0.23	5.10 ± 0.19
50	$\lambda = (10, 10^3)$	31.74 ± 1.71	1.77 ± 0.38	33.67 ± 2.53	26.69 ± 2.08	2.45 ± 0.35	4.53 ± 0.22
50	$\lambda = (10, 10^4)$	24.14 ± 0.85	1.15 ± 0.27	**32.27 ± 2.44**	25.43 ± 2.02	**2.06 ± 0.34**	4.78 ± 0.22
50	$\lambda = (10^2, 10^3)$	15.50 ± 0.84	2.16 ± 0.63	34.87 ± 2.56	28.06 ± 2.05	2.50 ± 0.38	4.31 ± 0.26
50	rand init	17.37 ± 0.34	47.87 ± 27.87	156.03 ± 5.19	118.60 ± 4.16	26.79 ± 1.15	10.64 ± 0.26
50	rand p	24.40 ± 1.03	1497.50 ± 407.17	255.90 ± 3.06	155.04 ± 2.35	85.71 ± 1.51	15.14 ± 0.33
50	$\lambda = 10^2$s	8.45 ± 0.55	2957.04 ± 69.41	69.24 ± 2.00	48.31 ± 1.57	19.01 ± 0.54	1.92 ± 0.15
50	$\lambda = (10, 10^3)$s	12.51 ± 0.68	2114.01 ± 51.61	56.68 ± 1.95	40.93 ± 1.55	13.62 ± 0.52	2.13 ± 0.14
50	$\lambda = 10^2$-	**5.39 ± 0.34**	16135.01 ± 389.36	56.18 ± 1.46	**8.10 ± 0.58**	47.85 ± 1.05	0.23 ± 0.05
50	$\lambda = (10, 10^3)$-	13.94 ± 0.76	15500.46 ± 378.96	44.66 ± 1.37	8.97 ± 0.57	35.54 ± 0.95	**0.15 ± 0.04**
50	$\lambda = 10^2$+	18.03 ± 0.78	2.31 ± 0.41	40.31 ± 2.39	32.10 ± 2.00	2.82 ± 0.24	5.39 ± 0.26
50	$\lambda = (10, 10^3)$+	44.13 ± 0.81	1.78 ± 0.46	33.41 ± 2.63	26.40 ± 2.15	2.56 ± 0.41	4.45 ± 0.21
100	$\lambda = 1$	52.40 ± 1.68	67.18 ± 6.59	215.70 ± 6.90	165.76 ± 5.56	30.96 ± 1.38	18.98 ± 0.40
100	$\lambda = 10$	62.17 ± 2.19	20.28 ± 4.99	122.05 ± 4.26	96.67 ± 3.62	10.23 ± 0.56	15.15 ± 0.35
100	$\lambda = 10^2$	41.29 ± 1.56	4.30 ± 0.99	89.93 ± 3.49	74.34 ± 3.04	4.64 ± 0.34	10.95 ± 0.33
100	$\lambda = 10^3$	60.88 ± 2.30	1.98 ± 0.86	92.12 ± 4.32	77.68 ± 3.69	4.21 ± 0.48	10.23 ± 0.32
100	$\lambda = 10^4$	94.68 ± 1.99	**-0.03 ± 0.22**	92.36 ± 3.36	78.60 ± 2.97	3.28 ± 0.30	10.48 ± 0.27
100	$\lambda = (10, 10^2)$	152.84 ± 6.00	4.68 ± 0.77	84.02 ± 3.58	68.57 ± 3.10	4.91 ± 0.39	10.54 ± 0.33
100	$\lambda = (10, 10^3)$	84.24 ± 3.26	2.61 ± 0.93	72.30 ± 3.80	60.12 ± 3.49	3.58 ± 0.30	8.60 ± 0.27
100	$\lambda = (10, 10^4)$	135.50 ± 3.39	0.88 ± 0.32	**66.13 ± 2.85**	54.66 ± 2.53	**2.92 ± 0.27**	8.55 ± 0.26
100	$\lambda = (10^2, 10^3)$	146.64 ± 5.87	2.64 ± 1.08	85.73 ± 5.71	72.07 ± 4.72	4.66 ± 0.83	9.00 ± 0.33
100	rand init	51.32 ± 0.31	49.17 ± 6.88	445.43 ± 16.00	367.63 ± 14.03	61.00 ± 2.32	16.80 ± 0.37
100	rand p	67.64 ± 2.14	3895.88 ± 1085.90	608.89 ± 6.27	409.29 ± 5.61	173.17 ± 2.24	26.43 ± 0.47
100	$\lambda = 10^2$s	43.98 ± 2.11	6319.57 ± 103.73	138.70 ± 2.61	97.83 ± 2.14	37.22 ± 0.69	3.65 ± 0.16
100	$\lambda = (10, 10^3)$s	106.88 ± 4.64	5285.84 ± 124.45	108.05 ± 2.74	78.33 ± 2.27	26.03 ± 0.60	3.69 ± 0.19
100	$\lambda = 10^2$-	**32.56 ± 1.53**	31457.27 ± 604.87	110.61 ± 1.75	14.13 ± 0.66	96.12 ± 1.36	0.36 ± 0.07
100	$\lambda = (10, 10^3)$-	121.89 ± 5.33	30154.20 ± 608.15	83.21 ± 1.61	**13.67 ± 0.75**	69.26 ± 1.12	**0.28 ± 0.06**
100	$\lambda = 10^2$+	84.24 ± 3.38	4.04 ± 0.93	89.64 ± 3.47	74.27 ± 3.04	4.41 ± 0.34	10.96 ± 0.31
100	$\lambda = (10, 10^3)$+	107.87 ± 3.07	2.60 ± 0.93	72.35 ± 3.81	60.17 ± 3.49	3.58 ± 0.30	8.60 ± 0.27

Table 6. Comparison of Different Algorithms on Linear Synthetic Datasets: results (mean ± standard error over 100 trials) for ER3-Gaussian Cases, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	NOTEARS	1.71 ± 0.07	**0.03 ± 0.01**	1.11 ± 0.21	0.55 ± 0.14	0.15 ± 0.05	0.41 ± 0.06
10	FGS	0.65 ± 0.07	–	6.34 ± 0.55	2.85 ± 0.37	0.98 ± 0.13	2.51 ± 0.18
10	CAM	8.46 ± 0.16	–	12.34 ± 0.61	5.05 ± 0.34	1.77 ± 0.17	5.52 ± 0.23
10	MMPC	0.89 ± 0.03	–	15.36 ± 0.36	0.68 ± 0.09	3.78 ± 0.30	10.90 ± 0.15
10	Eq+BU	0.57 ± 0.01	–	2.92 ± 0.21	2.91 ± 0.21	**0.01 ± 0.01**	**0.00 ± 0.00**
10	Eq+TD	0.58 ± 0.02	–	3.21 ± 0.23	3.20 ± 0.23	**0.01 ± 0.01**	**0.00 ± 0.00**
10	NoCurl-1s	**0.09 ± 0.00**	1.45 ± 0.02	3.11 ± 0.31	1.60 ± 0.18	0.95 ± 0.11	0.56 ± 0.08
10	NoCurl-2s	0.43 ± 0.01	0.53 ± 0.02	1.27 ± 0.20	0.66 ± 0.14	0.20 ± 0.05	0.41 ± 0.06
10	NoCurl-1	0.11 ± 0.00	0.09 ± 0.02	2.18 ± 0.28	1.24 ± 0.18	0.26 ± 0.06	0.68 ± 0.08
10	NoCurl-2	0.47 ± 0.01	0.06 ± 0.02	**1.08 ± 0.18**	**0.54 ± 0.12**	0.09 ± 0.03	0.45 ± 0.06
30	NOTEARS	37.25 ± 1.67	**-0.06 ± 0.02**	**4.42 ± 0.48**	2.85 ± 0.36	0.47 ± 0.11	1.10 ± 0.10
30	FGS	0.96 ± 0.04	–	15.16 ± 1.33	8.53 ± 1.05	1.98 ± 0.23	4.65 ± 0.24
30	CAM	45.87 ± 0.94	–	36.27 ± 1.17	18.23 ± 0.70	4.34 ± 0.31	13.70 ± 0.37
30	MMPC	1.74 ± 0.05	–	46.67 ± 0.68	**2.62 ± 0.18**	11.72 ± 0.60	32.33 ± 0.34
30	Eq+BU	2.12 ± 0.01	–	14.14 ± 0.75	14.12 ± 0.75	**0.02 ± 0.01**	**0.00 ± 0.00**
30	Eq+TD	2.07 ± 0.01	–	15.45 ± 0.82	15.43 ± 0.81	**0.02 ± 0.01**	**0.00 ± 0.00**
30	NoCurl-1s	**0.29 ± 0.01**	17.27 ± 0.20	13.39 ± 0.66	8.50 ± 0.53	3.89 ± 0.20	1.00 ± 0.10
30	NoCurl-2s	1.54 ± 0.04	10.98 ± 0.19	8.18 ± 0.61	5.26 ± 0.47	2.12 ± 0.16	0.80 ± 0.08
30	NoCurl-1	1.19 ± 0.04	0.33 ± 0.19	7.18 ± 0.61	5.05 ± 0.49	0.40 ± 0.07	1.73 ± 0.12
30	NoCurl-2	2.38 ± 0.06	0.07 ± 0.04	5.20 ± 0.49	3.63 ± 0.39	0.27 ± 0.05	1.30 ± 0.10
50	NOTEARS	253.96 ± 9.49	**-0.25 ± 0.04**	**8.39 ± 0.70**	5.56 ± 0.53	1.24 ± 0.18	1.59 ± 0.13
50	FGS	1.42 ± 0.06	–	26.90 ± 2.14	16.21 ± 1.85	3.48 ± 0.30	7.21 ± 0.26
50	CAM	75.11 ± 0.73	–	59.03 ± 1.58	29.31 ± 0.97	7.71 ± 0.39	22.01 ± 0.47
50	MMPC	3.88 ± 0.11	–	78.82 ± 0.86	**4.32 ± 0.23**	19.39 ± 0.68	55.11 ± 0.50
50	Eq+BU	4.59 ± 0.05	–	27.06 ± 1.12	26.99 ± 1.12	0.07 ± 0.04	**0.00 ± 0.00**
50	Eq+TD	4.29 ± 0.05	–	29.39 ± 1.24	29.33 ± 1.23	**0.06 ± 0.04**	**0.00 ± 0.00**
50	NoCurl-1s	**1.37 ± 0.06**	48.15 ± 1.02	25.53 ± 1.08	16.85 ± 0.81	7.02 ± 0.32	1.66 ± 0.12
50	NoCurl-2s	4.30 ± 0.10	24.66 ± 0.40	15.77 ± 0.71	10.77 ± 0.57	3.80 ± 0.19	1.20 ± 0.10
50	NoCurl-1	2.32 ± 0.09	0.05 ± 0.05	13.51 ± 1.00	9.78 ± 0.82	0.65 ± 0.10	3.08 ± 0.18
50	NoCurl-2	6.48 ± 0.16	-0.10 ± 0.05	8.92 ± 0.70	6.35 ± 0.57	0.41 ± 0.08	2.16 ± 0.14
100	NOTEARS	659.35 ± 10.91	**-1.65 ± 0.08**	22.26 ± 1.58	16.28 ± 1.22	3.77 ± 0.35	2.21 ± 0.14
100	FGS	**2.36 ± 0.09**	–	34.12 ± 2.04	16.16 ± 1.69	5.54 ± 0.38	12.42 ± 0.37
100	CAM	197.76 ± 1.89	–	104.99 ± 2.00	51.07 ± 1.24	12.21 ± 0.55	41.71 ± 0.68
100	MMPC	6.38 ± 0.16	–	159.40 ± 1.19	**10.00 ± 0.38**	32.39 ± 1.19	117.01 ± 0.84
100	Eq+BU	15.31 ± 0.14	–	52.94 ± 2.28	52.84 ± 2.27	**0.10 ± 0.05**	**0.00 ± 0.00**
100	Eq+TD	13.02 ± 0.10	–	58.34 ± 2.51	58.24 ± 2.50	**0.10 ± 0.05**	**0.00 ± 0.00**
100	NoCurl-1s	6.55 ± 0.30	100.83 ± 0.84	57.03 ± 1.49	38.63 ± 1.20	14.85 ± 0.38	3.55 ± 0.20
100	NoCurl-2s	21.11 ± 0.56	66.03 ± 0.66	35.75 ± 1.17	25.47 ± 0.95	7.96 ± 0.29	2.32 ± 0.15
100	NoCurl-1	18.20 ± 0.81	-0.82 ± 0.16	31.99 ± 1.66	24.09 ± 1.32	1.30 ± 0.16	6.60 ± 0.30
100	NoCurl-2	26.02 ± 0.76	-1.44 ± 0.09	**19.16 ± 1.10**	14.30 ± 0.89	0.62 ± 0.09	4.24 ± 0.23

Table 7. Comparison of Different Algorithms on Linear Synthetic Datasets: results (mean ± standard error over 100 trials) for ER4-Gaussian Cases, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	NOTEARS	3.35 ± 0.13	**0.08 ± 0.02**	**1.88 ± 0.26**	**0.89 ± 0.15**	0.35 ± 0.07	0.64 ± 0.08
10	FGS	0.80 ± 0.08	–	13.14 ± 0.69	6.88 ± 0.43	2.56 ± 0.20	3.70 ± 0.24
10	CAM	12.10 ± 0.17	–	19.06 ± 0.64	7.51 ± 0.31	4.35 ± 0.27	7.20 ± 0.28
10	MMPC	1.14 ± 0.04	–	21.13 ± 0.39	1.45 ± 0.12	9.16 ± 0.41	10.52 ± 0.19
10	Eq+BU	0.55 ± 0.01	–	4.73 ± 0.24	4.58 ± 0.23	0.15 ± 0.06	**0.00 ± 0.00**
10	Eq+TD	0.55 ± 0.01	–	4.81 ± 0.25	4.67 ± 0.23	**0.14 ± 0.05**	**0.00 ± 0.00**
10	NoCurl-1s	**0.10 ± 0.00**	4.44 ± 0.09	4.01 ± 0.36	2.02 ± 0.21	1.50 ± 0.16	0.49 ± 0.07
10	NoCurl-2s	0.56 ± 0.02	2.03 ± 0.07	2.39 ± 0.29	1.11 ± 0.16	0.57 ± 0.14	0.71 ± 0.08
10	NoCurl-1	0.25 ± 0.01	0.14 ± 0.03	2.51 ± 0.32	1.39 ± 0.19	0.45 ± 0.09	0.67 ± 0.08
10	NoCurl-2	0.43 ± 0.01	0.13 ± 0.03	2.22 ± 0.27	1.12 ± 0.17	0.33 ± 0.06	0.77 ± 0.08
30	NOTEARS	94.21 ± 5.25	**0.25 ± 0.11**	8.81 ± 1.08	6.11 ± 0.78	1.50 ± 0.28	1.20 ± 0.11
30	FGS	1.71 ± 0.10	–	50.37 ± 3.27	37.87 ± 2.75	5.50 ± 0.42	7.00 ± 0.31
30	CAM	61.92 ± 0.78	–	56.80 ± 1.69	29.98 ± 0.95	11.96 ± 0.61	14.86 ± 0.39
30	MMPC	1.58 ± 0.05	–	63.70 ± 0.80	**4.19 ± 0.21**	27.61 ± 1.00	31.90 ± 0.47
30	Eq+BU	2.42 ± 0.04	–	31.06 ± 1.33	30.79 ± 1.31	**0.27 ± 0.08**	**0.00 ± 0.00**
30	Eq+TD	2.36 ± 0.04	–	33.48 ± 1.46	33.16 ± 1.43	0.32 ± 0.08	**0.00 ± 0.00**
30	NoCurl-1s	**0.59 ± 0.03**	58.05 ± 1.08	20.27 ± 0.81	13.17 ± 0.61	6.09 ± 0.27	1.01 ± 0.09
30	NoCurl-2s	1.92 ± 0.06	37.25 ± 0.68	13.16 ± 0.87	8.67 ± 0.65	3.75 ± 0.25	0.74 ± 0.08
30	NoCurl-1	1.18 ± 0.06	0.31 ± 0.05	10.84 ± 0.72	7.97 ± 0.56	0.75 ± 0.09	2.12 ± 0.13
30	NoCurl-2	3.39 ± 0.11	0.40 ± 0.19	**7.91 ± 0.83**	5.69 ± 0.68	0.71 ± 0.12	1.51 ± 0.12
50	NOTEARS	209.49 ± 6.13	**0.19 ± 0.09**	19.98 ± 1.46	14.73 ± 1.12	3.45 ± 0.37	1.80 ± 0.14
50	FGS	3.41 ± 0.20	–	71.11 ± 4.15	54.36 ± 3.63	7.81 ± 0.44	8.94 ± 0.38
50	CAM	117.45 ± 1.92	–	91.13 ± 2.01	47.89 ± 1.23	18.91 ± 0.73	24.33 ± 0.49
50	MMPC	3.84 ± 0.15	–	106.73 ± 1.07	**6.07 ± 0.28**	44.99 ± 1.20	55.67 ± 0.58
50	Eq+BU	4.25 ± 0.03	–	64.18 ± 2.29	63.68 ± 2.26	**0.50 ± 0.12**	**0.00 ± 0.00**
50	Eq+TD	3.98 ± 0.03	–	69.71 ± 2.40	69.21 ± 2.37	**0.50 ± 0.12**	**0.00 ± 0.00**
50	NoCurl-1s	**3.33 ± 0.17**	164.65 ± 2.42	37.82 ± 1.19	25.68 ± 0.94	10.38 ± 0.36	1.76 ± 0.14
50	NoCurl-2s	5.80 ± 0.19	110.84 ± 1.86	26.26 ± 1.13	18.35 ± 0.89	6.55 ± 0.29	1.36 ± 0.12
50	NoCurl-1	4.13 ± 0.20	0.40 ± 0.13	19.76 ± 1.22	15.01 ± 1.01	1.02 ± 0.12	3.73 ± 0.19
50	NoCurl-2	7.55 ± 0.25	0.42 ± 0.21	**15.24 ± 1.27**	11.66 ± 1.04	0.81 ± 0.13	2.77 ± 0.20
100	NOTEARS	1265.47 ± 15.70	**-0.64 ± 0.14**	49.07 ± 2.55	37.86 ± 2.07	8.27 ± 0.49	2.94 ± 0.18
100	FGS	**10.17 ± 0.65**	–	93.24 ± 5.63	66.74 ± 4.67	12.98 ± 0.83	13.52 ± 0.46
100	CAM	258.39 ± 1.83	–	159.91 ± 3.10	81.10 ± 1.83	34.55 ± 1.23	44.26 ± 0.72
100	MMPC	14.10 ± 0.56	–	213.12 ± 1.49	**12.00 ± 0.39**	83.00 ± 1.84	118.12 ± 1.14
100	Eq+BU	15.21 ± 0.19	–	138.38 ± 5.52	137.61 ± 5.47	**0.77 ± 0.14**	**0.00 ± 0.00**
100	Eq+TD	12.69 ± 0.15	–	150.08 ± 6.03	149.31 ± 5.97	**0.77 ± 0.13**	**0.00 ± 0.00**
100	NoCurl-1s	12.90 ± 0.69	492.97 ± 7.35	82.46 ± 1.45	58.14 ± 1.25	21.13 ± 0.43	3.19 ± 0.17
100	NoCurl-2s	30.94 ± 1.13	348.21 ± 5.30	60.64 ± 1.79	43.99 ± 1.51	13.80 ± 0.42	2.85 ± 0.18
100	NoCurl-1	26.43 ± 1.46	-0.28 ± 0.26	44.43 ± 1.80	34.83 ± 1.50	1.74 ± 0.17	7.86 ± 0.28
100	NoCurl-2	43.99 ± 1.77	-0.32 ± 0.28	**37.11 ± 1.71**	29.28 ± 1.48	1.57 ± 0.16	6.26 ± 0.23

Table 8. Comparison of Different Algorithms on Linear Synthetic Datasets: results (mean ± standard error over 100 trials) for ER6-Gaussian Cases, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	NOTEARS	3.20 ± 0.20	**0.22 ± 0.04**	3.21 ± 0.31	1.11 ± 0.14	**1.00 ± 0.14**	1.10 ± 0.09
10	FGS	0.58 ± 0.02	–	19.77 ± 0.58	8.19 ± 0.28	5.42 ± 0.21	6.16 ± 0.35
10	CAM	10.46 ± 0.17	–	26.41 ± 0.46	6.07 ± 0.24	11.17 ± 0.30	9.17 ± 0.31
10	MMPC	1.14 ± 0.04	–	21.13 ± 0.39	1.45 ± 0.12	9.16 ± 0.41	10.52 ± 0.19
10	Eq+BU	0.56 ± 0.01	–	6.18 ± 0.31	4.79 ± 0.21	1.39 ± 0.22	**0.00 ± 0.00**
10	Eq+TD	0.57 ± 0.01	–	6.22 ± 0.30	4.85 ± 0.20	1.37 ± 0.23	**0.00 ± 0.00**
10	NoCurl-1s	0.37 ± 0.03	8.23 ± 0.15	4.79 ± 0.37	1.78 ± 0.18	2.32 ± 0.23	0.69 ± 0.08
10	NoCurl-2s	0.92 ± 0.03	5.55 ± 0.32	3.27 ± 0.35	1.10 ± 0.16	1.19 ± 0.17	0.98 ± 0.09
10	NoCurl-1	**0.34 ± 0.02**	0.54 ± 0.22	3.54 ± 0.37	1.37 ± 0.18	1.30 ± 0.17	0.87 ± 0.08
10	NoCurl-2	0.75 ± 0.03	0.36 ± 0.07	**3.07 ± 0.30**	**0.97 ± 0.13**	1.02 ± 0.14	1.08 ± 0.09
30	NOTEARS	102.46 ± 4.68	1.02 ± 0.18	20.85 ± 2.09	15.15 ± 1.58	3.75 ± 0.49	1.95 ± 0.14
30	FGS	5.44 ± 0.21	–	132.42 ± 3.71	105.01 ± 3.06	14.64 ± 0.52	12.77 ± 0.53
30	CAM	64.53 ± 0.75	–	105.49 ± 1.86	50.80 ± 1.08	35.69 ± 0.90	19.00 ± 0.45
30	MMPC	**1.58 ± 0.05**	–	63.70 ± 0.80	**4.19 ± 0.21**	27.61 ± 1.00	31.90 ± 0.47
30	Eq+BU	2.12 ± 0.03	–	68.38 ± 1.66	63.70 ± 1.40	4.68 ± 0.57	**0.00 ± 0.00**
30	Eq+TD	2.08 ± 0.02	–	70.82 ± 1.67	66.04 ± 1.40	4.78 ± 0.57	**0.00 ± 0.00**
30	NoCurl-1s	**1.58 ± 0.10**	824.91 ± 15.39	37.36 ± 1.34	24.83 ± 0.94	11.04 ± 0.47	1.49 ± 0.14
30	NoCurl-2s	4.69 ± 0.23	667.06 ± 17.54	29.88 ± 1.55	20.25 ± 1.12	8.01 ± 0.46	1.62 ± 0.13
30	NoCurl-1	3.34 ± 0.21	1.78 ± 0.38	21.44 ± 1.56	15.98 ± 1.16	2.41 ± 0.33	3.05 ± 0.19
30	NoCurl-2	7.68 ± 0.39	**0.97 ± 0.16**	**17.37 ± 1.18**	12.93 ± 0.92	**1.71 ± 0.19**	2.73 ± 0.16
50	NOTEARS	340.03 ± 6.99	1.97 ± 0.26	52.40 ± 3.24	40.53 ± 2.62	9.25 ± 0.67	2.62 ± 0.16
50	FGS	20.31 ± 1.00	–	235.85 ± 7.94	195.75 ± 6.77	23.79 ± 1.00	16.31 ± 0.56
50	CAM	129.50 ± 1.18	–	176.25 ± 2.94	89.77 ± 1.73	59.87 ± 1.27	26.61 ± 0.53
50	MMPC	**3.84 ± 0.15**	–	106.73 ± 1.07	**6.07 ± 0.28**	44.99 ± 1.20	55.67 ± 0.58
50	Eq+BU	3.97 ± 0.03	–	153.43 ± 3.76	145.16 ± 3.27	8.27 ± 0.93	**0.00 ± 0.00**
50	Eq+TD	**3.84 ± 0.02**	–	161.11 ± 4.00	152.68 ± 3.52	8.43 ± 0.97	**0.00 ± 0.00**
50	NoCurl-1s	8.45 ± 0.55	2957.04 ± 69.41	69.24 ± 2.00	48.31 ± 1.57	19.01 ± 0.54	1.92 ± 0.15
50	NoCurl-2s	12.51 ± 0.68	2114.01 ± 51.61	56.68 ± 1.95	40.93 ± 1.55	13.62 ± 0.52	2.13 ± 0.14
50	NoCurl-1	12.05 ± 0.77	2.31 ± 0.41	40.32 ± 2.40	32.10 ± 2.00	2.83 ± 0.24	5.39 ± 0.26
50	NoCurl-2	31.74 ± 1.71	**1.77 ± 0.38**	**33.67 ± 2.53**	26.69 ± 2.08	**2.45 ± 0.35**	4.53 ± 0.22
100	NOTEARS	2146.90 ± 31.22	**2.49 ± 0.37**	116.52 ± 4.39	92.10 ± 3.66	20.54 ± 0.84	3.88 ± 0.21
100	FGS	105.57 ± 5.35	–	421.53 ± 15.01	356.15 ± 13.33	43.64 ± 1.59	21.74 ± 0.62
100	CAM	290.21 ± 3.76	–	310.54 ± 4.54	156.83 ± 2.66	110.55 ± 2.03	43.16 ± 0.69
100	MMPC	14.10 ± 0.56	–	213.12 ± 1.49	**12.00 ± 0.39**	83.00 ± 1.84	118.12 ± 1.14
100	Eq+BU	13.15 ± 0.15	–	378.33 ± 8.50	365.13 ± 7.81	13.20 ± 1.20	**0.00 ± 0.00**
100	Eq+TD	**11.37 ± 0.11**	–	397.65 ± 8.66	383.96 ± 7.92	13.69 ± 1.20	**0.00 ± 0.00**
100	NoCurl-1s	43.98 ± 2.11	6319.57 ± 103.73	138.70 ± 2.61	97.83 ± 2.14	37.22 ± 0.69	3.65 ± 0.16
100	NoCurl-2s	106.88 ± 4.64	5285.84 ± 124.45	108.05 ± 2.74	78.33 ± 2.27	26.03 ± 0.60	3.69 ± 0.19
100	NoCurl-1	41.29 ± 1.56	4.30 ± 0.99	89.93 ± 3.49	74.34 ± 3.04	4.64 ± 0.34	10.95 ± 0.33
100	NoCurl-2	84.24 ± 3.26	2.61 ± 0.93	**72.30 ± 3.80**	60.12 ± 3.49	**3.58 ± 0.30**	8.60 ± 0.27

Table 9. Comparison of Different Algorithms on Linear Synthetic Datasets: results (mean ± standard error over 100 trials) for SF4-Gumbel Cases, where bold numbers highlight the best method for each case.

d	Method	Time	ΔF	SHD	#Extra E	#Missing E	#Reverse E
10	NOTEAR	5.26 ± 0.17	**-0.71 ± 0.01**	1.10 ± 0.22	0.80 ± 0.15	0.12 ± 0.05	0.18 ± 0.04
10	FGS	0.47 ± 0.02	–	5.30 ± 0.57	3.13 ± 0.44	0.99 ± 0.12	1.18 ± 0.11
10	CAM	11.76 ± 0.20	–	17.70 ± 0.73	9.22 ± 0.49	1.67 ± 0.13	6.81 ± 0.27
10	MMPC	0.52 ± 0.02	–	14.93 ± 0.18	0.88 ± 0.11	3.06 ± 0.17	10.99 ± 0.13
10	Eq+BU	0.67 ± 0.01	–	1.24 ± 0.13	1.24 ± 0.13	**0.00 ± 0.00**	**0.00 ± 0.00**
10	Eq+TD	0.55 ± 0.02	–	1.28 ± 0.13	1.27 ± 0.13	0.01 ± 0.01	**0.00 ± 0.00**
10	NoCurl-1s	**0.06 ± 0.00**	0.09 ± 0.01	0.94 ± 0.17	**0.67 ± 0.13**	0.16 ± 0.04	0.11 ± 0.03
10	NoCurl-2s	0.18 ± 0.00	-0.11 ± 0.00	0.97 ± 0.17	0.73 ± 0.13	0.03 ± 0.02	0.21 ± 0.05
10	NoCurl-1	0.14 ± 0.00	-0.58 ± 0.01	**0.93 ± 0.20**	0.69 ± 0.15	0.08 ± 0.04	0.16 ± 0.04
10	NoCurl-2	0.35 ± 0.01	-0.59 ± 0.01	1.08 ± 0.22	0.86 ± 0.19	0.04 ± 0.02	0.18 ± 0.04
30	NOTEARS	82.37 ± 1.57	**-3.55 ± 0.02**	2.68 ± 0.51	2.13 ± 0.43	0.11 ± 0.05	0.44 ± 0.07
30	FGS	1.01 ± 0.03	–	22.81 ± 1.70	12.18 ± 1.38	7.68 ± 0.46	2.95 ± 0.19
30	CAM	62.27 ± 0.91	–	62.80 ± 1.29	28.46 ± 0.96	13.71 ± 0.32	20.63 ± 0.42
30	MMPC	13.58 ± 3.40	–	54.24 ± 0.39	4.14 ± 0.23	18.61 ± 0.42	31.49 ± 0.34
30	Eq+BU	2.74 ± 0.05	–	9.46 ± 0.49	9.42 ± 0.49	**0.04 ± 0.02**	**0.00 ± 0.00**
30	Eq+TD	3.00 ± 0.03	–	9.91 ± 0.52	9.86 ± 0.52	0.05 ± 0.02	**0.00 ± 0.00**
30	NoCurl-1s	**0.29 ± 0.01**	5.57 ± 0.08	5.76 ± 0.59	4.85 ± 0.52	0.67 ± 0.10	0.24 ± 0.05
30	NoCurl-2s	1.13 ± 0.02	2.28 ± 0.07	5.26 ± 0.75	4.35 ± 0.65	0.37 ± 0.10	0.54 ± 0.07
30	NoCurl-1	0.76 ± 0.02	-3.30 ± 0.04	**2.57 ± 0.43**	**2.04 ± 0.37**	0.12 ± 0.04	0.41 ± 0.05
30	NoCurl-2	1.84 ± 0.05	-3.31 ± 0.02	4.42 ± 0.70	3.60 ± 0.62	0.09 ± 0.03	0.73 ± 0.09
50	NOTEARS	150.33 ± 1.98	**-7.08 ± 0.02**	**3.94 ± 0.77**	3.22 ± 0.70	0.18 ± 0.07	0.54 ± 0.07
50	FGS	2.35 ± 0.10	–	43.47 ± 2.77	19.25 ± 2.12	19.00 ± 0.95	5.22 ± 0.29
50	CAM	110.65 ± 1.42	–	103.32 ± 1.55	39.74 ± 1.10	30.54 ± 0.69	33.04 ± 0.52
50	MMPC	417.94 ± 349.20	–	96.70 ± 0.56	8.91 ± 0.38	38.39 ± 0.83	49.40 ± 0.69
50	Eq+BU	5.49 ± 0.05	–	23.64 ± 1.03	23.30 ± 1.02	0.33 ± 0.12	0.01 ± 0.01
50	Eq+TD	5.75 ± 0.04	–	24.52 ± 1.08	24.18 ± 1.07	0.34 ± 0.12	**0.00 ± 0.00**
50	NoCurl-1s	**0.99 ± 0.05**	24.11 ± 0.37	14.98 ± 1.13	12.82 ± 1.01	1.72 ± 0.15	0.44 ± 0.08
50	NoCurl-2s	5.94 ± 0.15	10.42 ± 0.21	13.73 ± 1.36	11.78 ± 1.24	0.56 ± 0.08	1.39 ± 0.12
50	NoCurl-1	3.45 ± 0.13	-6.74 ± 0.03	4.06 ± 0.64	**3.16 ± 0.56**	**0.14 ± 0.03**	0.76 ± 0.09
50	NoCurl-2	5.64 ± 0.14	-6.74 ± 0.03	8.38 ± 1.17	7.05 ± 1.08	0.18 ± 0.05	1.15 ± 0.10
100	NOTEARS	1113.10 ± 9.71	**-17.53 ± 0.05**	11.98 ± 2.18	10.40 ± 2.04	0.43 ± 0.11	1.15 ± 0.12
100	FGS	8.04 ± 0.54	–	91.32 ± 3.48	30.09 ± 2.59	52.39 ± 1.54	8.84 ± 0.34
100	CAM	240.04 ± 2.91	–	211.33 ± 2.25	74.66 ± 1.60	76.12 ± 0.90	60.55 ± 0.82
100	MMPC	40.22 ± 14.80	–	217.00 ± 0.82	32.41 ± 0.73	88.73 ± 1.12	95.86 ± 1.04
100	Eq+BU	21.50 ± 0.29	–	62.96 ± 2.20	61.33 ± 2.27	1.62 ± 0.30	0.01 ± 0.01
100	Eq+TD	17.46 ± 0.10	–	65.60 ± 2.27	63.98 ± 2.34	1.62 ± 0.30	**0.00 ± 0.00**
100	NoCurl-1s	**6.87 ± 0.33**	97.63 ± 1.31	38.66 ± 2.28	35.02 ± 2.09	3.03 ± 0.25	0.61 ± 0.09
100	NoCurl-2s	23.98 ± 0.87	55.31 ± 1.21	30.14 ± 2.42	26.66 ± 2.23	1.78 ± 0.20	1.70 ± 0.15
100	NoCurl-1	27.64 ± 0.82	-17.29 ± 0.05	**8.68 ± 1.09**	**7.06 ± 1.01**	0.19 ± 0.04	1.43 ± 0.13
100	NoCurl-2	49.83 ± 1.20	-17.19 ± 0.06	16.84 ± 1.66	14.78 ± 1.58	**0.17 ± 0.05**	1.89 ± 0.14

Table 10. Comparison of Different Algorithms on Nonlinear Synthetic datasets: results (mean ± standard error over 5 trails) on SHD and Run Time (in seconds), where bold numbers highlight the best method for each case.

Nonlinear Case 1: SHD							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	$\mathbf{3.0 \pm 1.2}$	3.2 ± 1.7	4.6 ± 0.6	21.4 ± 0.4	9.0 ± 3.9	7.4 ± 2.8	7.4 ± 3.2
20	$\mathbf{3.8 \pm 2.3}$	5.0 ± 1.8	12.6 ± 1.5	46.4 ± 0.7	12.6 ± 3.8	8.8 ± 3.1	8.6 ± 3.5
50	29.0 ± 6.5	$\mathbf{9.6 \pm 1.6}$	12.0 ± 1.4	110.6 ± 2.7	28.0 ± 3.7	26.4 ± 10.5	23.2 ± 9.8
100	$> 72h$	$> 72h$	$\mathbf{34.2 \pm 4.0}$	251.2 ± 6.1	$> 72h$	58.6 ± 15.9	54.0 ± 13.8
Nonlinear Case 2: SHD							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	$\mathbf{0.4 \pm 0.4}$	4.2 ± 2.4	7.0 ± 0.5	15.8 ± 0.3	7.8 ± 4.7	2.0 ± 1.5	5.6 ± 3.1
20	2.2 ± 1.0	8.0 ± 3.1	24.6 ± 1.0	37.0 ± 0.3	21.2 ± 11.8	5.0 ± 2.3	$\mathbf{2.0 \pm 0.8}$
50	20.8 ± 4.7	17.6 ± 5.8	41.4 ± 1.2	83.6 ± 0.4	54.6 ± 10.2	12.4 ± 4.6	$\mathbf{9.0 \pm 3.4}$
100	$> 72h$	26.0 ± 6.7	78.0 ± 2.4	179.0 ± 1.5	$> 72h$	21.4 ± 2.3	$\mathbf{18.6 \pm 3.0}$
Nonlinear Case 3: SHD							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	2.6 ± 1.2	8.8 ± 2.9	0.2 ± 0.0	15.8 ± 0.1	4.2 ± 0.8	$\mathbf{2.4 \pm 0.8}$	3.0 ± 0.6
20	$\mathbf{6.0 \pm 2.5}$	36.0 ± 11.1	13.6 ± 1.0	37.2 ± 0.3	18.8 ± 4.7	9.0 ± 1.4	10.4 ± 1.6
50	$\mathbf{14.8 \pm 0.4}$	60.8 ± 8.0	55.2 ± 2.1	$> 72h$	57.4 ± 5.2	27.6 ± 3.7	26.6 ± 1.8
100	$\mathbf{45.4 \pm 2.7}$	$> 72h$	73.0 ± 0.5	$> 72h$	$> 72h$	62.6 ± 6.6	60.8 ± 3.8
Nonlinear Case 1: Run Time							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	$2.3e3 \pm 4.5e2$	$7.2e2 \pm 5.2e1$	$3.9e1 \pm 0.8$	$\mathbf{1.5 \pm 0.1}$	$6.3e2 \pm 6.5e1$	$6.3e1 \pm 1.4e2$	$3.8e2 \pm 3.3e1$
20	$9.1e3 \pm 15.3e4$	$1.5e3 \pm 7.6e1$	$8.9e1 \pm 1.2$	$\mathbf{2.1 \pm 0.1}$	$1.9e3 \pm 3.2e2$	$1.1e3 \pm 2.0e2$	$4.9e2 \pm 2.9e1$
50	$6.0e4 \pm 3.0e4$	$5.3e3 \pm 4.2e2$	$2.5e2 \pm 1.9$	$\mathbf{1.3e1 \pm 1.4}$	$1.7e4 \pm 1.1e3$	$2.2e3 \pm 2.1e2$	$9.1e2 \pm 8.5e1$
100	$> 72h$	$> 72h$	$6.1e2 \pm 4.7$	$\mathbf{1.0e2 \pm 7.6}$	$> 72h$	$4.7e3 \pm 4.9e2$	$1.6e3 \pm 7.4e1$
Nonlinear Case 2: Run Time							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	$3.2e3 \pm 9.4e2$	$5.9e2 \pm 7.8e1$	$3.5e1 \pm 1.2$	$\mathbf{0.7 \pm 0.0}$	$6.7e2 \pm 3.8e1$	$9.5e2 \pm 5.4e1$	$3.1e2 \pm 1.3e1$
20	$2.0e4 \pm 1.5e3$	$1.3e3 \pm 5.3e2$	$1.0e2 \pm 2.3$	$\mathbf{1.1 \pm 0.0}$	$1.9e3 \pm 1.9e2$	$1.2e3 \pm 7.2e1$	$3.9e2 \pm 1.3e1$
50	$1.8e5 \pm 5.4e4$	$5.3e3 \pm 1.5e3$	$2.9e2 \pm 4.6$	$\mathbf{4.8 \pm 0.2}$	$1.2e4 \pm 1.4e3$	$2.9e3 \pm 2.6e2$	$8.5e2 \pm 4.1e1$
100	$> 72h$	$1.6e4 \pm 7.2e2$	$6.1e2 \pm 2.7$	$\mathbf{1.1e1 \pm 0.4}$	$> 72h$	$5.7e3 \pm 5.3e2$	$1.3e3 \pm 6.6e1$
Nonlinear Case 3: Run Time							
d	NOTEARS-MLP	GraN-DAG	CAM	MMPC	GSGES	DAG-GNN	DAG-GNN + NoCurl
10	$1.1e3 \pm 6.3e2$	$1.3e3 \pm 2.1e2$	$5.1e1 \pm 0.7$	$\mathbf{0.7 \pm 0.0}$	$3.3e2 \pm 1.4e1$	$1.5e2 \pm 1.9e1$	$3.6e2 \pm 3.2e1$
20	$1.2e4 \pm 1.0e4$	$2.2e3 \pm 4.4e2$	$1.4e2 \pm 3.4$	$\mathbf{4.5e1 \pm 6.7}$	$1.6e3 \pm 1.7e2$	$1.3e3 \pm 1.3e2$	$4.2e2 \pm 4.1e1$
50	$7.8e3 \pm 1.1e3$	$2.0e4 \pm 2.2e3$	$\mathbf{3.8e2 \pm 7.3}$	$> 72h$	$1.1e4 \pm 7.3e2$	$2.7e3 \pm 3.2e2$	$9.9e2 \pm 5.2e1$
100	$3.2e4 \pm 6.2e3$	$> 72h$	$\mathbf{7.9e2 \pm 5.5}$	$> 72h$	$> 72h$	$4.7e3 \pm 3.5e2$	$1.3e3 \pm 3.1e1$

Table 11. Accuracy Results on Protein Signaling Network, where bold number highlights the best method.

Method	FGS	NOTEARS	NOTEARS-MLP	DAG-GNN	GraN-DAG	CAM	DAG-GNN+NoCurl
# Edges	17	16	13	18	-	-	18
SHD	22	22	16	19	13^*	$\mathbf{12}^*$	16

CAM and GraN-DAG results adopted from (Lachapelle et al., 2019), without the number of edges reported.