LEARNING DAGS FROM FOURIER-SPARSE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a novel perspective on learning directed acyclic graphs (DAGs) from data, leveraging a recently proposed theory of causal Fourier analysis on DAGs. We build on prior work that learned DAGs from data generated by a structural equation model (SEM). First, we show that data generated by linear SEMs can be characterized in the frequency domain as having dense spectra with random coefficients. Then we propose the new problem of learning DAGs from approximately Fourier-sparse data, which we solve by minimizing the L^1 norm of the spectrum. We provide a motivation for this problem and compare our method to prior DAG learning methods, showing superior performance.

1 INTRODUCTION

In this work we study the problem of learning directed acyclic graphs (DAGs) from data using a Fourier analysis perspective. The DAG learning problem can be stated as learning the edges of an unknown DAG given data indexed by its nodes. DAGs can represent causal dependencies (edges) between events (nodes) in the sense that an event only depends on its predecessors. Thus, DAG learning has applications in causal discovery, which, however, is a more demanding problem that we do not consider here, as it requires further concepts of causality analysis, like interventions (Peters et al., 2017). Despite being conceptually simpler than causal learning, learning a DAG from data is still a very challenging problem and even NP-hard (Chickering et al., 2004). Thus, one has to make assumptions on the data generating process, to infer information about the underlying DAG.

One common assumption is that the data follow a structural equation model (SEM) (Shimizu, 2014; Zheng et al., 2018; Gao et al., 2021), meaning that the value of every node is recursively computed as a function of the values of its direct parents and noise. In this paper we will consider a novel form of generating data on DAGs that is based on the notion of Fourier-sparsity. In particular, our viewpoint on DAG learning is based on a recently proposed theory of causal Fourier analysis on DAGs (Seifert et al., 2022a;b). The theory instantiates all basic Fourier concepts including shift, filters, spectrum and Fourier transform for signals (or data) on DAGs, following the general approach of the algebraic signal processing theory (Püschel & Moura, 2006). In particular, the proposed Fourier transform for DAGs coincides with the weighted version of the Möbius transform from combinatorics (Rota, 1964). Interestingly, as we will show first, data produced by linear SEMs can then be viewed as having a dense, random spectrum of causes (in a sense being defined) with the data generation matrix being the inverse Fourier transform.

With this viewpoint we inherit a particular notion of Fourier-sparsity, i.e., data with an approximately sparse spectrum of causes. For prior, classical forms of Fourier transforms, Fourier-sparsity has played a significant role, including of course the discrete Fourier transform (DFT) (Hassanieh, 2018), the discrete cosine transform (DCT) where it enables JPEG compression (Wallace, 1991), or the Walsh-Hadamard transform for estimating set functions Stobbe & Krause (2012).

Contributions. In this work we leverage the concept of Fourier-sparsity associated with the Fourier analysis from (Seifert et al., 2022b) for solving the DAG learning problem. Our contributions consist of the following:

- We characterize data generated by linear SEMs in the spectral domain. We show that they cover the whole frequency spectrum and have random Fourier coefficients.
- Motivated by the concept of Fourier-sparsity, we pose the new problem of learning DAGs from data with approximately sparse Fourier-spectrum.

- To solve this problem, we propose a novel linear DAG learning method, called Möbius, that has as minimization objective the L^1 norm of the approximated spectrum. Analyzing Möbius experimentally, in comparison with recent linear DAG learning methods, we conclude that our method offers significant improvements on Fourier-sparse data.

2 Causal Fourier Analysis

In this section we provide the necessary background on the causal Fourier analysis on DAGs from (Seifert et al., 2022b;a) including a motivating example.

2.1 Background

DAG. We consider a DAG $\mathcal{G} = (V, E)$ with $|V| = d$ vertices and no loops. $\mathcal{G}$ induces a partial order on V: $i \leq j$ for $i, j \in V$ whenever there is a path from i to j. We assume the vertices to be sorted topologically and for simplicity assume then $V = \{1, 2, ..., d\}$. Further, we assume a weighted adjacency matrix $\boldsymbol{A}$ of the graph:

$$\boldsymbol{A} = (a_{ij})_{i,j \in V} = \begin{cases} a_{ij}, \text{ if } (i,j) \in E, \\ 0, \text{ else.} \end{cases} \tag{1}$$

We consider the nodes as events and say that i is a cause of j if $i \leq j$. Note that $\boldsymbol{A}^d = \boldsymbol{0}$.

Signal. A *signal* (or data) over $\mathcal{G}$ is defined as a vector $\boldsymbol{s} = (s_i)_{i \in V}$ indexed by the vertices of $\mathcal{G}$, where s_i has been measured at i. Further, the model assumes that every s_i is a linear combination

$$s_i = \sum_{j \leq i} c_j w_{ji}, \quad \text{for all } i \in V, \tag{2}$$

of unknown values c_j of its predecessors with associated weights w_{ji}. By slight extension of terminology, c_j, for $j \leq i$, is called a cause of s_i and w_{ji} captures the associated strength of influence. In matrix form, Eq. (2) becomes $\boldsymbol{s} = \boldsymbol{W}^T \boldsymbol{c}$ with $\boldsymbol{W} = (w_{ij})_{i,j}$. Note that the weights are present for all $j \leq i$, not just for the edges as in $\boldsymbol{A}$. Thus, $\boldsymbol{W}$ will be computed through a transitive closure of $\boldsymbol{A}$ as explained next.

Weighted transitive closure. The computation of $\boldsymbol{W}$ from $\boldsymbol{A}$ first requires a weighted transitive closure of $\boldsymbol{A}$, denoted with $\overline{\boldsymbol{A}}$. According to the theory in (Abdali & Saunders, 1985) there are several choices depending on the meaning of the weights in $\boldsymbol{A}$ which could be the shortest path's length, level of pollution, most reliable path, or capacity, and they can be computed using a modified Floyd-Warshall algorithm. Two examples of weighted transitive closures are the following:

$$\begin{aligned} \text{Trivial zero closure:} \quad & \overline{\boldsymbol{A}} = \boldsymbol{A} \\ \text{Floyd-Warshall closure:} \quad & \overline{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{A}^2 + ... + \boldsymbol{A}^{d-1} \end{aligned}$$

The trivial zero closure encodes a Markovian condition: only direct predecessors have an influence. The Floyd-Warshall closure can be computed by running a modified Floyd-Warshall algorithm (Lehmann, 1977). One intuitive interpretation of $\overline{\boldsymbol{A}}$ is a pollution model. Assume $\mathcal{G}$ is a river network and an edge weight $a_{ij} \in [0, 1]$, $(i, j) \in E$, captures what fraction of a pollutant inserted at i reaches the neighbour j. Then $\overline{a}_{ij}$ is the total fraction for every $i \leq j$, considering all paths from i to j.

It remains to determine the self-loop weights $w_{ii},\ i \in V$, which are all chosen as $= 1$. As a result,

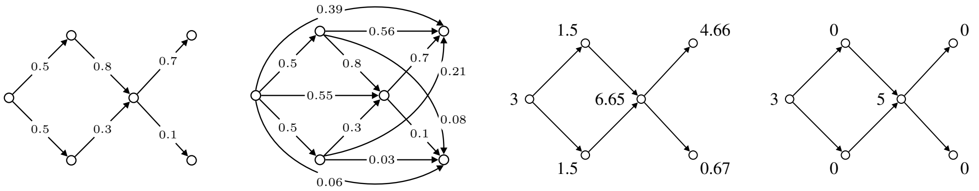
$$\boldsymbol{W} = \boldsymbol{I} + \overline{\boldsymbol{A}} \tag{3}$$

is a reflexive-transitive closure of $\boldsymbol{A}$ and invertible. One might ask the question whether there can exist multiple DAGs $\boldsymbol{A}$ that attain the same Floyd-Warshall transitive closure $\overline{\boldsymbol{A}}$. The answer is negative:

Lemma 1. *The Floyd-Warshall transitive closure $\boldsymbol{A} \mapsto \overline{\boldsymbol{A}}$ is a bijective mapping.*

Proof. It holds that

$$\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right)(\boldsymbol{I} - \boldsymbol{A}) = \left(\boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + ... + \boldsymbol{A}^{d-1}\right)(\boldsymbol{I} - \boldsymbol{A}) = \boldsymbol{I}, \tag{4}$$



(a) Original DAG (b) Transitive closure (c) Signal (d) Spectrum

Figure 1: (a) A DAG as example of a river network with weights that capture fraction of pollution transported between adjacent nodes. (b) shows its transitive closure whose weights are fractions of pollution transported between all pairs of nodes that succeed each other, (c) a possible signal measuring pollution, and (d) its spectrum, in this case sparse, revealing the causes.

and thus

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{B}} \Leftrightarrow (\boldsymbol{I} - \boldsymbol{A})^{-1} = (\boldsymbol{I} - \boldsymbol{B})^{-1} \Leftrightarrow \boldsymbol{A} = \boldsymbol{B}. \tag{5}$$

□

Fourier transform. In summary, the linear relationship (2) between the observed signal $\boldsymbol{s}$ and its causes $\boldsymbol{c}$ becomes

$$\boldsymbol{s} = \boldsymbol{W}^T \boldsymbol{c} = \left(\boldsymbol{I} + \overline{\boldsymbol{A}}^T\right) \boldsymbol{c}. \tag{6}$$

(Seifert et al., 2022a) argues that $\boldsymbol{c}$ can be interpreted as a form of spectrum of $\boldsymbol{s}$. This is done by providing a suitable notion of shift and associated shift-equivariant convolution whose eigenvectors are the columns of $\boldsymbol{I} + \overline{\boldsymbol{A}}^T$. Here, we only need and state the Fourier transform obtained by inverting (6):

$$\boldsymbol{c} = \boldsymbol{W}^{-T} \boldsymbol{s} = \left(\boldsymbol{I} + \overline{\boldsymbol{A}}^T\right)^{-1} \boldsymbol{s} = (\boldsymbol{I} - \boldsymbol{A}^T)\boldsymbol{s}$$

using the proof of Lemma 1. Within the presented model it computes the causes from the observed data. $\boldsymbol{c}$ is sparse if $\boldsymbol{s}$ has few causes.

2.2 Intuitive example

We expand the pollution in a river network mentioned before for an intuitive example.

DAG. We assume a DAG describing a river network. The acyclicity is guaranteed since flows only occur downstream. The DAG consists of 6 nodes and is depicted in Fig. 1a. The nodes $i \in V$ represent geographical points of interest, e.g. cities, and edges are rivers connecting them. We suppose that the cities can cause pollution to the rivers. The edge weight $a_{ij} \in [0, 1]$ describes what fraction of a pollutant at i reaches a neighbour j.

Transitive closure. Fig. 1b shows the transitive closure of the DAG in (a). The edge weight $\overline{a}_{ij} \in [0, 1]$ now represents the total fraction of a pollutant at i that reaches j via all connecting paths.

Signal and spectrum. Fig. 1c shows a possible signal $\boldsymbol{s}$ on the DAG, for example the total pollution measured at each node. Within the model, the associated spectrum $\boldsymbol{c}$ in Fig. 1d then shows the origin or causes of the pollution. In this case the spectrum is sparse, i.e., there are few causes. Fourier-sparsity will be the key assumption in our work.

Fourier transform. Fig. 2 shows the matrices associated with the example: the DAG's adjacency matrix, its reflexive-transitive closure, and its inverse-transpose, which is the Fourier transform matrix.

3 Linear Data on DAGs

In this section we interpret linear SEMs from a causal Fourier analysis perspective. In particular, we show that they can be interpreted as generating signals with a dense, random spectrum. This will then

$$\begin{pmatrix}0 & 0.5 & 0.5 & 0 & 0 & 0\\ 0 & 0 & 0 & 0.8 & 0 & 0\\ 0 & 0 & 0 & 0.3 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.7 & 0.1\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}$$

(a) $\boldsymbol{A}$

$$\begin{pmatrix}1 & 0.5 & 0.5 & 0.55 & 0.39 & 0.06\\ 0 & 1 & 0 & 0.8 & 0.56 & 0.08\\ 0 & 0 & 1 & 0.3 & 0.21 & 0.03\\ 0 & 0 & 0 & 1 & 0.7 & 0.1\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{pmatrix}$$

(b) $I + \overline{\boldsymbol{A}}$

$$\begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\\ -0.5 & 1 & 0 & 0 & 0 & 0\\ -0.5 & 0 & 1 & 0 & 0 & 0\\ 0 & -0.8 & -0.3 & 1 & 0 & 0\\ -0 & 0 & -0 & -0.7 & 1 & 0\\ 0 & -0 & 0 & -0.1 & 0 & 1\end{pmatrix}$$

(c) $\left(\boldsymbol{I} + \overline{\boldsymbol{A}}^{T}\right)^{-1} = \boldsymbol{I} - \boldsymbol{A}^{T}$

Figure 2: Numerical data for the DAG example. The first matrix is the weighted adjacency matrix of the DAG. The second is its reflexive-transitive closure, and the third, its inverse, is the Fourier matrix that computes causes from the signal in the presented model.

allow us to generalize the process of generating causal data to synthetically imposing (approximate) Fourier-sparsity. First, we recall the definition of a linear SEM.

Linear SEM. Consider a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ of n signals of dimension d indexed by the vertices of the DAG $\mathcal{G}$ with weighted adjacency matrix $\boldsymbol{A}$ and $\boldsymbol{N} \in \mathbb{R}^{n\times d}$ a matrix of independent random noise values. The data $\boldsymbol{X}$ follow a linear SEM if

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{N}$$

Linear SEMs (Peters et al., 2017) first appeared in the literature of econometrics in 1930's (Frisch & Waugh, 1933). Currently, several continuous DAG learning methods are based on the assumption that data $\boldsymbol{X}$ follow a linear SEM (Shimizu, 2014; Zheng et al., 2018; Ng et al., 2020). They assume the following data generating process: First, the values at the sources of the DAG are initialized with random noise. Then, the remaining nodes are processed in topological order and each node is assigned the linear combination of its parents' values in addition with independent noise. Formally, the procedure is described by Algorithm 1. Computationally, Algorithm 1 operates in a similar manner to the signal model (2) in the sense that the values of a node depend on itself and its ancestors. However, the main difference is that in the linear SEM only the direct parents are considered, and the node values occur linearly from signal values instead of spectral ones. The interesting connection between linear SEMs and the causal Fourier transform is analyzed with Lemma 2.

Algorithm 1: Linear SEM data generation

1. A DAG $\mathcal{G}(V, E)$ is given with weighted adjacency matrix $\boldsymbol{A}$.
2. Initialize the columns i that correspond to sources of the graph $\mathcal{G}$ with random noise $\boldsymbol{X}_{:,i} = \boldsymbol{N}_{:,i}$.
3. Compute the values for every other vertex in topological order. The value of the column corresponding to ith vertex is computed based on the values of its direct parents
$$\boldsymbol{X}_{:,i} = \boldsymbol{X}_{:,Pa(i)}\boldsymbol{A}_{Pa(i),i} + \boldsymbol{N}_{:,i}$$
4. Output the synthetic data $\boldsymbol{X}$ that satisfy $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{N}$.

Lemma 2. *The linear SEM data generation Algorithm 1 computes data $\boldsymbol{X}$ that satisfy*

$$\boldsymbol{X} = \boldsymbol{N}\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right) \tag{7}$$

Proof. Using $\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right)\left(\boldsymbol{I} - \boldsymbol{A}\right) = \boldsymbol{I}$, we derive

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{N} \Leftrightarrow \boldsymbol{X}\left(\boldsymbol{I} - \boldsymbol{A}\right) = \boldsymbol{N} \Leftrightarrow \boldsymbol{X} = \boldsymbol{N}\left(\boldsymbol{I} - \boldsymbol{A}\right)^{-1} \Leftrightarrow \boldsymbol{X} = \boldsymbol{N}\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right)$$

as desired. In the appendix, we also include a constructive proof of the above statement based on the iterations of Algorithm 1. □

Lemma 2 together with the analysis of the previous section yields the following interpretation of linear SEMs.

Theorem 1. *The linear SEM data generation Algorithm 1 is equivalent to the inverse causal Fourier transformation of the noise spectra $\boldsymbol{N}$, given by (7), when the weights $\boldsymbol{W}$ are computed according to the Floyd-Warshall closure. In essence, in (7) the rows of the noise matrix $\boldsymbol{N}$ contain the spectra of the signals that constitute the rows of $\boldsymbol{X}$.*

Fourier-sparse data. Motivated by Theorem 1, we define a novel data generating process for DAG signals based on the notion of Fourier sparsity. As previously, we consider a topologically sorted DAG $\mathcal{G}$ with weighted adjacency matrix $\boldsymbol{A}$ and transitive closure $\overline{\boldsymbol{A}}$. We define the matrices $\boldsymbol{C}, \boldsymbol{N}_f, \boldsymbol{N}_s \in \mathbb{R}^{n\times d}$ that correspond to the signal, spectra and the random noise for the frequency and signal domain respectively. According to the spectra and the noise, the data $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ are generated via the causal Fourier transform

$$\boldsymbol{X} = (\boldsymbol{C} + \boldsymbol{N}_f)\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right) + \boldsymbol{N}_s. \tag{8}$$

The data are characterized as approximately Fourier-sparse if

$$\begin{aligned} \min_{i,j}\{|c_{ij}|,\, c_{ij} \neq 0\} &\gg \max\{\|\boldsymbol{N}_f\|_\infty, \|\boldsymbol{N}_s\|_\infty\}, \text{ and} \\ nd &\gg \|\boldsymbol{C}\|_0. \end{aligned} \tag{9}$$

The L^0 norm promotes only a few spectral coefficients to be non-zero. Thus, in this context, being sparse in the Fourier domain simply translates into the signal being generated using only a few causes. Note that (8) provides an alternative way to express linear SEMs. Namely, for zero spectra $\boldsymbol{C} = \boldsymbol{0}$, zero noise on the signal domain $\boldsymbol{N}_s = \boldsymbol{0}$ and Floyd-Warshall closure $\overline{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{A}^2 + ... + \boldsymbol{A}^{d-1}$, (8) specializes to (7). Thus, our model (8) generalizes linear SEMs by potentially allowing non-zero spectra, signal noise, and other notions of transitive closure from (Abdali & Saunders, 1985).

Example. Consider the river network of our previous example. We assume that we take measurement for all d sensors of the network once in a day for n days and collect them in the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$. As before, the data $\boldsymbol{X}$ represent accumulated pollution in the nodes of the river network. The pollution is caused at nodes (cities) of the network. Fourier-sparsity means that every day only a small number of cities pollute, which is captured by $\boldsymbol{C}$. Negligible pollution from other sources is modeled as the noise $\boldsymbol{N}_f$. $\boldsymbol{N}_s$ models the noise in the sensor measurements. Thus, in this scenario we deduce that $\boldsymbol{C}, \boldsymbol{N}_f, \boldsymbol{N}_s$ satisfy (9), i.e., the measurements in $\boldsymbol{X}$ are approximately Fourier-sparse.

4 Method for Fourier-sparse Data

In this section we present our proposed continuous optimization problem for learning DAGs from Fourier-sparse data. We assume that the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is generated as the inverse Fourier transformation of sparse spectra with noise according to (8). The Fourier-sparsity is ensured when $\boldsymbol{C}$ and the noise $\boldsymbol{N}_f, \boldsymbol{N}_s$ satisfy the sparsity criterion (9). Notice that for both noise matrices $\boldsymbol{N}_f, \boldsymbol{N}_s$ we do not make any assumption (e.g., on their distribution) other than that the noise has low magnitude. The general approach for solving the DAG learning problem via continuous optimization consists of the following objective

$$\begin{aligned} \min_{\boldsymbol{A}\in\mathbb{R}^{d\times d}} &\ \ell(\boldsymbol{X}, \boldsymbol{A}) + R(\boldsymbol{A}) \\ \text{s.t.} &\quad h(\boldsymbol{A}) = 0, \end{aligned} \tag{10}$$

in which $\ell(\boldsymbol{A}, \boldsymbol{X})$ is the loss function corresponding to the data, $R(\boldsymbol{A})$ is a regularizer that promotes sparsity in the adjacency matrix and $h(\boldsymbol{A})$ is any continuous constraint for acyclicity. Current approaches assume that the data are generated via a linear SEM following the equation $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{A} + \boldsymbol{N}$ and the loss function in that case takes the form

$$\ell(\boldsymbol{X}, \boldsymbol{A}) = \frac{1}{2n}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{A}\|_2^2 = \frac{1}{2n}\left\|\hat{\boldsymbol{N}}\right\|_2^2$$

This loss function is called the mean-squared loss approximation and achieves the minimization of the L^2 norm of the approximated noise $\hat{\boldsymbol{N}}$. A popular choice for the regularizer is the L^1 norm $R(\boldsymbol{A}) = \lambda\|\boldsymbol{A}\|_1$ to promote sparsity in the adjacency matrix. The acyclicity constraint can be, for example, the one proposed by (Zheng et al., 2018), namely, $h(\boldsymbol{A}) = tr\left(e^{\boldsymbol{A}\odot\boldsymbol{A}}\right) - d$, or the one from (Yu et al., 2021), which is $h(\boldsymbol{A}) = tr\left((\boldsymbol{I} + \lambda\boldsymbol{A}\odot\boldsymbol{A})^d\right) - d$.

For our case we consider the following instantiation of the optimization problem (10):

$$\begin{aligned} \min_{\boldsymbol{A}\in\mathbb{R}^{d\times d}} \frac{1}{2n} \left\| \boldsymbol{X}\left(\boldsymbol{I}+\overline{\boldsymbol{A}}\right)^{-1} \right\|_1 + \lambda \|\boldsymbol{A}\|_1 & \\ \text{s.t.} \quad h\left(\boldsymbol{A}\right) = 0. & \end{aligned} \tag{11}$$

We call this method Möbius due to the fact that the Fourier transform in the causal setting coincides with the Möbius transformation for posets (Seifert et al., 2022a). Here the acyclicity constraint can be chosen freely among those previously mentioned. The choice of the L^1 norm for minimizing the approximated spectra $\boldsymbol{X}\left(\boldsymbol{I}+\overline{\boldsymbol{A}}\right)^{-1}$ is to promote sparsity. If we assume that we have found the true weighted adjacency matrix $\boldsymbol{A}$ then according to the Eq. (8) the approximated spectra can be computed explicitly as

$$\frac{1}{2n} \left\| \boldsymbol{X}\left(\boldsymbol{I}+\overline{\boldsymbol{A}}\right)^{-1} \right\|_1 = \frac{1}{2n} \left\| \boldsymbol{C} + \boldsymbol{N}_f + \boldsymbol{N}_s\left(\boldsymbol{I}+\overline{\boldsymbol{A}}\right)^{-1} \right\|_1 .$$

This quantity is sparse when assumption (9) holds and $\boldsymbol{N}_s\left(\boldsymbol{I}+\overline{\boldsymbol{A}}\right)^{-1}$ has entries of low magnitude.

Remark. Notice that, the optimization problem (11) becomes the conventional one in the linear SEM setting if we use the Floyd-Warshall transitive closure and interchange the L^2 with the L^1 norm. For the linear SEM case, the mean-squared loss is a more compatible loss function to consider in order to minimize the approximated noise. Therefore, our method is an extension of the current DAG learning literature in order to learn DAGs from Fourier-sparse data. Furthermore, it could also be used with different notions of transitive closure.

5 Related Work

Graph learning. We build on the idea to utilize Fourier analysis tools to learn graphs that are directed and acyclic. Related work applies graph signal processing (GSP) (Sandryhaila & Moura, 2013b; Ortega et al., 2018) concepts to extract the graph structure from data (Xia et al., 2021). Kalofolias & Perraudin (2017); Kalofolias (2016) learn graphs by applying a GSP smoothness criterion, Dong et al. (2016) learn the Laplacian of the graph, and Egilmez et al. (2017); Pavez et al. (2018); Koyakumaru et al. (2021) utilize Gaussian Markov random fields for graph learning. Prior GSP tools are not applicable to DAGs since the spectrum collapses (all eigenvalues of the adjacency matrix are zero) and no Fourier transform is available. The Fourier analysis from (Seifert et al., 2022b) that we use is fundamentally different from prior GSP and specifically designed for DAGs.

DAG learning. The approaches to solve the DAG learning problem fall into two categories, using combinatorial search or continuous relaxations. In combinatorics terms, DAG learning is an NP-hard problem, as the space of possible DAGs is super-exponential in the number of vertices (Chickering et al., 2004). Methods that search on the space of possible DAGs apply heuristic criteria in order to find the ground truth DAG (Ramsey et al., 2017; Chickering, 2002; Hauser & Bühlmann, 2012; Tsamardinos et al., 2006). Lately, with the computational advances of deep learning , researchers have been focusing on continuous optimization methods (Vowels et al., 2021). These methods model the data generation process using SEMs. Among the first methods to utilize SEMs were CAM (Bühlmann et al., 2014) and LINGAM (Shimizu, 2014) with the latter specializing in linear SEMs with Gaussian noise. NOTEARS (Zheng et al., 2018) described the combinatorial constraint of acyclicity as a continuous one. Despite some vulnerabilities, like lack of scale-invariance (Kaiser & Sipos, 2022; Reisach et al., 2021), it has inspired many subsequent DAG learning methods. A continuation of it (Zheng et al., 2020) utilizes neural networks to generalize NOTEARS to non-linear SEMs. Other nonlinear methods for DAG learning are DAG-GNN (Yu et al., 2019), in which also a more efficient acyclicity constraint than the one in NOTEARS is proposed, and DAG-GAN (Gao et al., 2021). DAG-NoCurl (Yu et al., 2021) proposes learning the DAG on the equivalent space of weighted gradients of graph potential functions. GOLEM (Ng et al., 2020) studies the role of the weighted adjacency matrix sparsity, the acyclicity constraint, and proposes to directly minimize the data likelihood. The method DYNOTEARS (Pamfil et al., 2020) implements a variation of NOTEARS compatible with time series data. A recent line of works consider permutation-based methods to parametrize the search space of the DAG (Charpentier et al., 2022; Zantedeschi et al., 2022). Our work considers DAG learning under the new assumption of Fourier-sparsity.

Fourier sparsity. The notion of sparsity in the frequency domain has been widely used in different forms of Fourier analysis. For the classical DFT (Hassanieh et al., 2012b; Indyk & Kapralov, 2014) it enables faster data processing in spectrum sensing (Hassanieh et al., 2014), magnetic resonance imaging (Shi et al., 2013), or GPS locking (Hassanieh et al., 2012a). For the DCT it enables JPEG compression. For the WHT and other set function Fourier transforms (Püschel & Wendler, 2020) it enables estimation from samples (Stobbe & Krause, 2012; Amrollahi et al., 2019; Wendler et al., 2021). In GSP it has enabled compression or estimation (Sandryhaila & Moura, 2013a; Ortega et al., 2018; Chen et al., 2015). Our work aims to leverage Fourier-sparsity instantiated for (Seifert et al., 2022a) for DAG learning.

6 Experiments

We experimentally evaluate our novel DAG learning method with both synthetically generated Fourier-sparse data and real data from (Sachs et al., 2005).[1]

Baselines. We compare against DAG learning methods suitable for data generated by classical linear SEMs with additive noise. In particular, we consider the prior NOTEARS (Zheng et al., 2018), GOLEM (Ng et al., 2020), the trivial baseline sortnregress (Reisach et al., 2021), DAG-NoCurl (Yu et al., 2021), greedy equivalence search (GES) (Chickering, 2002; Hauser & Bühlmann, 2012), fast greedy equivalence search (fGES) (Ramsey et al., 2017), max-min hill-climbing (MMHC) (Tsamardinos et al., 2006), and causal additive models (CAM) (Bühlmann et al., 2014).

Metrics. We use the following performance metrics that consider the unweighted approximation $\hat{\boldsymbol{U}}$ of the adjacency matrix: (a) the true positive rate (TPR), computed as the ratio of correct edges found by $\hat{\boldsymbol{U}}$, (b) the structural Hamming distance (SHD), which is the number of edge insertions, deletions or reverses needed to get from $\hat{\boldsymbol{U}}$ to the ground truth $\boldsymbol{U}$, and (c) the total number of nonzero edges (NNZ) proposed by the algorithm. The first two metrics accurately show whether an algorithm yields a good approximation while the last one only measures whether the algorithm proposes a number of edges close to the real number. For the weighted matrix approximation we compute the normalized mean-squared error (NMSE) as $\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_2 / \|\boldsymbol{A}\|_2$.

6.1 Evaluation on Fourier-sparse data

We perform our first evaluation on synthetic Fourier-sparse data on DAGs.

Data generating process. We first generate a random Erdös-Renyi graph with weights sampled uniformly at random from $(-b, -a) \cup (a, b)$, where $a, b \in \mathbb{R}^+$ are fixed. This results in the true adjacency matrix $\boldsymbol{A}$. Next, the sparse DAG spectra $\boldsymbol{C}$ are instantiated by setting each entry either to 0 with probability $1 - p$ or to some random uniform value from $(0, 1)$ with probability p. A low value of p yields higher sparsity, since p is the expected fraction of non-zero coefficients. Note that this means that the support (locations of nonzero values) varies between spectra. Finally, the data matrix $\boldsymbol{X}$ is computed according to Eq. (8) where only one of the noise matrices $\boldsymbol{N}_f, \boldsymbol{N}_s$ is non-zero. The non-zero noise is set to have low standard deviation $\sigma = 0.01$. This together with low value of p ensures that the inequalities (9) hold and that the data are truly Fourier-sparse. After executing the corresponding method on $\boldsymbol{X}$ we get its approximation $\hat{\boldsymbol{A}}$ of the true weighted adjacency matrix $\boldsymbol{A}$. For the evaluation we take the average and standard deviation over 10 repetitions, for all methods and all corresponding performance metrics.

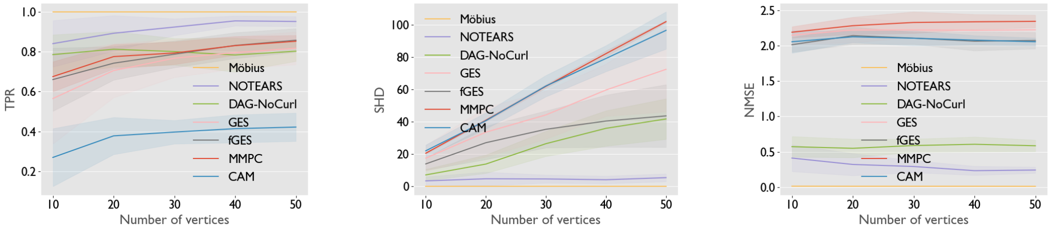
Experiment 1: Comparison of the methods. We consider first the experiment with the default experimental settings shown in the blue column of Table 4. We show the results in Fig. 3, where we see that Möbius achieves optimal performance and the rest of the methods perform worse. The reason for this behavior is that Möbius optimization problem (11) is well suited for these settings. Both the generated data are Fourier-sparse and the DAG is sparse in the sense it has low average degree.

Experiment 2: Challenging the methods. In Fig. 4, we challenge the algorithms by either using a small number of samples to learn the DAG or by restricting the Fourier spectra to be non-zero always the same locations (fixed support). While, for a small number of samples, Möbius performance degrades, it still achieves the best TPR among all methods.

[1] All of our code is available as supplementary material. We used the repositories of NOTEARS (Zheng et al., 2018), and DAG-NoCurl (Yu et al., 2021) and the causal discovery toolbox (Kalainathan & Goudet, 2019).

Hyperparameter	Values		True positive rate (TPR ↑)								
	Default	Current	Möbius	GOLEM	sortnregress	NOTEARS	DAG-NoCurl	GES	fGES	MMHC	CAM
Default settings			**1.00 ± 0.00**	0.95 ± 0.02	0.90 ± 0.05	0.95 ± 0.03	0.77 ± 0.06	0.85 ± 0.05	0.78 ± 0.12	0.85 ± 0.05	0.43 ± 0.07
Graph Type	Erdös-Renyi	Scale-Free	**1.00 ± 0.00**	0.97 ± 0.03	0.98 ± 0.02	0.97 ± 0.01	0.91 ± 0.05	0.80 ± 0.08	0.91 ± 0.07	0.71 ± 0.07	0.26 ± 0.08
Signal Noise	$\sigma = 0.01$	$\sigma = 0.1$	**0.96 ± 0.01**	0.89 ± 0.04	0.90 ± 0.06	0.86 ± 0.05	0.73 ± 0.08	0.82 ± 0.07	0.72 ± 0.10	0.82 ± 0.06	0.43 ± 0.05
Low Sparsity	30%	60%	0.55 ± 0.07	**0.97 ± 0.02**	0.92 ± 0.03	0.96 ± 0.02	0.78 ± 0.06	0.82 ± 0.07	0.75 ± 0.09	0.83 ± 0.05	0.51 ± 0.10
High Sparsity	30%	10%	**1.00 ± 0.00**	0.85 ± 0.06	0.90 ± 0.03	0.80 ± 0.06	0.78 ± 0.07	0.83 ± 0.05	0.77 ± 0.08	0.81 ± 0.04	0.29 ± 0.05
Small weight bounds	$(0.4, 0.8)$	$(0.2, 0.4)$	0.49 ± 0.07	0.34 ± 0.05	0.81 ± 0.05	0.20 ± 0.04	0.34 ± 0.07	0.76 ± 0.07	0.71 ± 0.09	**0.91 ± 0.02**	0.11 ± 0.04
Large weight bounds	$(0.4, 0.8)$	$(0.5, 2)$	0.97 ± 0.03	**0.98 ± 0.02**	0.95 ± 0.03	0.94 ± 0.06	0.93 ± 0.03	0.80 ± 0.06	0.82 ± 0.09	0.48 ± 0.03	0.42 ± 0.08
Edges / Vertices	2	3	**1.00 ± 0.00**	0.94 ± 0.04	0.89 ± 0.04	0.89 ± 0.04	0.77 ± 0.05	0.68 ± 0.10	0.58 ± 0.11	0.56 ± 0.05	0.37 ± 0.08
Standardization	No	Yes	0.34 ± 0.04	0.24 ± 0.08	0.48 ± 0.08	0.27 ± 0.03	0.12 ± 0.04	**0.83 ± 0.07**	0.78 ± 0.11	0.82 ± 0.04	0.41 ± 0.06
Transitive closure	Floyd-Warshall	Zero	0.69 ± 0.11	0.86 ± 0.05	0.83 ± 0.04	0.75 ± 0.04	0.73 ± 0.05	0.78 ± 0.06	0.61 ± 0.07	**0.88 ± 0.04**	0.44 ± 0.03
Noise Distribution	Gaussian	Gumbel	**1.00 ± 0.00**	0.94 ± 0.02	0.91 ± 0.03	0.93 ± 0.03	0.78 ± 0.07	0.82 ± 0.04	0.80 ± 0.10	0.81 ± 0.05	0.39 ± 0.10
Spectral Noise	$\boldsymbol{N}_f = 0,\ \boldsymbol{N}_s \neq 0$	$\boldsymbol{N}_f \neq 0,\ \boldsymbol{N}_s = 0$	**1.00 ± 0.00**	0.95 ± 0.02	0.90 ± 0.02	0.94 ± 0.02	0.77 ± 0.04	0.82 ± 0.07	0.81 ± 0.12	0.82 ± 0.03	0.43 ± 0.08
Samples	400	20	**0.52 ± 0.09**	0.26 ± 0.09	error	0.34 ± 0.07	0.49 ± 0.05	0.36 ± 0.04	error	0.21 ± 0.03	time-out
Fixed support	No	Yes	0.13 ± 0.05	0.14 ± 0.04	0.25 ± 0.10	0.14 ± 0.06	0.31 ± 0.10	0.30 ± 0.09	**0.32 ± 0.07**	0.23 ± 0.07	0.19 ± 0.07

Table 1: TPR metric for learning graph with 40 nodes. The blue column shows the default value of the corresponding parameter and the next column the current choice for the experiment, while all other parameters are set to default. Best results are marked in bold.



(a) TPR ↑

(b) SHD ↓

(c) NMSE ↓

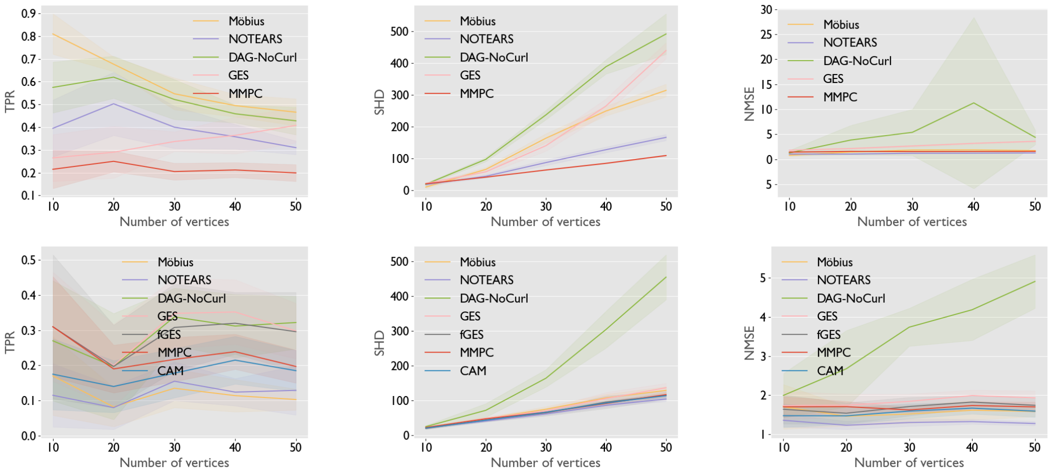
Figure 3: Plots illustrating performance metrics TPR ↑ (higher is better), SHD ↓ (lower is better) and NMSE ↓ (lower is better) on the default experimental settings.

For fixed Fourier-sparsity support all algorithms fail to learn the DAG for any number of vertices. One can imagine the fixed sparsity support on the the river pollution example. Fixed support means that there is only a fixed subset of cities that cause pollution. Intuitively, this will result in only giving information about the outgoing edges of these cities and not for the rest. This poses the question whether the problem of learning DAGs from Fourier-sparse data with fixed sparsity support is possible at all with a suitable algorithm.

Experiment 3: Sensitivity analysis on the hyperparameters. In Table 4, we present an extensive series of experiments each of which alters a particular hyperparameter indicated in the first column. For higher standard deviation, we see worse performance on all methods except MMHC and CAM. Our method still performs best. It is expected that for higher standard deviation the sparsity in the data breaks and our algorithm will not longer be efficient. Higher sparsity has no effect on our performance, whereas lower sparsity degrades our performance as expected. In contrast, imposing high sparsity is harmful for NOTEARS. For smaller weight bounds we see a deterioration in performance on most algorithms. The best performance in this case is achieved by MMHC. Larger weight bounds affect our performance only by a small percentage and our method is still the best in this case. For higher average degree we also achieve the best accuracy. Testing on more dense graphs would have a negative effect on methods that impose sparsity on the adjacency matrix, like ours. Next, the standardization of data is something that generally affects negatively algorithms with continuous objectives (Reisach et al., 2021) and is expected. Moreover, on the zero transitive closure our method is not the best but has decent performance close to the best MMHC. Finally, applying noise in the spectrum instead of the signal does not have impact our performance. The last two rows of the table simply repeat what we saw in the previous experiment, but with higher resolution on the TPR metric. Overall, our method achieves the best TPR in most scenarios.

6.2 Evaluation on real data

We also execute our method on the causal protein-signaling network provided by (Sachs et al., 2005). The dataset consists of 7466 samples from a network with 11 nodes that represent proteins and 17 edges showing the interaction between them. Even if the DAG of this network is relatively small, the task of learning it is rather difficult. This dataset has been a common benchmark for many prior DAG



(a) TPR ↑

(b) SHD ↓

(c) NMSE ↓

Figure 4: Deterioration of performance. The first row illustrates the performance when using only 20 data samples. The second row considers spectra with fixed sparsity support.

	Möbius	NOTEARS	GOLEM	sortnregress	DAG-NoCurl	GES	fGES	MMHC	CAM
SHD ↓	15	**11**	21	18	22	13	17	17	12
TPR ↑	0.35	0.41	0.35	0.35	0.35	0.26	0.35	**0.47**	0.35
NNZ	**16**	15	19	20	23	8	14	**16**	10

Table 2: Performance on the real dataset of (Sachs et al., 2005). NNZ shows the total number of non-zero edges proposed by the algorithm. The true number of edges of the unknown DAG is 17.

learning methods Ng et al. (2020); Gao et al. (2021); Yu et al. (2019); Zheng et al. (2018). Among these methods, the best performance has SHD that does not drop below ≈ 10 which is relatively large comparing to the 17 edges in total.

We report the performance metrics for all methods in Table 2. As seen in the synthetic experiment our method works best when the data are Fourier-sparse. However, for this dataset it is not clear whether this is true. Here, our method has decent performance. Among the other methods the best TPR is achieved by MMHC, which, however, has higher SHD. NOTEARS together with CAM have the best SHDs equal to 11 and 12 respectively, which is slightly better than ours which is 15.

7 Conclusion

We presented a novel form on learning DAGs from Fourier-sparse data building on a recently proposed causal Fourier analysis framework. We analyzed linear SEMs from a Fourier analysis point of view and discovered that the data they generate have random spectra in the Fourier domain. This observation allowed us to generalize the data generating process of linear SEMs by further capturing the notion of (approximate) sparsity in the spectral domain. Fourier-sparsity has been a frequently exploited feature in various types of Fourier domains, and we provide an intuitive example to demonstrate viability for DAGs.

To learn DAGs from Fourier-sparse data, we build on prior work to enforce acyclicity but propose an alternative L^1-loss on the approximated spectrum. Empirically, we show that our method has the best performance under the Fourier-sparsity assumption compared to NOTEARS, DAG-NoCurl, and other DAG learning baselines. Thus we hope that our contribution can serve as a valuable building block for learning DAGs in real case scenarios where current methods fail.

References

S. Kamal Abdali and B. David Saunders. Transitive closure and related semiring properties via eliminants. *Theor. Comput. Sci.*, 40:257–274, 1985.

Andisheh Amrollahi, Amir Zandieh, Michael Kapralov, and Andreas Krause. Efficiently learning fourier sparse set functions. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

Bertrand Charpentier, Simon Kibler, and Stephan Günnemann. Differentiable dag sampling. *arXiv preprint arXiv:2203.08509*, 2022.

Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević. Discrete signal processing on graphs: Sampling theory. *IEEE Transactions on Signal Processing*, 63(24):6510–6523, 2015. doi: 10.1109/TSP.2015.2469645.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Max Chickering, David Heckerman, and Chris Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23): 6160–6173, 2016.

Hilmi E. Egilmez, Eduardo Pavez, and Antonio Ortega. Graph learning from data under laplacian and structural constraints. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):825–841, 2017. doi: 10.1109/JSTSP.2017.2726975.

Ragnar Frisch and Frederick V Waugh. Partial time regressions as compared with individual trends. *Econometrica: Journal of the Econometric Society*, pp. 387–401, 1933.

Yinghua Gao, Li Shen, and Shu-Tao Xia. Dag-gan: Causal structure learning with generative adversarial nets. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3320–3324, 2021. doi: 10.1109/ICASSP39728.2021.9414770.

Haitham Hassanieh. *The Sparse Fourier Transform: Theory and Practice*, volume 19. Association for Computing Machinery and Morgan and Claypool, 2018.

Haitham Hassanieh, Fadel Adib, Dina Katabi, and Piotr Indyk. Faster gps via the sparse fourier transform. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 353–364, 2012a.

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1183–1194. SIAM, 2012b.

Haitham Hassanieh, Lixin Shi, Omid Abari, Ezzeldin Hamed, and Dina Katabi. Ghz-wide sensing and decoding using the sparse fourier transform. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 2256–2264. IEEE, 2014.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1): 2409–2464, 2012.

Piotr Indyk and Michael Kapralov. Sample-optimal sparse fourier transform in any constant dimension. In *FOCS*, 2014.

Marcus Kaiser and Maksim Sipos. Unsuitability of notears for causal graph discovery when dealing with dimensional quantities. *Neural Processing Letters*, pp. 1–9, 2022.

Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv:1903.02278*, 2019. URL `https://github.com/FenTechSolutions/CausalDiscoveryToolbox`.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pp. 920–929. PMLR, 2016.

Vassilis Kalofolias and Nathanaël Perraudin. Large scale graph learning from smooth signals. *arXiv preprint arXiv:1710.05654*, 2017.

Tatsuya Koyakumaru, Masahiro Yukawa, Eduardo Pavez, and Antonio Ortega. A graph learning algorithm based on gaussian markov random fields and minimax concave penalty. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5410–5414, 2021. doi: 10.1109/ICASSP39728.2021.9413850.

Daniel J Lehmann. Algebraic structures for transitive closure. *Theoretical Computer Science*, 4(1): 59–76, 1977.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems*, 33:17943–17954, 2020.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106 (5):808–828, 2018.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1595–1605. PMLR, 2020.

Gunwoong Park. Identifiability of additive noise models using conditional variances. *Journal of Machine Learning Research*, 21(75):1–34, 2020. URL `http://jmlr.org/papers/v21/19-664.html`.

Eduardo Pavez, Hilmi E. Egilmez, and Antonio Ortega. Learning graphs with monotone topology properties and multiple connected components. *IEEE Transactions on Signal Processing*, 66(9): 2399–2413, 2018. doi: 10.1109/TSP.2018.2813337.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Markus Püschel and José MF Moura. Algebraic signal processing theory. *arXiv preprint cs/0612077*, 2006.

Markus Püschel and Chris Wendler. Discrete signal processing with set functions. *IEEE Transactions on Signal Processing*, 69:1039–1053, 2020.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2):121–129, 2017.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021.

Gian-Carlo Rota. On the foundations of combinatorial theory i. theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(4):340–368, 1964.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs: Graph fourier transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6167–6170. IEEE, 2013a.

Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. *IEEE transactions on signal processing*, 61(7):1644–1656, 2013b.

Bastian Seifert, Chris Wendler, and Markus Püschel. Learning fourier-sparse functions on dags. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022a.

Bastian Seifert, Chris Wendler, and Markus Püschel. Causal fourier analysis on directed acyclic graphs and posets. *arXiv*, 2022b.

Lixin Shi, Ovidiu Andronesi, Haitham Hassanieh, Badih Ghazi, Dina Katabi, and Elfar Adalsteinsson. Mrs sparse-fft: Reducing acquisition time and artifacts for in vivo 2d correlation spectroscopy. In *ISMRM13, Int. Society for Magnetic Resonance in Medicine Annual Meeting and Exhibition*, 2013.

Shohei Shimizu. Lingam: Non-gaussian methods for estimating causal structures. *Behaviormetrika*, 41(1):65–98, 2014.

Peter Stobbe and Andreas Krause. Learning fourier sparse set functions. In *Artificial Intelligence and Statistics*, pp. 1125–1133. PMLR, 2012.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *arXiv preprint arXiv:2103.02582*, 2021.

Gregory K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, apr 1991. ISSN 0001-0782. doi: 10.1145/103085.103089. URL `https://doi.org/10.1145/103085.103089`.

Chris Wendler, Andisheh Amrollahi, Bastian Seifert, Andreas Krause, and Markus Püschel. Learning set functions that are sparse in non-orthogonal fourier bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10283–10292, 2021.

Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

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

Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. Dags with no curl: An efficient dag structure learning approach. In *International Conference on Machine Learning*, pp. 12156–12166. PMLR, 2021.

Valentina Zantedeschi, Jean Kaddour, Luca Franceschi, Matt Kusner, and Vlad Niculae. Dag learning on the permutahedron. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pp. 3414–3425. PMLR, 2020.

A PROOFS

Here we present an additional proof on Lemma 2 that allows us to get a better understanding of how the values are propagated from parent vertices to their children.

Proof. Consider the total ordering $1, 2, ..., d$ of the vertices of the graph. For each vertex i we will show by induction that $\boldsymbol{X}_{:,i} = \boldsymbol{N}\left(\boldsymbol{I}_{:,i} + \boldsymbol{W}_{:,i} + \boldsymbol{W}^2_{:,i} + ... + \boldsymbol{W}^k_{:,i}\right)$

Starting with the sources, we have that

$$\begin{aligned}
\boldsymbol{X}_{:,i} &= \boldsymbol{N}_{:,i} \\
&= \boldsymbol{N}\boldsymbol{I}_{:,i} \\
&= \boldsymbol{N}\left(\boldsymbol{I}_{:,i} + \boldsymbol{W}_{:,i} + \boldsymbol{W}^2_{:,i} + ... + \boldsymbol{W}^k_{:,i}\right)
\end{aligned}$$

since $\boldsymbol{W}_{:,i} = \mathbf{0}$ and $\boldsymbol{W}^j_{:,i} = \boldsymbol{W}^{j-1}\boldsymbol{W}_{:,i} = \mathbf{0}$, for all $j > 1$.

Now assume that the statement is true up to the $i-$th vertex according to the ordering. Consider, now the vertex $i + 1$. Then $Pa(i + 1) \subseteq \{1, 2, ..., i\}$ and according to the update rule of the algorithm.

$$\begin{aligned}
\boldsymbol{X}_{:,i+1} &= \boldsymbol{X}_{:,Pa(i+1)}\boldsymbol{W}_{Pa(i+1),i+1} + \boldsymbol{N}_{:,i+1} \\
&= \boldsymbol{N}\left(\boldsymbol{I}_{:,Pa(i+1)} + \boldsymbol{W}_{:,Pa(i+1)} + ... + \boldsymbol{W}^k_{:,Pa(i+1)}\right)\boldsymbol{W}_{Pa(i+1),i+1} + \boldsymbol{N}\boldsymbol{I}_{:,i+1} \\
&= \boldsymbol{N}\left(\boldsymbol{I}_{:,i+1} + \boldsymbol{W}_{:,i+1} + \boldsymbol{W}^2_{:,i+1} + ... + \boldsymbol{W}^k_{:,i+1} + \boldsymbol{W}^{k+1}_{:,i+1}\right) \\
&= \boldsymbol{N}\left(\boldsymbol{I}_{:,i+1} + \boldsymbol{W}_{:,i+1} + \boldsymbol{W}^2_{:,i+1} + ... + \boldsymbol{W}^k_{:,i+1}\right)
\end{aligned}$$

The relation

$$\boldsymbol{X}_{:,Pa(i+1)} = \left(\boldsymbol{I}_{:,Pa(i+1)} + \boldsymbol{W}_{:,Pa(i+1)} + ... + \boldsymbol{W}^k_{:,Pa(i+1)}\right)$$

holds by induction and

$$\boldsymbol{W}^l_{:,Pa(i+1)}\boldsymbol{W}_{Pa(i+1),i+1} = \boldsymbol{W}^l\boldsymbol{W}_{:,i+1} = \boldsymbol{W}^{l+1}_{:,i+1}$$

because $\boldsymbol{W}_{j,i+1} = 0$ for all $j \notin Pa(i + 1)$. Also $\boldsymbol{W}^{k+1} = \boldsymbol{O}$ since $\boldsymbol{W}$ is nilpotent. □

The previous proof leads to conclude that the polynomial $\boldsymbol{W} + \boldsymbol{W}^2 + ... + \boldsymbol{W}^k$ works as the transitive closure $\overline{\boldsymbol{W}}$ of the weighted adjacency matrix.

A.1 IDENTIFIABILITY

In addition to Lemma 1, we prove the following lemma to provide a condition that works as a deterministic criterion for the identifiability of the graph from Fourier-sparse data. This lemma simply states that under assumptions on the spectrum and the noise, the generated data correspond uniquely to the unknown DAG.

Lemma 3. *If the transitive closure of the graph is bijective and the data $\boldsymbol{X}$ are generated with fixed spectral data $\boldsymbol{C} + \boldsymbol{N}_f$ with full rank and zero signal noise, then any two graphs generating the same signal data $\boldsymbol{X}$ using zero noise are equal.*

Proof. Indeed, let $\boldsymbol{X}_{\boldsymbol{A}}$ be generated by the graph with weighted adjacency matrix $\boldsymbol{A}$ and $\boldsymbol{X}_{\boldsymbol{B}}$ similarly by $\boldsymbol{B}$. Then if $\boldsymbol{X}_{\boldsymbol{A}} = \boldsymbol{X}_{\boldsymbol{B}}$ we have that

$$\begin{gathered}
\left(\boldsymbol{C} + \boldsymbol{N}_f\right)\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right) = \left(\boldsymbol{C} + \boldsymbol{N}_f\right)\left(\boldsymbol{I} + \overline{\boldsymbol{B}}\right) \Leftrightarrow \\
\left(\boldsymbol{C} + \boldsymbol{N}_f\right)^T\left(\boldsymbol{C} + \boldsymbol{N}_f\right)\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right) = \left(\boldsymbol{C} + \boldsymbol{N}_f\right)^T\left(\boldsymbol{C} + \boldsymbol{N}_f\right)\left(\boldsymbol{I} + \overline{\boldsymbol{B}}\right) \Leftrightarrow \\
\left(\boldsymbol{I} + \overline{\boldsymbol{A}}\right) = \left(\boldsymbol{I} + \overline{\boldsymbol{B}}\right) \Leftrightarrow \\
\boldsymbol{A} = \boldsymbol{B}
\end{gathered}$$

□

Notice that the previous criterion doesn't make any assumption on the variances of the noise variables. In the special case when the signal noise is zero, our model coincides with the linear SEM formulation and in that case we can also use standard identifiability criterions from the literature. In that case

$$\mathbf{X} = \mathbf{X}\mathbf{A} + (\mathbf{C} + \mathbf{N_f}) \Leftrightarrow \mathbf{X} = (\mathbf{C} + \mathbf{N_f})\left(\mathbf{I} + \overline{\mathbf{A}}\right) \tag{12}$$

The way we generate the independent causes is the following. Each data point c in the spectrum is set to 0 with probability $(1-p)$, otherwise set uniformly at random to a value $[0,1]$. Then at each such point we add the corresponding noise n_f. So this means that for each node the variable $c + n_f$ has the same variance. Therefore, this case falls in the category of equal noise variances for linear SEMs and is thus identifiable according to (Peters & Bühlmann, 2014). This is supported empirically, by the fact that our method achieves very high accuracy on discovering the true DAG.

The scenario where we have fixed sparsity support is not generated with equal variances and thus is not justified to be identifiable. We are currently working towards analyzing the theory for the identifiability of our model. As preliminary result we may say that when the support of the sparsity is concentrated on the sink nodes then our model is identifiable due to the conditional variance criterion (Park, 2020).

B ADDITIONAL EXPERIMENTAL RESULTS

In addition to the TPR metric of Table 4 we provide the computation of SHD in the table below.

Hyperparameter	Values		Structural Hamming Distance (SHD ↓)								
	Default	Current	Möbius	GOLEM	sortnregress	NOTEARS	DAG-NoCurl	GES	fGES	MMHC	CAM
Default settings			$\mathbf{0.00 \pm 0.00}$	4.20 ± 2.40	96.80 ± 33.03	4.30 ± 2.61	38.50 ± 9.06	37.50 ± 14.11	64.50 ± 23.00	82.90 ± 1.45	76.60 ± 9.70
Graph Type	Erdös-Renyi	Scale-Free	$\mathbf{0.00 \pm 0.00}$	3.40 ± 4.34	45.10 ± 15.39	2.10 ± 0.83	16.60 ± 9.68	26.60 ± 9.93	42.70 ± 23.12	81.70 ± 2.33	86.10 ± 9.93
Signal Noise	$\sigma = 0.01$	$\sigma = 0.1$	$\mathbf{3.30 \pm 1.19}$	8.80 ± 3.63	105.20 ± 32.66	11.00 ± 4.29	31.80 ± 9.66	36.20 ± 13.70	72.00 ± 18.78	84.10 ± 1.64	70.50 ± 5.78
Low Sparsity	30%	60%	66.80 ± 10.50	$\mathbf{2.60 \pm 1.43}$	83.10 ± 24.05	3.40 ± 1.20	37.90 ± 8.99	44.30 ± 19.63	70.10 ± 29.38	83.00 ± 2.24	64.00 ± 11.36
High Sparsity	30%	10%	$\mathbf{0.00 \pm 0.00}$	18.40 ± 7.76	106.10 ± 27.46	16.30 ± 4.65	37.40 ± 12.55	45.20 ± 14.03	63.40 ± 17.97	81.90 ± 1.22	100.70 ± 8.75
Small weight bounds	$(0.4, 0.8)$	$(0.2, 0.4)$	41.00 ± 6.03	52.50 ± 4.08	48.50 ± 10.02	63.80 ± 2.99	53.20 ± 5.58	$\mathbf{23.50 \pm 7.37}$	43.00 ± 8.01	85.70 ± 1.95	99.80 ± 4.02
Large weight bounds	$(0.4, 0.8)$	$(0.5, 2)$	6.00 ± 8.28	$\mathbf{3.30 \pm 4.82}$	108.90 ± 27.25	9.30 ± 11.26	38.00 ± 12.54	58.30 ± 21.65	65.60 ± 27.25	84.80 ± 2.23	91.40 ± 14.30
Edges / Vertices	2	3	$\mathbf{0.00 \pm 0.00}$	6.10 ± 3.70	116.80 ± 25.48	11.30 ± 4.20	39.60 ± 7.89	86.80 ± 20.39	108.40 ± 21.32	94.60 ± 1.80	84.80 ± 13.67
Standardization	No	Yes	83.80 ± 4.47	71.50 ± 3.50	243.80 ± 36.15	85.90 ± 7.71	101.20 ± 10.02	$\mathbf{38.80 \pm 17.49}$	61.10 ± 20.71	82.30 ± 1.79	77.80 ± 6.78
Transitive closure	Floyd-Warshall	Zero	48.00 ± 13.47	41.00 ± 9.22	186.50 ± 26.10	$\mathbf{26.70 \pm 4.71}$	67.00 ± 16.73	81.10 ± 11.85	132.40 ± 19.66	88.00 ± 3.41	77.50 ± 5.48
Noise Distribution	Gaussian	Gumbel	$\mathbf{0.00 \pm 0.00}$	5.90 ± 3.36	85.20 ± 17.76	5.90 ± 2.62	35.90 ± 12.23	44.60 ± 12.22	61.50 ± 22.07	81.80 ± 0.98	79.90 ± 14.14
Spectral Noise	$\boldsymbol{N}_f = 0,\ \boldsymbol{N}_s \neq 0$	$\boldsymbol{N}_f \neq 0,\ \boldsymbol{N}_s = 0$	$\mathbf{0.00 \pm 0.00}$	4.80 ± 2.27	96.90 ± 17.13	5.30 ± 2.05	39.20 ± 9.25	40.50 ± 16.81	56.70 ± 26.72	82.00 ± 1.41	77.60 ± 10.96
Samples	400	20	259.50 ± 16.95	206.70 ± 22.77	error	128.90 ± 13.98	371.50 ± 25.48	265.40 ± 27.62	error	$\mathbf{86.90 \pm 1.38}$	time-out
Fixed support	No	Yes	99.60 ± 5.85	151.10 ± 22.13	184.00 ± 46.03	$\mathbf{82.00 \pm 5.40}$	300.40 ± 48.12	84.10 ± 10.90	96.60 ± 14.44	91.40 ± 3.80	88.70 ± 8.86

Table 3: SHD metric for learning graph with 40 nodes. The blue column shows the default value of the corresponding parameter and the next column the current choice for the experiment, while all other parameters are set to default. Best results are marked in bold.

Hyperparameter	Values		Avg. Varsortability
	Default	Current	
Default settings			0.98
Graph Type	Erdös-Renyi	Scale-Free	0.92
Signal Noise	$\sigma = 0.01$	$\sigma = 0.1$	0.93
Low Sparsity	30%	60%	0.95
High Sparsity	30%	10%	0.95
Small weight bounds	$(0.4, 0.8)$	$(0.2, 0.4)$	0.90
Large weight bounds	$(0.4, 0.8)$	$(0.5, 2)$	0.95
Edges / Vertices	2	3	0.96
Standardization	No	Yes	0.50
Transitive closure	Floyd-Warshall	Zero	0.90
Noise Distribution	Gaussian	Gumbel	0.96
Spectral Noise	$\boldsymbol{N}_f = 0,\ \boldsymbol{N}_s \neq 0$	$\boldsymbol{N}_f \neq 0,\ \boldsymbol{N}_s = 0$	0.96
Samples	400	20	0.93
Fixed support	No	Yes	0.66

Table 4: Average Varsortability measurement for different experimental scenarios.

Varsortability. Moreover, in the table above we compute the varsortability in each different experimental setting. We notice that we have high varsortability in general. However, our measurements are in general lower than the measurements reported in (Reisach et al., 2021) (Appendix G.1) for linear SEMs. In some cases is around 5% lower than those of linear SEMs, which is a sign that in our experimental setting it is not trivial to learn the DAG. This can also be deduced by the performance of sortnregress, which has decent performance but is not achieving the best results. It is also worth mentioning that the fixed sparsity support case, seems like the most interesting one as all the methods fail in this case and varsortability is very low. Therefore, there is at least one scenario where our model successfully proposes a hard problem for DAG learning.