Zero-padding on Connected Directed Acyclic Graphs for Spectral Processing

Ljubiša Stanković
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
ljubisa@ucg.ac.me

Miloš Daković
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
milos@ucg.ac.me

Miloš Brajović
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
milosb@ucg.ac.me

Isidora Stanković
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
isidoras@ucg.ac.me

Ali Bagheri Bardi
Faculty of Intelligent Systems Engineering and Data Science
Persian Gulf University
Busher, Iran
bagheri@pgu.ac.ir

***Abstract*—Directed acyclic graphs (DAGs) are often utilized for modeling causal relationships, dependencies, and flows in various systems. However, spectral analysis becomes impractical in this setting because the eigendecomposition of the adjacency matrix yields all eigenvalues equal to zero. This inherent property of DAGs results in an inability to differentiate between frequency components of signals on such graphs. To address this limitation, we propose, in this paper, a zero-padding approach for connected DAGs. This approach involves augmenting the original connected DAG with additional nodes that are connected to the existing structure. These added nodes are characterized by signal values set to zero and does not interfere with the original DAG except at the source and sink. The proposed technique enables the spectral evaluation of system outputs on DAGs, that is the computation of vertex-domain convolution (output of a graph filter) in without the adverse effects of aliasing due to the added nodes (added nodes do not change the original output of a filter on a DAG).**

***Index Terms*—DAG, Graphs, Graph signal processing, Graph Fourier transform, Graph filtering**

I. Introduction

Graph signal processing is an emerging research area at the intersection between signal processing and graph theory, dealing with analysis, processing and interpretation of data defined on graphs [1]–[8]. In many real-world applications, data structure can be represented with graphs, with edges representing relationships or interactions [6], [8].

One particular type of graph that is of great significance in various domains is the Directed Acyclic Graph (DAG) [9]–[14]. A DAG is a special type of graph where edges have a direction. Each edge points from one node to another, and there are no cycles. DAGs are commonly used to model causal relationships, dependencies, and flows in various systems [9]. In this context, DAGs are used to model and analyze dynamic systems where information or signals flow from one node to another, causing a cascade of effects. This is particularly relevant in fields like epidemiology, finance, neuroscience, and machine learning, where understanding the causal relationships between variables or events can lead to better predictions and interventions [9], [13].

Spectral analysis and processing based on the standard Graph Fourier Transform (GFT) is impossible for signals on DAGs, since all eigenvalues of the adjacency matrix are equal to zero, meaning that the frequency components of the analyzed signal cannot be distinguished [9]. In this paper, we present a new concept of zero padding, which can be implemented by adding nodes connected to the existing graph structure, with the corresponding signal values equal to zero. This leads to the possibility to calculate the output of the graph system (filtering based on convolution), without the aliasing (changing the original DAG output).

The paper is organized as follows. In Section II, basic theory regarding graphs and signals on graphs is presented. Next, in Section III we present some fundamental properties of DAGs. Systems on graphs are introduced in Section IV, while the new zero-padding for DAGs is presented in Section V. Section VI contains a numerical example, whereas Section VII concludes the paper.

II. Graph Fourier Transform

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a finite set $\mathcal{V}$ of N nodes (vertices), and a finite set $\mathcal{E}$ of edges, connecting the nodes, and reflecting their mutual relations. In the case of undirected graphs, the relation between two nodes $m, n \in \mathcal{V}$ is mutual, whereas in the case of directed graphs, the connectivity between the two nodes m and n is one-sided. Each edge $(m, n) \in \mathcal{E}$ can be characterized by its weight w_{mn}, with values between 0 and 1, therefore forming the weight matrix $\mathbf{W}$ to model the graph, that is, the node interconnections. It is very common that the edges are unweighted. In this case, the graph is modeled by the adjacency matrix $\mathbf{A}$. The elements a_{mn} of this matrix have value 1 if an edge between m and n exists, and value 0 if such edge does not exist.

This research is supported by Montenegrin Academy of Sciences and Arts (MASA – CANU).

For graph spectral analysis, the eigendecomposition of the real-valued adjacency matrix, $\mathbf{A}$, of a graph is considered,

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}, \tag{1}$$

leading to the matrix $\mathbf{U}$ consisting of the eigenvectors $\mathbf{u}_k$, $k = 1, 2, \ldots, N$, corresponding to the eigenvalues, λ_k, from the diagonal matrix $\mathbf{\Lambda}$. In general, the adjacency matrix of directed graphs is not symmetric. Therefore, the eigenvectors and eigenvalues are complex-valued. The eigenvectors are commonly nonorthogonal.

The GFT of a signal, $x(n)$, on nodes $n = 1, 2, \ldots, N$, is defined as

$$X(k) = \sum_{n=1}^{N} x(n) v_k(n). \tag{2}$$

Here, $v_k(n)$ represents the value of the kth column of matrix $\mathbf{U}^{-1}$, corresponding to a node n. The matrix form of the GFT reads $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$. The inverse graph Fourier transform (IGFT) is defined by

$$x(n) = \sum_{k=1}^{N} X(k) u_k(n), \tag{3}$$

or, in a matrix form $\mathbf{x} = \mathbf{U}\mathbf{X}$. Here, $u_k(n)$ is the value of the kth eigenvector (column of matrix $\mathbf{U}$), corresponding to a node n.

III. Directed Acyclic Graphs

Directed acyclic graph (DAG) is a special case of directed graph without cycles. This graph have specific nodes, named *sources* (without inbound edges), and *sinks* (without outbound edges). If DAG is connected it is easy to prove the next statements:

- Connected DAG has only one source node.
- Connected DAG has only one sink node.
- There exists a Hamiltonian path (path visiting all graph nodes) from source to sink.
- Nodes in the connected DAG of size N can be numbered in an unique way, form 1 (sink) to N (source) so that for every edge connecting node i with node j holds $i < j$. This numbering directly follows from the Hamiltonian path in the connected DAG.
- Adjacency matrix of connected DAG, with a proper node numbering, is an upper triangular matrix with ones on the super-diagonal (the diagonal above the main diagonal) and arbitrary values (0 or 1) above super-diagonal.
- All eigenvalues of the adjacency matrix of DAG are equal to 0.

A *path graph* is the special case of connected DAG. In the sequel we will assume that DAG is connected.

The GFT in (2) for directed graphs follows from eigendecomposition in (1) of the graph adjacency matrix. The eigenvectors determine basis functions while the eigenvalues determine the generalized frequency. Since all eigenvalues of DAG are equal to 0,

$$\lambda_k = 0, \quad k = 0, 1, \ldots, N-1,$$

frequency analysis is not possible, because we cannot distinguish frequency components of the analyzed signal.

It is well known that the classical Fourier analysis follows from a circular graph. This graph can be considered as a path graph with one additional edge connecting the sink to the source. This graph has adjacency matrix with ones at super-diagonal, and 1 at lower left corner. Eigenvalues of such matrix (of size N) are obtained as solutions of

$$\lambda^N = 1,$$

and they are equal to

$$\lambda_k = e^{j\frac{2\pi}{N}k}, \quad k = 0, 1, \ldots, N-1,$$

where k is the frequency index. For $0 \leq k < \frac{N}{2}$, we talk about "positive" frequencies, while indices $\frac{N}{2} \leq k < N$ are associated to "negative" frequencies.

IV. System on a graph

A system on a graph is defined by

$$\mathbf{y} = h_0\mathbf{x} + h_1\mathbf{A}\mathbf{x} + \cdots + h_{P-1}\mathbf{A}^{P-1}\mathbf{x}, \tag{4}$$

with $\mathbf{x}$ acting as the input graph signal, and $h_0, h_1, \ldots, h_{P-1}$ as system coefficients. System order is P, and it can be easily shown that $P < N$. The spectral domain representation is obtained using (1) by left multiplying the vertex domain relation (4) by $\mathbf{U}^{-1}$, producing

$$\mathbf{Y} = h_0\mathbf{X} + h_1\mathbf{\Lambda}\mathbf{X} + \cdots + h_{P-1}\mathbf{\Lambda}^{P-1}\mathbf{X}, \tag{5}$$

with $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$ and $\mathbf{Y} = \mathbf{U}^{-1}\mathbf{y}$ being the GFTs of the graph signals $\mathbf{x}$ and $\mathbf{y}$, respectively. The spectral domain representation of the considered system on a graph reads

$$\mathbf{Y} = H(\mathbf{\Lambda})\mathbf{X} = (h_0 + h_1\mathbf{\Lambda} + \cdots + h_{P-1}\mathbf{\Lambda}^{P-1})\mathbf{X}, \tag{6}$$

where $H(\mathbf{\Lambda})$ is a diagonal matrix representing the transfer function of the system.

The spectral domain tools based on the standard GFT cannot be used directly for signals on DAGs, since all eigenvalues are zero. This problem can be avoided by the proposed zero-padding procedure.

V. Graph zero-padding

Zero-padding is a well known and commonly used tool for avoiding aliasing. In classical signal analysis, signal length is extended by adding zero valued samples. This further permits to calculate the convolution in the spectral domain (with circular extension) of two discrete signals without aliasing.

In graph signal processing, the zero-padding approach requires additional nodes, on which the graph signal is zero valued. Here, we are focused on connected DAGs, with proper node numbering. Therefore, we have a path starting from the first node and ending at the last node. The zero-padding is implemented by adding M nodes connected in a path, where the last node of DAG is connected to the first node of the added path and last node of the added path is connected to the first node of DAG (see Fig. 1).

Adjacency matrix of zero-padded graph can be presented in a block form as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{D} & \mathbf{J} \end{bmatrix}, \tag{7}$$

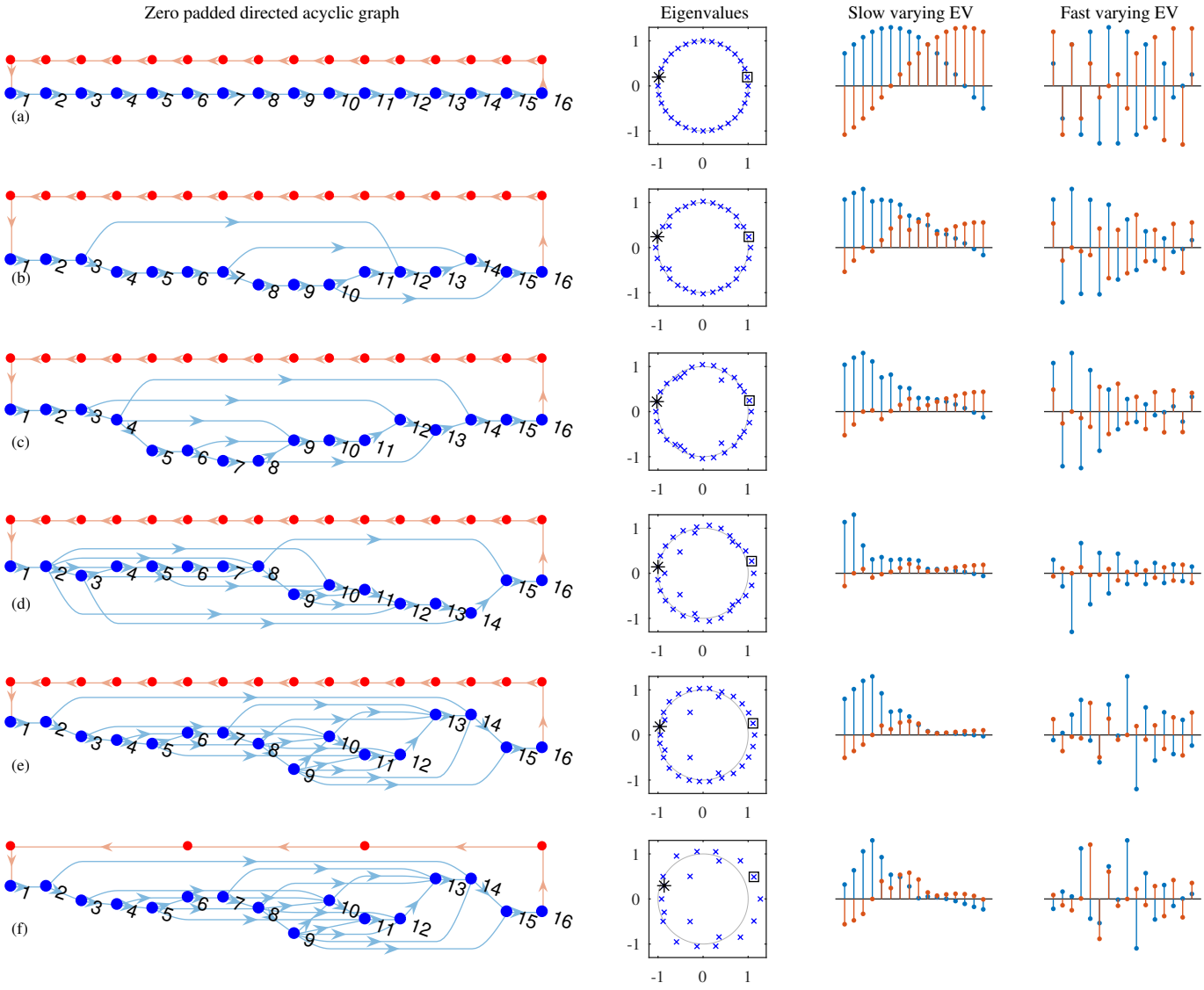

Fig. 1. Examples of zero padded connected DAG. Simple path graph (a); connected DAG with 3, 5, 10 and 15 additional edges compared to path graph (b-e); same DAG as in (e) with $M = 4$. Original DAGs are presented with blue nodes and edges. Nodes and edges added by zero-padding are in red color. In graphs (a-e) zero-padding with $M = N = 16$ is used, while system order $M = 4$ is used for case (f). For each zero-padded graph eigenvalues are presented. Two eigenvectors that corresponds to low and high frequency eigenvalues are also presented (real parts in blue and imaginary parts in red color). Corresponding eigenvalues are marked with square (low frequency) and star (high frequency) on the eigenvalues plot.

where $\mathbf{A}$ is $N \times N$ adjacency matrix of DAG, $\mathbf{J}$ is matrix of size $M \times M$, with ones at superdiagonal and zeros elsewhere. Matrix $\mathbf{C}$ is of size $N \times M$, with all zeros except lower left corner where it has value 1. This block represent connection of DAG last node to first node of added path. Matrix $\mathbf{D}$ is of size $M \times N$ with all zeros except lower left corner where it has value 1. This block represents edge form last node of added path to the first node of DAG.

To illustrate the concept, consider a DAG with six nodes, whose adjacency matrix is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

For zero-padding with $M = 3$ additional nodes, the matrix $\mathbf{Z}$ has the following form

$$\mathbf{Z} = \left[\begin{array}{cccccc|ccc} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

A system on a DAG can be analyzed in spectral domain of the corresponding zero-padded graph. In order to avoid aliasing, and obtain the same results using vertex domain system and the system in the spectral domain, the number of nodes added in zero-padding process should be higher than

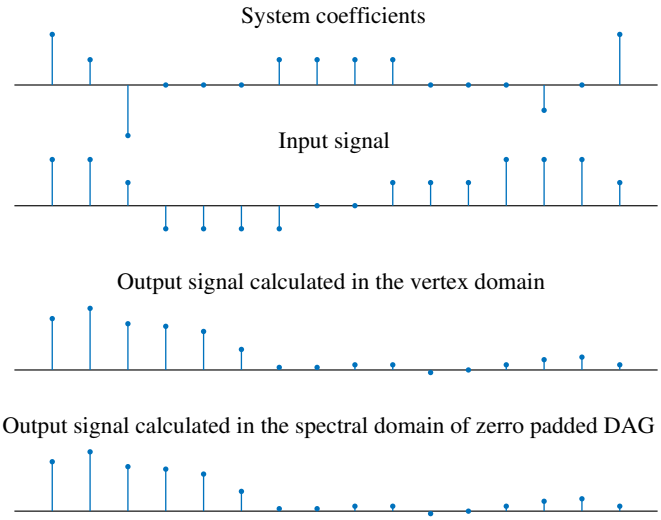

Fig. 2. System on the graph presented in Fig 1(c): system coefficients h_n (first), input signal $x(n)$ (second), output calculated in the vertex domain (third) and in the spectral domain of zero-padded graph (fourth).

or equal to the system order P. Since $P < N$, a safe choice is to use $M = N$ nodes for zero-padding.

VI. Examples

Five examples od zero-padded connected DAG are presented in Fig. 1. The first case is the most simple one, i.e. the simple path graph. This graph transforms into a cycle graph after the zero-padding. It is well known that the classical Fourier analysis follows from the eigendecomposition of acyclic graph adjacency matrix. The eigenvalues lie equally spaced on the unit circle.

In more complex DAG examples (Fig. 1 (b)-(f)), we add more edges to the path graph in order to keep connectivity and avoid cycles. Here, we present graphs with 3, 5, 10 and 15 additional edges. In all presented cases, we can see that the eigenvalues lie close to the unit circle, with uneven, but almost equal spacing, i.e. eigenvalues are very similar to the eigenvalues of the path graph.

Zero-padding with $M = N = 16$ is used for graphs presented in Fig. 1 (a)-(e), while $M = 4$ is used for graph presented in Fig. 1 (f). Using zero-padding with $M < N$ guaranties equality of output calculated in vertex domain and output calculated using spectral domain for systems of order up to M.

For each graph, two eigenvectors are presented in Fig. 1 (last two columns). Since eigenvectors are complex, we present real part in blue and imaginary part in red color. We chose one slow-varying eigenvector that corresponds to low frequency eigenvalue (marked by a square in the eigenvalue visualization in Fig. 1) and one fast varying eigenvector with the corresponding eigenvalue marked by a star. From these figures, we can observe that the concepts of slow-varying and fast-varying terms, inherently obvious in the classical Fourier domain (path graph), can be straight-forwardly generalized to more complex graphs.

Next, we consider a system on the graph from Fig 1(c), with the corresponding system coefficients h_n presented in Fig. 2. The input signal is presented in the same figure, as well as output signal calculated in the vertex domain using the DAG adjacency matrix, and in the spectral domain of zero-padded DAG. The same output signal is obtained in both cases.

Vertex domain calculation requires N matrix multiplications, resulting in $O(N^4)$ calculation complexity, while spectral domain calculation needs only three matrix-vector multiplications resulting in $O(N^2)$ calculation complexity.

VII. Conclusion

In this paper, we presented a zero-padding approach designed to overcome the inherent limitations associated with signal analysis and processing on DAGs, particularly the constraints in utilizing the GFT. Special case of connected DAG is considered, however this approach could be extended to arbitrary DAG. The proposed method facilitates the application of GFT-based spectral analysis and processing, while effectively alleviating the undesired aliasing effects in the vertex domain during the assessment of system outputs on a DAG.

References

[1] G. Leus, A.G. Marques, J. M. Moura, A. Ortega, and D. I. Shuman, "Graph Signal Processing: History, development, impact, and outlook." *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 49-60, 2023.

[2] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 128–138, 2020.

[3] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[4] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, pp. 83-98, 2013.

[5] LJ. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo-Dees, and A.G. Constantinides, "Data Analytics on Graphs - Part I: Graphs and Spectra on Graphs," *Foundations and Trends in Machine Learning*, Vol. 13: No. 1, 2020, pp 1-157. http://dx.doi.org/10.1561/2200000078-1

[6] LJ. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo-Dees, and A.G. Constantinides, "Data Analytics on Graphs Part II: Signals on Graphs, "*Foundations and Trends in Machine Learning*, Vol. 13: No. 2-3, 2020, pp. 158-331. http://dx.doi.org/10.1561/2200000078-2

[7] LJ. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo-Dees, S. Li, and A.G. Constantinides, "Data Analytics on Graphs - Part III: Machine Learning on Graphs, from Graph Topology to Applications,"*Foundations and Trends in Machine Learning*, Vol. 13: No. 4, 2020, pp. 332-530. http://dx.doi.org/10.1561/2200000078-3.

[8] L. Stankovic, D. Mandic, M. Dakovic, I. Kisil, E. Sejdic, and A. Constantinides, "Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach," *IEEE Signal Processing Magazine*, Vol. 36, No. 6, pp.133-145, 2019.

[9] B. Seifert, C. Wendler, M. Püschel, "Causal Fourier Analysis on Directed Acyclic Graphs and Posets", 10.48550/arXiv.2209.07970.

[10] T. Pegolotti, B. Seifert, M. Püschel, "Fast Möbius and Zeta Transforms," arXiv:2211.13706

[11] V. Mihal and M. Püschel, "Möbius Total Variation for Directed Acyclic Graphs," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095435.

[12] B. Seifert, C. Wendler, M. Püschel, "Learning Fourier-Sparse Functions on DAGs, ", *ICLR 2022 workshop on Objects, Structure and Causality*, 2022.

[13] S. Park and J. Kim, "DAG-GCN: Directed Acyclic Causal Graph Discovery from Real World Data using Graph Convolutional Networks," *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, Republic of Korea, 2023, pp. 318-319, doi: 10.1109/BigComp57234.2023.00065.

[14] S. Chen, A. Sandryhaila, J. M. Moura, J. Kovacevic, Signal denoising on graphs via graph filtering. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 872-876, Dec. 2014.