



Delft University of Technology

A Generalization of the Convolution Theorem and its Connections to Non-Stationarity and the Graph Frequency Domain

Natali, Alberto; Leus, Geert

DOI

[10.1109/TSP.2024.3423432](https://doi.org/10.1109/TSP.2024.3423432)

Publication date

2024

Document Version

Final published version

Published in

IEEE Transactions on Signal Processing

Citation (APA)

Natali, A., & Leus, G. (2024). A Generalization of the Convolution Theorem and its Connections to Non-Stationarity and the Graph Frequency Domain. *IEEE Transactions on Signal Processing*, 72, 3424-3437. <https://doi.org/10.1109/TSP.2024.3423432>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

A Generalization of the Convolution Theorem and Its Connections to Non-Stationarity and the Graph Frequency Domain

Alberto Natali , *Student Member, IEEE* and Geert Leus , *Fellow, IEEE*

Abstract—In this paper, we present a novel convolution theorem which encompasses the well known convolution theorem in (graph) signal processing as well as the one related to time-varying filters. Specifically, we show how a node-wise convolution for signals supported on a graph can be expressed as another node-wise convolution in a frequency domain graph, different from the original graph. This is achieved through a parameterization of the filter coefficients following a basis expansion model. After showing how the presented theorem is consistent with the already existing body of literature, we discuss its implications in terms of non-stationarity. Finally, we propose a data-driven algorithm based on subspace fitting to learn the frequency domain graph, which is then corroborated by experimental results on synthetic and real data.

Index Terms—Graph signal processing, convolution, non-stationarity, frequency domain.

I. INTRODUCTION

CONVOLUTION is the core operation in signal processing and machine learning systems. Its use is at the heart of digital filters [2] for audio applications and time series prediction [3], as well as for convolutional neural networks in deep learning [4], enabling scalable architectures and endowing a notion of locality among samples, properties exploited in, e.g., object recognition [5].

The three key operations defining a convolution are the *shift*, the *scale* and the *sum*. The shift is responsible to capture the underlying signal domain and brings the notion of locality and proximity among samples: in time, for instance, successive

applications of the shift (in that case corresponding to a delay) give previous time samples. The scale defines how the shifted samples are weighted before summing them, and different weighting schemes lead to different structural properties (such as invariants) of the architecture implementing the convolution. The notion of regularity in time and in space, which are two very well structured domains, is reflected in the definition of their frequency domain. Specifically, a signal in these domains can be decomposed into elementary building blocks (such as sine waves) which endow a physical interpretation with a well understood meaning of variability. In a less structured domain modeled by a graph, this definition is not tight and multiple interpretations are possible.

Graph signal processing (GSP) [6] extends the convolution principle to data residing on graphs by means of graph filters (GFs) [7], [8], [9], [10], architectures which are parametric on the mathematical structure defining the *shift* operation. While in temporal signal processing this shift operator is mathematically represented by the (lower) shift/delay matrix, in GSP the *graph shift operator* (GSO) depends on the underlying network domain. The eigendecomposition of the GSO reveals its respective frequencies: specifically, the eigenvalues of the shift are the frequencies. In temporal signal processing these frequencies are the well-known complex roots of unity which obey a natural ordering, and the associated (normalized) eigenvectors are the Fourier modes (remember that the delay matrix is a particular case of a circulant matrix). In GSP, however, different shifts have different eigenvalues and hence different frequencies. In this case, an ordering purely based on their numeric value might not be meaningful and a more structured frequency domain, for instance captured by a graph, might convey more information. This is the recent line of work explored in [11], [12], which we exploit here.

By relying on the novel notion of *dual graph* [11], which models the support of the frequency domain as a graph, in this work we introduce a new convolution theorem which generalizes the (graph) convolution theorem and the one related to time-varying filters. To do this, we adopt so-called node-varying graph filters (NV-GFs) [7] and we show how a node-varying convolution in one domain (captured by the *primal* graph) can be expressed as a node-variant convolution in the other domain (captured by the *dual* graph), while remaining consistent with the pre-existing body of literature. Based on the

Manuscript received 27 December 2023; revised 2 May 2024; accepted 12 June 2024. Date of publication 4 July 2024; date of current version 5 August 2024. This work was supported by GraSPA project (project 19497 within the TTW OTP programme), which was financed by Netherlands Organization for Scientific Research (NWO). An earlier version of this paper was presented at the 2022 56th Asilomar Conference on Signals, Systems, and Computers, 2022 [DOI: 10.1109/IEEECONF56349.2022.10052018]. The associate editor coordinating the review of this article and approving it for publication was Prof. Ivan Dokmanić. (*Corresponding author: Alberto Natali.*)

The authors are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: a.natali@tudelft.nl; g.j.t.leus@tudelft.nl).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSP.2024.3423432>, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3423432

proposed theorem, we outline models for non-stationary graph signals. Finally, we propose an algorithmic approach to learn the dual graph from data with a subspace fitting approach [13] resorting to sequential convex programming [14] to tackle the non-convexity of the problem. The validity of this approach is finally corroborated by simulations on synthetic and real data.

Remark 1: Differently from temporal signal processing, where the shift operator remains consistent in time and frequency domains, such uniformity does not typically hold in graph-based signal processing and the notion of shift becomes inherently more intricate. We believe that to construct a principled and elegant framework for addressing shift-variance and non-stationarity within a graph context, employing the concept of the dual graph is essential. This will be more evident in the following sections.

A. Related Works

Although this work is novel in its genre, many of the concepts it relies on have been recently introduced. Modeling the frequency domain through a graph has been proposed in [11], [12], and similar efforts to interpret such a domain differently from ordering the eigenvalues of the GSO is given by means of embeddings in [15], [16]. Node-variant graph filters as a way to extend classical time-varying filters to the graph setting have been introduced in [7], while graph signal stationarity arguments and their implications have been studied in [17], [18]. Non-stationarity of random graph signals has been exploited in [19] in the context of network topology identification.

B. Contributions

We summarize the specific contributions of this work as follows:

- 1) We propose a convolution theorem which encompasses the (graph) convolution theorem and the one related to time-varying filters; we show how the latter are specific instantiations of the proposed theorem for particular choices of the GSO and the scaling scheme;
- 2) We introduce novel non-stationary graph signal models;
- 3) We devise an algorithmic procedure to learn the dual graph from input-output and output-only data. The problem formulation can be cast as a blind polynomial regression, as such also applicable to graph-agnostic tasks, such as polynomial interpolations and jitter correction in communication applications. The solution approach relies on a subspace fitting method and it is accompanied by a theoretical study of the ambiguities present in the problem.
- 4) We showcase the validity of our findings on synthetic and real data with numerical simulations.

II. PRELIMINARIES

Graphs and Graph Signals. We consider data residing on a non-Euclidean domain, which we formally model by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes (or vertices), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and \mathbf{S} is an $N \times N$

matrix that represents the graph structure. The matrix \mathbf{S} is called the graph shift operator (GSO), since it plays a role akin to the delay operator in temporal signal processing. Specifically, its entries $[\mathbf{S}]_{ij} \in \mathbb{C}$ for $i \neq j$ are different from zero only if nodes i and j are connected by an edge; typical examples of such a matrix are the (weighted) adjacency matrix \mathbf{W} [20] and the graph Laplacian \mathbf{L} [6]. A graph signal is then the vector $\mathbf{x} \in \mathbb{C}^N$, where $x_i : \mathcal{V} \rightarrow \mathbb{C}$ is the value collected at node i .

In this manuscript, for the sake of simplicity, we consider the shift operator \mathbf{S} to admit an eigenvalue decomposition (EVD) written as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, with \mathbf{V} an invertible matrix collecting the eigenvectors and $\mathbf{\Lambda} = \text{Diag}(\boldsymbol{\lambda})$ a diagonal matrix collecting the eigenvalues $\boldsymbol{\lambda}$ of \mathbf{S} . A fundamental assumption in GSP is that the matrix \mathbf{V} provides a basis for expressing signals living on \mathbf{S} , and with favorable discrete Fourier transform (DFT)-like properties providing a notion of frequency similar to the one in temporal signal processing. For this reason, the matrix \mathbf{V}^{-1} is often referred to as the graph Fourier transform (GFT) and the projection of \mathbf{x} onto this basis, i.e., $\hat{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$ as the GFT signal.

Filtering on Graphs. Given a graph \mathbf{S} , a classical graph filter (C-GF) of order $L - 1$ is the matrix polynomial:

$$\mathbf{H}(\mathbf{p}, \mathbf{S}) = \sum_{l=0}^{L-1} p_l \mathbf{S}^l, \quad (1)$$

where $\mathbf{p} = [p_0, \dots, p_{L-1}]^\top \in \mathbb{C}^L$ collects the graph filter coefficients (taps). The application of the filter $\mathbf{H}(\mathbf{p}, \mathbf{S})$ on a signal \mathbf{x} to obtain a new signal \mathbf{y} , i.e., $\mathbf{y} = \mathbf{H}(\mathbf{p}, \mathbf{S})\mathbf{x}$, is often referred to as graph filtering or *graph convolution*, as it respects the scale-sum-shift principle of convolution. With a few simple calculations, it is easy to show that in the (graph) frequency domain, a graph convolution is expressed as a pointwise multiplication; this is the (graph) convolution theorem, which can be expressed as follows:

$$\mathbf{y} = \sum_{l=0}^{L-1} p_l \mathbf{S}^l \mathbf{x} \iff \hat{\mathbf{y}} = \sum_{l=0}^{L-1} p_l \mathbf{\Lambda}^l \hat{\mathbf{x}} \quad (2)$$

with $\hat{\mathbf{y}} = \mathbf{V}^{-1}\mathbf{y}$ the GFT of \mathbf{y} . Notice that such a filter is isotropic, meaning that for each $l = 0, \dots, L - 1$, the filter coefficient p_l is shared among all the nodes of the shifted signal $\mathbf{S}^l \mathbf{x}$; for this reason a C-GF is an example of a *node-invariant* graph filter.

A more versatile and flexible version of (1) is the so-called *node-variant* graph filter [7], which allows a per-node weighting scheme of each shifted version of the input signal. Due to its relevance in this work, we distinguish among two flavours of a NV-GF, henceforth referred to as *type-I* and *type-II*, defined, for a given graph \mathbf{S} and fixed order $L - 1$, respectively as:

$$\mathbf{H}_I(\mathbf{P}, \mathbf{S}) = \sum_{l=0}^{L-1} \text{Diag}(\mathbf{p}_l) \mathbf{S}^l, \quad (3)$$

$$\mathbf{H}_{II}(\mathbf{P}, \mathbf{S}) = \sum_{l=0}^{L-1} \mathbf{S}^l \text{Diag}(\mathbf{p}_l), \quad (4)$$

where $\mathbf{P} \in \mathbb{C}^{N \times L}$ collects the filter coefficients $\mathbf{P} = [\mathbf{p}_0, \dots, \mathbf{p}_{L-1}]$ with $\mathbf{p}_l := [p_{l1}, \dots, p_{lN}]^T \in \mathbb{C}^N$ the l -th hop filter tap vector. As a short-hand notation, we will use \mathbf{H}_I and \mathbf{H}_{II} to refer to the NV-GF in (3) and (4), respectively; when convenient for clarity of exposition, we will explicitly write $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ or $\mathbf{H}_{II}(\mathbf{P}, \mathbf{S})$ concordantly. The application of a NV-GF on a signal \mathbf{x} to obtain a new signal \mathbf{y} will be referred to as *node-variant graph convolution*. From a theoretical point of view, both NV-GF types have the same expressive behavior, yet the order of shifting and weighing is reversed. Specifically, in type-I, each node performs a linear combination of the (shifted) signal values of neighboring nodes, where the weights of the linear combination are neighbor-specific; in type-II, each node performs a linear combination of the (shifted) signal values of neighboring nodes, which have been already scaled by such nodes. Nonetheless, both can be implemented with the same complexity and in a distributed manner [7].

Dual Graph. Although often not explicitly stated in the academic literature, the support of the GFT signal $\hat{\mathbf{x}}$ is described by the eigenvalues $\boldsymbol{\lambda}$ of \mathbf{S} , which correspond to a discretization/sampling of a continuous domain, either the real line \mathbb{R} or the complex plane \mathbb{C} . This is consistent with the discrete signal processing notion of frequency domain: when \mathbf{S} represents a cycle graph, possibly capturing the time domain, its eigenvector matrix \mathbf{V}^{-1} coincides with the (normalized) DFT matrix, and its eigenvalues $\boldsymbol{\lambda}$ with the complex frequencies on the unit circle, i.e., $\boldsymbol{\lambda} = [1, e^{-j2\pi/N}, \dots, e^{-j2\pi(N-1)/N}]$.

However, a modern line of research attempts to model the (graph) frequency domain by means of a graph [11]. The motivation behind this line of research relies on the fact that classical signal processing tasks usually performed in the frequency domain, such as frequency-shifting, do not have their counterpart in GSP. Furthermore, given that a graph signal is inherently associated with a graph structure, it is desirable to establish a corresponding Fourier representation that is also inherently linked to a graph structure. This leads to the notion of a dual graph¹ $\mathbf{S}_f = \mathbf{V}_f \boldsymbol{\Lambda}_f \mathbf{V}_f^{-1}$, which represents the support for the GFT signal $\hat{\mathbf{x}}$. Because we want the GFT \mathbf{V}_f^{-1} associated to the dual graph to map $\hat{\mathbf{x}}$ back to the signal \mathbf{x} , i.e., $\mathbf{x} = \mathbf{V}_f^{-1} \hat{\mathbf{x}}$, and we know that $\mathbf{x} = \mathbf{V} \hat{\mathbf{x}}$, we must have $\mathbf{V}_f = \mathbf{V}^{-1}$.

Thus, the primal graph provides *spectral templates* for the graph frequency domain, i.e., the eigenvectors \mathbf{V}_f for the dual graph \mathbf{S}_f are known once we know those of \mathbf{S} . The only unknown is then the eigenvalue matrix $\boldsymbol{\Lambda}_f := \text{Diag}(\boldsymbol{\lambda}_f)$, which can be found, for instance, with an axiomatic or an optimization approach [11], or in a data-driven manner as we will show in Section IV. We anticipate that our criterion for estimating $\boldsymbol{\lambda}_f$ will be to express the graph filter coefficients \mathbf{P} as a low-degree polynomial in such a $\boldsymbol{\lambda}_f$. Although [12] proposes $\boldsymbol{\lambda}_f = \boldsymbol{\lambda}^*$, we do not see this as a favorable definition, since in our view it only holds when specified to the “temporal” graph; in all the other cases, especially for undirected graphs, it would imply that primal and dual eigenvalues always coincide. Such an

interpretation would be inconsistent with the desirable properties highlighted in [11, Axioms (1-3)].

III. AN ENCOMPASSING CONVOLUTION THEOREM

In this section we propose a convolution theorem which encompasses the graph convolution theorem [cf. (2)] introduced in Section II and the convolution theorem related to time-varying filters, which will be introduced later on to highlight similarities and differences. This generalization is made possible by using the node-variant graph filters (3) and (4) with an appropriate parametrization of the filter coefficients. Specifically, we show how a limited order NV-GF in the primal domain can be expressed as a limited order NV-GF in the frequency domain (henceforth referred to also as the “dual domain”). This is formally stated in the following theorem.

Theorem 1 (Node-variant convolution theorem): Consider a type-I NV-GF \mathbf{H}_I defined over the graph \mathbf{S} with filter taps $\{\mathbf{p}_l\}_{l=0}^{L-1}$, i.e., $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$, and assume that a dual graph \mathbf{S}_f with dual graph frequencies $\boldsymbol{\lambda}_f$ describing the dual domain is given. Assume also that each filter tap vector \mathbf{p}_l can be expressed as a polynomial of order $K-1$ in $\boldsymbol{\lambda}_f$. Then, there exists a set of coefficients $\{\hat{\mathbf{p}}_k\}_{k=0}^{K-1}$ for which the type-I NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ in the primal domain corresponds to a type-II NV-GF \mathbf{H}_{II} on the dual graph \mathbf{S}_f with filter taps $\{\hat{\mathbf{p}}_k\}_{k=0}^{K-1}$, i.e., $\mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f)$.

Proof: By multiplying both sides of (3) with the GFT matrix \mathbf{V}^{-1} , we have:

$$\hat{\mathbf{y}} = \mathbf{V}^{-1} \sum_{l=0}^{L-1} \text{Diag}(\mathbf{p}_l) \mathbf{S}^l \mathbf{x} = \mathbf{V}^{-1} \sum_{l=0}^{L-1} \text{Diag}(\mathbf{p}_l) \mathbf{V} \boldsymbol{\Lambda}^l \hat{\mathbf{x}}. \quad (5)$$

Next, we use a basis expansion model (BEM) [21] to express the NV filter coefficients $\{\mathbf{p}_l\}_{l=0}^{L-1}$ as a linear combination of a dual graph dependent basis. Specifically, we express each \mathbf{p}_l through powers of the dual eigenvalues $\boldsymbol{\lambda}_f$, representing our basis expansion; that is:

$$\mathbf{p}_l = \sum_{k=0}^{K-1} c_{lk} \boldsymbol{\lambda}_f^k = \boldsymbol{\Psi}_f \mathbf{c}_l \quad (6)$$

with $\boldsymbol{\Psi}_f := [\mathbf{1}, \boldsymbol{\lambda}_f, \dots, \boldsymbol{\lambda}_f^{K-1}]$ the Vandermonde matrix of dual eigenvalues and $\mathbf{c}_l := [c_{l0}, \dots, c_{l(K-1)}]^T$ the expansion coefficients for the l -th primal filter tap vector \mathbf{p}_l . With this choice, substituting (6) in (5), we have:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{V}^{-1} \sum_{l=0}^{L-1} \text{Diag} \left(\sum_{k=0}^{K-1} c_{lk} \boldsymbol{\lambda}_f^k \right) \mathbf{V} \boldsymbol{\Lambda}^l \hat{\mathbf{x}} \\ &= \sum_{k=0}^{K-1} \mathbf{V}^{-1} \text{Diag}(\boldsymbol{\lambda}_f^k) \mathbf{V} \text{Diag} \left(\sum_{l=0}^{L-1} c_{lk} \boldsymbol{\lambda}^l \right) \hat{\mathbf{x}} \\ &= \sum_{k=0}^{K-1} \mathbf{S}_f^k \text{Diag}(\hat{\mathbf{p}}_k) \hat{\mathbf{x}} \end{aligned} \quad (7)$$

where $\hat{\mathbf{p}}_k := \sum_{l=0}^{L-1} c_{lk} \boldsymbol{\lambda}^l = \boldsymbol{\Psi} \hat{\mathbf{c}}_k$ is the k -th hop filter tap vector on the *dual* graph, with $\boldsymbol{\Psi} := [\mathbf{1}, \boldsymbol{\lambda}, \dots, \boldsymbol{\lambda}^{L-1}]$ the Vandermonde matrix of primal eigenvalues, and $\hat{\mathbf{c}}_k := [c_{0k}, \dots, c_{(L-1)k}]^T$ the expansion coefficients for the k -th dual filter

¹Not to be confused with the dual graph notion in graph theory, as the graph which has a vertex for each face of the original graph.

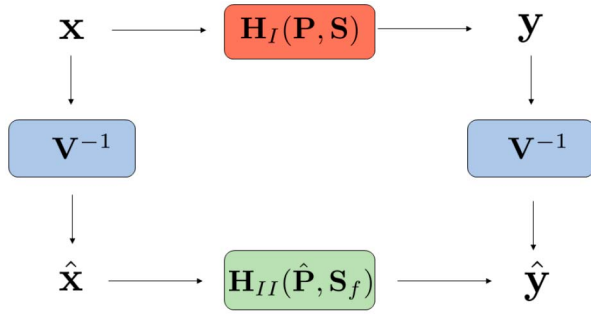


Fig. 1. General convolution theorem. A node-variant graph convolution in the primal domain is equivalent to a node-variant graph convolution in the dual domain.

tap vector $\hat{\mathbf{p}}_k$. So in the frequency domain, we also obtain a NV-GF denoted as $\mathbf{H}_{II} = \sum_{k=0}^{K-1} \mathbf{S}_f^k \text{diag}(\hat{\mathbf{p}}_k)$. Whenever the dependency on the dual filter coefficients and shift operator is necessary, we use $\mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f)$, where $\hat{\mathbf{P}}$ is the $N \times K$ matrix of coefficients $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_0, \dots, \hat{\mathbf{p}}_{K-1}]$. \square

This theorem allows us to delineate a general convolution theorem encompassing (2) as a special case, which relies on node-variant graph filtering and is pictorially described in Fig. 1, as follows:

$$\begin{aligned} \mathbf{y} &= \sum_{l=0}^{L-1} \text{Diag}(\mathbf{p}_l) \mathbf{S}^l \mathbf{x} \iff \hat{\mathbf{y}} = \sum_{k=0}^{K-1} \mathbf{S}_f^k \text{Diag}(\hat{\mathbf{p}}_k) \hat{\mathbf{x}} \quad (8) \\ \mathbf{p}_l &= \Psi_f \mathbf{c}_l \iff \hat{\mathbf{p}}_k = \Psi \hat{\mathbf{c}}_k \quad (9) \end{aligned}$$

The connection between the primal and the dual node-variant graph filters defined in (8) is given by the $K \times L$ expansion coefficients conveniently stored in the matrix $\mathbf{C} = [\mathbf{c}_0, \dots, \mathbf{c}_{L-1}] = [\hat{\mathbf{c}}_0, \dots, \hat{\mathbf{c}}_{K-1}]^\top$. This enables also to concisely express the node-variant coefficients in the primal and dual domain as $\mathbf{P} = \Psi_f \mathbf{C}$ and $\hat{\mathbf{P}} = \Psi \mathbf{C}^\top$, respectively.

Remark 2: The same type of theorem construction can be obtained by reversing the types of filters adopted in the primal and dual domain; that is, applying a type-II NV-GF in the primal domain is equivalent to applying a type-I NV-GF in the dual domain, with the graph filter coefficients following the parametrization in (9).

Corollary 1: Given a graph signal \mathbf{x} , the application of a node-variant graph filter $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ in the primal domain followed by the GFT \mathbf{V}^{-1} is equivalent to the application of the GFT followed by a node-variant graph filter $\mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f)$ in the dual domain. In other words, it holds (see also Fig. 1):

$$\mathbf{V}^{-1} \mathbf{H}_I(\mathbf{P}, \mathbf{S}) = \mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f) \mathbf{V}^{-1}. \quad (10)$$

In temporal signal processing, the frequency representation of windowing in the time domain is the convolution between the spectra of the signal and the window. Because a node-variant convolution of order $L - 1$ is nothing else than the application of L windows on shifted versions of the input graph signal \mathbf{x} , a similar result can be derived in the graph setting; the following corollary expresses this.

Corollary 2: Given an input graph signal \mathbf{x} and a type-I NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$, with each $\{\mathbf{p}_l\}_{l=0}^{L-1}$ parametrized as in (9), a node-variant graph convolution of order $L - 1$ in the primal domain is equivalent to the sum of L classical graph convolutions in the dual domain, each one with as input a (modulated) version of $\hat{\mathbf{x}}$; that is:

$$\hat{\mathbf{y}} = \sum_{l=0}^{L-1} \mathbf{H}(\mathbf{c}_l, \mathbf{S}_f)(\boldsymbol{\lambda}^l \odot \hat{\mathbf{x}}) \quad (11)$$

$$= \mathbf{H}(\mathbf{c}_0, \mathbf{S}_f) \hat{\mathbf{x}} + \dots + \mathbf{H}(\mathbf{c}_{L-1}, \mathbf{S}_f)(\boldsymbol{\lambda}^{L-1} \odot \hat{\mathbf{x}}) \quad (12)$$

Proof: From the first equality of (7), we have:

$$\begin{aligned} \hat{\mathbf{y}} &= \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} c_{lk} \mathbf{V}^{-1} \text{Diag}(\boldsymbol{\lambda}_f^k) \mathbf{V} \boldsymbol{\Lambda}^l \hat{\mathbf{x}} \\ &= \sum_{l=0}^{L-1} \left(\sum_{k=0}^{K-1} c_{lk} \mathbf{S}_f^k \right) \boldsymbol{\Lambda}^l \hat{\mathbf{x}} \\ &= \sum_{l=0}^{L-1} \mathbf{H}(\mathbf{c}_l, \mathbf{S}_f)(\boldsymbol{\lambda}^l \odot \hat{\mathbf{x}}) \end{aligned} \quad (13)$$

Notice how the filter coefficients in (13) are the expansion coefficients \mathbf{c}_l associated to the primal filter coefficients \mathbf{p}_l . \square

An important consequence of Corollary 2 is that windowing in the primal domain is equivalent to a C-GF in the dual domain; we will rely upon this when introducing non-stationary signal models in Section III-C.

A. Consistency With the Graph Convolution Theorem

Because a C-GF is a NV-GF with constant filter taps, we expect that our introduced theory encompasses the existing one. Indeed, we can formally show that the graph convolution theorem (2) falls within the introduced theory. To see this, consider $\mathbf{p}_l = p_l \mathbf{1}$, $\forall l$, i.e., the case in which each vector of filter taps \mathbf{p}_l is constant over the nodes, thus corresponding to the C-GF as in (1). The column space of \mathbf{P} is then one-dimensional, specifically spanned by the all-one vector. By construction, the first column of the Vandermonde matrix is the all-one vector. Thus, from (9), we have that \mathbf{c}_l necessarily needs to be $\mathbf{c}_l = [p_l, \mathbf{0}^\top]^\top$, i.e.,:

$$[p_1 \mathbf{1}, \dots, p_{L-1} \mathbf{1}] = \begin{bmatrix} 1 & \dots & \lambda_{0,f}^{K-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & \lambda_{f,N-1}^{(K-1)} \end{bmatrix} \begin{bmatrix} p_1 & \dots & p_{L-1} \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \quad (14)$$

meaning that only the first row $\hat{\mathbf{c}}_0$ of \mathbf{C} is different from zero. In other words, any $\boldsymbol{\lambda}_f$ can be used, as it will be zeroed-out by the matrix \mathbf{C} . As we will see later on, this also means that we cannot learn any dual graph from stationary graph signals, since any $\boldsymbol{\lambda}_f$ would suffice. From the right equations in (8)-(9), we have that:

$$\hat{\mathbf{y}} = \mathbf{S}_f^0 \text{Diag}(\hat{\mathbf{p}}_0) \hat{\mathbf{x}} = \text{Diag}(\Psi \mathbf{p}) \hat{\mathbf{x}}, \quad (15)$$

which shows how the proposed theory fits within the principle that a classical graph convolution (node-invariant GF) is a pointwise multiplication in the frequency domain.

Likewise, one can similarly show that if $\hat{\mathbf{p}}_k = \hat{p}_k \mathbf{1}$, $\forall k$, i.e., the case in which each vector of filter taps $\hat{\mathbf{p}}_k$ in the frequency domain is constant over the nodes (frequencies), then only $\mathbf{c}_0 = \hat{\mathbf{p}} := [\hat{p}_0, \dots, \hat{p}_{K-1}]^\top$ is different from zero. This leads to a pointwise multiplication (windowing) [cf. (9) left] in the primal domain:

$$\mathbf{y} = \text{Diag}(\mathbf{p}_0) \mathbf{S}^0 \mathbf{x} = \text{Diag}(\Psi_f \hat{\mathbf{p}}) \mathbf{x}, \quad (16)$$

which also complies with Corollary 2.

Another interesting relation arises when considering a NV-GF with $p_{li} = p_i$ for all l ; i.e., the case in which each node i uses the same weight p_i (possibly different from p_j of node j) for the diffused sequence $\{\mathbf{x}, \mathbf{S}\mathbf{x}, \dots, \mathbf{S}^{L-1}\mathbf{x}\}$. In this case, multiple λ_f and \mathbf{C} satisfy the theorem; for instance we can choose $\lambda_f = [p_1, \dots, p_N]^\top$ and $\mathbf{c}_l = [0, 1, \mathbf{0}^\top]^\top$, i.e.,:

$$\begin{bmatrix} p_1 \mathbf{1}^\top \\ \vdots \\ p_N \mathbf{1}^\top \end{bmatrix} = \begin{bmatrix} 1 & p_1 & \cdots & p_1^{K-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p_N & \cdots & p_N^{K-1} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (17)$$

This implies that:

$$\hat{\mathbf{y}} = \mathbf{S}_f \text{Diag}(\hat{\mathbf{p}}_1) \hat{\mathbf{x}} = \mathbf{S}_f \text{Diag}(\Psi \mathbf{1}) \hat{\mathbf{x}}, \quad (18)$$

so that each \hat{x}_i is multiplied with $\hat{p}_{1i} = [1, \lambda_i, \dots, \lambda_i^{L-1}] \mathbf{1}$.

B. Relationship With Time-Varying Channel Propagation

The proposed theory also generalizes, to the graph setting, concepts which are familiar in the context of time-varying channel propagation [22], arising for instance in mobile communication scenarios. In that case, the received signal y at time n , i.e., $y[n]$, is modeled as²:

$$y[n] = \sum_{l=0}^{L-1} p[n, l] x[n-l], \quad (19)$$

where $p[n, l]$ denotes the channel impulse response of the l -th path at the n -th time instant, and $x[n-l]$ is the transmitted signal at the $(n-l)$ -th time instant. The gains associated to the different paths are assumed to be time-varying and approximated by a basis expansion model [21]; specifically:

$$\mathbf{p}_l = \sum_{k=0}^{K-1} c_{lk} \mathbf{b}_k, \quad (20)$$

where $\mathbf{p}_l = [p[0, l], \dots, p[N-1, l]]^\top$ stores the evolution of the filter impulse response over the N time instants, $\mathbf{b}_k \in \mathbb{R}^N$ is the k -th basis function, and c_{lk} is the coefficient associated to the l -th path and the k -th basis function. This alleviates the effort of having to deal with NL channel coefficients (usually a very high number), and converts the model into a simpler one with only LK BEM coefficients.

²We use square brackets to indicate that the argument is a time index and not a graph node.

It is easy to show that we can write (19) in matrix-vector form, by taking into account (20), as:

$$\mathbf{y} = \sum_{k=0}^{K-1} \text{Diag}(\mathbf{b}_k) \left(\sum_{l=0}^{L-1} c_{lk} \mathbf{D}^l \right) \mathbf{x} \quad (21)$$

where $\mathbf{x} = [x[0], \dots, x[N-1]]^\top$ and \mathbf{D} is the $N \times N$ lower delay matrix; notice how the matrix $\sum_{l=0}^{L-1} c_{lk} \mathbf{D}^l$ implements a standard convolutional filter in time and observe its similarities with the left equation in (8). Next, denote with $\mathbf{F} \in \mathbb{C}^{N \times N}$ the normalized DFT matrix, and with \mathbf{f}_k its k th column; the classical complex exponential BEM uses the Fourier basis as basis functions in (20), i.e., $\mathbf{b}_k = \sqrt{N} \mathbf{f}_k$. As such the gains \mathbf{p}_l associated to the l -th path [cf. (20)] are modelled as smoothly-varying over time and hence expressed with a small number K of Fourier basis functions. Increasing K accommodates for faster changes.

With this choice, (21) can be expressed in the frequency domain as:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{F} \mathbf{y} = \sum_{k=0}^{K-1} \mathbf{F} \text{Diag}(\sqrt{N} \mathbf{f}_k) \sum_{l=0}^{L-1} c_{lk} \mathbf{F}^H \text{Diag}(\sqrt{N} \mathbf{f}_l) \mathbf{F} \mathbf{x} \\ &= \sum_{l=0}^{L-1} \left(\sum_{k=0}^{K-1} c_{lk} \mathbf{D}^k \right) \text{Diag}(\sqrt{N} \mathbf{f}_l) \hat{\mathbf{x}}. \end{aligned} \quad (22)$$

While in (21) the matrix \mathbf{D} shifts in the time domain, in (22) it shifts in the frequency domain; however, such shift matrix is the same in both domains. This is different when looking at the graph counterpart in (7), where the two shift matrices might be different.

Remark 3: It is worthwhile to point out that the notion of smoothness for signals and filter coefficients is different in temporal and graph signal processing. In the time domain the basis to express smoothness for signals and filter coefficients is the same, both coinciding with the normalized DFT matrix. In GSP, the basis to express smoothness of graph signals and graph filter coefficients is different. Our theory shows that smoothness of graph signals is determined by the eigenvectors of the primal graph, while smoothness of the filter coefficients is determined by the Vandermonde matrix containing the dual graph frequencies.

All in all (22) is the time domain counterpart of (7), by choosing the primal eigenvector matrix \mathbf{V} to be $\mathbf{V} = \mathbf{F}^H$ and the basis functions λ_f^k to be $\lambda_f^k = \sqrt{N} \mathbf{f}_k$.

C. Non-Stationarity

In this section we study how and where (non-)stationarity of graph signals stands within the introduced theory. From [17], a process \mathbf{y} is said to be weakly stationary on a GSO \mathbf{S} if the covariance matrix $\mathbf{C}_y := \mathbb{E}[\mathbf{y} \mathbf{y}^H]$ commutes with \mathbf{S} or, equivalently, if \mathbf{y} can be written as the output of a C-GF \mathbf{H} [cf. (1)] when excited with a white input \mathbf{x} , i.e., $\mathbf{y} = \mathbf{H} \mathbf{x}$. As a consequence, the covariance matrix $\mathbf{C}_{\hat{y}}$ of the GFT process $\hat{\mathbf{y}}$ is diagonal, revealing the power spectral density of the process \mathbf{y} on its diagonal.

While [17] offers conditions to identify and model stationary graph signals, it does not explore non-stationary signal models;

this is our first attempt in that direction. Worth to mention is the work of [19], where a network topology identification approach is put forth to learn the GSO \mathbf{S} given a set of realizations of a non-stationary graph signal \mathbf{y} modelled with a classical graph convolution (2). There, however, non-stationarity is only considered with respect to node-invariant GFs, thus restricting the non-stationarity model taxonomy. Indeed, the (non-)stationarity of a random graph signal \mathbf{y} obtained as $\mathbf{y} = \mathbf{H}\mathbf{x}$ for a generic graph filter \mathbf{H} and an excitation input \mathbf{x} , depends either on the type of graph filter or the properties of the input \mathbf{x} . Precisely a non-stationary graph signal \mathbf{y} can be either modelled as the output of a shift-invariant graph filter with a non-stationary input or as the output of a node-variant graph filter when excited with a white input. The following properties and propositions formally describe these claims.

Property 1. Given input \mathbf{x} and output $\mathbf{y} = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{x}$, it holds:

$$\mathbf{C}_y = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{C}_x\mathbf{H}_I(\mathbf{P}, \mathbf{S})^H \quad (23)$$

with $\mathbf{C}_x := \mathbb{E}[\mathbf{x}\mathbf{x}^H]$. Moreover:

$$\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{V}^{-1}\mathbf{C}_y\mathbf{V}. \quad (24)$$

Depending on the structure of the graph filter $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ and the input signal \mathbf{x} , the ensuing propositions can be derived. Unless explicitly stated differently, we assume that the graph filter coefficient matrix \mathbf{P} respects the parametrization in (9), i.e., $\mathbf{P} = \Psi_f\mathbf{C}$, for some order L, K .

Proposition 1: If \mathbf{x} is a white graph signal, then [cf. Corollary 1]:

$$\mathbf{C}_y = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{H}_I(\mathbf{P}, \mathbf{S})^H \quad (25)$$

$$\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f)\mathbf{H}_{II}(\hat{\mathbf{P}}, \mathbf{S}_f)^H. \quad (26)$$

In general, this means that \mathbf{y} is non-stationary on \mathbf{S} and $\hat{\mathbf{y}}$ is non-stationary on \mathbf{S}_f .

Proposition 2: If $L = 1$ and \mathbf{x} is a white signal (stationary by definition), then $\hat{\mathbf{y}}$ is stationary on \mathbf{S}_f .

From (24), we can see that if \mathbf{C}_y is diagonal, then \mathbf{S}_f commutes with $\mathbf{C}_{\hat{\mathbf{y}}}$ and as such $\hat{\mathbf{y}}$ is stationary on \mathbf{S}_f , with a power spectral density (in the primal domain) equal to the eigenvalues of $\mathbf{C}_{\hat{\mathbf{y}}}$. This can happen only if $L = 1$ and \mathbf{x} is a white signal, where the convolution simply becomes $\mathbf{y} = \text{Diag}(\mathbf{p}_0)\mathbf{x}$, i.e., a windowing in the primal domain. The covariance matrix \mathbf{C}_y is then $\mathbf{C}_y = \text{Diag}(\mathbf{p}_0)^2$ and the cross correlation in \mathbf{y} is zero. In this case the covariance matrix $\mathbf{C}_{\hat{\mathbf{y}}}$ of the process $\hat{\mathbf{y}}$ in the dual domain is not diagonal in general, since it reads as:

$$\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{V}^{-1}\text{Diag}(\mathbf{p}_0)^2\mathbf{V}. \quad (27)$$

From (27), we can then conclude that \mathbf{y} is non-stationary on \mathbf{S} and, more importantly, $\mathbf{S}_f\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{C}_{\hat{\mathbf{y}}}\mathbf{S}_f$, i.e., \mathbf{S}_f commutes with $\mathbf{C}_{\hat{\mathbf{y}}}$. This implies that $\hat{\mathbf{y}}$ can be expressed as the output of a node-invariant graph filter in the dual domain when excited with white input, i.e., $\hat{\mathbf{y}} = \mathbf{H}(\hat{\mathbf{p}}, \mathbf{S}_f)\hat{\mathbf{x}}$ for some limited order filter coefficients $\hat{\mathbf{p}}$, rendering the estimation of the dual graph (see Section IV) a node-invariant graph filter estimation problem [7], [23]. This extends the classical notion of windowing in time domain (for instance used in power spectral density estimation),

which corresponds to a frequency-invariant convolution in the frequency domain. A generalization of this is given for $L > 1$ and a general signal \mathbf{x} , for which Corollary 2 applies.

Proposition 3: If $L = 1$ and \mathbf{x} is a non-white yet stationary graph signal with covariance $\mathbf{C}_x = \mathbf{V}\mathbf{\Lambda}_x\mathbf{V}^H$, then $\hat{\mathbf{y}}$ is not stationary on \mathbf{S}_f since:

$$\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{H}(\mathbf{c}; \mathbf{S}_f)\mathbf{\Lambda}_x\mathbf{H}(\mathbf{c}; \mathbf{S}_f)^H \quad (28)$$

for some coefficients $\mathbf{c} \in \mathbb{R}^K$, does not commute with \mathbf{S}_f .

Proposition 4: If $\mathbf{p}_l = p_l\mathbf{1}$, $\forall l$, and \mathbf{x} is not stationary on \mathbf{S} , then \mathbf{y} is not stationary on \mathbf{S} and $\hat{\mathbf{y}}$ is not stationary on \mathbf{S}_f . Indeed we have that the covariance matrix of \mathbf{y} :

$$\mathbf{C}_y = \mathbf{H}(\mathbf{p}; \mathbf{S})\mathbf{C}_x\mathbf{H}(\mathbf{p}; \mathbf{S})^H \quad (29)$$

does not commute with \mathbf{S} ; likewise \mathbf{S}_f does not commute with $\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{V}^{-1}\mathbf{C}_y\mathbf{V}$.

In other words, a node-invariant graph convolution of a non-stationary process \mathbf{x} results in a non-stationary process \mathbf{y} on \mathbf{S} (and non-stationary GFT $\hat{\mathbf{y}}$ on \mathbf{S}_f). While this result is not novel, we include it here to ensure a comprehensive coverage.

The introduced propositions provide a way to artificially generate non-stationary graph signals on a given GSO \mathbf{S} by filtering white noise, as explained next.

Generating non-stationary graph signals. Remember that $\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{V}^{-1}\mathbf{C}_y\mathbf{V}$ has to be diagonal for \mathbf{y} to be stationary on \mathbf{S} . Thus, non-stationary graph signals on \mathbf{S} can be generated as long as \mathbf{C}_y is not diagonalizable by \mathbf{V} (which would render $\mathbf{C}_{\hat{\mathbf{y}}}$ diagonal). In particular, if we want our GFT random process $\hat{\mathbf{y}}$ to have a specific covariance matrix equal to $\mathbf{C}_{\hat{\mathbf{y}}} = \mathbf{V}^{-1}\mathbf{C}_y\mathbf{V}$, for some positive semidefinite (PSD) \mathbf{C}_y we can generate random samples $\hat{\mathbf{y}}$ as:

$$\hat{\mathbf{y}} = \mathbf{V}^{-1}\mathbf{R}\mathbf{V}\hat{\mathbf{x}} \quad (30)$$

where \mathbf{R} is a matrix such that $\mathbf{C}_y = \mathbf{R}\mathbf{R}^H$ and $\hat{\mathbf{x}}$ is a white input; equivalently, in the primal domain:

$$\mathbf{y} = \mathbf{R}\mathbf{x}. \quad (31)$$

A simple example of such generation process is given by setting $\mathbf{C}_y = \text{Diag}(\mathbf{p})$ for some $\mathbf{p} \succcurlyeq \mathbf{0}$, so that (31) is a windowing operation in the primal domain, corresponding to a node-invariant graph convolution [cf. (30)] in the dual domain. With this choice, however, $\hat{\mathbf{y}}$ is stationary on \mathbf{S}_f [cf. Proposition 2]. If this is not sought-after, a general non diagonal PSD \mathbf{C}_y should be used.

IV. DUAL GRAPH IDENTIFICATION

So far we have assumed the knowledge of \mathbf{S}_f . In this section, we put forth a data-driven procedure to learn the dual graph eigenvalues λ_f in such a way that the resulting graph \mathbf{S}_f respects the theory developed in Section III. For simplicity, we restrict our attention to the case of an undirected primal graph with real-valued GSO \mathbf{S} and real-valued graph signals and filter coefficients. The problem setting is the following: consider T graph signals $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$ which can be modelled as non-stationary on the graph \mathbf{S} as the result of filtering T (possibly

unknown) input graph signals $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ with a NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$, i.e., $\mathbf{Y} = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{X}$. In particular, similar to temporal signal processing, we assume that the orders K and L are much smaller than N . Then the problem of identifying the dual graph can be formalized as follows:

Given \mathbf{Y} (and possibly \mathbf{X}), find a dual graph \mathbf{S}_f which is consistent with Theorem 1. In other words, find the dual eigenvalues λ_f such that a NV-GF \mathbf{H}_I on \mathbf{S} with $L \ll N$ can be expressed as a NV-GF \mathbf{H}_{II} on \mathbf{S}_f with $K \ll N$.

To tackle this problem, we adopt a two-step approach: in *step i*) we learn the filter taps \mathbf{P} of the NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ which best fit the available data, developing two distinct approaches for the input-output and output-only scenarios; in the second *step ii*) we find the dual eigenvalues λ_f by exploiting (9), i.e., we fit the model $\mathbf{P} = \Psi_f \mathbf{C}$, which is a specific structured matrix factorization with a Vandermonde factor. We solve this problem by following the recently proposed approach in [24] relying on a subspace method followed by successive convex programming.

A. Graph Filter Estimation With Input-Output Data

The goal of this section is to estimate the graph filter coefficients \mathbf{P} from a set of data pairs $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{y}_t) | t = 1, \dots, T\}$, all obtained using the same node-varying graph filter. By vectorizing the expression $\mathbf{Y} = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{X}$ we obtain:

$$\begin{aligned} \mathbf{y} &= \sum_{l=0}^{L-1} \text{vec}(\text{Diag}(\mathbf{p}_l) \mathbf{S}^l \mathbf{X}) \\ &= \sum_{l=0}^{L-1} ((\mathbf{S}^l \mathbf{X})^\top \circ \mathbf{I}_N) \mathbf{p}_l \\ &= [\mathbf{X}^\top \circ \mathbf{I}_N, \dots, (\mathbf{S}^{L-1} \mathbf{X})^\top \circ \mathbf{I}_N] \text{vec}(\mathbf{P}) \\ &= \mathbf{A} \text{vec}(\mathbf{P}) \end{aligned} \quad (32)$$

where $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_T^\top]^\top$, $\mathbf{A} = [\mathbf{X}^\top \circ \mathbf{I}_N, \dots, (\mathbf{S}^{L-1} \mathbf{X})^\top \circ \mathbf{I}_N] \in \mathbb{C}^{NT \times NL}$ and \circ denotes the Khatri-Rao product. An estimate of \mathbf{P} can then be obtained as:

$$\mathbf{P} = \text{unvec}(\mathbf{A}^\dagger \mathbf{y}), \quad (33)$$

where $\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$ is the pseudo-inverse of \mathbf{A} . The computational complexity of this step is dominated by the method employed to solve (33), which in MATLAB is based on the QR decomposition, incurring a cost of $\mathcal{O}(N^3 L^2 T)$. Since \mathbf{A} is very sparse due to the Hadamard structure, ad-hoc implementations can potentially lower this cost.

B. Graph Filter Estimation With Output-Only Data

The goal of this section is to estimate the graph filter coefficients \mathbf{P} with the only knowledge of the output graph signals $\mathbf{Y} \in \mathbb{R}^{N \times T}$. We assume that each \mathbf{y}_t can be modelled as the output of a NV-GF when excited with a white input $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is not directly observable. A viable approach is then to fit the (empirical) second order information of the process

which, together with the filter parametrization (3), leads to the following optimization problem:

$$\min_{\mathbf{P}} \|\hat{\mathbf{C}}_y - \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{H}_I(\mathbf{P}, \mathbf{S})^\top\|_F^2 \quad (34)$$

where $\hat{\mathbf{C}}_y = (1/T)\mathbf{Y}\mathbf{Y}^\top$ is the sample covariance matrix (\mathbf{Y} is already centered).

Problem (34) is non-convex in \mathbf{P} ; thus, to alleviate the non-convexity, we consider instead the simpler (yet, again non-convex) problem:

$$\min_{\mathbf{P}, \mathbf{U}} \|\mathbf{R} - \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{U}\|_F^2 \quad \text{s.t. } \mathbf{U}^\top \mathbf{U} = \mathbf{I} \quad (35)$$

where \mathbf{R} is a square matrix such that $\hat{\mathbf{C}}_y = \mathbf{R}\mathbf{R}^\top$, and \mathbf{U} is an $N \times N$ orthogonal matrix. By exploiting the SVD of the matrix \mathbf{Y} , i.e., $\mathbf{Y} = \mathbf{U}_y \mathbf{\Lambda}_y \mathbf{V}_y^\top$, possible choices for \mathbf{R} are $\mathbf{R} = (1/\sqrt{T})\mathbf{U}_y \mathbf{\Lambda}_y \mathbf{U}_y^\top$ and $\mathbf{R} = (1/\sqrt{T})\mathbf{U}_y \mathbf{\Lambda}_y$.

Observe that the solutions of (34) and (35) coincide in the absence of noise and as $T \rightarrow \infty$. In other words, \mathbf{P}^* is a solution of (34) if and only if it is also a solution of (35), for some orthogonal \mathbf{U}^* . This equivalence is formally stated in the following lemma, where we denote $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ as $\mathbf{H}(\mathbf{P})$ for simplicity.

Lemma: The system of equations $\hat{\mathbf{C}}_y = \mathbf{H}(\mathbf{P})\mathbf{H}(\mathbf{P})^\top$ has solution \mathbf{P}^* if and only if the system of equations $\mathbf{R} = \mathbf{H}(\mathbf{P})\mathbf{U}$ has solution $(\mathbf{P}^*, \mathbf{U}^*)$ for an orthogonal matrix \mathbf{U}^* .

Proof: (\Leftarrow) Let $\mathbf{P}^*, \mathbf{U}^*$ be the solution of $\mathbf{R} = \mathbf{H}(\mathbf{P})\mathbf{U}$. This is clearly also a solution of the other system of equations, since:

$$\begin{aligned} \mathbf{H}(\mathbf{P}^*)\mathbf{U}^* \mathbf{U}^{*\top} \mathbf{H}(\mathbf{P}^*)^\top &= \mathbf{H}(\mathbf{P}^*)\mathbf{H}(\mathbf{P}^*)^\top \\ &= \mathbf{R}\mathbf{R}^\top = \hat{\mathbf{C}}_y. \end{aligned} \quad (36)$$

(\Rightarrow) Let \mathbf{P}^* be the solution of $\hat{\mathbf{C}}_y = \mathbf{H}(\mathbf{P})\mathbf{H}(\mathbf{P})^\top$. The (thin) SVD of matrix \mathbf{R} and $\mathbf{H}(\mathbf{P}^*)$ can be written as:

$$\mathbf{R} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top \quad (37)$$

$$\mathbf{H}(\mathbf{P}^*) = \mathbf{U}_h \mathbf{\Sigma}_h \mathbf{V}_h^\top \quad (38)$$

from which it follows that $\mathbf{R} = \mathbf{H}(\mathbf{P}^*)\mathbf{U}^*$ with $\mathbf{U}^* = \mathbf{V}_h \mathbf{V}_r^\top$. \square

Despite the fact that (35) is not jointly convex in \mathbf{P} and \mathbf{U} , it is convex in \mathbf{P} for a fixed \mathbf{U} ; moreover, for a fixed \mathbf{P} , it reduces to the well-studied orthogonal Procrustes problem [25], for which a closed form solution exists albeit its non-convexity. Thus, an alternating minimization over \mathbf{P} and \mathbf{U} can be put forth, as follows: *a*) first, given the estimate of \mathbf{U} at the $(n-1)$ th iteration, i.e., $\mathbf{U}^{(n-1)}$, the estimation problem at the n th iteration for the filter taps matrix \mathbf{P} reads as:

$$\mathbf{P}^{(n)} = \argmin_{\mathbf{P}} \|\mathbf{R} - \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{U}^{(n-1)}\|_F^2, \quad (36a)$$

which can be solved in closed form by considering $\mathbf{Y} = \mathbf{R}$ and $\mathbf{X} = \mathbf{U}^{(n-1)}$ in (32). The solution of (36a) is then *b*) used in the next step to refine the estimate of the unitary matrix \mathbf{U} , i.e.:

$$\begin{aligned} \mathbf{U}^{(n)} &= \argmin_{\mathbf{U}} \|\mathbf{R} - \mathbf{H}_I(\mathbf{P}^{(n)}, \mathbf{S})\mathbf{U}\|_F^2 \\ \text{s.t. } \mathbf{U}^\top \mathbf{U} &= \mathbf{I} \end{aligned} \quad (36b)$$

for which the closed form solution is $\mathbf{U}^{(n)} = \mathbf{V}_p \mathbf{U}_p^\top$, where \mathbf{U}_p and \mathbf{V}_p are the left and right singular vector matrices,

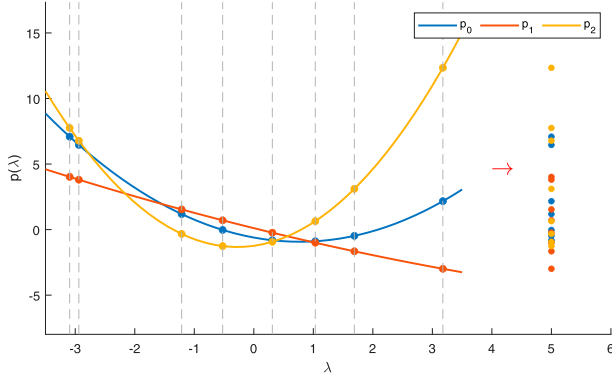


Fig. 2. Illustration of the problem: a number of functions ($L = 3$) are sampled (filled points) in $N = 8$ locations (gray dotted lines). The problem is to recover the sampling locations and the polynomial coefficients with as little knowledge as possible, the function they belong to, and ordered such that the sampling points for all functions are aligned (shown on the right in the figure at $\lambda = 5$).

respectively, of the matrix product $\mathbf{R}^\top \mathbf{H}_I(\mathbf{P}^{(n)}, \mathbf{S})$, that is, $\mathbf{R}^\top \mathbf{H}_I(\mathbf{P}^{(n)}, \mathbf{S}) = \mathbf{U}_p \mathbf{\Lambda}_p \mathbf{V}_p^\top$, with $\mathbf{\Lambda}_p$ the matrix of singular values. The solution of (36b) is then fed again into (36a), unless a predefined number of iterations or stopping criterion is reached. The main computational load per iteration stems from calculating matrix \mathbf{P} in (36a), typically incurring a cost of $\mathcal{O}(N^4)$; overall the cost is then $\mathcal{O}(IN^4)$ with I the number of iterations.

C. Dual Graph Frequency Estimation

The next step is to learn the dual graph frequencies λ_f , which are the only unknowns of the dual GSO \mathbf{S}_f . In order to do this, consider again the matrix form of (6):

$$\mathbf{P} = \Psi_f(\lambda_f) \mathbf{C} \quad (39)$$

where we explicitly wrote Ψ_f as a function of λ_f to highlight the fact that the matrix is entirely determined by its second column λ_f , representing our unknown. Then, the problem we aim to solve can be formally stated as follows:

Given the matrix $\mathbf{P} \in \mathbb{R}^{N \times L}$, recover the input vector $\lambda_f \in \mathbb{R}^N$ and the coefficient matrix $\mathbf{C} \in \mathbb{R}^{K \times L}$ such that (39) holds as accurately as possible.

Although the problem can be approached from a pure algebraic point of view as a structured matrix factorization, a pleasing geometrical interpretation of (39) is given by interpreting the vectors $\mathbf{p}_0, \dots, \mathbf{p}_{L-1}$ as function values obtained by sampling L distinct polynomials $p_0(\lambda_f), \dots, p_{L-1}(\lambda_f)$, all with degree $K - 1$, in the same N unknown locations $\lambda_{f,0}, \dots, \lambda_{f,N-1}$. The goal is to recover the original locations (and polynomial coefficients) from the available sampled function values. The only side information we have about these sampling points is *i*) which function they belong to and *ii*) that they are ordered in such a way that the related sampling points are aligned. See Fig. 2 for an illustration.

Subspace Fitting. Assume we start by considering the following optimization problem to estimate λ_f and \mathbf{C} :

$$\min_{\lambda_f, \mathbf{C}} \frac{1}{2} \|\mathbf{P} - \Psi_f(\lambda_f) \mathbf{C}\|_F^2, \quad (40)$$

which can be solved, for instance, with an alternating minimization approach, without guarantee of convergence to a global optimum.

An alternative subspace method can be devised when $L \geq K$. We then consider the economy-size SVD of matrix \mathbf{P} , i.e., $\mathbf{P} = \mathbf{U} \mathbf{\Sigma} \mathbf{Z}^\top$, where $\mathbf{U} \in \mathbb{R}^{N \times K}$ and $\mathbf{Z} \in \mathbb{R}^{L \times K}$ are the left and right singular vectors, respectively, and $\mathbf{\Sigma} \in \mathbb{R}^{K \times K}$ is the diagonal matrix of singular values. Since both $\Psi_f(\lambda_f)$ and \mathbf{U} represent a basis for the column space of \mathbf{P} , there exists a non-singular matrix $\mathbf{Q} \in \mathbb{R}^{K \times K}$ such that $\Psi_f = \mathbf{U} \mathbf{Q}$. The subspace fitting [13] problem reads then as:

$$\min_{\lambda_f, \mathbf{Q}} \frac{1}{2} \|\Psi_f(\lambda_f) - \mathbf{U} \mathbf{Q}\|_F^2, \quad (41)$$

which, upon substituting the pseudoinverse solution $\mathbf{Q} = \mathbf{U}^\dagger \Psi_f$ into (41), can be casted as the following equivalent problem:

$$\min_{\lambda_f} \{f(\lambda_f) := \frac{1}{2} \|\Pi \Psi_f(\lambda_f)\|_F^2\}, \quad (42)$$

with $\Pi := \mathbf{I}_N - \mathbf{U} \mathbf{U}^\dagger$ the orthogonal projection matrix onto the orthogonal complement of \mathbf{U} . In other words, problem (42) aims to find a vector λ_f such that the Vandermonde matrix $\Psi_f(\lambda_f)$ is orthogonal to the subspace spanned by the orthogonal complement of \mathbf{U} . Notice that we require $L \geq K$, so that the matrix \mathbf{P} can have rank K revealing the subspace of the Vandermonde matrix $\Psi_f(\lambda_f)$ we want to estimate.

Problem (41) is not convex in λ_f due to the polynomial degree K (unless $K = 2$, i.e., the model in (39) is linear in λ_f). To tackle the non-convexity of the problem, we resort to sequential convex programming (SCP) [14], a local optimization method that leverages convex optimization, where the non-convex portion of the problem is modeled by convex functions that are (at least locally) accurate. As in any non-convex problem, the initial starting point plays a big role; thus, if no prior information on the variable is given, a multi-starting point approach is advisable. The SCP formulation can be found in [24] and in the Supplemental Material. Overall, the computational complexity is dominated by computing the left singular vectors of matrix \mathbf{P} , tantamount to a cost of $\mathcal{O}(NL^2)$.

Remark 4: The nature of the problem and the formulation (42) share similarities with the MUSIC algorithm [26]. There, however, the problem considers Ψ_f^\top instead of Ψ_f and N independent 1-dimensional searches can be carried out to find λ_f (which would be contained in the second row of Ψ_f^\top). In our case, each column contains all the N variables and an N -dimensional search is needed, rendering a “scanning” of the vector variable λ_f infeasible, unless N is very small.

D. Ambiguities

Notice that (39) is not free of model ambiguities, since different pairs (λ_f, \mathbf{C}) may lead to the same observation matrix \mathbf{P} .

Because we assume that \mathbf{P} has rank K , if Ψ_f and \mathbf{C} are the true matrix factors satisfying (39), then for any $K \times K$ invertible matrix \mathbf{T} , it holds:

$$\mathbf{P} = \Psi_f \mathbf{T} \mathbf{T}^{-1} \mathbf{C} = \Psi'_f \mathbf{C}'. \quad (43)$$

However, Ψ'_f needs to be Vandermonde in order for (43) to respect the model structure in (39). As we proved in [24, Theorem 2], matrix \mathbf{T} needs in general to be a Pascal matrix.

The consequence of such theorem is that without any added constraint on Ψ_f or \mathbf{C} in (39), every shifted and scaled version of the groundtruth parameter λ_f (for an appropriate \mathbf{C}), perfectly fits the observation model and is a valid solution for (42). This is satisfactory for our purpose: a shift and scale of the graph eigenvalues maintains the same topological structure of the original graph (removing the self loops caused by the shift); we will illustrate this in Section V.

V. NUMERICAL EXPERIMENTS

In this section we perform numerical experiments relying on the theory outlined in Section IV. Specifically, we first test our learning algorithm on synthetic data to assess the validity of the approach and its drawbacks; finally, we use it on real data.

A. Synthetic Data

The goal of this section is to validate the approach outlined in Section IV, by assuming the knowledge of the dual eigenvalues λ_f , which however is not used during the training phase but only used to compute the performance metrics. We have the following *generation* and *learning* phases (also depicted in the Supplemental Material for visual clarity):

1) Generation.

- (a) **Dual eigenvalues λ_f :** in order to have control over the spacing between different eigenvalues, we generate them with the following strategy. First, we uniformly sample the eigenvalue domain leading to the grid $\mathbf{u} := [u_0, u_1, \dots, u_{N-1}]^\top$, where the distance among two samples $P := u_n - u_{n-1}$ is constant for all n . Then, the actual eigenvalues λ_f are generated as $\lambda_{f,n} = u_n + j_n$ where $j_n \sim \mathcal{TN}(0, (\delta P/2)^2)$, with $\mathcal{TN}(0, \sigma_j^2)$ a Gaussian distribution with zero mean and standard deviation σ_j , yet truncated at σ_j . The (jitter) parameter $\delta > 0$ specifies the randomness of the eigenvalues. If $\delta < 1$ the eigenvalues maintain the same ranking order as the uniform samples, while if $\delta \geq 1$ they can overlap to the adjacent ones. Notice that in this synthetic setup we are not interested in creating a “meaningful” dual graph, but rather assessing whether our algorithmic routine is able to identify such dual graph.
- (b) **Filter taps \mathbf{P} :** we generate the primal node-varying filter taps $\mathbf{P} \in \mathbb{R}^{N \times L}$ as $\mathbf{P} = \Psi_f \mathbf{C}$, with Ψ_f the Vandermonde matrix associated to λ_f generated in step (a), and $\mathbf{C} \in \mathbb{C}^{K \times L}$ a random expansion coefficient matrix [cf. (9)] generated as $\text{vec}(\mathbf{C}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{KL})$. To increase the curvature of the considered polynomials (thus avoiding to have almost flat curves in the domain of

interest), we perform an extra weighting scheme by increasing the weight of higher order monomials; that is, we multiply the matrix \mathbf{C} with a mask matrix as $\mathbf{C} \leftarrow [\mathbf{1}, 2\mathbf{1}, \dots, K\mathbf{1}]^\top \odot \mathbf{C}$, where the i th column of the mask matrix is the constant vector containing in all its entries the value i .

- (c) **Input-Output Data \mathbf{X}, \mathbf{Y} :** we generate T input graph signals $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ and stack them in the matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$. Then, we filter \mathbf{X} with the type-I NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ to obtain T new (possibly noisy) graph signals $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T] = \mathbf{H}_I(\mathbf{P}, \mathbf{S})\mathbf{X} + [\mathbf{n}_1, \dots, \mathbf{n}_T]$, with $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ the measurement noise.

2) Learning. The goal in this phase is to recover the original λ_f of step 1(a) from the input-output data $\{\mathbf{X}, \mathbf{Y}\}$. This is achieved with the algorithmic routine introduced in Section IV-A and Section IV-C; namely:

- (a) **Filter Taps $\tilde{\mathbf{P}}$:** we find an estimate $\tilde{\mathbf{P}}$ of the filter taps \mathbf{P} through (33);
- (b) **Dual eigenvalues $\tilde{\lambda}_f$:** we find an estimate $\tilde{\lambda}_f$ of λ_f with the procedure outlined in Section IV-C, namely, casting the problem as a subspace fitting problem and solving it with SCP. During this step, we also estimate the coefficient matrix $\tilde{\mathbf{C}}$ of the expansion model leading to $\tilde{\mathbf{C}}$.

Metrics. To assess the validity of the proposed approach, we consider two different performance metrics, one relative to the estimation of the filter coefficients \mathbf{P} and one relative to the estimation of the dual eigenvalues λ_f . For the filter coefficients, we consider the normalized squared error (NSE), computed as:

$$\text{NSE}(\tilde{\mathbf{P}}, \mathbf{P}) = \frac{\|\tilde{\mathbf{P}} - \mathbf{P}\|_F^2}{\|\mathbf{P}\|_F^2}. \quad (44)$$

For the dual eigenvalues, recall that we can recover the solution of problem (42) up to a shift and scaling of the true positions [cf. Section IV-D]. Thus, as performance metric, we use the normalized error modulo Pascal (PNE), defined as:

$$\text{PNE}(\tilde{\lambda}_f, \lambda_f) = \min_{t_0, t_1} \frac{\|\lambda_f - (t_0 \mathbf{1} + t_1 \tilde{\lambda}_f)\|_2^2}{\|\lambda_f\|_2^2} \quad (45)$$

which measures how far the true eigenvalues are from a linear transformation of the recovered estimates. Clearly (45) is zero whenever $\tilde{\lambda}_f$ is a solution for (42).

Results. We generate our primal graph \mathbf{S} as a random sensor network with $N = 40$ nodes, using the CVX toolbox [27]. In this setup, each node is randomly sampled from the unit square, and a 10-nearest neighbor graph is created with edge weights determined by a Gaussian kernel. Additionally, we set $u_0 = -1$ and $u_{N-1} = +1$ [cf. 1(a)]. We run the algorithm for different parameter configurations, specifically: the order of the filter in the primal domain $L \in \{2, \dots, 9\}$, which is also the polynomial degree of the primal eigenvalues (cf. (7)); the order of the filter in the dual domain $K \in \{2, \dots, 9\}$, with $K \leq L$, which is also the polynomial degree of the dual eigenvalues; the jitter parameter $\delta \in \{1, 10, 100, 1000\}$; and the noise standard deviation $\sigma \in \{0, 0.5, 5, 50\}$. The number of input (and output)

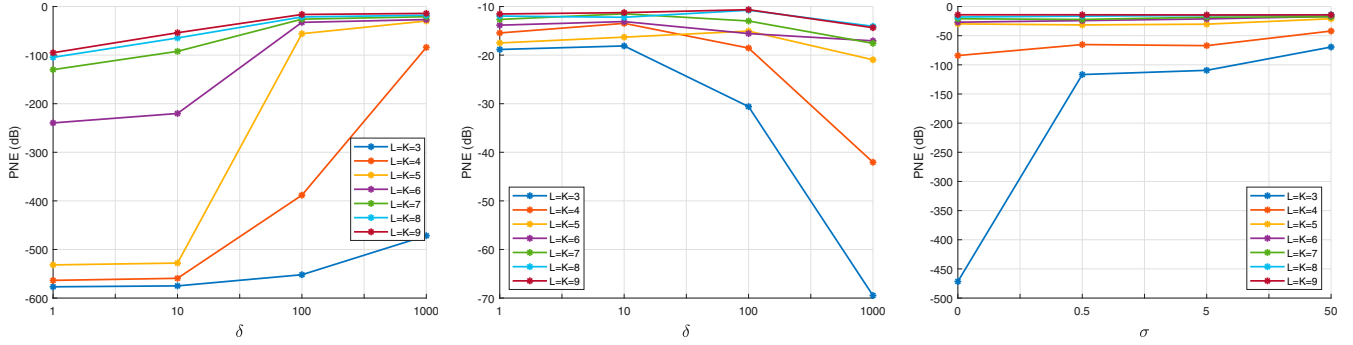


Fig. 3. PNE (in dB) as a function of δ with $\sigma = 0$ (left) and $\sigma = 50$ (center), for different values of L, K ; and PNE (in dB) as a function of σ for $\delta = 1000$ (right).

graph signals is set to $T = 3000$. Due to the non-convexity of the cost function (42), we run the algorithm with 5 different starting points λ_f^0 , one of which is the uniform grid \mathbf{u} ; this, together with the jitter parameter δ [cf. Generation (a)] helps us also understanding the “magnitude” of the objective function’s non-convex landscape: if the objective function is highly non-convex, even an initial starting point λ_f^0 close to the real λ_f (meaning a small δ) might incur a very high objective value and likely end-up in a local minimum. In such a case, a random starting point might be beneficial. The magnitude of non convexity of the objective function increases by increasing K . Finally, we compute the performance metrics relative to the solution associated to the different starting points.

In Fig. 3 we show the PNE (in dB) as a function of δ for different $L = K^3$, in the noiseless case (left figure) and noisy case (middle figure); in addition, we show (right) how the PNE varies as a function of the noise σ for fixed $\delta = 1000$. We can make the following observations: as expected, for low-degree polynomials, the algorithm performs better, since the non-convexity of the problem increases with increasing polynomial order. This is visible from the left and middle figure: for a small perturbation δ , also high orders yield a good performance; which degrades by increasing δ . However, when noise is present in the observations, a random initial starting points (i.e. higher δ) seems to be beneficial. Moreover, noise in the measurement (right figure) has obviously a negative impact in the learning performance, which however enables us to reconstruct the graph as we can see next.

To visually assess the algorithm’s performance, in Fig. 4 we show, for the noiseless case, the ambiguity-corrected estimated eigenvalues $\tilde{\lambda}_f^c$ (red crosses), together with the original eigenvalues λ_f (green circles) and the initial starting point of the algorithm λ_f^0 (blue diamonds) for $L = K = 3$ (top row) and $L = K = 9$ (bottom row), both with $\delta = 1 \times 10^4$, which corresponds to a completely random configuration of the eigenvalues. The ambiguity correction is explicitly performed as:

$$\tilde{\lambda}_f^c = [\mathbf{1} \tilde{\lambda}_f][\mathbf{1} \tilde{\lambda}_f]^\dagger \lambda_f, \quad (46)$$

³For all the plots we set $L = K$ for visualization clarity. To reproduce the experiments with different parameter settings, we make available the source code: <https://github.com/albertonati/genConv>

which is the closest point to λ_f up to a linear transformation dictated by the optimal t_0 and t_1 minimizing (45).

In the $L = K = 3$ case, the NSE is $\text{NSE}(\tilde{\mathbf{P}}, \mathbf{P}) = 2.41 \times 10^{-29}$ and the PNE is $\text{PNE}(\tilde{\lambda}_f, \lambda_f) = 3.34 \times 10^{-25}$; in the $L = K = 9$ case the NSE is $\text{NSE}(\tilde{\mathbf{P}}, \mathbf{P}) = 9.20 \times 10^{-23}$ and the PNE is $\text{PNE}(\tilde{\lambda}_f, \lambda_f) = 1.03 \times 10^{-1}$. In both cases the inferred dual GSO, shown in the middle column of Fig. 4, correctly reveals the structure of the true dual GSO \mathbf{S}_f , shown in the right column; this even though the eigenvalue reconstruction (left column) is perfect only in the first case. In other words, even with an inexact (but not random) reconstruction of the eigenvalues, the algorithm seems to adequately capture the connections present in the true GSO \mathbf{S}_f . The reason behind the difference in error estimation among the two cases is mainly due to the high polynomial degree K , which on the analytic side renders the objective function (42) highly non-convex (and hence easier for the algorithm to end up in a local minimum); on the algebraic side it increases the numerical instability of performing the pseudoinverse of the matrix \mathbf{A} required for a correct estimation of the filter parameter matrix \mathbf{P} . Hence, even a perfect dual graph frequency estimation step fitting perfectly the estimated \mathbf{P} , might fail to perfectly reconstruct the true eigenvalues λ_f . For $\delta = 10$, the algorithm improves the PNE by two orders of magnitude and the inferred eigenvalues $\tilde{\lambda}_f$ nearly overlap the true ones λ_f .

For the noisy scenario, in Fig. 5 we show the results obtained by considering the two cases described above (i.e. $L = K = 3$ and $L = K = 9$ with $\delta = 1000$), but with a measurement noise having $\sigma = 50$. In this case the NSE is 8.88×10^{-6} and the PNE is 8.72×10^{-5} , while for the latter the NSE is 9.74×10^{-16} and the PNE is 2.00×10^{-1} . The eigenvalues and graph reconstruction is successful despite the fact that the reconstruction of the filter taps is not perfect as in the noiseless case, which gives us hope for the robustness of the algorithm when measuring noisy data.

Overall we can state that from an algorithmic point of view, the algorithm is robust in presence of a considerable eigenvalue perturbation δ and noise σ , especially for low polynomial degree. In instances where the polynomial degree is high, the inherent non-convex nature of the problem introduces substantial complexity into the optimization process, which however leads to a dual graph resembling the original one. In addition, we

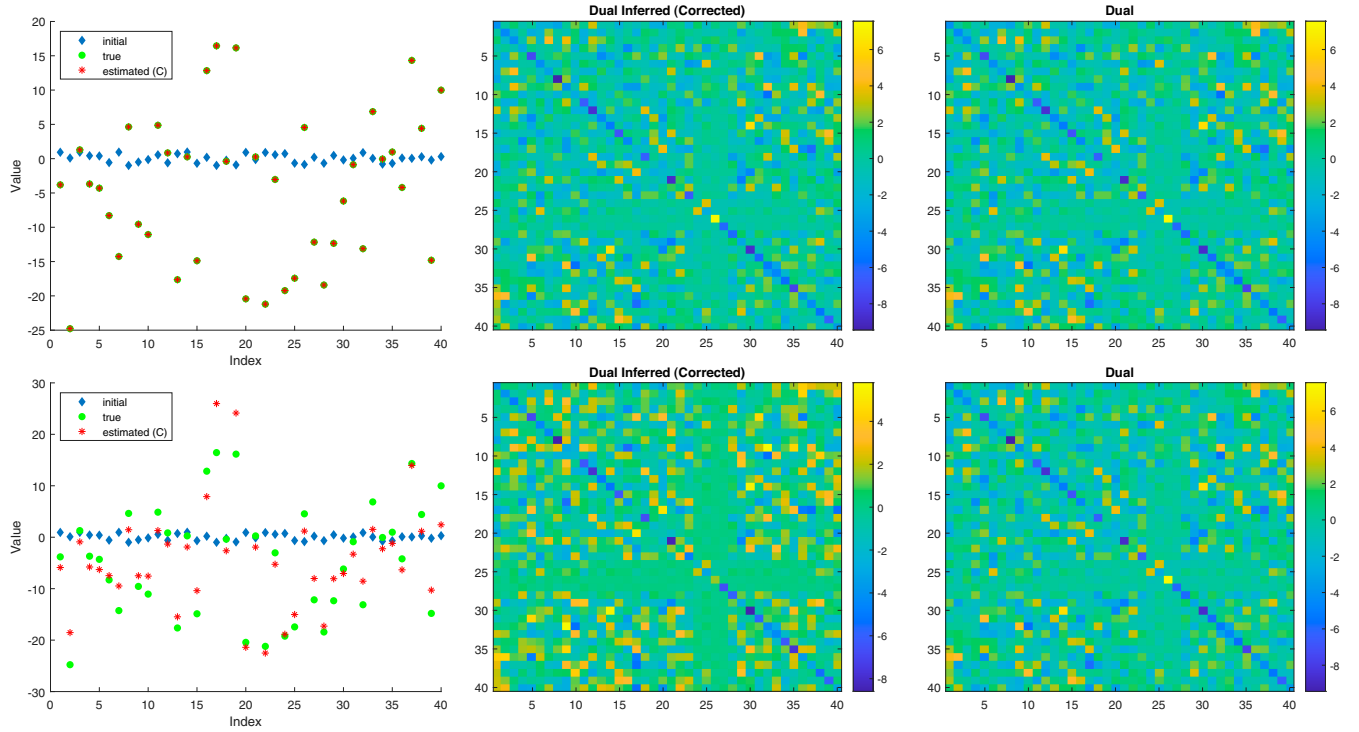


Fig. 4. (Noiseless case $\sigma = 0$) Results for $L = K = 3$ (top row) and $L = K = 9$ (bottom row) with $\delta = 1 \times 10^4$. (Left Column) True eigenvalues (green circles), inferred eigenvalues (red crosses) and initial starting point of the algorithm (blue diamonds); (Center Column) Inferred dual GSO with ambiguity-correction; (Right Column) True dual GSO \mathbf{S}_f .

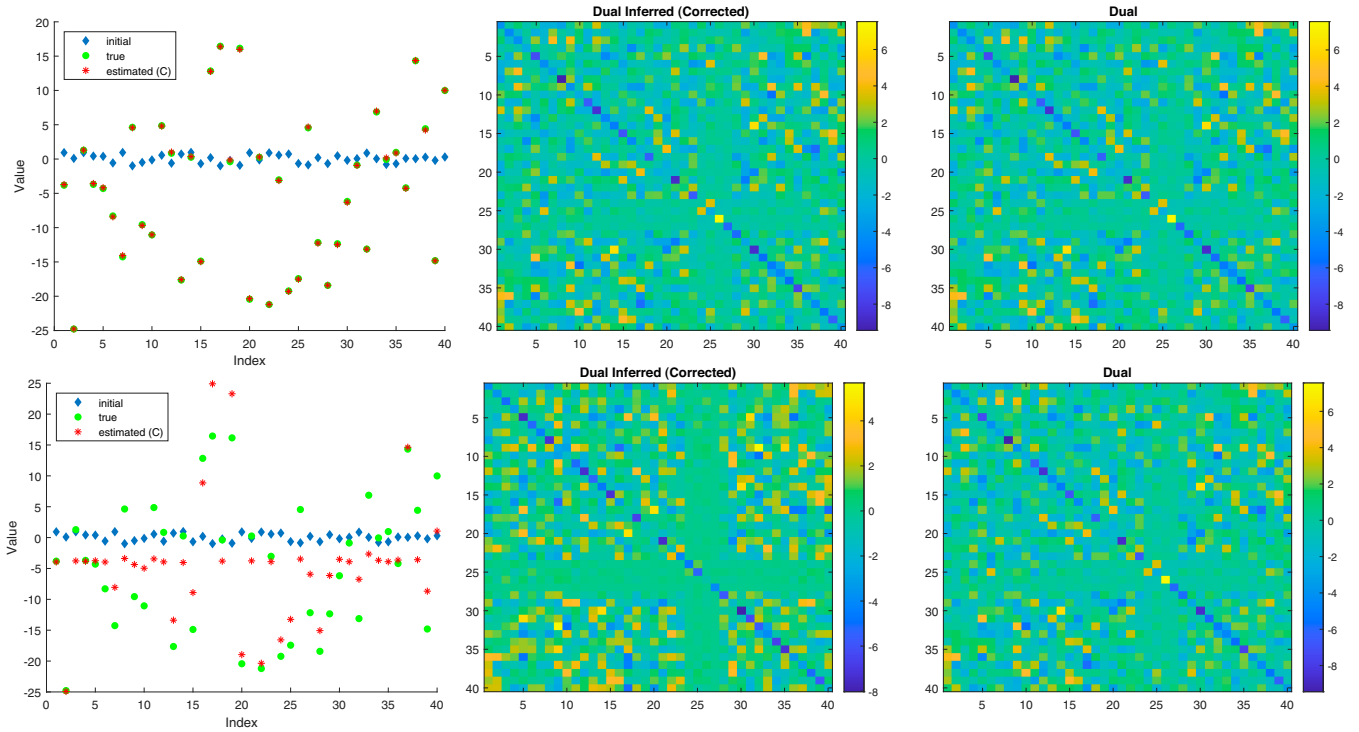


Fig. 5. (Noisy case $\sigma = 50$) Results for $L = K = 3$ (top row) and $L = K = 9$ (bottom row) with $\delta = 1 \times 10^3$. (Left Column) True eigenvalues (green circles), inferred eigenvalues (red crosses) and initial starting point of the algorithm (blue diamonds); (Center Column) Inferred dual GSO with ambiguity-correction; (Right Column) True dual GSO \mathbf{S}_f .

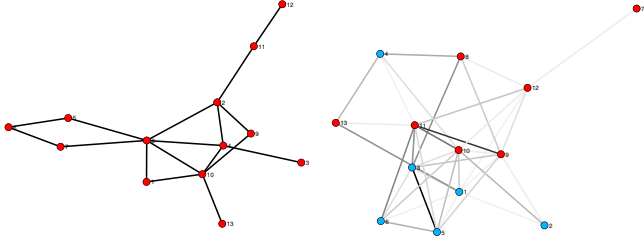


Fig. 6. (Left): primal graph of the road network; (Right) dual graph of the road network, where each node represents a graph frequency. Red (blue) nodes are the standard low (high) frequencies.

observe that when the NSE is high, meaning that \mathbf{P} has not been properly reconstructed, the PNE is also usually high, which is somehow expected since we rely on \mathbf{P} to estimate the dual eigenvalues λ_f .

B. Real Data

In this section we exploit the theory developed in Section III and Section IV to infer a dual graph from real data. To evaluate the stationarity level of the given data \mathbf{Y} for a given GSO \mathbf{S} , we use the proxy-measure $\rho = \|\text{diag}(\mathbf{V}^{-1}\mathbf{C}_y\mathbf{V})\|_2^2 / \|\mathbf{V}^{-1}\mathbf{C}_y\mathbf{V}\|_F^2$, which measures the “diagonal dominance” of the spectral covariance matrix; a value of 1 indicates that the data are stationary.

As performance metrics we monitor two errors: *i*) the $\text{NSE}(\mathbf{Y}, \mathbf{H}_I(\tilde{\mathbf{P}}, \mathbf{S})\mathbf{X})$ for the input-output case [cf. IV-A] or the $\text{NSE}(\mathbf{R}, \mathbf{H}_I(\tilde{\mathbf{P}}, \mathbf{S})\tilde{\mathbf{U}})$ for the output-only case [cf. IV-B] (recall the definition of the NSE in (44)); and *ii*) the “corollary error” [cf. Corollary 1 in (10)]:

$$\varepsilon_c = \frac{\|\mathbf{V}^{-1}\mathbf{H}_I(\tilde{\mathbf{P}}, \mathbf{S}) - \mathbf{H}_{II}(\tilde{\mathbf{P}}, \mathbf{S}_f)\mathbf{V}^{-1}\|_F^2}{\|\mathbf{V}^{-1}\mathbf{H}_I(\tilde{\mathbf{P}}, \mathbf{S})\|_F^2} \quad (47)$$

which assesses whether the inferred \mathbf{S}_f is a “valid” dual graph consistent with our theorem, i.e., how much the upper and lower branches of Fig. 1 diverge from each other. To make things more clear, the $\tilde{\mathbf{P}}$ refers to the estimation of the primal filter tap matrix \mathbf{P} either following (33) or (36a), while $\hat{\mathbf{P}}$ refers to the estimation of the dual filter tap matrix $\hat{\mathbf{P}}$, which is here computed as $\hat{\mathbf{P}} = \Psi^\dagger \tilde{\mathbf{C}}^\top$, with $\tilde{\mathbf{C}} = \Psi_f(\tilde{\lambda}_f)^\dagger \tilde{\mathbf{P}}$, where $\tilde{\lambda}_f$ is the estimation of the dual eigenvalues λ_f solving (42). All in all, error *i*) concerns the graph filter estimation, while *ii*) concerns the dual graph estimation.

1) *Traffic Volume*: we consider a subsampled version of the open dataset in [28] which contains $T = 1259$ traffic volume measurements at intervals of 15 minutes at $N = 13$ sensor locations along two major highways in Northern Virginia/Washington, D.C.; in addition, the physical (road) network is available, see Fig. 6(left). We denote with \mathbf{S} the adjacency matrix representing the given road network, and with $\mathbf{Y} \in \mathbb{R}^{N \times T}$ the (centered) graph signals corresponding to the traffic volume measurements. These signals exhibit a non-stationary behavior captured by $\rho = 0.54$.

In our experiment, we explore both input-output and output-only scenarios. In the former scenario, we define the input

matrix \mathbf{X} by aligning it with \mathbf{Y} , shifting each column two positions to the left; consequently, the learning task revolves around forecasting the traffic volume 30 minutes ahead. We initialize the parameters with $L = K = 3$ and execute the algorithm. Notice that our data was deliberately not partitioned into training, validation, and test sets, since our primary focus is optimizing the fitting of our graph filter to the provided data, rather than evaluating its forecasting performance.

First, we learn the filter coefficients \mathbf{P} for both scenarios, yielding a NSE equal to 8×10^{-2} for both. Subsequently, we use the inferred $\tilde{\mathbf{P}}$ to learn the dual eigenvalues λ_f by solving (42). The associated dual GSO $\mathbf{S}_f = \mathbf{V}^{-1} \text{Diag}(\tilde{\lambda}_f) \mathbf{V}$, achieving a corollary error [cf. (47)] $\varepsilon_c = 1 \times 10^{-1}$ for scenario 1 and $\varepsilon_c = 3 \times 10^{-1}$ for scenario 2, is shown in Fig. 6(right), where we only display the 50% biggest edges (in absolute values) to ease the visualization. We color with blue the first half of the nodes, representing what are usually considered “high pass” frequencies, and with red the second half of the nodes, representing what are usually considered the “low pass” frequencies (remember that \mathbf{S} is the adjacency matrix). Adjacent nodes in the graph are not necessarily among consecutive graph frequencies, as it is commonly assumed in GSP. This shows that the frequency ordering that is commonly assumed does not fit our theory and we might obtain a more expressive way to embed the different graph frequencies.

2) *MNIST Dataset*: We consider the MNIST dataset of gray-scale handwritten digits from 0 to 9, focusing on the digit 5⁴ containing $T = 5949$ images of size 18×18 . We model each pixel of the image as a node of the primal graph \mathbf{S} and its pixel intensity as the graph signal value at that node. As a preprocessing step, we remove the mean from each pixel and vectorize the images, thus obtaining a graph signal matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$, with $N = 324$. As GSO \mathbf{S} , we consider the normalized Laplacian matrix of the 18×18 grid graph, for which the 5 exhibits a non-stationary behavior, since $\rho = 0.4$.

We assume that each \mathbf{y}_t is the output of a NV-GF $\mathbf{H}_I(\mathbf{P}, \mathbf{S})$ when excited with a white input $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; our goal is then to learn the filter taps \mathbf{P} and subsequently the dual eigenvalues λ_f from the available data \mathbf{Y} . In this particular scenario, it is important to highlight that our access is limited to the output data \mathbf{Y} , since the corresponding noise input \mathbf{X} is not available. Consequently, the sole viable approach becomes the output-only procedure [cf. Section IV-B]. Nevertheless, to navigate this limitation and to be also able to use the input-output approach of Section IV-A, we can employ the ensuing rationale to derive pairs $(\mathbf{x}_t, \mathbf{y}_t)$:

- (a) compute the covariance matrix $\mathbf{C}_y = \mathbf{Y}\mathbf{Y}^\top / T$ and decompose it as $\mathbf{C}_y = \mathbf{R}\mathbf{R}^\top$;
- (b) filter a white input signal \mathbf{x}_t to generate a new signal \mathbf{y}_t as $\mathbf{y}_t = \mathbf{R}\mathbf{x}_t$.

It follows that the \mathbf{y}_t vector generated in this way follow the same distribution of the original \mathbf{Y} and still represent the digit 5 (up to a sign ambiguity). This time, however, we have the

⁴Similar results can be obtained with the other digits, see <https://github.com/albertonat/genConv>

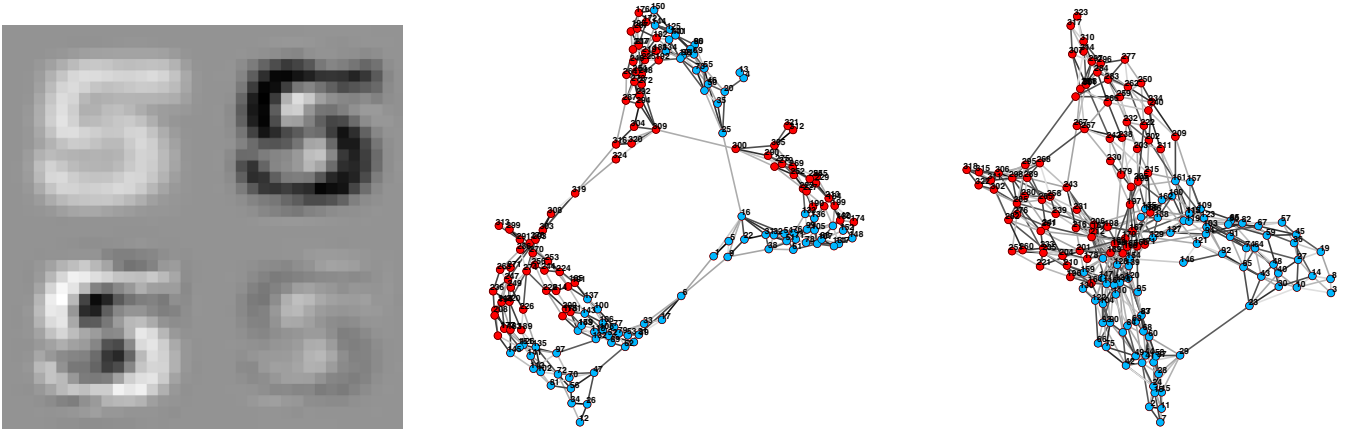


Fig. 7. (Left) Illustration of each filter tap $\{\mathbf{p}_l\}_{l=0}^3$, following a row-wise stacking, reshaped to have the same shape of the input image. Increasing l renders the entries of \mathbf{p}_l less influential for the construction of the final digit; (Right) Dual Graph of the 5-digit. The label indicates the index of the eigenvalues λ_i , and the node color indicates whether it is part of the first half (blue) or the second half (red), commonly associated to the low and high pass bands, respectively. Notice that the graph is connected but due to the edge thresholding (only for the visualization) it is split in two.

associated input \mathbf{X} . We run the proposed algorithm for different orders L and K , with and without input \mathbf{X} .

In Fig. 7(left) we show the inferred filter taps $\tilde{\mathbf{P}}$, for $L = 4$, corresponding to the solution of problem (35) and yielding a $\text{NSE} = 3 \times 10^{-2}$; each filter tap $\{\mathbf{p}_l\}_{l=0}^3$ has been reshaped to have the same size of the input image and stacked in a row-wise fashion. It is interesting to notice how each \mathbf{p}_l has a digit-shape look, with a decreasing pixel intensity for increasing filter tap order L ; this indicates pixel-locality as an important factor for the creation of the final pixel intensity. Since each (reshaped) filter tap image in Fig. 7 pointwise-multiplies a shifted white input (noise) image of same size (to be finally aggregated), it is visible how the most influential filter content gathers around the digit shape for the digit-formation, and decreases its importance for higher shifts, meaning that the process is local on the pixel and there are no long term-influences. The same NSE and \mathbf{P} -profile is obtained with the input-output approach.

Once we have the filter tap matrix $\tilde{\mathbf{P}}$, we then learn the dual eigenvalues $\tilde{\lambda}_f$ by solving (42). The associated dual GSO $\mathbf{S}_f = \mathbf{V}^{-1} \text{Diag}(\tilde{\lambda}_f) \mathbf{V}$, achieving a corollary error [cf. (47)] $\varepsilon_c = 9 \times 10^{-2}$, is shown in Fig. 7(right), where we only display 2% of the most significant edges in absolute values⁵. The node label i indicates the index associated to λ_i . As in the previous experiment, we color the first half of the nodes (representing now the “low frequency” eigenvalues) with blue, and the other half with red. Similar results and conclusions can be made by following the input-output approach, where we obtain $\varepsilon_c = 3 \times 10^{-2}$ and the graph is similarly structured as the one in Fig. 7 (see Supplementary Material).

Together with the previous experiment, the following observations can be made:

- The connections between the eigenvalues do not follow the linear ordering as assumed by the traditional real-line interpretation (in that case, we would only have red-red

and blue-blue connections without interactions); this has consequences when designing graph filters based solely on the value of the λ_i , since now the concept of “bands” needs also to account for the topology.

- Because the dual graph represents the support of the GFT signals, we can now inspect which neighborhood is influential for a particular frequency during a convolution on \mathbf{S}_f ; this was not possible with the standard real-line interpretation, as the convolution operation was a simple pointwise multiplication.

All in all, these results confirm the commutative nature of the two branches of Fig. 1, thus rendering the dual convolution a preferred approach when $K < L$, or when \mathbf{S}_f and/or the GFT signals exhibit sparsity, in addition to delivering an elegant theoretical framework.

VI. CONCLUSION

In this work we proposed a convolution theorem which extends the classical convolution theorem in (graph) signal processing and the one related to time-varying filters. More precisely, we illustrate how a convolution in the primal graph domain can be redefined as a distinct convolution in the dual graph (frequency) domain, given a suitable filter parametrization. After illustrating the implications of such a theorem in terms of non-stationarity of signals, and generative models thereof, we devise an algorithmic approach based on subspace fitting and non-convex programming techniques to learn the dual graph from data when this is not a priori known. We evaluated the proposed theory and algorithms on synthetic data, as well as on real data.

While our current theoretical framework holds promise for practical applications in the future, there are notable challenges that merit further exploration. A significant gap lies in the absence of a one-shot procedure to construct the dual GSO directly from a primal GSO, along with potential graph signals associated with it. This limitation curtails the broader

⁵For a graph with $N = 324$ nodes there would be more than 50k edges possible.

applicability of the proposed theoretical insights. Nonetheless, multiple extensions of this work are possible. From an algebraic point of view, an interesting line of research would involve exploring the connections between Vandermonde and Hankel matrices, as well as with Krylov subspaces, potentially unveiling new algorithmic solutions to learn the dual eigenvalues. From a modeling point of view, an interesting extension of this work would include the node-varying graph filter coefficients also to be time-varying; holding promise for utilization in graph autoregressive models. In such cases, leveraging the basis expansion model technique across the temporal dimension becomes a potential avenue for further investigation. From an optimization point of view, the use of orthogonal polynomials might alleviate the ill-conditioning of the Vandermonde matrix.

Our hope is that in the coming years, further exploration and refinement of this research direction will reveal new insights and methodologies to process signals defined on graphs in a way previously unfeasible.

REFERENCES

- [1] A. Natali and G. Leus, "A general convolution theorem for graph data," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 48–52.
- [2] A. V. Oppenheim, J. R. Buck, and R. W. Schaffer, *Discrete-time signal processing*, vol. 2. Upper Saddle River, NJ: Prentice Hall, 2001.
- [3] H. Lütkepohl, *Introduction to Multiple Time Series Analysis*. Berlin, Heidelberg: Springer Science & Business Media, 2013.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [5] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*. Cambridge, USA: MIT Press, 1998, pp. 255–258.
- [6] D. I. Shuman, S. K. 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, Mar. 2013.
- [7] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.
- [8] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 4, pp. 736–751, Dec. 2018.
- [9] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *IEEE Trans. Signal Process.*, early access, Jan. 10, 2024, doi: 10.1109/TSP.2024.3349788.
- [10] D. Wei and Z. Yan, "Generalized sampling of graph signals with the prior information based on graph fractional Fourier transform," *Signal Process.*, vol. 214, 2024, Art. no. 109263.
- [11] G. Leus, S. Segarra, A. Ribeiro, and A. G. Marques, "The dual graph shift operator: Identifying the support of the frequency domain," *J. Fourier Anal. Appl.*, vol. 27, no. 3, pp. 1–20, 2021.
- [12] J. Shi and J. M. Moura, "Graph signal processing: Dualizing GSP sampling in the vertex and spectral domains," *IEEE Trans. Signal Process.*, vol. 70, pp. 2883–2898, 2022.
- [13] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1110–1121, May 1991.
- [14] S. Boyd, *Sequential Convex Programming*, in Lecture Notes. Stanford, CA, USA: Stanford Univ., 2008.
- [15] N. Saito, "How can we naturally order and organize graph Laplacian eigenvectors?" in *Proc. IEEE Statist. Signal Process. Workshop (SSP)*, Piscataway, NJ, USA: IEEE Press, 2018, pp. 483–487.
- [16] A. Cloninger, H. Li, and N. Saito, "Natural graph wavelet packet dictionaries," *J. Fourier Anal. Appl.*, vol. 27, no. 3, pp. 1–33, 2021.
- [17] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.
- [18] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017.
- [19] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Identifying the topology of undirected networks from diffused non-stationary graph signals," *IEEE Open J. Signal Process.*, vol. 2, pp. 171–189, 2021.
- [20] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- [21] G. B. Giannakis and C. Tepedelenlioglu, "Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels," *Proc. IEEE*, vol. 86, no. 10, pp. 1969–1986, Oct. 1998.
- [22] Z. Tang, R. C. Cannizzaro, G. Leus, and P. Banelli, "Pilot-assisted time-varying channel estimation for OFDM systems," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2226–2238, May 2007.
- [23] A. Natali, M. Coutino, and G. Leus, "Topology-aware joint graph filter and edge weight identification for network processes," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Piscataway, NJ, USA: IEEE Press, 2020.
- [24] A. Natali and G. Leus, "Blind polynomial regression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 1–5.
- [25] B. F. Green, "The orthogonal approximation of an oblique structure in factor analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.
- [26] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.
- [27] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," CVX. [Online]. Available: <https://cvxr.com/cvx>
- [28] L. Zhao, O. Gkountouna, and D. Pfoser, "Spatial auto-regressive dependency interpretable learning based on spatial topological constraints," *ACM Trans. Spatial Algorithms Syst.*, vol. 5, no. 3, pp. 1–28, 2019.