

Robust Graph Filter Identification and Graph Denoising from Signal Observations

Samuel Rey, *Student Member, IEEE*, Victor M. Tenorio, *Student Member, IEEE*, and Antonio G. Marques, *Senior Member, IEEE*

Abstract—When facing graph signal processing tasks, it is typically assumed that the graph describing the support of the signals is known. However, in many relevant applications *the available graph suffers from observational errors and perturbations*. As a result, any method that relies on the graph topology and ignores the presence of perturbations may yield suboptimal results. Motivated by this, we propose a novel approach for handling perturbations on the links of the graph and apply it to the problem of robust graph filter (GF) identification from input-output observations. Different from existing works, we formulate a non-convex optimization problem that operates in the vertex domain and jointly performs GF identification and graph denoising. As a result, on top of learning the desired GF, an estimate of the graph is obtained as a byproduct. To handle the resulting bi-convex problem, we design an algorithm that blends techniques from alternating optimization and majorization minimization, showing its convergence to a stationary point. The second part of the paper i) generalizes the design to a robust setup where several GFs are jointly estimated, and ii) introduces an alternative algorithmic implementation that reduces the computational complexity. Finally, the detrimental influence of the perturbations and the benefits resulting from the robust approach are numerically analyzed over synthetic and real-world datasets, comparing them with other state-of-the-art alternatives.

Index Terms—Graph Filter Identification, Graph Denoising, Robust Graph Signal Processing, Graph Perturbations.

I. INTRODUCTION

NOWADAYS, a significant number of datasets are defined over an irregular (heterogeneous) support that can be conveniently represented by a graph. As a result, the data at hand can be readily understood as graph signals (alternatively, network processes) whose structure and properties depend on the topology of the underlying graph. Illustrative examples include measurements from power, communications, social, biological, or financial networks, to name a few [1], [2], [3], [4], [5]. Characterizing and modeling graph and network processes entails a prevalent and relevant task that not only enhances our understanding of the data at hand, but also opens the door to more sophisticated processing and knowledge

extraction schemes. A popular approach within the graph signal processing (GSP) framework [4], [6], [7], [8], [9] is to model the data defined over the graph as graph signals generated by applying a (low-pass, smooth, bandlimited...) *graph filter* (GF) to a simple (sparse, white, constant...) input. GFs are topology-aware linear operators whose output can be interpreted as the outcome of a network diffusion or spreading process [10], [11], [12]. GFs can be expressed as polynomials of the graph-shift operator (GSO), a matrix encoding the topology of the graph, and on top of its theoretical interest, the task of GF identification is practically relevant to, e.g., understanding the dynamics of network diffusion processes [11], [13], [6], as well as explaining the structure of real-world datasets [14], [15]. It is worth mentioning that, while alternative definitions of GFs exist in the literature (see for example ARMA, Chebyshev, or spectral GFs [16]), based on the Cayley-Hamilton theorem [17], every GF can be expressed as a polynomial of a sufficiently large degree of the GSO, hence rendering the identification of polynomial GFs particularly relevant.

Since GSP is a relatively recent area of research, it is not surprising that most GSP works focus on how to harness the graph structure while assuming that the topology of the graph is perfectly known. Nevertheless, this assumption is unlikely to hold in many practical setups, where graphs suffer from *imperfections and perturbations*. When networks are given explicitly, perturbations may be due to observational noise and errors (e.g., link failures in power or wireless networks [18]). When in lieu of physical entities, the graphs model (statistical) pairwise relationships among the observed variables, they need to be inferred from the data [19], [20], [21], [22], [23]. Since this is a challenging (oftentimes ill-posed) task, the estimated graphs inherit the imperfections (limitations) of the graph learning scheme adopted (e.g., the thresholding operation implemented in correlation networks [2]). Intuitively, the presence of perturbations hinders any GSP scheme or GSP tool applied to the data. While this is true regardless of the task at hand, it is even more relevant for those involving spectral transforms and GFs, since eigenvectors and high-order matrix polynomials are more sensitive to errors in the matrix codifying the graph.

To address these challenges, this paper investigates the problem of estimating a GF from input-output signal pairs assuming that both the signals and the supporting graph have errors. The proposed approach is formulated in the vertex domain, avoiding the numerical instability of computing large polynomials and, at the same time, bypassing the challenges

Manuscript received 15 October 2022; revised 16 May 2023 and 20 July 2023; accepted 20 July 2023. This work was supported in part by the Spanish AEI under Grants PID2019-105032GB-I00, PID2022-136887NB-I00, FPU17/04520, and FPU20/05554 and in part by the Young Researchers R&D Project under ref. num. F861 (CAM and URJC). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. David I. Shuman. (Corresponding author: XXX.)

The authors are with the Department of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain (e-mail: samuel.rey.escudero@urjc.es; victor.tenorio@urjc.es; antonio.garcia.marques@urjc.es).

Digital Object Identifier 10.1109/TSP.2023.3300632

associated with robust *spectral* graph theory. We recast the robust estimation as a joint optimization problem where the GF identification objective is augmented with a graph-denoising regularizer, so that, on top of the desired GF, we also obtain an enhanced estimate of the supporting graph. The joint formulation leads to a non-convex bi-convex optimization problem, for which a provably-convergent efficient (alternating minimization) algorithm able to find an approximate solution is developed. Furthermore, to address scenarios where multiple GFs are present (e.g., when dealing with vector autoregressive (AR) spatio-temporal processes or in setups where nodes collect multi-feature vector measurements), we generalize our framework so that multiple GFs, all defined over the same graph, are jointly identified.

Contributions and outline. After reviewing preliminary concepts and results, we analyze the influence of edge perturbations in polynomial GFs and state the robust GF identification problem in Section III. Then, our main contributions are:

- 1) We elaborate on different types of edge perturbations and propose appropriate functions to harness the relation between the true and the perturbed GSO (Section III-A).
- 2) We formulate a non-convex optimization problem to jointly estimate the GF and the graph topology, develop an alternating optimization algorithm to solve it, and prove its convergence to a stationary point (Section IV).
- 3) We address the more general problem of jointly learning several GFs by leveraging the fact that they are polynomials of the same GSO (Section V).
- 4) We introduce an efficient implementation of the robust GF identification algorithm to handle graphs with a large number of nodes (Section VI).

The algorithms are evaluated numerically in Section VII, and some concluding remarks are provided in Section VIII. Last but not least, while we focus on GF identification from input-output pairs, the approach put forth in this paper can be generalized to other GSP tasks, which is a research path we plan to pursue in the near future.

A. Related works

Despite their theoretical and practical relevance, the number of robust GSP works is limited, due in part to the challenges emanating from the presence of graph perturbations.

Starting with the spectral domain, an initial approach considered in [24] employed a small perturbation analysis to characterize the impact of perturbations in the spectrum of the graph Laplacian. Similarly, [25], [26] studied the stability of spectral GFs when a small number of edges were rewired, upper-bounding the error stemming from the perturbations in the GSO. More recently, [27] analyzed the influence of rewiring a large number of edges, providing an enhanced upper-bound at the expense of restricting the analysis to low-pass GFs. While analyzing the influence of perturbations is pertinent, these works do not offer a robust formulation to handle perturbations, which is particularly relevant in the context of GFs.

Moving on to the vertex domain, [28], [29] postulates a graphon-based perturbation model and studies how pertur-

bations affect GFs. Unfortunately, the analysis is limited to filters of order 1. Later on, [30] combines structural equation models (SEMs) with total least squares (TLS) to jointly infer signals and perturbations. Nonetheless, as shown in the numerical evaluation (Section VII), this approach has limited applicability due to the SEM assumption. Another alternative is presented in [31], which, assuming that the support of the graph is known, estimates the weights of the network topology and the coefficients of the GF by solving a non-convex optimization problem using a sequential convex programming (SCP) algorithm. The main limitation is the assumption of the support of the true GSO being known, which may not be true in practical settings. In contrast with previous works, the approach put forth in this paper considers more lenient assumptions. We do not require perfect knowledge of the true graph (although it can be incorporated when available), and we consider that the observed data is explained through a GF without any assumption of the particular filter type (other than being a polynomial on the GSO), which includes the SEM as a particular case and involves dealing with polynomials of a perturbed matrix. In this direction is our preliminary work from [32], which also studies the problem of GF identification with imperfect topology knowledge. However, its contribution is limited since it only considers one type of perturbation, does not provide guarantees for the convergence of the optimization method, does not consider settings with several GFs, and the resulting method scales poorly with the size of the graph. All the previous limitations are addressed in this paper.

Finally, the presence of perturbations has also been considered in non-linear GSP tasks. An alternative definition of GFs robust to perturbations that replace the power of the GSO by different neighborhoods is proposed in [33], and the transferability of GFs when employed in graph neural networks is studied in [34], [35], [36], [37].

II. GSP PRELIMINARIES

Graphs and graph signals. Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ formed by the set of N nodes (vertices) collected in \mathcal{V} and the set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, such that $(i, j) \in \mathcal{E}$ if node i is connected to node j . The topology of \mathcal{G} can be represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, a sparse matrix whose elements A_{ij} are non-zero if and only if $(i, j) \in \mathcal{E}$. When \mathcal{G} is weighted, the entries A_{ij} capture the strength of the link between nodes i and j . Otherwise, the elements of \mathbf{A} are binary. Together with the graph \mathcal{G} , we focus on modeling the data as signals defined on the nodes in \mathcal{V} . Formally, a (nodal) *graph signal* is a function from the vertex set to the real field $x : \mathcal{V} \rightarrow \mathbb{R}$, which can be alternatively represented as an N -dimensional vector $\mathbf{x} \in \mathbb{R}^N$ whose i -th entry x_i denotes the value of the signal at node i . The foundational assumption of GSP is that incorporating the information of the graph \mathcal{G} provides an advantage when processing the signal \mathbf{x} . While a graph may capture any type of pair-wise relation, let us consider the simple example where \mathcal{G} captures the similarity between nodes. Then a high value of A_{ij} implies that the values of x_i and x_j are expected to be akin to each other, which can be exploited when processing graph signals.

Graph-shift operator (GSO). The GSO associated with a graph \mathcal{G} of N nodes is as a generic matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ that: i) captures the topology of the underlying graph and ii) represents a *local* and *linear* transformation that can be applied to graph signals defined over \mathcal{G} . The entries of \mathbf{S} satisfy $S_{ij} \neq 0$ only if $(i, j) \in \mathcal{E}$ or $i = j$ and, as a result, the application of the GSO to a graph signal involves mixing values among one-hop neighborhoods. Two typical choices for the GSO are the adjacency matrix \mathbf{A} and the graph Laplacian $\mathbf{L} := \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$ [4], [6], where $\text{diag}(\cdot)$ transforms a vector into a diagonal matrix and $\mathbf{1}$ is the vector of all ones. We assume that \mathbf{S} is diagonalized as $\mathbf{S} = \mathbf{V}\text{diag}(\boldsymbol{\lambda})\mathbf{V}^{-1}$, where the $N \times N$ matrix \mathbf{V} collects N orthogonal eigenvectors of \mathbf{S} , and the vector $\boldsymbol{\lambda}$ collects the associated eigenvalues. The matrix \mathbf{V}^{-1} is commonly adopted as the Graph Fourier Transform (GFT) for graph signals with $\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$ being the graph frequency representation of \mathbf{x} [5].

Graph filtering. GFs are topology-aware operators whose inputs and outputs are graph signals. More specifically, GFs implement a linear transformation that can be written as a polynomial of \mathbf{S}

$$\mathbf{H} = \sum_{r=0}^{N-1} h_r \mathbf{S}^r = \mathbf{V}\text{diag}(\boldsymbol{\Psi}\mathbf{h})\mathbf{V}^{-1} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}, \quad (1)$$

where $\mathbf{h} = [h_0, \dots, h_{N-1}]$ is the vector collecting the GF coefficients, and the second equality follows from the eigendecomposition of \mathbf{S} . The $N \times N$ Vandermonde matrix $\boldsymbol{\Psi}$ defined as $\Psi_{ij} := \lambda_i^{j-1}$ represents the GFT for GFs, and thus, $\tilde{\mathbf{h}} = \boldsymbol{\Psi}\mathbf{h}$ is the vector of size N representing the frequency response of \mathbf{H} [5], [11]. Since \mathbf{S}^r encodes the r -hop neighborhood of the graph, a graph signal given by $\mathbf{y} = \sum_{r=0}^{N-1} h_r \mathbf{S}^r \mathbf{x} = \mathbf{H}\mathbf{x}$ can be interpreted as a version of the input signal \mathbf{x} diffused across $N-1$ neighborhoods with h_r being the coefficients of the linear combination [11]. Moreover, there are scenarios where $h_r = 0$ for $r \geq R$. In those cases, the order of the filter is R and, if more convenient, \mathbf{h} and $\boldsymbol{\Psi}$ can be redefined so that the number of elements of \mathbf{h} (columns of $\boldsymbol{\Psi}$) is R in lieu of N .

Graph stationarity. A zero-mean random graph signal \mathbf{x} is said to be stationary on \mathcal{G} if its covariance matrix $\mathbf{C}_{\mathbf{x}} = \mathbb{E}[\mathbf{x}\mathbf{x}^T]$ is a positive-semidefinite matrix that can be written as a polynomial of the GSO [38]. In such a case, both $\mathbf{C}_{\mathbf{x}}$ and \mathbf{S} have the same eigenvectors and, as a consequence, the two matrices commute, i.e., $\mathbf{S}\mathbf{C}_{\mathbf{x}} = \mathbf{C}_{\mathbf{x}}\mathbf{S}$. A common example of stationary graph signals arises when \mathbf{x} is the output of a linear graph diffusion process whose input (initial condition) is a white signal $\boldsymbol{\nu}$ and whose diffusion dynamics can be accurately represented by a GF. Mathematically, if we have that $\mathbf{x} = \mathbf{H}\boldsymbol{\nu}$ with \mathbf{H} being a GF [cf. (1)] and $\mathbf{C}_{\boldsymbol{\nu}} = \mathbb{E}[\boldsymbol{\nu}\boldsymbol{\nu}^T] = \mathbf{I}$, it follows that the covariance of \mathbf{x} is $\mathbf{C}_{\mathbf{x}} = \mathbf{H}\mathbf{H}^T = \sum_{r=0, r'=0}^{N-1, N-1} h_r h_{r'} \mathbf{S}^{r+r'}$. Since the latter is a polynomial of the GSO, it follows that \mathbf{x} is stationary on \mathcal{G} .

A. GF identification from input-output pairs

In the context of linear operators, let us consider that we observe M input and output pairs $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_M]$ and $\mathbf{Y} := [\mathbf{y}_1, \dots, \mathbf{y}_M]$ whose relation is given by

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}, \quad (2)$$

with \mathbf{W} being a zero-mean random matrix (typically assumed to have i.i.d. entries) that accounts for noisy measurements and model inaccuracies. Note that if the interest is in M signals with F features described by the same \mathbf{H} , this is equivalent to considering MF different input-output pairs. Leveraging (2), the GF identification task amounts to using the input-output pairs to estimate \mathbf{H} under the model in (1), which, if the GSO \mathbf{S} is known, boils down to estimating the GF coefficients collected in $\mathbf{h} \in \mathbb{R}^N$.

Hence, we can approach the GF identification task in the node domain by solving the convex problem

$$\min_{\mathbf{h}} \left\| \mathbf{Y} - \sum_{r=0}^{N-1} h_r \mathbf{S}^r \mathbf{X} \right\|_F^2. \quad (3)$$

Leveraging the frequency definition of GFs in (1), we use the GFT matrices \mathbf{V}^{-1} and $\boldsymbol{\Psi}$ and the vectorization operation to rewrite the least-squares (LS) cost in (3) and obtain its (closed-form) solution as

$$\begin{aligned} \hat{\mathbf{h}} &= \underset{\mathbf{h}}{\text{argmin}} \left\| \text{vec}(\mathbf{Y}) - \text{vec}(\mathbf{V}\text{diag}(\boldsymbol{\Psi}\mathbf{h})\mathbf{V}^{-1}\mathbf{X}) \right\|_2^2 \\ &= \underset{\mathbf{h}}{\text{argmin}} \left\| \text{vec}(\mathbf{Y}) - ((\mathbf{V}^{-1}\mathbf{X})^T \otimes \mathbf{V}) \text{vec}(\text{diag}(\boldsymbol{\Psi}\mathbf{h})) \right\|_2^2 \\ &= \underset{\mathbf{h}}{\text{argmin}} \left\| \text{vec}(\mathbf{Y}) - ((\mathbf{V}^{-1}\mathbf{X})^T \odot \mathbf{V}) \boldsymbol{\Psi}\mathbf{h} \right\|_2^2 \\ &= \underset{\mathbf{h}}{\text{argmin}} \left\| \text{vec}(\mathbf{Y}) - \boldsymbol{\Theta}\mathbf{h} \right\|_2^2 = \boldsymbol{\Theta}^\dagger \text{vec}(\mathbf{Y}), \end{aligned} \quad (4)$$

where $\text{vec}(\cdot)$ denotes the vectorization operation, \otimes the Kronecker product, \odot the Khatri-Rao product, \dagger the pseudoinverse, $\boldsymbol{\Theta}$ is defined as the $NM \times N$ matrix $((\mathbf{V}^{-1}\mathbf{X})^T \odot \mathbf{V})\boldsymbol{\Psi}$, and we have used the property $\text{vec}(\mathbf{A}\mathbf{B}\mathbf{C}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B})$.

From (4) we observe that estimating \mathbf{H} is straightforward under the assumptions of: i) $\boldsymbol{\Theta}$ being full rank (i.e., the inputs are sufficiently rich) and ii) \mathbf{S} being perfectly known. However, the (critical) assumption in ii) does not hold in most practical settings. The remainder of the paper approaches the GF identification problem assuming imperfect knowledge of the GSO.

III. GF IDENTIFICATION WITH IMPERFECT GRAPH KNOWLEDGE

This section introduces and discusses the problem of estimating a GF $\mathbf{H} = \sum_{r=0}^{N-1} h_r \mathbf{S}^r$ from noisy input-output signal pairs $(\mathbf{X} \in \mathbb{R}^{N \times M}, \mathbf{Y} \in \mathbb{R}^{N \times M})$ assuming that we have access to an *imperfect* GSO $\tilde{\mathbf{S}} \in \mathbb{R}^{N \times N}$, which can be modeled as

$$\tilde{\mathbf{S}} = \mathbf{S} + \boldsymbol{\Delta}, \quad (5)$$

where $\mathbf{S} \in \mathbb{R}^{N \times N}$ represents the true GSO and $\boldsymbol{\Delta} \in \mathbb{R}^{N \times N}$ is a *perturbation matrix*. Before discussing models for the perturbation matrix, we find illustrative to demonstrate the impact of $\boldsymbol{\Delta}$ on the GSP problem at hand.

As pointed out in the introduction, the presence of uncertainties in the topology of \mathcal{G} is particularly relevant when dealing with GFs. Indeed, due to the polynomial definition of \mathbf{H} , even small perturbations can lead to significant errors when $\tilde{\mathbf{S}}$ (and not \mathbf{S}) is used as the true GSO. To see this more clearly, Figs. 1(a)-(c) provide an example that illustrates how the errors encoded in $\boldsymbol{\Delta}$ propagate across different matrix powers, demonstrating that the discrepancies between $\tilde{\mathbf{S}}^r$ and \mathbf{S}^r increase swiftly as the power r grows. More rigorously, let

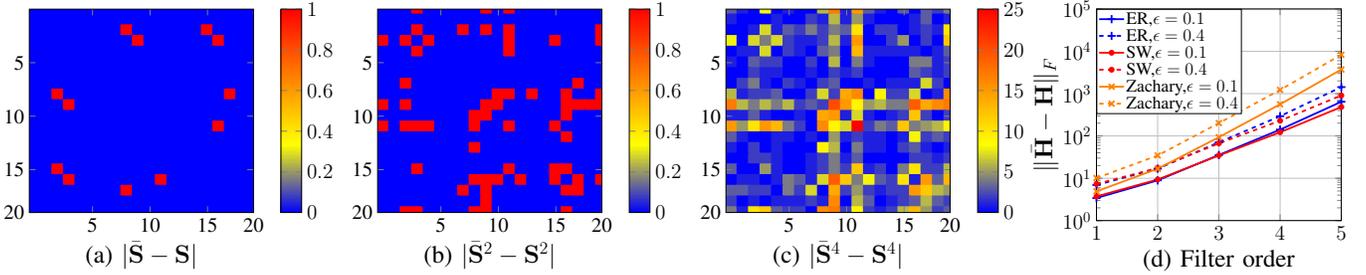


Fig. 1: (a)-(c) Absolute error for different powers of the matrix \mathbf{S} and its perturbed version $\bar{\mathbf{S}}$. The true GSO is the adjacency matrix of an Erdős-Rényi (ER) graph with link probability of 0.15, and $\bar{\mathbf{S}}$ is perturbed by creating and destroying links independently with a probability of 0.05. (d) Error between \mathbf{H} and $\bar{\mathbf{H}}$, which are polynomials of \mathbf{S} and $\bar{\mathbf{S}}$ respectively, as the order of the filter increases. The graphs considered are ER, Small World, and the Zachary Karate club graph [39]. We consider different perturbation probabilities ϵ and set $\mathbf{h} = \mathbf{1}$.

C be a positive constant such that $\|\mathbf{S}\| \leq C$ and $\|\bar{\mathbf{S}}\| \leq C$, and define $\bar{\mathbf{H}} := \sum_{r=0}^{N-1} h_r \bar{\mathbf{S}}^r$. Then, according to [34], [35], the error generated by the perturbations is upper-bounded by

$$\|\bar{\mathbf{H}} - \mathbf{H}\| \leq \sum_{r=1}^{N-1} |h_r| \|\bar{\mathbf{S}}^r - \mathbf{S}^r\| \leq \sum_{r=1}^{N-1} |h_r| r C^{r-1} \|\Delta\|, \quad (6)$$

where $\|\cdot\|$ denotes any matrix norm (e.g., the Frobenius norm). Put in words, the upper-bound of the error between the true \mathbf{H} and the perturbed $\bar{\mathbf{H}}$ increases *exponentially* with the degree of the GF. Furthermore, as we show numerically in Fig. 1(d), not only the upper-bound but the actual error $\|\bar{\mathbf{H}} - \mathbf{H}\|_F$ may also grow exponentially with the filter order, a behavior observed both in synthetic and real-world graphs.

From the previous discussion, it is not surprising that the imperfect knowledge of the graph topology is also relevant when estimating the filter coefficients. In fact, ignoring the errors in Δ and attempting to estimate \mathbf{h} solving (4) when $\bar{\mathbf{S}}$ is used in lieu of the true (unknown) \mathbf{S} leads to a poor solution, as we illustrate numerically in Section VII. Motivated by this, we approach the GF identification problem from a robust perspective by taking into account the imperfect knowledge of the GSO. The resultant robust estimation task is formally stated next.

Problem 1: Let \mathcal{G} be a graph with N nodes, let $\mathbf{S} \in \mathbb{R}^{N \times N}$ be the true (unknown) GSO, and let $\bar{\mathbf{S}} \in \mathbb{R}^{N \times N}$ be the perturbed (observed) GSO. Moreover, let $\mathbf{X} \in \mathbb{R}^{N \times M}$ and $\mathbf{Y} \in \mathbb{R}^{N \times M}$ collect the M observed input and output signals defined over \mathcal{G} and related by the model in (2). Our goal is to use the triplet $(\mathbf{X}, \mathbf{Y}, \bar{\mathbf{S}})$ to: i) learn the GF \mathbf{H} that best fits the model in (2) and ii) recover an enhanced estimation of \mathbf{S} . To that end, we make the following assumptions: (AS1) \mathbf{H} is a polynomial of \mathbf{S} [cf. (1)]. (AS2) \mathbf{S} and $\bar{\mathbf{S}}$ are close according to some metric $d(\mathbf{S}, \bar{\mathbf{S}})$, i.e., the observed perturbations are “small” in some sense.

On top of the previous two assumptions, we also consider that the norm of the noise observation matrix \mathbf{W} in (2) is small, which is a standard procedure when dealing with noise in the observations. Similar to classical GF identification approaches, (AS1) limits the degrees of freedom of the linear operator in (2). However, the fact of the true \mathbf{S} being unknown adds uncertainty to the problem and, as a result, additional signal observations are required to achieve an identification performance comparable to the one obtained when $\bar{\mathbf{S}} = \mathbf{S}$. Regarding the recovery of the true GSO, (AS2) accounts

for the hypothesis that $\bar{\mathbf{S}}$ is a perturbed observation of \mathbf{S} and, hence, matrices \mathbf{S} and $\bar{\mathbf{S}}$ are not extremely different. Intuitively, this guarantees that “some” information about the true \mathbf{S} is available, so that (AS1) can be effectively leveraged. While not exploited in our formulation, additional assumptions constraining the GSO could also be incorporated into the problem. Finally, the metric $d(\cdot, \cdot)$ employed to quantify the similarity between \mathbf{S} and $\bar{\mathbf{S}}$ should depend on the model for the perturbation Δ , a subject that is briefly discussed next.

A. Modeling graph perturbations

The development and analysis of graph perturbation models that combine practical relevance and analytical tractability constitutes an interesting yet challenging open line of research [28], [29]. Due to its flexibility and tractability, this paper considers an additive perturbation model [cf. (5)], so that the focus is constrained to understanding the structural (statistical) properties of the matrix $\Delta = \bar{\mathbf{S}} - \mathbf{S}$.

Consider first the case where perturbations only *create or destroy links* independently. If \mathcal{G} is an *unweighted graph*, a simple approach is to consider perturbations modeled as independent Bernoulli variables with possibly different creation/destruction probabilities. In this case, the entries of Δ would be

$$\Delta_{ij} = \begin{cases} 1 & \text{if link } (i, j) \text{ is created,} \\ -1 & \text{if link } (i, j) \text{ is destroyed,} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Since Δ models the creation and destruction of links, it is worth noting that $\Delta_{ij} = 1$ only if $S_{ij} = 0$ and $\Delta_{ij} = -1$ only if $S_{ij} = 1$. In the more general case of \mathcal{G} being a *weighted graph*, $\Delta_{ij} = -S_{ij}$ destroys an existing link while $\Delta_{ij} = z$ creates a new link. Here, z is a random value sampled from a particular distribution (typically mimicking the weight distribution of the true \mathbf{S}). When facing this type of perturbations, a suitable distance function is the ℓ_0 pseudonorm

$$d(\mathbf{S}, \bar{\mathbf{S}}) = \|\mathbf{S} - \bar{\mathbf{S}}\|_0, \quad (8)$$

with the ℓ_1 norm $\|\mathbf{S} - \bar{\mathbf{S}}\|_1$ being a prudent convex relaxation.

Alternatively, rather than creating or destroying links, perturbations may represent uncertainty over the edge weights. This entails the support of matrix Δ matching that of \mathbf{S} and $\bar{\mathbf{S}}$, and the non-zero entries of Δ being sampled from a distribution that models the observation noise. For example,

if the noise is zero-mean, Gaussian and white, it holds that $\Delta_{ij} \sim \mathcal{N}(0, \sigma^2)$ when $\mathbf{S}_{ij} \neq 0$ and $\Delta_{ij} = 0$ when $\mathbf{S}_{ij} = 0$. Under this setting, an appropriate distance metric is given by

$$d(\mathbf{S}, \bar{\mathbf{S}}) = \|\mathbf{S}_\mathcal{E} - \bar{\mathbf{S}}_\mathcal{E}\|_2^2, \quad (9)$$

where $\mathbf{S}_\mathcal{E}$ and $\bar{\mathbf{S}}_\mathcal{E}$ are vectors containing the non-zero entries (edges) in \mathbf{S} and $\bar{\mathbf{S}}$. Additionally, one can have setups where the two types of perturbations are present. That is, perturbations may create and destroy links while the actual value of the existing links is also uncertain. In such a case, a combination of ℓ_1 and ℓ_2 norms like in elastic nets [40] is adequate.

The models described thus far assume that perturbations are independent across edges, but in some scenarios perturbations may be correlated. Consider for example a communication network. If the power supply of a node stalls, the signal-to-noise ratio of all its links will be poor, and hence, links involving that node will be more likely to fail. Perturbations dependent across links can be modeled by means of a multivariate correlated Bernoulli distribution, an Ising model, or more sophisticated random graph models [41]. When prior information about the dependence of the perturbations is available, it can be incorporated into the function $d(\mathbf{S}, \bar{\mathbf{S}})$ to extract the information encoded in $\bar{\mathbf{S}}$ more effectively. Additionally, in applications where perturbations due to adversarial attacks are a concern, this distance can be designed to minimize the harmful influence of attackers.

IV. ROBUST GF IDENTIFICATION

This section presents the optimization problem and the proposed algorithm to estimate \mathbf{H} and \mathbf{S} under the setting described in Problem 1. Given the matrices \mathbf{X} , \mathbf{Y} , and $\bar{\mathbf{S}}$ (recall that $\bar{\mathbf{S}} = \mathbf{S} + \mathbf{\Delta}$), we approach the robust GF identification task by means of the following non-convex optimization

$$\begin{aligned} \hat{\mathbf{H}}, \hat{\mathbf{S}} = \operatorname{argmin}_{\mathbf{H}, \mathbf{S}} \quad & \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \lambda d(\mathbf{S}, \bar{\mathbf{S}}) + \beta \|\mathbf{S}\|_0 \\ \text{s. to:} \quad & \mathbf{S} \in \mathcal{S}, \quad \mathbf{S}\mathbf{H} = \mathbf{H}\mathbf{S}, \end{aligned} \quad (10)$$

where s. to stands for subject to and \mathcal{S} represents a (desired) family of GSOs such as the set of adjacency matrices with no self-loops (\mathcal{S} is the set of matrices with non-negative entries whose diagonal entries are zero) or the set of combinatorial graph Laplacians (matrices with non-positive off-diagonal entries and zero row-sum). The first term in the objective promotes the linear input-output relation in (2), encouraging the norm of $\mathbf{W} = \mathbf{Y} - \mathbf{H}\mathbf{X}$ to be small. The use of the Frobenius norm is well-justified when the observation noise is Gaussian and white, but other types of noise could be accommodated by using a different norm. The second term incorporates the assumption (AS2) as a regularizer to obtain an estimate $\hat{\mathbf{S}}$ that is related to the given GSO $\bar{\mathbf{S}}$. The ℓ_0 pseudonorm in the third term promotes the sparsity of $\hat{\mathbf{S}}$, a canonical assumption when learning the graph topology [21], [23]. Clearly, if additional information about \mathbf{S} is available, it can be incorporated into (10), either as a regularizer (e.g., a statistical prior quantifying the log-likelihood of a class of GSOs) or as a constraint that *must* be satisfied (e.g., the GSO being symmetric). The latter is indeed the role of $\mathbf{S} \in \mathcal{S}$ in (2). Finally, the (key) constraint $\mathbf{S}\mathbf{H} = \mathbf{H}\mathbf{S}$ captures the fact

that \mathbf{H} is a polynomial of the true \mathbf{S} (AS1), which constitutes the only assumption about the GF being estimated. Note first that the constraint is pertinent, if \mathbf{H} is a polynomial of \mathbf{S} , then \mathbf{H} and \mathbf{S} have the same eigenvectors and, as a result, their product commutes [11]. More importantly for the GF-identification at hand, when the GSO is perfectly known the model

$$\mathbf{H} = h_0 \mathbf{I} + h_1 \mathbf{S} + \dots + h_{N-1} \mathbf{S}^{N-1} \quad (11)$$

is linear in the unknown \mathbf{h} . As a result, a formulation that estimates \mathbf{h} directly (as carried out in classical non-robust approaches) is well-motivated. However, when both \mathbf{h} and \mathbf{S} are unknown, the powers of \mathbf{S} render the model highly non-linear, challenging the development of a tractable solution that jointly estimates \mathbf{h} and \mathbf{S} . Our formulation bypasses this problem by recasting the optimization variables as \mathbf{H} and \mathbf{S} , leading to the (more tractable) bilinear constraint in (10) and, at the same time, harnessing the fact that \mathbf{H} is a polynomial of \mathbf{S} . Nonetheless, if learning \mathbf{h} is the ultimate goal, this can be readily achieved from $\hat{\mathbf{H}}$ and $\hat{\mathbf{S}}$ by vectorizing (11) as

$$\operatorname{vec}(\mathbf{H}) = (\operatorname{vec}(\mathbf{I}), \operatorname{vec}(\mathbf{S}), \dots, \operatorname{vec}(\mathbf{S}^{N-1})) \mathbf{h}, \quad (12)$$

and then finding the filter coefficients as

$$\hat{\mathbf{h}} = (\operatorname{vec}(\mathbf{I}), \operatorname{vec}(\hat{\mathbf{S}}), \dots, \operatorname{vec}(\hat{\mathbf{S}}^{N-1}))^\dagger \operatorname{vec}(\hat{\mathbf{H}}). \quad (13)$$

The approach put forth in (10) has two main advantages. First, while most works formulate the recovery of the GF in the spectral domain relying on the eigendecomposition of the GSO, our formulation operates in the vertex domain. Working on the spectral domain would imply, first of all, that the GSO \mathbf{S} is diagonalizable, which may not be the case for directed graphs. It also requires finding the Vandermonde GFT matrix Ψ . Since this matrix involves high-order polynomials of the eigenvalues of the GSO, it is prone to numerical instability and error accumulation [6]. Even if approaches that bypass this issue by estimating the graph-frequency response $\hat{\mathbf{h}} = \Psi \mathbf{h}$ in lieu of \mathbf{h} are adopted, the estimation would still be challenging since they require computing the eigenvectors \mathbf{V} , which are known to be highly sensitive to errors in the GSO (especially those associated with small eigenvalues)[42], [24]. On top of this, characterizing the spectral errors and incorporating those to the optimization is not a trivial task. The second advantage emanates from casting the true GSO \mathbf{S} as an explicit optimization variable. As already explained, this approach is robust to error accumulation and facilitates the incorporation of the (additive) effect of the perturbations into the optimization. An additional benefit is that we obtain a denoised version (enhanced estimation) of the true GSO, which can be practically relevant in most real-world applications.

In a nutshell, in the context of robust GF identification, choosing a formulation that: i) works entirely in the vertex domain, ii) considers \mathbf{S} as an explicit optimization variable, and iii) codifies the GF structure via the constraint $\mathbf{H}\mathbf{S} = \mathbf{S}\mathbf{H}$, exhibits multiple advantages. However, it must be noted that the number of optimization variables is larger than in classical approaches (adding computational complexity), and the bilinear filtering constraint $\mathbf{H}\mathbf{S} = \mathbf{S}\mathbf{H}$, while more tractable than

its polynomial counterpart, is still non-convex. Alternatives to deal with these issues are discussed in later sections.

A. Alternating minimization for robust GF identification

This section presents a systematic efficient approach to find an approximate solution to (10). Since the goal is to design specific algorithms, from this section onwards, we particularize the GSO distance to $d(\mathbf{S}, \bar{\mathbf{S}}) = \|\mathbf{S} - \bar{\mathbf{S}}\|_0$, so that, according to the discussion in Section III-A, the focus is on graph perturbations that create and destroy links. In addition to its practical relevance, the reason for choosing the ℓ_0 pseudonorm as a distance is also motivated by its more intricate (challenging) structure. Indeed, the algorithms presented next can be easily adapted to (more tractable) distances associated with alternative perturbation models discussed in Section III-A. Having clarified this, the main obstacle to solving (10) is its lack of convexity, which emanates from two different *sources*: (s1) the ℓ_0 pseudonorms in the objective, and (s2) the bilinear constraint involving \mathbf{S} and \mathbf{H} . Next, we explain the strategy adopted to solve (10) via a succession of convex problems.

- Regarding (s1), it is customary to rely on the ℓ_1 norm as a convex surrogate to promote sparsity. However, in relevant settings this norm may not be enough to obtain sparse solutions [43], [44] and more sophisticated (non-convex) alternatives may be preferred. The one chosen in this paper is to approximate the ℓ_0 pseudonorm of a generic matrix $\mathbf{Z} \in \mathbb{R}^{I \times J}$ using the logarithmic penalty

$$\|\mathbf{Z}\|_0 \approx r_\delta(\mathbf{Z}) := \sum_{i=1}^I \sum_{j=1}^J \log(|Z_{ij}| + \delta), \quad (14)$$

with δ being a small positive constant [43]. The non-convexity of the logarithm can be handled efficiently by relying on a majorization-minimization approach (MM) [45], which considers an iterative linear approximation via a first order Taylor approximation, leading to an iterative reweighted ℓ_1 norm. It is worth noting that, since we will consider an iterative algorithm to deal with the bilinearity of (10), the iterative nature of the reweighted ℓ_1 norm will not impose a significant computational burden. Details on the exact form of this sparse regularizer will be provided soon (see equation (18) in the graph denoising step).

- To deal with the bilinear terms in (s2), we adopt an alternating optimization approach [46] resulting in an iterative algorithm where the optimization variables \mathbf{H} and \mathbf{S} are updated in *two separate iterative steps*. At each step, we optimize over one of the optimization variables with the other remaining fixed, resulting in two simpler problems that can be solved efficiently. The details about the specific steps will be provided shortly (see Step 1 and Step 2 below and, in particular, equations (16) and (18)).

Taking into account these considerations, the first task to implement our approach is to rewrite the problem in (10) as

$$\min_{\mathbf{S} \in \mathcal{S}, \mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \lambda r_{\delta_1}(\mathbf{S} - \bar{\mathbf{S}}) + \beta r_{\delta_2}(\mathbf{S}) + \gamma \|\mathbf{S}\mathbf{H} - \mathbf{H}\mathbf{S}\|_F^2, \quad (15)$$

Algorithm 1: Robust GF id. with graph denoising.

Input: $\mathbf{X}, \mathbf{Y}, \bar{\mathbf{S}}, \gamma, \lambda, \beta, \delta_1, \delta_2$
Output: $\hat{\mathbf{H}}, \hat{\mathbf{S}}$.
1 Initialize $\mathbf{S}^{(0)}$ as $\mathbf{S}^{(0)} = \bar{\mathbf{S}}$.
2 **for** $t = 0$ to $t_{max} - 1$ **do**
3 Compute $\mathbf{H}^{(t+1)}$ by solving (17) fixing $\mathbf{S}^{(t)}$.
4 Update $\bar{\Omega}^{(t)}$ and $\bar{\Omega}^{(t)}$ as in (19).
5 Compute $\mathbf{S}^{(t+1)}$ by solving (18) using $\mathbf{H}^{(t+1)}$, $\bar{\Omega}^{(t)}$, and $\bar{\Omega}^{(t)}$.
6 **end**
7 $\hat{\mathbf{H}} = \mathbf{H}^{(t_{max})}$, $\hat{\mathbf{S}} = \mathbf{S}^{(t_{max})}$.

where we recall that $r_\delta(\cdot)$ was introduced in (14). Note that: i) the logarithmic penalty has also been used to promote sparsity in the term $\mathbf{S} - \bar{\mathbf{S}}$ since we selected the ℓ_0 pseudonorm as the distance between \mathbf{S} and $\bar{\mathbf{S}}$; and ii) the constraint $\mathbf{S}\mathbf{H} = \mathbf{H}\mathbf{S}$ was relaxed and rewritten as a regularizer, a formulation more amenable to an alternating optimization approach.

The next task is to solve (15) by means of an iterative algorithm that blends techniques from alternating optimization and MM approaches. Specifically, for a maximum of t_{max} iterations, we run the following two steps at each iteration $t = 0, \dots, t_{max} - 1$.

Step 1: GF Identification. We estimate the block of N^2 variables collected in \mathbf{H} while the current estimate of the GSO, denoted as $\mathbf{S}^{(t)}$, remains fixed. This results in the convex optimization problem

$$\mathbf{H}^{(t+1)} = \arg \min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \gamma \|\mathbf{S}^{(t)}\mathbf{H} - \mathbf{H}\mathbf{S}^{(t)}\|_F^2, \quad (16)$$

an LS minimization whose closed-form solution is

$$\text{vec}(\mathbf{H}^{(t+1)}) = (\mathbf{X}\mathbf{X}^\top \otimes \mathbf{I} + \gamma(\mathbf{S}\mathbf{S}^\top \oplus \mathbf{S}^\top \mathbf{S} - \mathbf{S}^\top \otimes \mathbf{S} - \mathbf{S} \otimes \mathbf{S}))^{-1} \times (\mathbf{X} \otimes \mathbf{I}) \text{vec}(\mathbf{Y}). \quad (17)$$

Here, \otimes is the Kronecker product, \oplus is the Kronecker sum, and \mathbf{I} is the identity matrix of size $N \times N$. Also note that (17) omitted the iteration superscript in $\mathbf{S}^{(t)}$ to alleviate notation.

Step 2: Graph Denoising. Following an MM scheme, we optimize an upper bound of (15) where the logarithmic penalties are linearized. Then, we estimate the block of N^2 variables collected in \mathbf{S} while the current estimate of the GF $\mathbf{H}^{(t+1)}$ remains fixed. This yields

$$\mathbf{S}^{(t+1)} = \arg \min_{\mathbf{S} \in \mathcal{S}} \sum_{i=1}^N \sum_{j=1}^N (\lambda \bar{\Omega}_{ij}^{(t)} |S_{ij} - \bar{S}_{ij}| + \beta \Omega_{ij}^{(t)} |S_{ij}|) + \gamma \|\mathbf{S}\mathbf{H}^{(t+1)} - \mathbf{H}^{(t+1)}\mathbf{S}\|_F^2, \quad (18)$$

where $\bar{\Omega}^{(t)}$ and $\Omega^{(t)}$ are the derivatives of the logarithm function for the Taylor approximation. These are computed in an entry-wise fashion based on the GSO estimate from the previous iteration as

$$\bar{\Omega}_{ij}^{(t)} = (|S_{ij}^{(t)} - \bar{S}_{ij}| + \delta_1)^{-1}, \quad \Omega_{ij}^{(t)} = (|S_{ij}^{(t)}| + \delta_2)^{-1}, \quad (19)$$

where δ_1 and δ_2 are small positive constants to prevent a division by 0. Recall that if a different function $d(\mathbf{S}, \bar{\mathbf{S}})$ were considered then the terms related to the reweighted ℓ_1 norm of $\mathbf{S} - \bar{\mathbf{S}}$ would be replaced by the selected function.

A relevant element in the proposed algorithm is the weight γ . If γ is set to a value that is too large, the GF estimated in the first iteration $\mathbf{H}^{(1)}$ will be an (almost exact) polynomial of $\bar{\mathbf{S}}$ so that the algorithm will converge quickly to the same solution as that of the non-robust design [cf. (10) with $\mathbf{S} = \bar{\mathbf{S}}$]. On the other hand, if γ is too close to zero the two problems decouple and the solution converges quickly to that of the two separated problems [cf. (16) and (18) with $\gamma = 0$]. In this context, schemes that start with a small γ to encourage the exploration during the warm-up phase, and then increase γ as the iteration index grows to guarantee that the final $\hat{\mathbf{H}}$ and $\hat{\mathbf{S}}$ commute are a suitable alternative for the setup at hand. In any case, the value of the parameter must be chosen carefully, paying attention both to the initial value of γ and to the strategy followed to increase it. The approach that works best is subject to the subtleties of the application at hand and, in this work, we dealt with it as a hyperparameter tuning task.

The overall alternating algorithm is summarized in Alg. 1, where a fixed number of iterations is considered. The algorithm starts by initializing the GSO as $\mathbf{S}^{(0)} = \bar{\mathbf{S}}$ (other options, such as setting $\mathbf{S}^{(0)}$ to the all 1s matrix, could also be used), and then, it iterates between Steps 1 and 2 for a fixed number of epochs (or until some stopping criterion is met). In this regard, a key feature of the algorithm is its guaranteed convergence to a stationary point, as is formally stated next.

Theorem 1. *Denote as $f(\mathbf{H}, \mathbf{S})$ the objective function in (15), and let \mathcal{Z}^* be the set of stationary points of f . Let $\mathbf{z}^{(t)} = [\text{vec}(\mathbf{H}^{(t)})^\top, \text{vec}(\mathbf{S}^{(t)})^\top]^\top$ represent the solution provided by Alg. 1 after t iterations. Assuming that i) the GSO does not have repeated eigenvalues and ii) every row of $\tilde{\mathbf{X}} = \mathbf{V}^{-1}\mathbf{X}$ has at least one nonzero entry, then $\mathbf{z}^{(t)}$ converges to a stationary point of f as t goes to infinity, i.e.,*

$$\lim_{t \rightarrow \infty} d(\mathbf{z}^{(t)} | \mathcal{Z}^*) = 0,$$

with $d(\mathbf{z} | \mathcal{Z}^*) := \min_{\mathbf{z}^* \in \mathcal{Z}^*} \|\mathbf{z} - \mathbf{z}^*\|_2$.

The proof relies on the convergence results shown in [47, Th. 1b] and the details are provided in App. A. Note that the convergence of the algorithm was not self-evident since the original optimization problem in (15) is non-convex and Step 2 is minimizing an upper-bound of the original objective function. The sufficient conditions in i) and ii) guarantee that every graph frequency is excited so that the GF is identifiable and (16) has a unique solution, which is a requirement for convergence (see Prop. 1 in App. A for details). Clearly, condition ii) is fulfilled even for $M = 1$ if all the entries of the vector $\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$ are nonzero. Alternatively, when $M > 1$ and ii) is satisfied, condition i) can be relaxed (for further details, see App. A).

Finally, one drawback of the proposed robust GF identification algorithm is that the optimization problems in (16) and (18) may be slow when dealing with large graphs, since its computational complexity scales with the number of nodes as N^7 . However, we will mitigate this issue by introducing an efficient implementation that reduces the computational complexity of the overall algorithm (see Section VI).

B. Leveraging stationary observations

The alternating convex approximation in Alg. 1 exploits the fact that \mathbf{X} and \mathbf{Y} are linearly related via \mathbf{H} , which is a polynomial of \mathbf{S} . However, in setups where the perturbations in $\bar{\mathbf{S}}$ are very large, obtaining accurate estimates of \mathbf{S} and \mathbf{h} from $\hat{\mathbf{H}}$ may still be challenging. One alternative to overcome this issue is to leverage the additional structure potentially present in our data. Indeed, as detailed in the introduction, it is common to consider setups where the signals exhibit additional properties depending on the supporting graph, with notable examples including graph-bandlimited signals [4], [5], diffused sparse graph signals [13], [14], or graph stationary signals [38], [48], [22]. Clearly, incorporating such additional information into the optimization problem would enhance its estimation performance.

This section explores this path, restricting our attention to the case where the observed signals are stationary on \mathcal{G} . The motivation for this decision is that, due to the tight connection between graph-stationary signals and GFs (see Section II), the formulation in (15) and Alg. 1 require relatively minor modifications to incorporate the assumption of \mathbf{X} and \mathbf{Y} being stationary on \mathbf{S} , leaving the incorporation of additional signal models as future work. To formulate the updated problem, recall that the covariance matrix of a stationary graph signal can be expressed as a polynomial of the GSO (see Section II) and, as a consequence, both matrices commute. Therefore, incorporating stationarity calls for modifying (15) as

$$\begin{aligned} \min_{\mathbf{S} \in \mathcal{S}, \mathbf{H}} \quad & \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \lambda r_{\delta_1}(\mathbf{S} - \bar{\mathbf{S}}) + \beta r_{\delta_2}(\mathbf{S}) + \gamma \|\mathbf{S}\mathbf{H} - \mathbf{H}\mathbf{S}\|_F^2 \\ \text{s. to :} \quad & \|\mathbf{C}_y \mathbf{S} - \mathbf{S} \mathbf{C}_y\|_F^2 \leq \epsilon_y, \|\mathbf{C}_x \mathbf{S} - \mathbf{S} \mathbf{C}_x\|_F^2 \leq \epsilon_x, \end{aligned} \quad (20)$$

where \mathbf{C}_y and \mathbf{C}_x denote the covariance matrices of \mathbf{Y} and \mathbf{X} , respectively. If the covariances are perfectly known, then the corresponding parameters ϵ_y and ϵ_x are set to zero. Alternatively, if the \mathbf{C}_y and \mathbf{C}_x are the sample estimates of the true covariances, then the values of ϵ_y and ϵ_x must be selected based on the quality of the estimators (accounting, e.g., for the number of available observations M).

The constraints in (20) capture the graph-stationarity assumption by promoting the commutativity with the true GSO. Therefore, such constraints are considered in the graph denoising step [cf. (18)]. In addition, since \mathbf{C}_y , \mathbf{C}_x and \mathbf{H} are all polynomials of \mathbf{S} , the equalities $\mathbf{C}_y \mathbf{H} = \mathbf{H} \mathbf{C}_y$ and $\mathbf{C}_x \mathbf{H} = \mathbf{H} \mathbf{C}_x$ must hold as well, so it is also possible to augment the GF identification step [cf. (16)] with the corresponding constraints. While in the interest of brevity, we do not spell out all the possible formulations here, the impact of several of these alternatives is numerically analyzed in Section VII. Finally, it is important to note that, since the stationarity constraints are quadratic and convex, the convergence described in Theorem 1 also holds true for the iterative algorithm associated with (20).

V. JOINT ROBUST IDENTIFICATION OF MULTIPLE GFs

In Section IV, we approached the problem of identifying a single GF \mathbf{H} defined over a single graph \mathcal{G} . However, in a variety of situations we encounter several signals (or

signals with multiple features) whose dynamics are modeled by *different* processes over the same graph \mathcal{G} . Consider for example a network of weather stations measuring the temperature, humidity, and wind speed. Each of these measurements corresponds to observations of a different process, all of them taking place over a common graph. Intuitively, since all the GFs are related by the underlying graph \mathcal{G} , we propose a *joint* GF identification approach that exploits this relationship to enhance the quality of the estimation. We focus first on the case where the input-output signals associated with each GF (graph process) are observed separately. Later in the section, we address a slightly more involved case where the GFs model the (AR) dynamics of a time-varying graph signal and, as a result, the observed signals are intertwined.

Consider a set of K unknown GFs $\{\mathbf{H}_k\}_{k=1}^K$, all represented by $N \times N$ matrices and defined over the graph \mathcal{G} . To be consistent with Problem 1, we assume that: i) the true \mathbf{S} is unknown and only the perturbed version $\bar{\mathbf{S}}$ is available; ii) all \mathbf{H}_k are polynomials of the *same* GSO \mathbf{S} ; and iii) for each k , matrices $\mathbf{X}_k \in \mathbb{R}^{N \times M_k}$ and $\mathbf{Y}_k \in \mathbb{R}^{N \times M_k}$ collect the observed input and output graph signals and are related via

$$\mathbf{Y}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{W}_k, \quad (21)$$

with $\mathbf{H}_k = \sum_{r=0}^{N-1} h_{r,k} \mathbf{S}^r$ and \mathbf{W}_k being a white random matrix capturing observation noise and model inaccuracies. Then, we aim at estimating the GFs $\{\mathbf{H}_k\}_{k=1}^K$ in a joint fashion while taking into account the inaccuracies in the topology of \mathcal{G} . This is summarized in the following problem statement.

Problem 2: Let \mathcal{G} be a graph with N nodes, let $\mathbf{S} \in \mathbb{R}^{N \times N}$ be the true (unknown) GSO, and let $\bar{\mathbf{S}} \in \mathbb{R}^{N \times N}$ be the perturbed (observed) GSO. Moreover, let $\mathbf{X}_k \in \mathbb{R}^{N \times M_k}$ and $\mathbf{Y}_k \in \mathbb{R}^{N \times M_k}$ collect the M_k observed input and output signals associated with $k = 1, \dots, K$ network processes, all defined over \mathcal{G} and adhering to the model in (21). Our goal is to use $\{\mathbf{X}_k\}_{k=1}^K$, $\{\mathbf{Y}_k\}_{k=1}^K$, and $\bar{\mathbf{S}}$ to learn the K GFs $\{\mathbf{H}_k\}_{k=1}^K$ that best fit the data, along with an enhanced estimation of \mathbf{S} . To that end, we make the following assumptions: (AS2) \mathbf{S} and $\bar{\mathbf{S}}$ are close according to some metric $d(\mathbf{S}, \bar{\mathbf{S}})$, i.e., the observed perturbations are “small” in some sense. (AS3) Every \mathbf{H}_k is a polynomial of \mathbf{S} .

Assumption (AS2), which was also considered in Problem 1, promotes the tractability of the problem by ensuring that \mathbf{S} and $\bar{\mathbf{S}}$ are related. As discussed in Section III-A, the distance function $d(\cdot, \cdot)$ must be selected depending on the perturbation model at hand. (AS3) captures the key fact that all the matrices \mathbf{H}_k are GFs of the *same* GSO, establishing a link that can be leveraged via a joint estimation (optimization) of the K GFs. Implementing an approach similar to that in Section IV (i.e., working on the vertex domain, considering the true GSO as an explicit optimization variable, accounting for the GF structure via a commutativity constraint, and assuming that the graph perturbations create and destroy links), the multi-

filter counterpart to (15) that codifies Problem 2 is

$$\begin{aligned} \min_{\mathbf{S} \in \mathcal{S}, \{\mathbf{H}_k\}_{k=1}^K} & \sum_{k=1}^K \alpha_k \|\mathbf{Y}_k - \mathbf{H}_k \mathbf{X}_k\|_F^2 + \lambda r_{\delta_1} (\mathbf{S} - \bar{\mathbf{S}}) \\ & + \beta r_{\delta_2} (\mathbf{S}) + \sum_{k=1}^K \gamma \|\mathbf{S} \mathbf{H}_k - \mathbf{H}_k \mathbf{S}\|_F^2. \end{aligned} \quad (22)$$

Ideally, the value of the positive weight α_k must be selected based on the norm of \mathbf{W}_k (e.g., prior information on the noise level and the number of signal pairs M_k). If none is available, then $\alpha_k = 1$ for all k . Equally important, the fact of pursuing a joint optimization implies that each \mathbf{H}_k contributes with a regularization term $\|\mathbf{S} \mathbf{H}_k - \mathbf{H}_k \mathbf{S}\|_F^2$, promoting the commutativity of the k -th GF with the *single* \mathbf{S} . Intuitively, having the same \mathbf{S} in all these terms couples the optimization across k and contributes to reduce the uncertainty over \mathbf{S} , leading to enhanced estimates of both \mathbf{S} and $\{\mathbf{H}_k\}_{k=1}^K$. As a result, the joint GF identification approach is expected to provide better results than estimating each \mathbf{H}_k separately by solving K instances of (15). We validate this hypothesis numerically via the experiments in Section VII.

Following a motivation similar to that in the previous section, we deal with the non-convex minimization in (22) designing an alternating optimization algorithm that breaks the bilinear terms $\mathbf{S} \mathbf{H}_k$ and $\mathbf{H}_k \mathbf{S}$, and approximates the logarithmic terms with a linear upper-bound. The resulting algorithm solves iteratively the following two subproblems for $t = 1, \dots, t_{\max}$ iterations.

Step 1: Multiple GF Identification. Given the current estimate $\mathbf{S}^{(t)}$, we solve the optimization problem in (22) with respect to each $\mathbf{H}^{(k)}$. This yields

$$\mathbf{H}_k^{(t+1)} = \operatorname{argmin}_{\mathbf{H}_k} \alpha_k \|\mathbf{Y}_k - \mathbf{H}_k \mathbf{X}_k\|_F^2 + \gamma \|\mathbf{S}^{(t)} \mathbf{H}_k - \mathbf{H}_k \mathbf{S}^{(t)}\|_F^2, \quad (23)$$

whose closed-form solution can be found using (17) replacing γ with γ/α_k , \mathbf{X} with \mathbf{X}_k , and \mathbf{Y} with \mathbf{Y}_k . Note that since the only coupling across GFs is via the GSO, (23) estimates each $\mathbf{H}_k^{(t+1)}$ separately from the other GFs, solving K LS problems (each with N^2 unknowns). Furthermore, if multiple processors are available, (23) can be run in parallel across k .

Step 2: Graph Denoising. Given the current estimates of the GFs $\{\mathbf{H}_k^{(t+1)}\}_{k=1}^K$, we follow an MM scheme that, minimizing a linear upper-bound of the logarithmic penalties, yields the estimate of the GSO via

$$\begin{aligned} \mathbf{S}^{(t+1)} = \operatorname{argmin}_{\mathbf{S} \in \mathcal{S}} & \sum_{ij=1}^N (\lambda \bar{\Omega}_{ij}^{(t)} |S_{ij} - \bar{S}_{ij}| + \beta \Omega_{ij}^{(t)} |S_{ij}|) \\ & + \sum_{k=1}^K \gamma \|\mathbf{S} \mathbf{H}_k^{(t+1)} - \mathbf{H}_k^{(t+1)} \mathbf{S}\|_F^2, \end{aligned} \quad (24)$$

where Ω and $\bar{\Omega}$ are obtained as in (19).

The solution to Problem 2 is simply given by $\hat{\mathbf{S}} = \mathbf{S}^{(t_{\max})}$ and $\hat{\mathbf{H}}_k = \mathbf{H}_k^{(t_{\max})}$ for every k . Similar to (15), convergence to a stationary point of (22) is guaranteed.

Corollary 1: Denote as $f(\{\mathbf{H}_k\}_{k=1}^K, \mathbf{S})$ the objective function in (22). If $\mathbf{z}^{(t)} = [\operatorname{vec}(\mathbf{H}_1^{(t)})^\top, \dots, \operatorname{vec}(\mathbf{H}_K^{(t)})^\top, \operatorname{vec}(\mathbf{S})^\top]^\top$

represents the solution provided by the iterative algorithm (23)-(24) after t iterations and every \mathbf{X}_k excites all graph frequencies, then $\mathbf{z}^{(t)}$ converges to a stationary point of f as the number of iterations t goes to infinity.

The key to prove Theorem 1, which established the convergence to a stationary point for the robust estimation of a single GF, was to show that the optimization problem in (15) and the proposed algorithm satisfied the conditions in [47, Th. 1b]. The formulation we put forth for the multi-filter case resembles closely that of the single-filter case, and, as a result, it is not difficult to show that those conditions also hold true for the problem in (22) (see App. A for details).

The discussion and formulations in Section IV-B dealing with incorporating additional information about the input-output signals into the optimization are also pertinent for the setup in this section. The details of such a formulation are omitted for brevity, but it will be explored in our simulations.

A. Joint GF identification for time series

A slightly different, practically relevant, setup where multiple GFs need to be estimated is that of graph-based multivariate time series. In that setup, each variable is associated with a node of the graph and the multiple graph-signal observations correspond to different instants of a time-varying graph signal. AR and moving-average (MA) modeling of time series has a long tradition, with common approaches to decrease the degrees of freedom including limiting the memory of the series and assuming that matrices of coefficients relating different time instants are low rank [49]. In the context of graph signals and network processes, a natural approach is to constrain the matrices of coefficients to be GFs, all defined over the same graph [50], [51]. This section introduces a variation of the problem in (22) tailored to this setup.

To introduce the multiple-graph identification problem formally, let \mathbf{X}_κ and \mathbf{Y}_κ denote a collection of M_κ graph signals corresponding to measurements of a network process for $\kappa = 1, \dots, \kappa_{max}$ time instants. Suppose now that \mathbf{Y}_κ can be accurately modeled by an AR dynamics with memory K so, at every instant κ , the observations \mathbf{Y}_κ satisfy the equation

$$\mathbf{Y}_\kappa = \sum_{k=1}^K \mathbf{H}_k \mathbf{Y}_{\kappa-k} + \mathbf{X}_\kappa, \text{ with } \mathbf{H}_k = \sum_{r=0}^{N-1} h_{r,k} \mathbf{S}^r, \quad (25)$$

where \mathbf{X}_κ is the exogenous input, and the GF \mathbf{H}_k models the influence that the signal observations from the time instant $\kappa - k$ exert on the (current) signal at time κ .

Suppose now that: i) we have access to an estimated (imperfect) graph $\bar{\mathbf{S}}$; ii) the value of the graph signals at different time instants is available; and iii) our goal is to estimate the set of matrices (GFs) $\{\mathbf{H}_k\}_{k=1}^K$ in (25) that describe the dynamics of the multivariate time series. This can be accomplished as

$$\begin{aligned} \min_{\mathbf{S} \in \mathcal{S}, \{\mathbf{H}_k\}_{k=1}^K} & \sum_{\kappa=K+1}^{\kappa_{max}} \left\| \mathbf{Y}_\kappa - \mathbf{X}_\kappa - \sum_{k=1}^K \mathbf{H}_k \mathbf{Y}_{\kappa-k} \right\|_F^2 \\ & + \lambda r_{\delta_1} (\mathbf{S} - \bar{\mathbf{S}}) + \beta r_{\delta_2} (\mathbf{S}) + \sum_{k=1}^K \gamma \left\| \mathbf{S} \mathbf{H}_k - \mathbf{H}_k \mathbf{S} \right\|_F^2. \quad (26) \end{aligned}$$

The main difference relative to (22) is in the first term, which accounts for the new observation model [cf. (21) vs. (25)]. Note that we assume that the exogenous input \mathbf{X}_κ is observed. If that were not the case, it would suffice to remove \mathbf{X}_κ from the objective (possibly updating the Frobenius norm in case statistical knowledge about \mathbf{X}_κ were available). Albeit the differences, the problem in (26) is closely related to (22), with the sources of non-convexities being the same. As a result, we approach its solution with a modified version of Alg. 1 which, at each iteration t , runs two steps. In the first one, we estimate each of the K GFs by solving

$$\begin{aligned} \mathbf{H}_k^{(t+1)} = \operatorname{argmin}_{\mathbf{H}_k} & \sum_{\kappa=K+1}^{\kappa_{max}} \left\| \mathbf{Y}_\kappa - \mathbf{X}_\kappa - \mathbf{H}_k \mathbf{Y}_{\kappa-k} - \sum_{k' < k} \mathbf{H}_{k'}^{(t+1)} \mathbf{Y}_{\kappa-k'} \right. \\ & \left. - \sum_{k' > k} \mathbf{H}_{k'}^{(t)} \mathbf{Y}_{\kappa-k'} \right\|_F^2 + \sum_{k=1}^K \gamma \left\| \mathbf{S}^{(t)} \mathbf{H}_k - \mathbf{H}_k \mathbf{S}^{(t)} \right\|_F^2, \quad (27) \end{aligned}$$

which is different from the previous GF identification step [cf. (23)]. In contrast, the graph-denoising step in (24) remains the same. Note that (27) updates each GF separately in a cyclic way by solving an LS problem with N^2 unknowns. Alternative implementations include using $\mathbf{H}_{k'}^{(t)}$ in lieu of $\mathbf{H}_{k'}^{(t+1)}$ for all $k' < k$ (so that a parallel implementation is enabled) as well as considering a single LS problem with KN^2 unknowns.

Finally, it is worth emphasizing that the formulation introduced in this section can be used as a starting point to design more general robust schemes for multivariate time series defined over a graph. Dealing with both AR and MA matrices, assuming that the memory of the system is unknown, having only partial/statistical information on the exogenous input, and observing the signals at only a subset of nodes are all examples of setups of interest. Since our goal in this section was to demonstrate the relevance of a robust multiple GF formulation in the context of multivariate time series, to facilitate exposition we restricted our discussion to the relatively simple case in (25), but a variety of setups (including those previously listed) will be subject of our future work.

VI. EFFICIENT IMPLEMENTATION OF THE ROBUST GF IDENTIFICATION ALGORITHM

The algorithms proposed up to this point are able to find a solution to the robust GF identification problem in polynomial time. However, their computational complexity scales with the number of nodes as N^7 . To facilitate the deployment in setups where N is large, this section puts forth an efficient implementation that reduces the number of operations.

The new algorithm (summarized in Alg. 2) preserves the core structure of Alg. 1, with an outer loop that, at each iteration, runs two steps: one involving the estimation of the GF(s) and another one dealing with the estimation of the GSO. The main difference is that now, instead of finding the exact solution to those two problems, we obtain an approximate solution. While the details, which are step-dependent, will be specified in the next paragraphs, the overall idea is that for each of the steps we run a few simple (gradient/proximal) iterations. Although Alg. 2 involves two nested loops, the complexity of the problems in the inner loop is cut down

significantly, so that the overall computational overhead is reduced.

To be specific, we describe next the two steps that, at each iteration of the outer loop $t=0, \dots, t_{max}-1$, Alg. 2 runs.

Step 1: Efficient GF Identification. Solving the GF-identification step with the closed-form solution presented in (17) involves inverting a matrix of size $N^2 \times N^2$, which requires $\mathcal{O}(N^6)$ operations. To explain our alternative implementation, let $f_1(\mathbf{H}|\mathbf{S}^{(t)})$ denote the objective function in (16). Since f_1 is strictly convex and smooth, it can be efficiently optimized using a gradient descent approach [52].

To that end, for each iteration t of the outer loop, we define the inner iteration index τ as well as the sequence of variables $\check{\mathbf{H}}^{(\tau)}$ with $\tau = 0, \dots, \tau_{max_1}$, which is initialized as $\check{\mathbf{H}}^{(0)} = \mathbf{H}^{(t)}$. With this notation at hand, at each iteration $\tau = 0, \dots, \tau_{max_1} - 1$ of the inner loop, we update $\check{\mathbf{H}}^{(\tau+1)}$ via

$$\check{\mathbf{H}}^{(\tau+1)} = \check{\mathbf{H}}^{(\tau)} - \mu \nabla f_1(\check{\mathbf{H}}^{(\tau)}|\mathbf{S}^{(t)}). \quad (28)$$

Here, $\mu > 0$ is the step size and ∇f_1 denotes the gradient of f_1 with respect to \mathbf{H} , which is given by

$$\begin{aligned} \nabla f_1(\mathbf{H}|\mathbf{S}^{(t)}) = & 2 \left(\mathbf{H}\mathbf{X}\mathbf{X}^\top - \mathbf{Y}\mathbf{X}^\top \right) \\ & + 2\gamma \left(\mathbf{S}^{(t)\top} (\mathbf{S}^{(t)}\mathbf{H} - \mathbf{H}\mathbf{S}^{(t)}) - (\mathbf{S}^{(t)}\mathbf{H} - \mathbf{H}\mathbf{S}^{(t)})\mathbf{S}^{(t)\top} \right). \end{aligned} \quad (29)$$

When the τ_{max_1} gradient updates are computed, we conclude the GF-identification step by setting $\mathbf{H}^{(t+1)} = \check{\mathbf{H}}^{(\tau_{max_1})}$.

Computing each gradient involves multiplying $N \times N$ matrices, leading to a computational complexity of $\mathcal{O}(\tau_{max_1} N^3)$, which may go down to $\mathcal{O}(\tau_{max_1} N^{2.4})$ if an efficient multiplication algorithm is employed [53]. For large values of N , this complexity is substantially smaller than that required to find the inverse of an $N^2 \times N^2$ matrix.

Step 2: Efficient graph denoising. Since the optimization in (18) involves N^2 variables (the entries in \mathbf{S}), using an off-the-shelf convex solver incurs a computational complexity of $\mathcal{O}(N^7)$ [52]. Inspired by the Lasso regression algorithm [54], we optimize individually over each entry S_{ij} in an iterative manner. The main idea is running multiple rounds of N^2 efficient scalar optimizations rather than dealing with a single but demanding N^2 -dimensional problem. To provide the details of the scheme developed to estimate \mathbf{S} , we need to specify the set of constraints \mathcal{S} and introduce some definitions. Let us focus on the set of adjacency matrices $\mathcal{S}_{\mathcal{A}} := \{\mathbf{S} | S_{ij} \geq 0, S_{ii} = 0\}$ and define the vector $\mathbf{s} := \text{vec}(\mathbf{S})$, the vector $\bar{\mathbf{s}} := \text{vec}(\bar{\mathbf{S}})$, and the matrix $\boldsymbol{\Sigma}^{(t)} := \mathbf{H}^{(t+1)\top} \oplus -\mathbf{H}^{(t+1)}$. With these definitions in place, the minimization in (18) is equivalent to solving

$$\begin{aligned} \min_{\mathbf{s}} \sum_{i=1}^{N^2} & \left(\lambda \bar{\omega}_i^{(t)} |s_i - \bar{s}_i| + \beta \omega_i^{(t)} s_i \right) + \gamma \|\boldsymbol{\Sigma}^{(t)} \mathbf{s}\|_2^2, \\ \text{s. to: } & \mathbf{s} \geq 0, \mathbf{s}_{\mathcal{D}} = 0, \end{aligned} \quad (30)$$

where $\mathbf{s}_{\mathcal{D}}$ collects the elements in the diagonal of \mathbf{S} , and the vectors $\bar{\omega}^{(t)}$ and $\omega^{(t)}$ are computed according to (19) but with $\bar{\mathbf{s}}^{(t)}$ and $\mathbf{s}^{(t)}$ in lieu of $\bar{\mathbf{S}}^{(t)}$ and $\mathbf{S}^{(t)}$. The constraint $\mathbf{s}_{\mathcal{D}} = 0$, implies that only the $N^2 - N$ elements of \mathbf{s} representing the off-diagonal entries of \mathbf{S} need to be optimized. The key

point to find those $N^2 - N$ values is to leverage that the non-differentiable part of the cost in (30) is separable across s_i , postulate $N^2 - N$ scalar optimization problems (coupled via the ℓ_2 term in the cost), and address the optimization following a projected cyclic coordinate descent scheme.

To define clearly the operation of Step 2 at each iteration t of the outer loop, we need to introduce some notation. First, let us denote as τ the iteration index for the inner loop, define the sequence of variables $\check{s}^{(\tau)}$ where $\tau = 0, \dots, \tau_{max_2}$, and initialize the sequence as $\check{s}^{(0)} = \mathbf{s}^{(t)}$. Moreover, with $\ell \notin \mathcal{D}$ denoting an index of the off-diagonal elements of the GSO, let $\boldsymbol{\sigma}_\ell \in \mathbb{R}^{N^2}$ denote the associated ℓ -th column of $\boldsymbol{\Sigma}^{(t)}$, $\omega_\ell \geq 0$ and $\bar{\omega}_\ell \geq 0$ the associated entries of $\boldsymbol{\omega}^{(t)}$ and $\bar{\boldsymbol{\omega}}^{(t)}$, and $\check{s}_\ell^{(\tau)} \in \mathbb{R}$ the associated entry of $\check{s}^{(\tau)}$ (note that dependence on t was omitted to facilitate readability). Then, at every iteration $\tau = 0, \dots, \tau_{max_2} - 1$ of the inner loop, Alg. 2 optimizes over each \check{s}_ℓ separately in a cyclic (successive) way. The advantage of this approach is that the solution to the *scalar* optimization over \check{s}_ℓ is given in closed form by the following projected soft-thresholding operation

$$\check{s}_\ell^{(\tau+1)} = \begin{cases} \left(-\bar{\lambda}_\ell + u_\ell^{(\tau)} \right)^+ & \text{if } \bar{s}_\ell < -\bar{\lambda}_\ell + u_\ell^{(\tau)}, \\ \left(\bar{\lambda}_\ell + u_\ell^{(\tau)} \right)^+ & \text{if } \bar{s}_\ell > \bar{\lambda}_\ell + u_\ell^{(\tau)}, \\ \bar{s}_\ell & \text{otherwise,} \end{cases} \quad (31)$$

$$\text{with } \bar{\lambda}_\ell = \frac{\lambda \bar{\omega}_\ell}{\gamma \boldsymbol{\sigma}_\ell^\top \boldsymbol{\sigma}_\ell} \text{ and } u_\ell^{(\tau)} = \frac{-\beta \omega_\ell - \gamma \boldsymbol{\sigma}_\ell^\top \mathbf{r}_\ell^{(\tau)}}{\gamma \boldsymbol{\sigma}_\ell^\top \boldsymbol{\sigma}_\ell}.$$

Here, $(\cdot)^+$ denotes the operation $(x)^+ = \max(0, x)$, and

$$\mathbf{r}_\ell^{(\tau)} := \sum_{j < \ell} \boldsymbol{\sigma}_j \check{s}_j^{(\tau+1)} + \sum_{j > \ell} \boldsymbol{\sigma}_j \check{s}_j^{(\tau)}. \quad (32)$$

Note that (31) is a soft-thresholding operation with respect to the term $|s_i - \bar{s}_i|$. Also, the constraints in $\mathcal{S}_{\mathcal{A}}$ are satisfied due to the projection operator $(\cdot)^+ := \max\{\cdot, 0\}$, and because we do not optimize over the elements of the diagonal of \mathbf{S} .

At first sight, computing each \check{s}_ℓ seems to require roughly N^2 operations, resulting in a computational complexity of $\mathcal{O}(N^4)$ for estimating the entire vector \mathbf{s} . However, closer examination of vectors $\boldsymbol{\sigma}_\ell$ reveals that at most $2N$ of their entries are non-zero since they are the columns of the Kronecker sum of two $N \times N$ matrices, resulting in a final computational complexity of $\mathcal{O}(2\tau_{max_2} N^3)$ for the graph denoising step.

The pseudocode describing the efficient implementation of Steps 1 and 2 is provided in Alg. 2. The summary is as follows. We postulate a nested algorithm with two loops. The outer loop runs t_{max} iterations. The inner loop runs two steps: Step 1, with τ_{max_1} iterations, and Step 2, with τ_{max_2} iterations. While the complexity for Alg. 1 scaled as $\mathcal{O}(t_{max} N^7)$, with t_{max} being typically small, the overall computational complexity of Alg. 2 is roughly $\mathcal{O}(t_{max}(\tau_{max_1} + \tau_{max_2})N^3)$, which is encouraging, since $2N^2$ variables are optimized and it scales with N significantly better than Alg. 1. Moreover, note that the operations required by the non-robust solution in (4) is $\mathcal{O}(MN^2)$, which is comparable to the computational complexity of our robust algorithm when $M \propto N$. Solving Steps 1 and 2 optimally requires setting large values for τ_{max_1} and τ_{max_2} . Nonetheless, we observe that in most tested setups

Algorithm 2: Reduced-complexity robust GF identification.

Input: $\mathbf{X}, \mathbf{Y}, \bar{\mathbf{S}}, \gamma, \lambda, \beta, \delta_1, \delta_2, \mu$
Output: $\hat{\mathbf{H}}, \hat{\mathbf{S}}$

- 1 Initialize $\mathbf{H}^{(0)}$ and $\mathbf{S}^{(0)}$
- 2 $\bar{\mathbf{s}} = \text{vec}(\bar{\mathbf{S}})$
- 3 **for** $t = 0$ to $t_{max} - 1$ **do**
- // GF-identification step
- 4 $\check{\mathbf{H}}^{(0)} = \mathbf{H}^{(t)}$
- 5 **for** $\tau = 0$ to $\tau_{max_1} - 1$ **do**
- 6 $\check{\mathbf{H}}^{(\tau+1)} = \check{\mathbf{H}}^{(\tau)} + \mu \nabla f_1(\check{\mathbf{H}}^{(\tau)} | \mathbf{S}^{(t)})$
- 7 **end**
- 8 $\mathbf{H}^{(t+1)} = \check{\mathbf{H}}^{(\tau_{max_1})}$
- // Graph denoising step
- 9 $[\sigma_1, \dots, \sigma_{N_2}] = \mathbf{H}^{(t+1)\top} \oplus \mathbf{H}^{(t+1)}$
- 10 $\check{\mathbf{s}}^{(0)} = \text{vec}(\mathbf{S}^{(t)})$
- 11 Update $\check{\omega}^{(t)}, \omega^{(t)}$ via (19) using $\bar{\mathbf{s}}$ and $\check{\mathbf{s}}^{(0)}$
- 12 **for** $i = 0$ to $\tau_{max_2} - 1$ **do**
- 13 **for** $\ell \notin \mathcal{D}$ **do**
- 14 Obtain $\mathbf{r}_\ell^{(\tau)}$ via (32)
- 15 Obtain $\check{\mathbf{s}}_\ell^{(\tau+1)}$ via (31) using $\sigma_\ell, \mathbf{r}_\ell^{(\tau)}, \omega_\ell, \bar{\omega}_\ell$
- 16 **end**
- 17 **end**
- 18 $\mathbf{S}^{(t+1)} = \text{unvec}(\check{\mathbf{s}}^{(\tau_{max_2})})$
- 19 **end**
- 20 $\hat{\mathbf{H}} = \mathbf{H}^{(t_{max})}, \hat{\mathbf{S}} = \mathbf{S}^{(t_{max})}$.

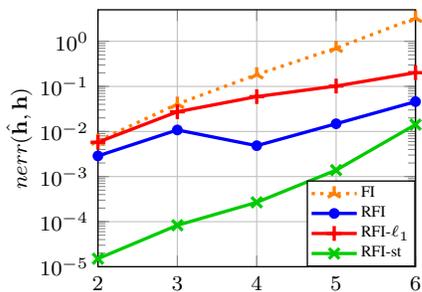


Fig. 2: Error of the estimated filter coefficients $\hat{\mathbf{h}}$ as the order of the filter increases for robust and non-robust GF identification methods.

the approach of setting small values for τ_{max_1} and τ_{max_2} (at the cost of setting a slightly higher value for t_{max}) typically yields a faster convergence. Finally, implementations where the number of iterations is not fixed but selected based on some convergence criterion are also sensible alternatives.

We close the section noting that we developed Alg. 2 for the setting described in Problem 1 because the notation was simpler and facilitated the discussion. Nonetheless, an analogous approach may be followed for the joint estimation of K GFs (cf. Section V), resulting in an algorithm with complexity per GF similar to that for Alg. 2.

VII. NUMERICAL RESULTS

This section discusses several numerical experiments to gain insights and assess the performance of the robust GF identification algorithms. Unless specified otherwise, for a variable of interest Ξ , we report its normalized estimation error defined as

$$nerr(\hat{\Xi}, \Xi) := \|\hat{\Xi} - \Xi\|_F^2 / \|\Xi\|_F^2, \quad (33)$$

where $\hat{\Theta}$ and Θ denote the estimated and the true value, respectively. The code implementing our algorithms and the experiments presented next are available on GitHub (https://github.com/reysam93/graph_denoising). The interested reader is referred there for additional details.

A. Synthetic experiments

We start by evaluating our algorithms with synthetic data, which is key to gain intuition. Unless otherwise stated, graphs are sampled from an Erdős-Rényi (ER) random graph model with a link probability of $p = 0.2$ and $N = 20$ nodes; $\bar{\mathbf{S}}$ is obtained by randomly creating and destroying 10% of the links in \mathbf{S} ; filter coefficients \mathbf{h} are sampled independently and uniformly at random from $[-1, 1]$; $M = 50$ signals \mathbf{x}_m and \mathbf{y}_m , collected in the matrices \mathbf{X} and \mathbf{Y} , are generated according to (2), with the columns of \mathbf{X} being drawn from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, so the signals \mathbf{Y} are stationary on \mathbf{S} ; signals in \mathbf{Y} are corrupted with white Gaussian noise with a normalized power of $\eta_{\mathbf{W}} = 0.05$; and the reported error corresponds to the median of $nerr$ across 64 realizations of graphs and graph signals.

Test case 1. The first experiment evaluates the influence of perturbations as the order of the GF R increases. The number of observed pairs of signals considered is $M = 100$ and 10% of the edges in \mathbf{S} are perturbed. Results are reported in Fig. 2, where the x-axis represents R and the y-axis $nerr(\hat{\mathbf{h}}, \mathbf{h})$. The algorithms considered are: (i) the GF identification algorithm that ignores perturbations [see (4)], denoted as “FI”; (ii) the robust GF identification algorithm from Alg. 1 (“RFI”); (iii) a variation of “RFI” where the reweighted ℓ_1 norm is replaced with the standard ℓ_1 norm (“RFI- ℓ_1 ”); and (iv) the robust GF identification algorithm accounting for the stationarity of \mathbf{Y} (“RFI-ST”). First, we observe that the error of the “FI” algorithm, while small for low values of R , increases rapidly as R grows. This is aligned with the discussion of high-order polynomials in Section III and illustrates the merits of the robust algorithms. Moreover, “RFI-ST” presents the best performance illustrating the importance of exploiting additional structure when it is available. Finally, comparing the error of “RFI” and “RFI- ℓ_1 ” showcases the benefits of replacing the ℓ_1 norm with its reweighted version.

Test case 2. The next experiment tests the influence of different types of perturbations in the robust and non-robust GF identification algorithms. Fig. 3 illustrates the quality of the estimated GF $\hat{\mathbf{H}}$ and the denoised GSO $\hat{\mathbf{S}}$ as the ratio of perturbed links in $\bar{\mathbf{S}}$ increases. Graphs are sampled from the small world [55] random graph model and $\bar{\mathbf{S}}$ is obtained by creating new links, destroying existing links, or simultaneously creating and destroying links, which are respectively denoted as “C”, “D”, and “C/D” in the legend. Analyzing the normalized error of $\hat{\mathbf{H}}$ reported in Fig. 3(a), we observe that the robust algorithm clearly outperforms the non-robust “FI” alternative. Interestingly, the results also demonstrate that destroying links has the most detrimental effect. This observation is consistent with Figs. 3(b) and 3(c), which respectively depict the normalized error and the f_{score}

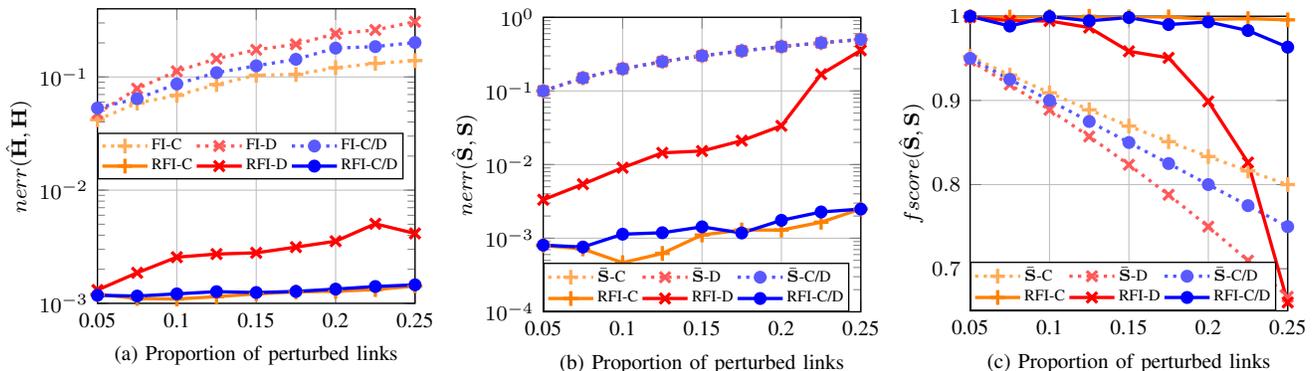


Fig. 3: Assessing the performance of different robust GF identification algorithms. The panels report: (a) the error of the estimated GF; (b) the error of the estimated GSO; and (c) the f_{score} of the estimated GSO. In panels (b) and (c) “ \bar{S} -C”, “ \bar{S} -D”, and “ \bar{S} -C/D” report the discrepancies between \mathbf{S} and $\bar{\mathbf{S}}$ when perturbations create, destroy, or create and destroy links.

of $\hat{\mathbf{S}}$ and the perturbed $\bar{\mathbf{S}}$ for each perturbation type. Since “FI” does not perform any graph-denoising step, the errors between the true GSO and the perturbed $\bar{\mathbf{S}}$ are reported as reference (see “ \bar{S} -C”, “ \bar{S} -D”, and “ \bar{S} -C/D” in the legend). Specifically, the GSO estimated in the “RFI-D” case deteriorates faster as the proportion of perturbed links increases. This can be attributed to the following reasons: i) “ \bar{S} -D” has the lowest f_{score} , indicating that removing links poses a more challenging scenario, as it implies removing ones in the entries of \mathbf{S} , which are elements from an underrepresented class since graphs are sparse; and ii) only removing edges is prone to produce disconnected graphs. Nevertheless, the results show the resilience of the “RFI” algorithm, which provides low-error estimates $\hat{\mathbf{H}}$ and $\hat{\mathbf{S}}$ even when more than 20% of the links are perturbed.

Test case 3. Next, we compare the performance of our algorithms with other robust alternatives. Fig. 4(a) reports, for each algorithm, $n_{err}(\hat{\mathbf{H}}, \mathbf{H})$ as the ratio of perturbed links increases. The baselines considered are the TLS-SEM algorithm from [30], and LLS-SCP from [31]. We note that the TLS-SEM algorithm is tailored to graph signals adhering to

$$\mathbf{Y} = \mathbf{A}\mathbf{Y} + \mathbf{X} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{X}, \quad (34)$$

where the observations at the i -th node are represented by the values of the neighbors of i and an exogenous input. As a result, the TLS-SEM algorithm may not be well suited to deal with signals generated according to the more general model in (2). Taking this into account, to offer a more favorable comparison we consider two types of graph signals: (i) signals generated according to (34), denoted as “SEM”; and (ii) signals generated according to (2), denoted as “H”. It is worth noting that the “SEM” can be considered as a particular case of the model “H” when the GF $\mathbf{H}_{SEM} = (\mathbf{I} - \mathbf{A})^{-1}$ is employed.

Analyzing the results in Fig. 4(a) yields the following findings. When the “SEM” model is considered, “TLS-SEM”, which is tailored for this setting, obtains the best performance when the perturbation probability is small. However, when the perturbation probability grows larger, the performance of “TLS-SEM” and that of “RFI” become comparable. This showcases that our algorithm is well suited to deal with a large number of perturbed links. On the other hand, when the

“H” model is considered, we observe that the “RFI” algorithm consistently outperforms the baselines in the presence of perturbations. The good performance of the “RFI” algorithm on both signal models highlights the flexibility of the proposed formulation since any GF can be expressed as a polynomial GFs.

Test case 4. Now, we compare the performance of the standard and the efficient implementation of the robust identification algorithm, as described in Algs. 1 and 2. The results are shown in Figs. 4(b) and 4(c), where the figures depict the running time measured in seconds and $n_{err}(\hat{\mathbf{H}}, \mathbf{H})$ as N increases. The legend identifies first the algorithm employed, then the number of iterations of the outer loop (t_{max}), and finally the iterations of the inner loops (with $\tau_{max1} = \tau_{max2}$). As expected, Fig. 4(b) shows that Alg. 2 is remarkably faster than Alg. 1 even with medium-sized graphs, achieving a running time 10^3 times smaller when $N = 100$. On the other hand, in Fig. 4(c) we observe that “Eff-5-50” has an error that is close to the standard implementation (“Stand-5”) even though it is considerably faster. Furthermore, the trade-off between speed and estimation accuracy is also evident. “Eff-5-10” is the fastest implementation but the quality of its estimated GF may not be enough for graphs with more than 40 nodes.

Test case 5. To study the benefits of harnessing the information available in the perturbed observation $\bar{\mathbf{S}}$, we compare the performance of our algorithm with that of a least squares (“LS”) approach that estimates the filter as $\arg \min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2$. Results are depicted in Fig. 5(a), where we report $n_{err}(\hat{\mathbf{H}}, \mathbf{H})$ as the normalized noise power increases. $M = 50$ input-output pairs are available, graphs are drawn from an ER model, and two different link probabilities ($p = 0.2$ and $p = 0.5$) are considered. We observe that “RFI” consistently outperforms “LS” even for low values of noise, showcasing the benefits of leveraging the fact that \mathbf{H} is a GF. We also observe that “RFI-st”, which takes advantage of the stationarity of the data, outperforms all the alternatives. Finally, RFI algorithms work slightly better with sparser graphs.

Test case 6. The last experiment with synthetic data studies the benefits of the joint GF estimation. All the GFs are polynomials of the same \mathbf{S} , and for each \mathbf{H}_k we consider $M_k = 15$ noisy observations with $\eta_w = 0.01$. Fig. 5(b) shows

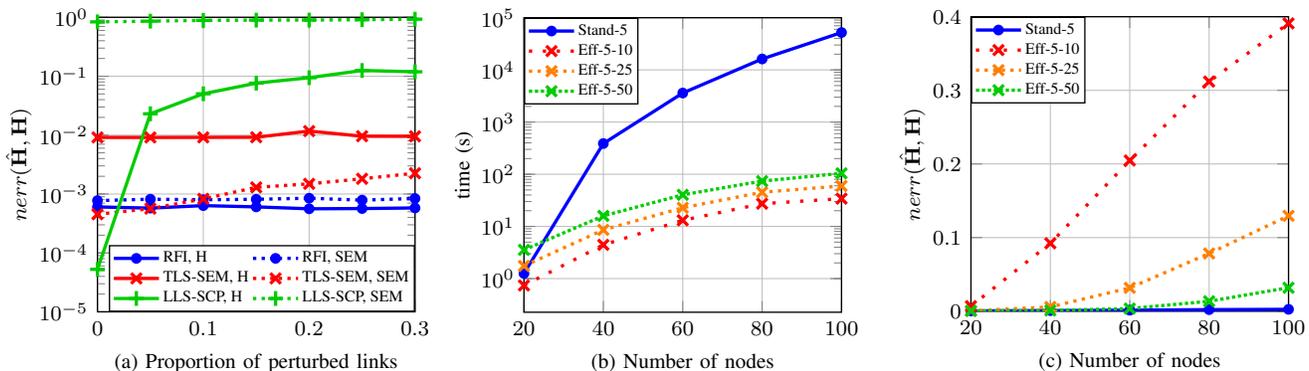


Fig. 4: Comparing the performance of several robust GF identification algorithms. (a) shows the error of $\hat{\mathbf{H}}$ when estimated with the proposed algorithm and with other baselines as the ratio of perturbed links increases. Different graph-signal models are considered. (b) and (c) respectively show the running time and error of $\hat{\mathbf{H}}$ using Alg. 1 and Alg. 2 as the number of nodes increases. Different values for the maximum number of iterations of the inner loops are considered.

Models	1-Step		3-Step	
	$\tau=0,25$	$\tau=0,5$	$\tau=0,25$	$\tau=0,5$
LS	$3.8e-3 \pm 3.7e-3$	$3.3e-3 \pm 3.7e-3$	$1.0e-2 \pm 1.0e-2$	$8.0e-3 \pm 1.0e-2$
LS-GF	$3.5e-3 \pm 4.2e-3$	$3.6e-3 \pm 4.2e-3$	$9.1e-3 \pm 1.0e-2$	$8.8e-3 \pm 1.1e-2$
TLS-SEM	$9.7e-2 \pm 3.1e-2$	$3.0e-2 \pm 1.6e-2$	$4.7e-1 \pm 4.2e-1$	$4.4e-2 \pm 3.4e-2$
RFI	$3.4e-3 \pm 4.1e-3$	$3.4e-3 \pm 4.2e-3$	$8.6e-3 \pm 1.0e-2$	$8.5e-3 \pm 1.1e-2$
VAR-RFI	$3.1e-3 \pm 3.7e-3$	$3.0e-3 \pm 3.6e-3$	$8.0e-3 \pm 9.4e-3$	$7.6e-3 \pm 9.8e-3$

TABLE I: Error performance in predicting the temperature for 2 prediction horizons (1 and 3) and 2 values (25% and 50%) of τ .

the results, with the y-axis representing the normalized error averaged across the K graphs, i.e., $\frac{1}{K} \sum_{k=1}^K nerr(\hat{\mathbf{H}}_k, \mathbf{H}_k)$, and the x-axis representing K . We compare the performance of estimating the GFs jointly (marked as “J” in the legend) or separately for the three algorithms (“RFI- ℓ_1 ”, “RFI”, and “RFI-st”) described in Test case 1. Note that “RFI-J” corresponds to the formulation in (22). The first thing we observe from the results in Fig. 5(b) is that the error decreases as K increases when a joint algorithm is employed. This is aligned with the discussion in Section V and illustrates the benefit of exploiting the common structure. In addition, algorithms accounting for the stationarity of \mathbf{Y} outperform the non-stationary alternatives even though we only have $M = 15$ signals to estimate the covariance $\hat{\mathbf{C}}_{\mathbf{y}}$.

B. Real-world datasets

To close the numerical evaluation, we test our robust GF identification algorithms over two real-world datasets.

Weather station network. This test case evaluates the ability of our algorithms to predict the temperature measured by a network of stations using the data from previous days. The data comes from the “Global Summary of the Day” dataset of the National Centers for Environmental Information (<https://www.ncei.noaa.gov/data/global-summary-of-the-day/archive/>) and we used daily temperature measurements from $N = 19$ stations in California during 2017 & 2018. Specifically, with $\mathbf{y}_{\kappa} \in \mathbb{R}^N$ collecting the measurements of the 19 stations at day κ , we consider an AR model without exogenous inputs, so that $\mathbf{y}_{\kappa} \approx \sum_{k=1}^K \mathbf{H}_k \mathbf{y}_{\kappa-k}$. The M measured signals were divided

into two subsets. The first τM samples (training) were used to obtain the GFs \mathbf{H}_k , where τ represents the fraction of samples used for training. The remaining samples (evaluation) were used to assess the performance and the generalization of the estimated GFs. This non-random train-test split mimics a real situation where we collect data for a specific continuous period of time, infer the graph filter modeling the evolution of the data, and then employ the estimated filters to infer the behavior of unobserved signals. Also, the data is normalized so that the signal at each station for all time samples has unitary norm.

The underlying \mathcal{G} was constructed as the unweighted 5-nearest neighbors graph, using the geographical distance between stations. Since temperature relations across stations are likely to be due to a range of factors (including, e.g., altitude), the considered adjacency (based only on geographical positions) may be imperfect, rendering our robust algorithms better suited for this task.

The estimation performance of the different algorithms is reported in Table I. The metrics shown are the average and standard deviation of the normalized error at each timestep $\frac{1}{M} \sum_{\kappa=1}^M nerr(\hat{\mathbf{y}}_{\kappa}, \mathbf{y}_{\kappa})$ for the test dataset. Because the ground-truth GF is not known in this setting, we use the error of the observed signals to evaluate the performance of our algorithms. To build the prediction $\hat{\mathbf{y}}_{\kappa}$ for day κ , we use our estimated filters $\hat{\mathbf{H}}_k, k = 1, \dots, K$ and the measurements for the previous K days, and calculate $\hat{\mathbf{y}}_{\kappa} = \sum_{k=1}^K \hat{\mathbf{H}}_k \mathbf{y}_{\kappa-h-k+1}$, where h is our prediction horizon. Then, to assess the quality of the schemes, we compare our prediction with the signal measurements calculating $nerr(\mathbf{y}_{\kappa}, \hat{\mathbf{y}}_{\kappa})$. The algorithms evaluated are “LS”, “LS-GF” (which postulates a GF with coefficients $\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \|\mathbf{Y} - \sum_{\ell} h_{\ell} \mathbf{S}^{\ell} \mathbf{X}\|_F^2$), “TLS-SEM”, “RFI” (which assumes an AR(1) process where $K = 1$, so that only one lag for a prediction horizon of h is considered and the output estimate is given by $\hat{\mathbf{y}}_{\kappa} = \hat{\mathbf{H}} \mathbf{y}_{\kappa-h}$, leading to the adoption of Alg. 1) and “AR(3)-RFI” (where $K = 3$ lags are considered). Two values of τ (0.25 and 0.50) and two prediction horizons (1 and 3) are considered. The main observation is that “AR(3)-RFI” yields the best performance in all settings. Additionally, the results for $\tau=0.25$ demonstrate the benefits of considering the underlying graph in the low-sample regime, since even “LS-GF”, which relies on the

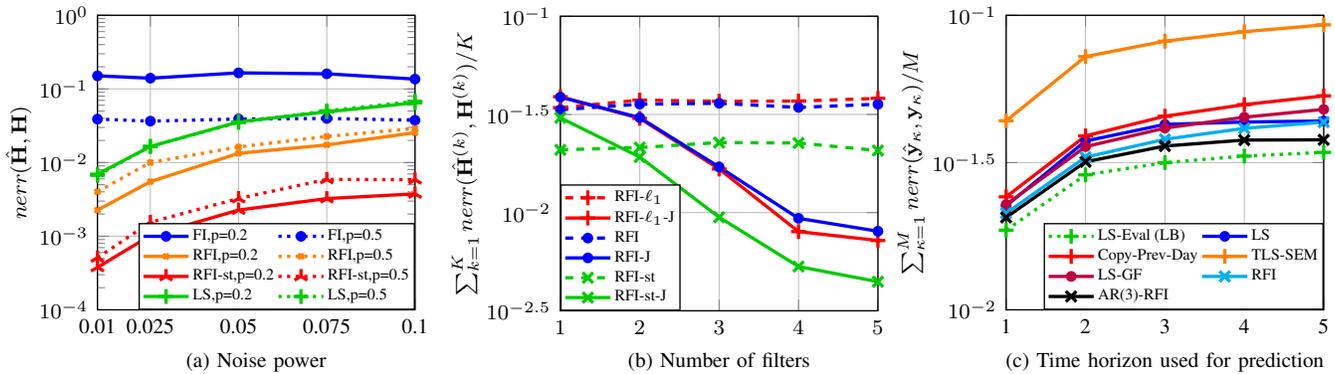


Fig. 5: Comparison of the performance of the RFI algorithms in the joint filter identification setup and assessing the influence of noise in the data. (a) Evolution of the normalized filter error as we increase the power of the noise introduced to the data. P values given in the legend represent the connection probability of the ER graphs. (b) Error performance when estimating K GFs using the separate and joint approach for different values of K . (c) Performance of the algorithms predicting ozone levels in the AirData station network, as the time horizon of the prediction increases.

imperfect $\bar{\mathbf{S}}$, outperforms “LS”.

Air quality station network. In this experiment we use 2018 & 2019 data from the United States Environmental Protection Agency (<https://www.epa.gov/outdoor-air-quality-data>) to predict the ozone levels in a network of 17 outdoor stations in California. The considered setup is similar to that of the weather station data (signals are assumed to follow an AR model and a nearest neighbor graph based on geographical distances is used). The stations chosen were those with at least 330 measurements each year for a selection of pollutants, and missing data was filled via first-order interpolation. The goal here is to analyze how the prediction horizon affects the prediction error. The value of τ was set to 0.5, i.e. evaluation data represented 50% of the samples. Fig. 5 shows the performance of the algorithms when predicting ozone levels. As a baseline, “LS-Eval(LB)” shows the error measured on the evaluation data when obtaining the GF also using evaluation data, therefore representing a lower bound for the LS error using AR models of order 1. Also, “Copy-Prev-Day” represents the error obtained by the “identity GF”, which copies the previous day’s measurement. As in the previous example, the best performing algorithm is “AR(3)-RFI”, whose performance is close to the baseline, followed by “RFI”.

VIII. CONCLUDING REMARKS

This paper puts forth a framework dealing with estimation problems in GSP where the information about (the links of) the supporting graph is uncertain. Specifically, we addressed the problem of estimating a GF (i.e., a polynomial of the GSO) from input and output graph signals under the key assumption that only a perturbed version of the true GSO was available. In contrast to the majority of existing approaches that operate on the spectral domain, we recast the true graph as an additional estimation variable and formulated an optimization problem that *jointly* estimated the GF and the true (unknown) GSO. We focused first on the case where only one GF needed to be estimated and, then, shifted to (multi-feature and AR graph signal) setups where multiple GFs have to be jointly identified. The formulated optimizations

operated completely in the vertex domain and bypassed the problem of computing high-order polynomials, avoiding the challenges of dealing with the influence of perturbations in the graph spectrum as well as the numerical instability and error propagation associated with high-order matrix polynomials. While non-convex, upon blending techniques from alternating optimization and MM, the proposed algorithm was shown to be capable to find a stationary point in polynomial time. This algorithm was later modified so that the scaling of the computational complexity with respect to the number of nodes in the graph is reduced. Future work includes delving into the robust estimation of ARMA time-varying graph signals, consideration of additional graph perturbation models, and application of our robust estimation framework to other GSP problems, to name a few.

APPENDIX A: PROOF OF TH. 1

The proof is based on the convergence of the Block Successive Upper-bound Minimization (BSUM) algorithm [47]. In summary, BSUM algorithms tackle optimization problems with an objective function $f(\mathbf{z})$ by splitting the vector of variables \mathbf{z} into B blocks \mathbf{z}_b , with $b = 1, \dots, B$, and then sequentially minimizing an upper bound $u(\mathbf{z}_b)$ of the original objective function for each block of variables. Then, [47, Th. 1] proves that BSUM algorithms converge to a stationary point when the following conditions are fulfilled:

- (C1) Each function $u_b(\mathbf{z}_b)$ must be a global upper bound of $f(\mathbf{z})$ and the first-order behavior of $u_b(\mathbf{z}_b)$ and $f(\mathbf{z})$ must be the same.
- (C2) $f(\mathbf{z})$ must be regular (cf. [47]) at every point in \mathcal{Z}^* .
- (C3) The level set $\mathcal{Z}^{(0)} = \{\mathbf{z} \mid f(\mathbf{z}) \leq f(\mathbf{z}^{(0)})\}$ is compact.
- (C4) At least one of the problems in (16) and (18) must have a unique solution.

Since Alg. 1 falls into the BSUM framework, it suffices to show that the previous conditions are met in our case. To that end, recall that $f(\mathbf{z})$ is the objective function in (15), and let the $B = 2$ block of variables be given by $\mathbf{z}_1 := \text{vec}(\mathbf{H})$ and $\mathbf{z}_2 := \text{vec}(\mathbf{S})$. Moreover, note that at each step, the function $f(\mathbf{z})$ is approximated by $u_1(\mathbf{z}_1)$ and $u_2(\mathbf{z}_2)$, which

respectively correspond to the objective functions in (16) and (18). Next, we prove that Alg. 1 satisfies the four conditions.

Condition (C1) requires the surrogate functions $u_b(\mathbf{z}_b)$ to be global upper bounds of $f(\mathbf{z})$ and exhibit the same first-order behavior. For the first block ($b = 1$), it readily follows that $u_1(\mathbf{z}_1)$ is obtained from $f(\mathbf{z})$ simply by dropping the terms that depend on \mathbf{z}_2 (see (16) in Step 1), hence the conditions are trivially satisfied. Regarding the second block, we note that $u_2(\mathbf{z}_2)$ can be obtained from $f(\mathbf{z})$ by dropping the terms that only depend on \mathbf{z}_1 , keeping the quadratic terms that depend on \mathbf{z}_2 , and replacing the logarithmic penalties that depend on \mathbf{z}_2 by their first-order Taylor approximation. This implies that: i) the first-order behavior of $u_2(\mathbf{z}_2)$ and $f(\mathbf{z})$ with respect to \mathbf{z}_2 is the same and ii) since the log is a *concave* differentiable function, its Taylor series of order one constitutes a *global upper* bound. As a result, $u_1(\mathbf{z}_1)$ and $u_2(\mathbf{z}_2)$ satisfy the requirement, and hence, (C1) is met.

To prove (C2), according to [47], a function $f(\mathbf{z})$ is regular if its non-smooth components are separable across the different blocks of variables. To show this, we decompose f as $f = g_A + g_B$, with functions g_A and g_B being defined as

$$g_A(\mathbf{H}, \mathbf{S}) = \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \gamma\|\mathbf{H}\mathbf{S} - \mathbf{S}\mathbf{H}\|_F^2,$$

$$g_B(\mathbf{S}) = \lambda \sum_{i,j=1}^N \log(|S_{ij} - \bar{S}_{ij}| + \delta_2) + \beta \sum_{i,j=1}^N \log(|S_{ij}| + \delta_1).$$

Since g_A is a smooth function and the non-smooth function g_B only depends on the variables on the second block, $\mathbf{z}_2 = \text{vec}(\mathbf{S})$, it follows that $f(\mathbf{z})$ is a regular function for the whole feasible set of the optimization problem.

Next, we show that the level set $\mathcal{Z}^{(0)} = \{\mathbf{z} \mid f(\mathbf{z}) \leq f(\mathbf{z}^{(0)})\}$ is compact as required by (C3). We start by noting that the entries of \mathbf{S} are continuous subsets of \mathbb{R} , (e.g., $S_{ij} \in \mathbb{R}_+$ when $\mathbf{S} = \mathbf{A}$), and that $\mathbf{H} \in \mathbb{R}^{N \times N}$, so $f(\mathbf{z})$ is continuous. Moreover, $f(\mathbf{z}) \leq f(\mathbf{z}^{(0)})$ implies that the functions $\|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2$ and $\log(|S_{ij}| + \delta_1)$ are all bounded, rendering the domain of $f(\mathbf{z})$ bounded. It follows then that the level set $\mathcal{Z}^{(0)}$ is compact.

Finally, we need to prove that either (16) or (18) have a unique solution, so that (C4) is fulfilled. Prop. 1 (see below) states that, under the two conditions required by Th. 1 (i.e., \mathbf{S} does not have repeated eigenvalues, and the graph signals \mathbf{X} excite every graph frequency), the solution to (16) is unique. This confirms that (C4) is satisfied, concluding the proof.

Proposition 1. *Let $\mathbf{H} \in \mathbb{R}^{N \times N}$, $\mathbf{S} = \mathbf{V}\text{diag}(\boldsymbol{\lambda})\mathbf{V}^{-1} \in \mathbb{R}^{N \times N}$, and $\mathbf{X} \in \mathbb{R}^{N \times M}$ be the GF, the GSO, and the input signals in (16). Then, (16) has a unique solution w.r.t. \mathbf{H} if the following conditions are satisfied:*

- 1) $\lambda_i \neq \lambda_{i'}$, for all $i \neq i'$ and $(i, i') \in \{1, \dots, N\}^2$.
- 2) Every row of $\tilde{\mathbf{X}} = \mathbf{V}^{-1}\mathbf{X}$ has at least one non-zero entry.

Proof. To simplify exposition, we focus first on the (most restrictive) setup of having only $M = 1$ input-output pairs. Defining $\hat{\mathbf{h}} := \text{vec}(\mathbf{H})$, we can reformulate (16) as

$$\min_{\hat{\mathbf{h}} \in \mathbb{R}^{N^2}} \gamma \|(\mathbf{I} \otimes \mathbf{S} - \mathbf{S}^T \otimes \mathbf{I})\hat{\mathbf{h}}\|_2^2 + \|\mathbf{y} - (\mathbf{x}^T \otimes \mathbf{I})\hat{\mathbf{h}}\|_2^2, \quad (35)$$

where lowercase symbols \mathbf{y} and \mathbf{x} are used to emphasize that the output and input signals are a single N -dimensional vector. Upon defining $\mathbf{D} := \mathbf{I} \otimes \mathbf{S} - \mathbf{S}^T \otimes \mathbf{I}$, and $\mathbf{E} := \mathbf{x}^T \otimes \mathbf{I}$, solving (35) is equivalent to solving

$$\min_{\hat{\mathbf{h}} \in \mathbb{R}^{N^2}} \left\| \begin{bmatrix} \mathbf{0}_{N^2} \\ \mathbf{y} \end{bmatrix} - \mathbf{F}\hat{\mathbf{h}} \right\|_2^2 \text{ with } \mathbf{F} := \begin{bmatrix} \gamma \mathbf{D} \\ \mathbf{E} \end{bmatrix} \quad (36)$$

To prove that (36) has a unique solution, it suffices to show that \mathbf{F} is full column rank, i.e. $\nexists \mathbf{n} \in \mathbb{R}^{N^2}$ such that $\mathbf{F}\mathbf{n} = \mathbf{0}_{N+N^2}$. To show this, we first identify $\mathcal{N}(\mathbf{D})$, the null space of \mathbf{D} , and then show that $\mathbf{E}\mathbf{n} \neq \mathbf{0}_N$ for all $\mathbf{n} \in \mathcal{N}(\mathbf{D}) \setminus \{\mathbf{0}_{N^2}\}$.

We start with the characterization of $\mathcal{N}(\mathbf{D})$. Given the Kronecker structure of \mathbf{D} , each of its N^2 eigenvalues has the form $\lambda_k - \lambda_{k'}$, with $(\mathbf{V}^{-1})^T \otimes \mathbf{V}$ being the associated eigenvectors. Leveraging that $\lambda_i \neq \lambda_{i'}$ for $i \neq i'$, it follows that only when $i = i'$ the eigenvalue of \mathbf{D} is zero. As a result, $\text{rank}(\mathbf{D}) = N^2 - N$ and $\dim(\mathcal{N}(\mathbf{D})) = N$. Equally important, the N eigenvectors associated with the N zero eigenvalues are given by $(\mathbf{V}^{-1})^T \odot \mathbf{V}$, which, as a result, constitutes a basis spanning $\mathcal{N}(\mathbf{D})$. More formally, we concluded that $\mathcal{N}(\mathbf{D}) = \{((\mathbf{V}^{-1})^T \odot \mathbf{V})\boldsymbol{\theta} \mid \forall \boldsymbol{\theta} \in \mathbb{R}^N\}$.

Thus, to show that \mathbf{F} in (36) is full column rank we just need to prove that the only element $\mathbf{n} \in \mathcal{N}(\mathbf{D})$ that renders $\mathbf{E}\mathbf{n} = \mathbf{0}_N$ is the all-zero vector $\mathbf{0}_{N^2}$. To do so, we leverage the characterization of $\mathcal{N}(\mathbf{D})$ and write $\mathbf{E}\mathbf{n}$ as

$$\begin{aligned} \mathbf{E}\mathbf{n} &= (\mathbf{x}^T \otimes \mathbf{I})((\mathbf{V}^{-1})^T \odot \mathbf{V})\boldsymbol{\theta} = (\mathbf{x}^T (\mathbf{V}^{-1})^T \odot \mathbf{V})\boldsymbol{\theta} \\ &= \mathbf{V}\text{diag}(\boldsymbol{\theta})(\mathbf{x}^T (\mathbf{V}^{-1})^T) = \mathbf{V}\text{diag}(\boldsymbol{\theta})\mathbf{V}^{-1}\mathbf{x} \\ &= \mathbf{V}\text{diag}(\boldsymbol{\theta})\tilde{\mathbf{x}} = \mathbf{V}(\boldsymbol{\theta} \circ \tilde{\mathbf{x}}), \end{aligned} \quad (37)$$

where \circ represents the element-wise vector product (Hadamard product) and we used the property $(a \otimes b)(c \odot d) = ac \odot bd$. Since \mathbf{V} is invertible, the first and last terms in (37) demonstrate that $\mathbf{E}\mathbf{n} = \mathbf{0}_N$ requires $\boldsymbol{\theta} \circ \tilde{\mathbf{x}} = \mathbf{0}_N$. However, condition 2) in Prop. 1 states that $\tilde{x}_i \neq 0$ for all i ; hence, $\boldsymbol{\theta} \circ \tilde{\mathbf{x}} = \mathbf{0}_N$ requires $\boldsymbol{\theta} = \mathbf{0}_N$. This implies that the only element in $\mathcal{N}(\mathbf{D})$ that renders $\mathbf{E}\mathbf{n} = \mathbf{0}_N$ is $\mathbf{n} = (\mathbf{V}^{-1})^T \odot \mathbf{V}\mathbf{0}_N = \mathbf{0}_{N^2}$, concluding the proof.

The proof can be generalized for $M > 1$. In that case, \mathbf{E} has size $MN \times N^2$ and the counterpart to (37) establishes that having $\mathbf{E}\mathbf{n} = \mathbf{0}$ requires $\text{vec}(\text{diag}(\boldsymbol{\theta})\tilde{\mathbf{X}}) = \mathbf{0}_{MN}$. Since Prop. 1 assumes that each row of $\tilde{\mathbf{X}}$ has at least one nonzero entry, it follows that $\boldsymbol{\theta} = \mathbf{0}_{MN}$, concluding the proof. \square

In other words, if the two conditions stated in Prop. 1 hold, the result guarantees that the problem in (16) has a unique solution. Please note that, if $M > N$ ($MN > N^2$) and the samples are sufficiently rich (e.g., i.i.d.), \mathbf{E} could be full column rank and, as a consequence, \mathbf{F} would be full column rank as well. Therefore, we could omit the first condition in the proposition (\mathbf{S} having distinct eigenvalues) and the problem would still have a unique solution. In the intermediate case $1 < M < N$ we could also relax condition 1, allowing some eigenvalues to be repeated.

ACKNOWLEDGMENT

The authors would like to thank Pablo Espana-Gutierrez for his help reviewing the literature and developing the early versions of the convergence results.

REFERENCES

- [1] K. Nodop, R. Connolly, and F. Girardi, "The field campaigns of the European Tracer Experiment (etex): Overview and results," *Atmospheric Environment*, vol. 32, no. 24, pp. 4095–4108, 1998.
- [2] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY: Springer, 2009.
- [3] O. Sporns, *Discovering the Human Connectome*. MIT Press, 2012.
- [4] D. Shuman et al., "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, 2013.
- [5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.
- [6] P. Djuric and C. Richard, *Cooperative and Graph Signal Processing: Principles and Applications*. Academic Press, 2018.
- [7] F. J. Iglesias et al., "Demixing and blind deconvolution of graph-diffused sparse signals," in *IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2018, pp. 4189–4193.
- [8] A. Ortega et al., "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [9] A. G. Marques et al., "Graph signal processing: Foundations and emerging directions (editorial)," *IEEE Signal Process. Mag.*, vol. 37, Nov. 2020.
- [10] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [11] 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, 2017.
- [12] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Trans. Signal Process. Inf. Netw.*, vol. 5, no. 1, pp. 47–60, 2018.
- [13] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters," *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1146–1159, 2017.
- [14] S. Rey, F. J. Iglesias, C. Cabrera, and A. G. Marques, "Sampling and reconstruction of diffused sparse graph signals from successive local aggregations," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1142–1146, 2019.
- [15] Y. He and H. Wai, "Detecting central nodes from low-rank excited graph signals via structured factor analysis," *IEEE Trans. Signal Process.*, vol. 70, pp. 2416–2430, 2022.
- [16] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 47–60, 2019.
- [17] G. F. R., *Matrix theory*. Chelsea, New York, 1959, vol. 21.
- [18] G. Egeland and P. E. Engelstad, "The availability and reliability of wireless multi-hop networks with stochastic link failures," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1132–1146, 2009.
- [19] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [20] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.
- [21] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, 2017.
- [22] A. Buciuiea, S. Rey, and A. G. Marques, "Learning graphs from smooth and graph-stationary signals with hidden variables," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 273–287, 2022.
- [23] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.
- [24] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," *IEEE Trans. Signal Process.*, vol. 68, pp. 1558–1573, 2020.
- [25] H. Kenlay, D. Thanou, and X. Dong, "Interpretable stability bounds for spectral graph filters," in *Int. Conf. Mach. Learn. (ICML)*, vol. 139, 18–24 Jul 2021, pp. 5388–5397.
- [26] H. Kenlay, D. Thanou, and X. Dong, "On the stability of graph convolutional neural networks under edge rewiring," in *IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2021, pp. 8513–8517.
- [27] H. S. Nguyen, Y. He, and H. T. Wai, "On the stability of low pass graph filter with a large number of edge rewires," in *IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2022, pp. 5568–5572.
- [28] J. Miettinen, S. A. Vorobyov, and E. Ollila, "Graph error effect in graph signal processing," in *IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2018, pp. 4164–4168.
- [29] —, "Modelling graph errors: Towards robust graph signal processing," *arXiv preprint arXiv:1903.08398*, 2019.
- [30] E. Ceci, Y. Shen, G. B. Giannakis, and S. Barbarossa, "Graph-based learning under perturbations via total least-squares," *IEEE Trans. Signal Process.*, vol. 68, pp. 2870–2882, 2020.
- [31] A. Natali, M. Coutino, and G. Leus, "Topology-aware joint graph filter and edge weight identification for network processes," in *IEEE Intl. Wrkshp. Mach. Learn. Signal Process. (MLSP)*. IEEE, 2020, pp. 1–6.
- [32] S. Rey and A. G. Marques, "Robust graph-filter identification with graph denoising regularization," in *IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2021, pp. 5300–5304.
- [33] V. M. Tenorio et al., "A robust alternative for graph convolutional neural networks via graph neighborhood filters," in *Conf. Signals, Syst., Computers (Asilomar)*. IEEE, 2021, pp. 1573–1578.
- [34] R. Levie, E. Isufi, and G. Kutyniok, "On the transferability of spectral graph filters," in *Int. Conf. Sampling Theory Appl. (SampTA)*. IEEE, 2019, pp. 1–5.
- [35] R. Levie et al., "Transferability of spectral graph convolutional neural networks," *J. Mach. Learn. Res.*, vol. 22, pp. 272–1, 2021.
- [36] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability, and transferability," *Proc. IEEE*, vol. 109, no. 5, pp. 660–682, 2021.
- [37] N. Keriven, A. Bietti, and S. Vaiter, "Convergence and stability of graph convolutional networks on large random graphs," *Advances Neural Inf. Process. Syst.*, vol. 33, pp. 21 512–21 523, 2020.
- [38] 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, 2017.
- [39] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [40] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
- [41] B. Dai, S. Ding, and G. Wahba, "Multivariate Bernoulli distribution," *Bernoulli*, vol. 19, no. 4, pp. 1465–1483, 2013.
- [42] S. Segarra and A. Ribeiro, "Stability and continuity of centrality measures in weighted graphs," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 543–555, 2015.
- [43] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5–6, pp. 877–905, 2008.
- [44] J. Ying, J. V. M. Cardoso, and D. P. Palomar, "Does the ℓ_1 -norm learn a sparse graph under laplacian constrained graphical models?" 2020.
- [45] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, 2016.
- [46] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, 2007.
- [47] M. Hong, M. Razaviyayn, Z. Luo, and J. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, 2015.
- [48] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, vol. 13, no. 9, p. 228, 2020.
- [49] G. C. Reinsel, *Elements of multivariate time series analysis*. Springer Science & Business Media, 2003.
- [50] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, 2017.
- [51] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with varma recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.
- [52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [53] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," in *Proc. ACM Symp. Theory Comput.*, 1987, pp. 1–6.
- [54] T. Hastie, R. Tibshirani, and M. Wainwright, "Statistical learning with sparsity," *Monographs Statistics Appl. Probability*, vol. 143, p. 143, 2015.

- [55] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Phys. Lett. A*, vol. 263, no. 4-6, pp. 341-346, 1999.