Learning Graphs from Smooth and Graph-Stationary Signals with Hidden Variables

Andrei Buciulea *Student Member, IEEE*, Samuel Rey *Student Member, IEEE*, and Antonio G. Marques, *Senior Member, IEEE*

***Abstract*—Network-topology inference from (vertex) signal observations is a prominent problem across data-science and engineering disciplines. Most existing schemes assume that observations from all nodes are available, but in many practical environments, only a subset of nodes is accessible. A natural (and sometimes effective) approach is to disregard the role of unobserved nodes, but this ignores latent network effects, deteriorating the quality of the estimated graph. Differently, this paper investigates the problem of inferring the topology of a network from nodal observations while taking into account *the presence of hidden (latent) variables*. Our schemes assume the number of observed nodes is considerably larger than the number of hidden variables and build on recent graph signal processing models to relate the signals and the underlying graph. Specifically, we go beyond classical correlation and partial correlation approaches and assume that the signals are *smooth* and/or *stationary* in the sought graph. The assumptions are codified into different constrained optimization problems, with the presence of hidden variables being explicitly taken into account. Since the resulting problems are ill-conditioned and non-convex, the block matrix structure of the proposed formulations is leveraged and suitable convex-regularized relaxations are presented. Numerical experiments over synthetic and real-world datasets showcase the performance of the developed methods and compare them with existing alternatives.**

***Index Terms*—Network-topology inference, hidden nodes, latent variables, graphical Lasso, graph stationarity.**

I. Introduction

Recent years have witnessed the rise of problems involving datasets with non-Euclidean support. A popular approach to deal with this type of data consists in exploiting graphs to generalize a wide range of classical information-processing techniques to those irregular domains. This graph-based perspective has been successfully applied to a number of applications (with power, communications, social, geographical, genetics, and brain networks being notable examples [2]–[6]) and has attracted the attention of researchers from different areas, including statistics, machine learning and signal processing (SP). For the latter case, graph SP (GSP) has been capable of generalizing a number of tools originally conceived to process signals with regular support (time or space) to signals defined on heterogeneous domains represented by a graph, providing new insights and efficient algorithms [3], [7]–[10]. The core assumption of GSP is that the properties of the graph signals can be explained by the influence of the network, whose topology is codified in the so-called graph-shift operator (GSO), a square matrix whose non-zero entries identify the edges of the graph.

Although networks may exist as physical entities, oftentimes they are abstract mathematical representations with nodes describing variables and links describing pairwise relationships between them. More importantly for the paper at hand, such relationships may not be always known a priori. In the scenarios where the graph is unknown, it is possible to learn the graph from a set of nodal observations under the fundamental assumption that there exists a relationship between the properties of the observed signals and the topology of the sought graph. The described task represents a prominent problem commonly referred to as *network topology inference*, which is also known as *graph learning* [11]–[16]. Noteworthy approaches include correlation networks [2], partial correlations and (Gaussian) Markov random fields [2], [17]–[19], sparse structural equation models [20], [21], GSP-based approaches [12]–[14], [22], [23], as well as their non-linear generalizations [24], [25], to name a few.

The standard network-inference approach in the aforementioned works is to assume that observations from all the nodes of the graph are available. In certain environments, however, only observations from a subset of nodes are available, with the remaining nodes being unobserved or *hidden*. The existence of hidden/latent nodes constitutes a relevant and challenging problem since closely related values from two observed nodes may be explained not only by an edge between the two nodes but by a third latent node connected to both of them. Moreover, because there are no observations from the hidden nodes, modeling their influence renders the network inference problem substantially more challenging and ill-posed. Except for direct pairwise methods, which can be trivially generalized to the setup at hand, most of the existing approaches require important modifications to deal with hidden nodes. Network-inference works that have looked at the problem of hidden variables include examples in the context of Gaussian graphical model selection [26], [27], inference of linear Bayesian networks [28], nonlinear regression [29], or brain connectivity [30] to name a few. Nonetheless, there are still a number of effective network-inference methods (including most in the context of GSP) that have not considered the presence of latent unobserved nodes.

Motivated by the previous discussion, in this paper we approach the problem of network topology inference with hidden variables by leveraging two fundamental concepts of the GSP framework: smoothness [3] and stationarity [31], [32]. A signal being smooth on a graph implies that the signal values at two neighboring nodes are close so that the signal varies slowly across the graph. This fairly general assumption has been successfully exploited to infer the topology of the graph

Work supported by the Spanish NSF Grants SPGraph (PID2019-105032GB-I00) and FPU17/04520, by the Grants F661-MAPPING-UCI and F663-AAGNCS funded by the Comunidad de Madrid (CAM) and King Juan Carlos University (URJC), and by the Grants F649-1209 and PREDOC20-003 funded by the CAM and URJC. All the authors are with the Dept. of Signal Theory and Comms., King Juan Carlos University, Madrid, Spain. Email contact author: antonio.garcia.marques@urjc.es. An early preliminary version of this work was presented as a conference paper in [1].

when values from all nodes are observed [23], [33], [34]. From a different perspective, assuming that a random process is stationary on a graph is tantamount to assuming that the covariance matrix of the random process is a polynomial of the GSO, which has been leveraged in the context of network-inference to develop new algorithms and establish important links between graph stationarity and classical correlation and partial-correlation approaches [13], [35], [36]. Although the assumptions of smoothness and stationarity have been successfully adopted in the context of the network-topology inference problem, a formulation robust to the presence of hidden variables is still missing. To fill this gap, this paper builds on our previous work and investigates how the presence of the hidden variables impacts the classical definitions of graph smoothness and stationarity. Then, it formulates the network-recovery problem as a constrained optimization that accounts explicitly for the modified definitions. A key in our formulation is the consideration of a block matrix factorization approach and exploitation of the low rankness and the sparsity pattern present in the blocks related to hidden variables. A range of formulations are presented and suitable (convex and non-convex) relaxations to deal with the sparsity and low-rank terms are considered. While our focus is to learn the connections among observed nodes, some of our approaches also reveal information related to links involving hidden nodes. A further investigation of this matter is left as future work.

To summarize, our contributions are as follows: (i) we analyze the influence of hidden variables on graph smoothness and graph stationarity; (ii) we propose several optimization problems to solve the topology inference problem with hidden variables when the observed signals are smooth, stationary, or both; and (iii) we present an extensive evaluation of the proposed models through both synthetic and real experiments.

The remainder of the paper is organized as follows. Section II introduces basic GSP concepts leveraged during the paper. Section III formalizes the problem at hand. Sections IV and V respectively detail the proposed topology inference algorithms for smooth and stationary signals, with Section VI combining both assumptions and considering that the signals are both smooth and stationary. The numerical evaluation of the proposed methods is presented in Section VII, and Section VIII provides some concluding remarks.

II. Fundamentals of graph signal processing

In this section, we introduce basic GSP concepts that help to explain the relationship between the observed signals and the topology of the underlying graph.

GSO and graph signals. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an undirected and weighted graph with N nodes where $\mathcal{V}$ and $\mathcal{E}$ represent the vertex and edge set, respectively. The weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a sparse matrix encoding the topology of the graph $\mathcal{G}$, with A_{ij} capturing the weight of the edge between the nodes i and j, and with $A_{ij} = 0$ if i and j are not connected. A general representation of the graph is the GSO $\mathbf{S} \in \mathbb{R}^{N\times N}$, where $S_{ij} \neq 0$ if and only if $i = j$ or $(i, j) \in \mathcal{E}$. Typical choices for the GSO are the adjacency matrix $\mathbf{A}$ [7], the combinatorial graph Laplacian $\mathbf{L} := \text{diag}(\mathbf{A1}) - \mathbf{A}$ [3], and their degree-normalized variants. Since the graph is undirected, the GSO is symmetric and can be diagonalized as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where the orthogonal matrix $\mathbf{V} \in \mathbb{R}^{N\times N}$ collects the eigenvectors of the GSO and the diagonal matrix $\mathbf{\Lambda}$ its eigenvalues. A graph signal can be denoted as a vector $\mathbf{x} \in \mathbb{R}^N$ where x_i represents the signal value observed at node i. A common tool to model the relationship between the signal $\mathbf{x}$ and its underlying graph are the graph filters. A graph filter $\mathbf{H} \in \mathbb{R}^{N\times N}$ is a linear operator defined as a polynomial of the GSO of the form

$$\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l = \mathbf{V} \sum_{l=0}^{L-1} h_l \mathbf{\Lambda}^l \mathbf{V}^\top = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top, \quad (1)$$

where the filter degree is $L-1$, $\{h_l\}_{l=0}^{L-1}$ represent the filter coefficients, and $\tilde{\mathbf{h}} \in \mathbb{R}^N$ denotes the frequency response of the graph filter. Since $\mathbf{H}$ is a polynomial of $\mathbf{S}$, it readily follows that both matrices have the same eigenvectors.

Graph stationarity. A *random* graph signal $\mathbf{x}$ is stationary on the graph $\mathcal{G}$ if it can be represented as the output of a graph filter $\mathbf{H}$ with a zero mean white signal $\mathbf{w} \in \mathbb{R}^N$ as input, i.e., the covariance of $\mathbf{w}$ is $\mathbb{E}[\mathbf{w}\mathbf{w}^\top] = \mathbf{I}$ and $\mathbf{x} = \mathbf{H}\mathbf{w}$. In turn, if $\mathbf{x}$ is stationary, then its covariance $\mathbf{C}$ is given by

$$\mathbf{C} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{H}\mathbb{E}[\mathbf{w}\mathbf{w}^\top]\mathbf{H}^\top = \mathbf{H}\mathbf{H}^\top = \mathbf{H}^2. \quad (2)$$

In the spectral domain, it can be seen from (2) that the GSO $\mathbf{S}$ and the covariance matrix $\mathbf{C}$ share the same eigenvectors $\mathbf{V}$ [31], [32], [37]. Therefore, graph stationarity implies that the matrices $\mathbf{S}$ and $\mathbf{C}$ commute, i.e., $\mathbf{CS} = \mathbf{SC}$, which is a relevant property to be exploited later on.

Graph smoothness. A graph signal is considered smooth on a graph $\mathcal{G}$ if the signal value at two connected nodes is "close", or equivalently, if the difference between the signal value at neighboring nodes is small. A common approach to quantify the smoothness of a graph signal is by means of the quadratic form [11]

$$\sum_{(i,j)\in\mathcal{E}} A_{ij}(x_i - x_j)^2 = \mathbf{x}^\top \mathbf{L}\mathbf{x}, \quad (3)$$

which quantifies how much the signal $\mathbf{x}$ changes with respect to the notion of similarity encoded in the weights of $\mathbf{A}$. This measure will be referred to as "local variation" (LV) of $\mathbf{x}$. Note that, if the goal is to obtain the mean LV of M graph signals collected in the $N \times M$ matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, this can be achieved by computing

$$\frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m^\top \mathbf{L}\mathbf{x}_m = \frac{1}{M}\sum_{m=1}^{M} \text{tr}(\mathbf{x}_m\mathbf{x}_m^\top\mathbf{L}) = \text{tr}(\hat{\mathbf{C}}\mathbf{L}), \quad (4)$$

where $\hat{\mathbf{C}} := \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m\mathbf{x}_m^\top = \frac{1}{M}\mathbf{X}\mathbf{X}^\top$ denotes the sample estimate of the covariance of $\mathbf{X}$.

III. Influence of hidden variables in the topology inference model

The current section is devoted to formally posing the topology-inference problem when only observations from a subset of nodes of the graph are available. We present a general formulation and highlight the influence of the hidden variables.

Denote as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M] \in \mathbb{R}^{N\times M}$ the collection of M signals defined on top of the *unknown* graph $\mathcal{G}$ with N nodes. Then, we consider that we only observe the values of $\mathbf{X}$ from a subset of nodes $\mathcal{O} \subset \mathcal{V}$ with cardinality $O < N$. In contrast, the values corresponding to the remaining $H = N - O$ nodes

in the subset $\mathcal{H} = \mathcal{V} \setminus \mathcal{O}$ stay hidden[1]. For simplicity and without loss of generality, let the observed nodes correspond to the first O nodes of the graph, so the values of the given signals at $\mathcal{O}$ are collected in the submatrix $\mathbf{X}_{\mathcal{O}} \in \mathbb{R}^{O \times M}$, which is formed by the first O rows of the matrix $\mathbf{X}$. As explained in the previous section, these observations can be used to form the sample covariance matrix. When doing so, it is important to notice the matrices $\mathbf{S} \in \mathbb{R}^{N \times N}$ and $\hat{\mathbf{C}} \in \mathbb{R}^{N \times N}$, which respectively represent the GSO and the sample covariance matrix associated with the full graph $\mathcal{G}$, and the signals $\mathbf{X}$, present the following block structure

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{\mathcal{O}} \\ \mathbf{X}_{\mathcal{H}} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathcal{O}} & \mathbf{S}_{\mathcal{O}\mathcal{H}} \\ \mathbf{S}_{\mathcal{H}\mathcal{O}} & \mathbf{S}_{\mathcal{H}} \end{bmatrix}, \ \hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}_{\mathcal{O}} & \hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}} \\ \hat{\mathbf{C}}_{\mathcal{H}\mathcal{O}} & \hat{\mathbf{C}}_{\mathcal{H}} \end{bmatrix}. \quad (5)$$

The $O \times O$ matrix $\mathbf{S}_{\mathcal{O}}$ denotes the GSO describing the connections between the observed nodes, while the remaining blocks model the edges involving hidden nodes. Similarly, $\hat{\mathbf{C}}_{\mathcal{O}} = \frac{1}{M}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^{\top}$ denotes the sample covariance of the observed signals, and the other blocks denote the submatrices of $\hat{\mathbf{C}}$ involving signal values from the hidden nodes. Since $\mathcal{G}$ is undirected, both $\mathbf{S}$ and $\hat{\mathbf{C}}$ are symmetric, and thus, $\mathbf{S}_{\mathcal{H}\mathcal{O}} = \mathbf{S}_{\mathcal{O}\mathcal{H}}^{\top}$ and $\hat{\mathbf{C}}_{\mathcal{H}\mathcal{O}} = \hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}}^{\top}$.

With the previous definitions in place, the problem of graph learning/network topology inference in the presence of hidden variables is formally introduced next.

Problem 1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with N nodes and GSO $\mathbf{S} \in \mathbb{R}^{N \times N}$, and suppose that $\{\mathcal{V}, \mathcal{E}, N, \mathbf{S}\}$ are all unknown. Given the nodal subset $\mathcal{O} \subset \mathcal{V}$ with cardinality $|\mathcal{O}| = O$, and the observations $\mathbf{X}_{\mathcal{O}} \in \mathbb{R}^{O \times M}$ corresponding to the values of M graph signals observed at the nodes in $\mathcal{O}$, find the underlying graph structure encoded in $\mathbf{S}_{\mathcal{O}} \in \mathbb{R}^{O \times O}$ under the assumptions that:*
(AS1) The number of hidden variables (nodes) is substantially smaller than the number of observed nodes, i.e., $O \lessapprox N$; and
(AS2) There exists a (known) property relating the full graph signals $\mathbf{X} \in \mathbb{R}^{N \times M}$ to the GSO $\mathbf{S}$.

Despite having observations from O nodes, there are still $H = N - O$ nodes that remain unseen and influence the observed signals $\mathbf{X}_{\mathcal{O}}$, rendering the inference problem challenging and severely ill-conditioned. To make the problem more tractable, (AS1) ensures that the number of hidden variables is small. Assumption (AS2) is more generic and establishes that there is a known relationship between the graph signals $\mathbf{X}$ and the full graph $\mathbf{S}$. The particular relationship is further developed in the following sections, where we assume that $\mathbf{X}$ is either smooth (Section IV) or stationary (Section V) on $\mathbf{S}$. The key issue to address is how (AS2), which involves the full signals and GSO, translates to the submatrices $\mathbf{X}_{\mathcal{O}}$, $\mathbf{S}_{\mathcal{O}}$, and $\mathbf{C}_{\mathcal{O}}$ in (5).

Given the above considerations, a general formulation to solve Problem 1 is as follows

$$\begin{aligned} \hat{\mathbf{S}}_{\mathcal{O}} = \operatorname{argmin}_{\mathbf{S}_{\mathcal{O}}} \ & f(\mathbf{S}_{\mathcal{O}}) \\ \text{s. t.} \quad & \mathbf{X}_{\mathcal{O}} \in \mathcal{X}(\mathbf{S}), \\ & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \end{aligned} \quad (6)$$

[1]With a slight abuse of notation, we use H to denote the number of hidden nodes and the square matrix $\mathbf{H}$ to denote a generic graph filter.

where $f(\cdot)$ is a (preferably convex) function that promotes desirable properties on the sought graph. Typical examples include the ℓ_1 norm, the Frobenius norm, the spectral radius, or linear combinations of those [15]. Note that the first constraint in (6) (referred to as observation constraint) takes into account that $\mathcal{X}$ involves the full matrices $\mathbf{X}$ and $\mathbf{S}$ but only $\mathbf{X}_{\mathcal{O}}$ is observed. It is also important to remark that, as will be apparent in the following sections, for observations that are either smooth or stationarity in the graph, the constraint $\mathbf{X}_{\mathcal{O}} \in \mathcal{X}(\mathbf{S})$ can be reformulated in terms of the (sample) covariance matrices $\hat{\mathbf{C}}_{\mathcal{O}} = \frac{1}{M}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^{\top}$ and $\mathbf{C}_{\mathcal{O}} = \mathbb{E}\left[\mathbf{x}_{\mathcal{O}}\mathbf{x}_{\mathcal{O}}^{\top}\right]$. Regarding the second constraint in (6), the set $\mathcal{S}$ collects the requirements for $\mathbf{S}$ to be a specific type of GSO. A typical example is the set of adjacency matrices

$$\mathcal{A} := \{A_{ij} \geq 0; \ \mathbf{A} = \mathbf{A}^{\top}; \ A_{ii} = 0; \ \mathbf{A}\mathbf{1} \geq \mathbf{1}\}, \quad (7)$$

where we require the GSO to have non-negative weights, be symmetric, and have no self-loops, and the last constraint rules out the trivial 0 solution by imposing that every node has at least one neighbor. Analogously, the set of combinatorial Laplacian matrices is

$$\mathcal{L} := \{L_{ij} \leq 0 \text{ for } i \neq j; \ \mathbf{L} = \mathbf{L}^{\top}; \ \mathbf{L}\mathbf{1} = \mathbf{0}; \ \mathbf{L} \succeq 0\}, \quad (8)$$

where we require the GSO to be a positive semidefinite matrix, have non-positive off-diagonal values, have positive entries on its diagonal, and have the constant vector as an eigenvector (i.e, the sum of the entries of each row to be zero). Lastly, we want to stress that the objective $f(\mathbf{S}_{\mathcal{O}})$ and the constraint $\mathbf{S}_{\mathcal{O}} \in \mathcal{S}$ can be alternatively formulated based on the full GSO $\mathbf{S}$, provided that we know that the structural properties (for instance sparsity in the objective and positive entries in the constraints) hold also for the non-observed parts of $\mathbf{S}$. Such an approach is suitable when the interest goes beyond $\mathbf{S}_{\mathcal{O}}$ and spans the estimation of the links involving the nodes in $\mathcal{H}$.

Hidden variables in correlation and partial-correlation networks: Before discussing our specific solutions to Problem 1, a relevant question is how classical topology-inference approaches (namely correlation and partial-correlation networks) handle the problem of latent nodal variables. The so-called direct methods consider that a link between nodes i and j exists based only on a pairwise similarity metric between the signals observed at i and j. Within this class of methods, correlation networks set the similarity metric to the correlation and, as a result, $\mathbf{S}$ corresponds to a (thresholded) version of $\mathbf{C}$. Given their simplicity, the generalization of direct methods to setups where hidden variables are present is straightforward and simply given by $\mathbf{S}_{\mathcal{O}} = \hat{\mathbf{C}}_{\mathcal{O}}$. Nevertheless, a high correlation between two nodes can be due to global network effects rather than to the direct influence among pairs of neighbors, calling for more involved topology-inference methods. To that end, partial-correlation methods, including the celebrated graphical Lasso (GL) algorithm [2], propose estimating the graph as a matrix of partial correlation coefficients, which boils down to assuming that the connectivity patterns can be identified as $\mathbf{S} = \mathbf{C}^{-1}$, with $\mathbf{C}^{-1}$ being known as the precision matrix. When hidden variables are present, the submatrix of the precision matrix is given by $\mathbf{C}_{\mathcal{O}}^{-1} = \mathbf{S}_{\mathcal{O}} - \mathbf{B}$, with $\mathbf{B} = \mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{H}}^{-1}\mathbf{S}_{\mathcal{H}\mathcal{O}}$ being a low-rank matrix since $H \ll O$. Leveraging this structure, the

authors in [26] modified the GL algorithm to deal with hidden variables via a maximum-likelihood estimator augmented with a nuclear-norm regularizer to promote low rankness in $\mathbf{B}$. The resulting algorithm is known as latent variable graphical Lasso (LVGL) and is given by

$$\begin{aligned}\max_{\mathbf{S}_{\mathcal{O}}-\mathbf{B}\succeq\mathbf{0},\,\mathbf{B}\succeq\mathbf{0}} \log\det(\mathbf{S}_{\mathcal{O}}-\mathbf{B})-\text{trace}(\hat{\mathbf{C}}_{\mathcal{O}}(\mathbf{S}_{\mathcal{O}}-\mathbf{B}))\\ -\lambda_1\|\mathbf{S}_{\mathcal{O}}\|_1-\lambda_2\|\mathbf{B}\|_*,\end{aligned} \tag{9}$$

where $\hat{\mathbf{C}}_{\mathcal{O}}$ represents the sample covariance of the observed data and λ_1 and λ_2 are regularization constants [26].

Rather than assuming that the relation between $\mathbf{X}$ and $\mathbf{S}$ postulated in (AS2) is given by either correlations or partial-correlations, this paper looks at setups where the operational assumption is that the observed signals are: i) smooth on the graph; ii) stationary on the graph; and iii) both smooth and stationary. Sections IV-VI deal with each of those three setups. Section VII evaluates numerically the performance of the developed algorithms and compares it with that of classical correlation and LVGL schemes.

IV. Topology inference from smooth signals

In this section, we address Problem 1 by particularizing (6) to the case of the signals $\mathbf{X}$ being smooth on $\mathcal{G}$.

As explained in Section II, a natural way of measuring the smoothness of (a set of) graph signals is to leverage the graph Laplacian and compute their LV as $\frac{1}{M}\text{tr}(\mathbf{X}\mathbf{X}^\top\mathbf{L})$ [cf. (4)]. As a result, in this section we set $\mathbf{S}=\mathbf{L}$ and focus on $\hat{\mathbf{C}}=\frac{1}{M}\mathbf{X}\mathbf{X}^\top$. Recall that, due to the existence of hidden variables, the whole covariance matrix is not observed. To account for this and leveraging the block definition of $\hat{\mathbf{C}}$ and $\mathbf{S}$ introduced in (5), we can rewrite the LV of our dataset as

$$\text{tr}(\hat{\mathbf{C}}\mathbf{L})=\text{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}})+2\text{tr}(\hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^\top)+\text{tr}(\hat{\mathbf{C}}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}}), \tag{10}$$

where only $\hat{\mathbf{C}}_{\mathcal{O}}=\frac{1}{M}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^\top$ is assumed to be known and the influence of the hidden variables in the LV has been made explicit.

Although the block-wise smoothness presented in (10) could be directly employed to approach the network-topology inference as an optimization problem, most of the submatrices are not known and need to be estimated. Incorporating the terms $\mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^\top$ and $\mathbf{C}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}}$ would directly render the problem non-convex. To circumvent this issue, we lift the problem by defining the matrix $\mathbf{K}:=\mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^\top\in\mathbb{R}^{O\times O}$. Since (AS1) guarantees that $\text{rank}(\mathbf{K})\leq H\ll O$, we exploit the low-rank structure of the matrix $\mathbf{K}$ in our formulation. Correspondingly, we also define the matrix $\mathbf{R}:=\mathbf{C}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}}\in\mathbb{R}^{H\times H}$ and note that, since $\mathbf{R}$ is the product of two positive semidefinite matrices, it has positive eigenvalues and, as a result, it holds that $\text{tr}(\mathbf{R})\geq 0$.

With these considerations in mind, the network topology inference from smooth signals is formulated as

$$\begin{aligned}\min_{\mathbf{L}_{\mathcal{O}},\mathbf{K},\mathbf{R}}\ & \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}})+2\text{tr}(\mathbf{K})+\text{tr}(\mathbf{R})+\alpha\|\mathbf{L}_{\mathcal{O}}\|_{F,off}^2\\ &-\beta\log(\text{diag}(\mathbf{L}_{\mathcal{O}}))+\gamma\|\mathbf{K}\|_*\\ \text{s. t.}\ & \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}})+2\text{tr}(\mathbf{K})+\text{tr}(\mathbf{R})\geq 0,\\ & \text{tr}(\mathbf{R})\geq 0,\\ & \mathbf{L}_{\mathcal{O}}\in\bar{\mathcal{L}},\end{aligned} \tag{11}$$

where $\|\cdot\|_{F,off}^2$ denotes the Frobenius norm excluding the elements of the diagonal. This term, together with $\log(\text{diag}(\mathbf{L}_{\mathcal{O}}))$, serves to control the sparsity of $\mathbf{L}_{\mathcal{O}}$. Furthermore, the logarithmic barrier rules out the trivial solution of $\mathbf{L}_{\mathcal{O}}=\mathbf{0}$. The nuclear norm $\|\cdot\|_*$ is a convex regularizer that promotes low-rank solutions for the matrix $\mathbf{K}$ and it is typically employed as a surrogate of the (non-convex) rank constraint. The adoption of the nuclear norm, together with the consideration of the matrices $\mathbf{K}$ and $\mathbf{R}$, ensure the convexity of (11) so a globally optimum solution can be efficiently found. The weights $\alpha,\beta,\gamma\geq 0$ control the trade-off between the regularizers, the first constraint ensures that the LV is non-negative, and the second constraint captures that fact of matrix[2] $\mathbf{R}$ being PSD.

The last point to discuss in detail is the form of $\bar{\mathcal{L}}$. Mathematically, the set $\bar{\mathcal{L}}$ is equivalent to the set of combinatorial Laplacians $\mathcal{L}$, but replacing the condition $\mathbf{L1}=\mathbf{0}$ with $\mathbf{L1}\geq\mathbf{0}$, i.e., $\bar{\mathcal{L}}:=\{L_{ij}\leq 0 \text{ for } i\neq j;\ \mathbf{L}=\mathbf{L}^\top;\ \mathbf{L1}\geq\mathbf{0};\ \mathbf{L}\succeq 0\}$. The modification is required because, strictly speaking, $\mathbf{L}_{\mathcal{O}}$ is not a combinatorial Laplacian. The existence of links between the elements in $\mathcal{O}$ and the hidden nodes in $\mathcal{H}$ give rise to non-zero (negative) entries in $\mathbf{L}_{\mathcal{O}\mathcal{H}}$ and, as a result, the sum of the off-diagonal elements of $\mathbf{L}_{\mathcal{O}}$ can be smaller than the value of the associated diagonal elements (which account for the links in both $\mathcal{O}$ and $\mathcal{H}$). Intuitively, the more relaxed condition $\mathbf{L}_{\mathcal{O}}\mathbf{1}\geq\mathbf{0}$ enlarges the set of feasible solutions rendering the inference process harder to solve, an issue that has been observed when running the numerical experiments. Moreover, when estimating the diagonal of $\mathbf{L}_{\mathcal{O}}$ we are indirectly estimating the number of edges between observed and the hidden nodes. This could be potentially leveraged to estimate links with non-observed nodes, but this entails a more challenging problem that goes beyond the scope of the paper. An approach to bypass some of these issues is analyzed next.

A. Exploiting the Laplacian of the observed adjacency matrix

The Laplacian $\mathbf{L}$ offers a neat way to measure the smoothness of graph signals [cf. (3)]. However, when addressing the problem of estimating the Laplacian from smooth signals under the presence of hidden nodes, we must face the challenges associated with the fact of the submatrix $\mathbf{L}_{\mathcal{O}}$ not being a Laplacian itself. As discussed in the preceding paragraphs, this requires dropping some of the Laplacian constraints from the optimization, leading to a looser recovery framework. To circumvent these issues, rather than estimating $\mathbf{L}_{\mathcal{O}}$, this section looks at the problem of estimating $\tilde{\mathbf{L}}_{\mathcal{O}}:=\text{diag}(\mathbf{A}_{\mathcal{O}}\mathbf{1})-\mathbf{A}_{\mathcal{O}}$, the Laplacian associated with the observed adjacency matrix $\mathbf{A}_{\mathcal{O}}\in\mathbb{R}^{O\times O}$. In contrast to $\mathbf{L}_{\mathcal{O}}$, the matrix $\tilde{\mathbf{L}}_{\mathcal{O}}$ is a proper combinatorial Laplacian ($\tilde{\mathbf{L}}_{\mathcal{O}}\in\mathcal{L}$) and, hence, the original Laplacian constraints can be restored. The remaining of this section is devoted to reformulating (11) in terms of $\tilde{\mathbf{L}}_{\mathcal{O}}$.

Upon defining the $O\times O$ diagonal matrices $\mathbf{D}_{\mathcal{O}}:=\text{diag}(\mathbf{A}_{\mathcal{O}}\mathbf{1})$ and $\mathbf{D}_{\mathcal{O}\mathcal{H}}:=\text{diag}(\mathbf{A}_{\mathcal{O}\mathcal{H}}\mathbf{1})$, which count the number of observed and hidden neighbors for the nodes in $\mathcal{O}$, the matrix $\mathbf{L}_{\mathcal{O}}$ is expressed as $\mathbf{L}_{\mathcal{O}}=\mathbf{D}_{\mathcal{O}}+\mathbf{D}_{\mathcal{O}\mathcal{H}}-\mathbf{A}_{\mathcal{O}}=$

[2]From an algorithmic point of view, it is worth noticing that the matrix $\mathbf{R}$ always appears as $\text{tr}(\mathbf{R})$ in (11). As a result, if convenient to reduce the numerical burden, one can replace $\text{tr}(\mathbf{R})$ with r and optimize over r in lieu of $\mathbf{R}$. See the related formulation in (13) for details.

$\tilde{\mathbf{L}}_{\mathcal{O}} + \mathbf{D}_{\mathcal{O}\mathcal{H}}$. With this equivalence, the smoothness penalty in (11) is rewritten as

$$\begin{aligned}\text{tr}(\mathbf{C}\mathbf{L}) &= \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}) + 2\text{tr}(\mathbf{K}) + \text{tr}(\mathbf{R}) \\ &= \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + \text{tr}(\mathbf{R}), \end{aligned} \tag{12}$$

where $\tilde{\mathbf{K}} := \mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}/2 + \mathbf{K}$. Because the entries of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ depend on the presence of edges between the observed and the hidden nodes, if the graph is sparse, the matrix $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ will be a low-rank matrix. Furthermore, since the sparsity pattern of the diagonal of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ depends on the matrix $\mathbf{A}_{\mathcal{O}\mathcal{H}} = -\mathbf{L}_{\mathcal{O}\mathcal{H}}$, it follows that the column sparsity pattern of $\mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}$ matches that of $\mathbf{K}$, and thus, $\tilde{\mathbf{K}}$ is also low rank.

With these considerations in mind, we reformulate the optimization in (11) replacing $\mathbf{L}_{\mathcal{O}}$ with $\tilde{\mathbf{L}}_{\mathcal{O}}$, resulting in the following convex optimization problem

$$\begin{aligned}\min_{\tilde{\mathbf{L}}_{\mathcal{O}},\tilde{\mathbf{K}},r} \; & \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + r + \alpha\|\tilde{\mathbf{L}}_{\mathcal{O}}\|_{F,off}^2 \\ & - \beta\log(\text{diag}(\tilde{\mathbf{L}}_{\mathcal{O}})) + \gamma_*\|\tilde{\mathbf{K}}\|_* + \gamma_{2,1}\|\tilde{\mathbf{K}}\|_{2,1} \\ \text{s. t.} \quad & \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + r \geq 0, \\ & r \geq 0 \\ & \tilde{\mathbf{L}}_{\mathcal{O}} \in \mathcal{L}, \end{aligned} \tag{13}$$

where $\bar{\mathcal{L}}$ in (11) has been replaced with $\mathcal{L}$ in (13), which is the set of all valid combinatorial Laplacian matrices defined in (8). Moreover, knowing that the matrix $\mathbf{R}$ only appears as $\text{tr}(\mathbf{R})$ we replace it with the nonnegative variable r to alleviate the numerical burden. Note that, although we replaced $\mathbf{K}$ with $\tilde{\mathbf{K}}$, the terms previously associated with $\mathbf{K}$ in (11) remain unchanged in (13). Nonetheless, while the original matrix $\mathbf{K} \in \mathbb{R}^{O\times O}$ is low rank because it is the product of a tall $O \times H$ matrix and a fat $H \times O$ matrix, the low-rankness of $\tilde{\mathbf{K}}$ is a byproduct of the sparsity of the graph. More precisely, the matrix $\tilde{\mathbf{K}}$ involves the product of the square (full rank) matrix $\mathbf{C}_{\mathcal{O}}$ and the diagonal matrix $\mathbf{D}_{\mathcal{O}\mathcal{H}}$. Since the diagonal of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ is sparse, such a product gives rise to a matrix with several zero columns, with the rank of the resultant matrix coinciding with the number of non-zero columns. We exploit this structure by further regularizing the matrix $\tilde{\mathbf{K}}$ with the $\ell_{2,1}$ norm.

Indeed, two different configurations of (13) can be obtained depending on the values of the regularization constants. Setting $\gamma_{2,1} = 0$ we promote a solution with a low rank on $\tilde{\mathbf{K}}$ by applying the nuclear norm regularization. Since the nuclear norm minimization does not ensure the desired column-sparsity of $\tilde{\mathbf{K}}$, an alternative is to set $\gamma_* = 0$ and rely on the penalty $\|\mathbf{K}\|_{2,1}$. The computation of $\|\tilde{\mathbf{K}}\|_{2,1}$ can be understood as a two-step process where one first obtains the ℓ_2 norm of each of the columns of $\tilde{\mathbf{K}}$ and, then, the ℓ_1 norm of the resulting row vector is computed. This regularization is commonly known as the group Lasso penalty [38], [39] and has been used in a number of sparse-recovery problems. The results in Section VII will illustrate that the formulation in (13) succeeds in promoting the desired column-sparsity pattern when using the appropriate values for the hyperparameters γ_* and $\gamma_{2,1}$. Note also that, by looking at the non-zero columns of $\tilde{\mathbf{K}}$, the nodes in $\mathcal{O}$ with connections to hidden nodes can be identified.

V. Topology inference from stationary signals

In this section, instead of relying on the smoothness of the signals $\mathbf{X}$, we approach Problem 1 by modifying (AS2) and considering that the data is stationary on the sought graph. The assumption of $\mathbf{X}$ being stationary on $\mathcal{G}$ is tantamount to the matrices $\mathbf{C}$ and $\mathbf{S}$ sharing the same eigenvectors $\mathbf{V}$ [31]. As a result, the approach for the fully observable case is to use the observations to estimate the sample covariance $\hat{\mathbf{C}}$ and then rely on the sample covariance to estimate the eigenvectors $\mathbf{V}$ [13]. However, when dealing with hidden variables, there is no obvious way to obtain $\mathbf{V}_{\mathcal{O}}$, the submatrix of the eigenvectors of the full covariance, using as input the submatrix $\hat{\mathbf{C}}_{\mathcal{O}}$. To bypass this problem, instead of requiring the eigenvectors of $\mathbf{C}$ and $\mathbf{S}$ being the same, our approach is to require that $\mathbf{C}$ and $\mathbf{S}$ commute, i.e., that the equation $\mathbf{C}\mathbf{S} = \mathbf{S}\mathbf{C}$ must hold [40]. To see why this condition leads to a more tractable formulation, let us leverage the block structure of $\mathbf{C}$ and $\mathbf{S}$ described in (5). It follows readily that the upper left submatrix of size $O \times O$ in both sides of the equality $\mathbf{C}\mathbf{S} = \mathbf{S}\mathbf{C}$ is given by

$$\mathbf{C}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{O}\mathcal{H}}^{\top} = \mathbf{S}_{\mathcal{O}}\mathbf{C}_{\mathcal{O}} + \mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{C}_{\mathcal{O}\mathcal{H}}^{\top}. \tag{14}$$

The above expression succeeds in relating the sought $\mathbf{S}_{\mathcal{O}}$ with $\mathbf{C}_{\mathcal{O}}$, which can be efficiently estimated using $\mathbf{X}_{\mathcal{O}}$. Furthermore, (14) reveals that when hidden variables are present, we cannot simply ask $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{C}_{\mathcal{O}}$ to commute, but we also need to account for the associated terms $\mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{O}\mathcal{H}}^{\top}$ and $\mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{C}_{\mathcal{O}\mathcal{H}}^{\top}$.

Implementing steps similar to those in Section IV, we can lift the problem defining the matrix $\mathbf{K} = \mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{O}\mathcal{H}}^{\top} \in \mathbb{R}^{O\times O}$ and leverage the fact that $\text{rank}(\mathbf{K}) \leq H \ll O$, due to (AS1). Note that the matrix $\mathbf{K}$ is equivalent to the one defined in Section IV with the only difference that now we use a block from the generic GSO $\mathbf{S}_{\mathcal{O}\mathcal{H}}$ instead of the Laplacian $\mathbf{L}_{\mathcal{O}\mathcal{H}}$. Moreover, since both $\mathbf{C}$ and $\mathbf{S}$ are symmetric matrices, we have that $\mathbf{K}^{\top} = \mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{C}_{\mathcal{O}\mathcal{H}}^{\top}$. Then, under the general assumption that graphs are typically sparse, we can approach Problem 1 with stationary observations by solving

$$\begin{aligned}\min_{\mathbf{S}_{\mathcal{O}},\mathbf{K}} \; & \|\mathbf{S}_{\mathcal{O}}\|_0 \\ \text{s. t.} \quad & \mathbf{C}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{K} = \mathbf{S}_{\mathcal{O}}\mathbf{C}_{\mathcal{O}} + \mathbf{K}^{\top}, \\ & \text{rank}(\mathbf{K}) \leq H, \\ & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \end{aligned} \tag{15}$$

where the ℓ_0 norm promotes sparse solutions, the equality constraint ensures commutativity of the GSO and the covariance while accounting for latent nodes, and the rank constraint captures the low rank of $\mathbf{K}$ due to (AS1).

Regarding the specific choice of the GSO, when the interest is in the Laplacian matrix we set $\mathbf{S}_{\mathcal{O}} = \tilde{\mathbf{L}}_{\mathcal{O}}$, with $\tilde{\mathbf{L}}_{\mathcal{O}}$ denoting the Laplacian of the observed adjacency matrix. Then, the matrix $\mathbf{K}$ is replaced with $\tilde{\mathbf{K}} = \mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}} + \mathbf{K}$, which accounts for the fact of using $\tilde{\mathbf{L}}_{\mathcal{O}}$ instead of $\mathbf{L}_{\mathcal{O}}$ in (14). This was further motivated in Section IV-A, and the discussion provided there also applies here.

The presence of the rank constraint and the ℓ_0 norm renders (15) non-convex and computationally hard to solve. Furthermore, the first constraint assumes perfect knowledge of $\mathbf{C}_{\mathcal{O}}$, which may not always represent a practical setup. These issues are addressed in the next section.

A. Convex and robust stationary topology inference

A natural approach to deal with (15) is to relax the non-convex terms, replacing the ℓ_0 norm with the ℓ_1 norm and the

rank constraint with the nuclear norm, their closest convex surrogates. Furthermore, in most practical scenarios the ensemble covariance $\mathbf{C}_{\mathcal{O}}$ is not known and one must rely on its sampled counterpart $\hat{\mathbf{C}}_{\mathcal{O}}$. This requires relaxing the equality constraint $\mathbf{C}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{K} = \mathbf{S}_{\mathcal{O}}\mathbf{C}_{\mathcal{O}} + \mathbf{K}^\top$ and replacing it with a constraint that guarantees that the terms on the left-hand side and right-hand side are similar but not necessarily the same. Taking all these considerations into account, the relaxed convex topology-inference problem is

$$\begin{aligned} \min_{\mathbf{S}_{\mathcal{O}},\mathbf{K}} \ & \|\mathbf{S}_{\mathcal{O}}\|_1 + \eta\|\mathbf{K}\|_* \\ \text{s. t. } & \|\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{K} - \mathbf{S}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}} - \mathbf{K}^\top\|_F^2 \leq \epsilon, \\ & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \end{aligned} \tag{16}$$

where $\eta \geq 0$ controls the low rankness of $\mathbf{K}$. Regarding the (relaxed) stationarity constraint, the squared Frobenius norm has been adopted to measure the similarity between the matrices at hand, but other (convex) distances could be alternatively used. It is also important to note that the value of the non-negative constant ϵ should be selected based on prior knowledge on the noise level present in the observations and, more importantly, the number of samples M used to estimate the covariance. Clearly, if $M < O$, the matrix is not full rank, increasing notably the size of the feasible set. On the other hand, if $M \to \infty$, one can set $\epsilon = 0$. This reduces drastically the degrees of freedom of the formulation and, as a result, renders more likely the solution to (16) to coincide with the actual GSO.

Remark 1 (Reweighted algorithm): The formulation in (16) is convex and robust. However, while replacing the original ℓ_0 norm with the convex ℓ_1 norm constitutes a common approach, it is well-known that non-convex surrogates can lead to sparser solutions. Indeed, a more sophisticated alternative in the context of sparse recovery is to define δ as a small positive number and replace the ℓ_0 norm with a (non-convex) logarithmic penalty $\|\mathbf{S}_{\mathcal{O}}\|_0 \approx \sum_{i,j=1}^{O} \log(|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \delta)$ [41]. An efficient way to handle the non-convexity of the logarithmic penalty is to rely on a majorization-minimization (MM) approach [42], which considers an iterative linear approximation to the concave objective and leads to an *iterative* re-weighted ℓ_1 minimization. To be specific, with $t = 1, ..., T$ being the iteration index, adopting such an approach for the problem in (16) results in

$$\begin{aligned} \mathbf{S}_{\mathcal{O}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{O}},\mathbf{K}}{\operatorname{argmin}} \ & \sum_{i,j=1}^{O} [\mathbf{W}^{(t)}]_{ij}|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \eta\|\mathbf{K}\|_* \\ \text{s. t.} \quad & \mathbf{C}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{K} = \mathbf{S}_{\mathcal{O}}\mathbf{C}_{\mathcal{O}} + \mathbf{K}^\top, \\ & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \end{aligned} \tag{17}$$

with $\mathbf{W}^{(t)}$ being defined as $[\mathbf{W}^{(t)}]_{ij} = \left(\left|\left[\mathbf{S}_{\mathcal{O}}^{(t-1)}\right]_{ij}\right| + \delta\right)^{-1}$. Since the iterative algorithm penalizes (assigns a larger weight to) entries of $\mathbf{S}_{\mathcal{O}}$ that are close to zero, the obtained solution is typically sparser at the expense of a higher computational cost. Finally, note that the absolute values can be removed whenever the constraint $[\mathbf{S}_{\mathcal{O}}]_{ij} \geq 0$ is enforced.

B. Exploiting structure through alternating optimization

In the previous section, the product of the unknown matrices $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$ was absorbed into matrix $\mathbf{K}$. Since such a matrix is low rank, the convex nuclear norm was used to promote low-rank solutions while achieving convexity. However, when implementing this approach, there were other properties (such as $\mathbf{S}_{\mathcal{OH}}$ being sparse) that were ignored. A reasonable question is, hence, if the judicious incorporation of the additional information outperforms the potential loss of convexity. In this section, we propose an efficient alternating *non-convex* algorithm that accounts for the additional structure present in our setup. Its associated recovery performance (along with comparisons to its convex counterparts) will be tested in Section VII.

A well-established approach in low-rank optimization is to factorize the matrix of interest as the product of a tall and fat matrix, which boils down to replacing $\mathbf{K}$ with the original submatrices $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$. Moreover, when the value of H is unknown, which determines the size of $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$, a principled approach is to rely on an upper bound on H and add the Frobenius terms $\|\mathbf{C}_{\mathcal{OH}}\|_F$ and $\|\mathbf{S}_{\mathcal{OH}}\|_F$ to the objective function (see, e.g., [43] for a formal derivation of this approach). In our particular setup, this factorization has the additional benefit of $\mathbf{S}_{\mathcal{OH}}$ being sparse. Then, the resulting non-convex optimization problem is given by

$$\begin{aligned} \min_{\mathbf{S}_{\mathcal{O}},\mathbf{C}_{\mathcal{OH}},\mathbf{S}_{\mathcal{OH}}} \ & \sum_{i,j=1}^{O} \log(|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \delta) + \eta\|\mathbf{S}_{\mathcal{OH}}\|_F^2 \\ & + \nu \sum_{i,j=1}^{O,H} \log(|[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \delta) + \eta\|\mathbf{C}_{\mathcal{OH}}\|_F^2 \\ & + \rho\|\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top - \mathbf{S}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}} - \mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top\|_F^2 \\ \text{s. t.} \quad & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \ \ \mathbf{S}_{\mathcal{OH}} \in \mathcal{S}_{\mathcal{OH}}, \end{aligned} \tag{18}$$

Clearly, problem (18) guarantees that the rank of the matrix $\mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top$ is upper bounded by the size of its composing factors $\mathbf{S}_{\mathcal{OH}}$ and $\mathbf{C}_{\mathcal{OH}}$. In this case, the sparse solutions for $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{S}_{\mathcal{OH}}$ are promoted by means of the (concave) logarithmic penalty, introduced on Remark 1. The robust commutativity constraint is placed on the objective function as a penalty term, and the set $\mathcal{S}_{\mathcal{OH}}$ captures the fact that $\mathbf{S}_{\mathcal{OH}}$ is a block from the GSO. In its simplest form, we have that $\mathcal{S}_{\mathcal{OH}} := \{S_{ij} \geq 0\}$ if the GSO is the adjacency matrix, and $\mathcal{S}_{\mathcal{OH}} := \{S_{ij} \leq 0\}$ if it is set to the Laplacian matrix.

The main drawback associated with the formulation in (18) is that the presence of the bilinear term $\mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top$ and the logarithmic penalty render the problem non-convex. To address this issue, we implement a Block Successive Upper bound Minimization (BSUM) algorithm [44], an iterative approach that blends techniques from MM and alternating optimization. Then, we find a solution to (18) by iterating between the following thee steps.

Step 1. Given the estimates $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$, we substitute $\mathbf{C}_{\mathcal{OH}} = \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\mathbf{S}_{\mathcal{OH}} = \hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$ into (18) and solve it to estimate $\mathbf{S}_{\mathcal{O}}$. This yields

$$\begin{aligned} \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{O}} \in \mathcal{S}}{\operatorname{argmin}} \ & \sum_{i,j=1}^{O} [\mathbf{W}_{\mathcal{O}}^{(t)}]_{ij}|[\mathbf{S}_{\mathcal{O}}]_{ij}| \\ & + \rho\|\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}[\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}]^\top - \mathbf{S}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}} - \hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}[\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}]^\top\|_F^2, \end{aligned} \tag{19}$$

where the logarithmic penalty is approximated by the re-weighted ℓ_1 norm as detailed after (17).

Step 2. Given the estimate $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ from the previous iteration, and leveraging the estimate $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ from the last step, we estimate the matrix $\mathbf{S}_{\mathcal{OH}}$ by solving

$$\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{OH}} \in \mathcal{S}_{\mathcal{OH}}}{\operatorname{argmin}} \sum_{i,j=1}^{O,H} [\mathbf{W}_{\mathcal{OH}}^{(t)}]_{ij} |[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \eta \|\mathbf{S}_{\mathcal{OH}}\|_F^2 + \rho \|\hat{\mathbf{C}}_{\mathcal{O}} \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} + \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)} \mathbf{S}_{\mathcal{OH}}^{\top} - \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} \hat{\mathbf{C}}_{\mathcal{O}} - \mathbf{S}_{\mathcal{OH}} [\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}]^{\top} \|_F^2. \tag{20}$$

Step 3. With the estimates from the previous steps in place, the last step involves estimating the matrix $\mathbf{C}_{\mathcal{OH}}$ by solving

$$\hat{\mathbf{C}}_{\mathcal{OH}}^{(t+1)} := \underset{\mathbf{C}_{\mathcal{OH}}}{\operatorname{argmin}} \quad \eta \|\mathbf{C}_{\mathcal{OH}}\|_F^2 \\ \|\hat{\mathbf{C}}_{\mathcal{O}} \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} + \mathbf{C}_{\mathcal{OH}} [\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}]^{\top} - \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} \hat{\mathbf{C}}_{\mathcal{O}} - \hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)} \mathbf{C}_{\mathcal{OH}}^{\top} \|_F^2. \tag{21}$$

The alternating algorithm is initialized by solving (16) to obtain $\hat{\mathbf{K}}$ and setting $\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(0)}$ as

$$\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)} = \mathbf{F}_{\mathcal{OH}} \mathbf{\Sigma}_{\mathcal{H}}^{\frac{1}{2}} \text{ and } \hat{\mathbf{S}}_{\mathcal{OH}}^{(0)} = \mathbf{G}_{\mathcal{OH}} \mathbf{\Sigma}_{\mathcal{H}}^{\frac{1}{2}}, \tag{22}$$

where $\mathbf{F}_{\mathcal{OH}}$ and $\mathbf{G}_{\mathcal{OH}}$ are the left and right singular vectors associated with the top H singular values, $\mathbf{\Sigma}_{\mathcal{H}}$, obtained from the singular value decomposition $\hat{\mathbf{K}} = \mathbf{F} \mathbf{\Sigma} \mathbf{G}^{\top}$. A summary of the proposed iterative algorithm is presented in Algorithm 1.

The three steps proposed in (19)-(21) are iterated until convergence to a stationary point is achieved, a result that is formally stated next.

Proposition 1. *Denote with f the objective function in* (18)*. Let $\mathcal{Y}^*$ be the set of stationary points of* (18)*, and let $\mathbf{y}^{(t)} = [\text{vec}(\mathbf{S}_{\mathcal{O}}^{(t)})^{\top}, \text{vec}(\mathbf{S}_{\mathcal{OH}}^{(t)})^{\top}, \text{vec}(\mathbf{C}_{\mathcal{OH}}^{(t)})^{\top}]^{\top}$ be the solution generated after running the 3 steps in* (19)-(21) *t times. Then, the solution generated by the iterative algorithm* (19)-(21) *converges to a stationary point of f as t goes to infinity, i.e.,*

$$\lim_{t \to \infty} d(\mathbf{y}^{(t)}, \mathcal{Y}^*) = 0,$$

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

Note that convergence was not obvious since at least one of the steps does not have a unique minimizer, and the first and second steps employ an approximation of the objective function in (18). The details of the proof, which relies on convergence results for BSUM schemes [44, Th. 1b], are provided in Appendix A.

While incurring additional computational costs (see Remark 2 for more details), the numerical tests in Section VII confirm that the supplemental structure incorporated by replacing $\mathbf{K}$ with $\mathbf{S}_{\mathcal{OH}}$ and $\mathbf{C}_{\mathcal{OH}}$ together with the re-weighted ℓ_1 approach for encouraging sparsity give rise to a better network reconstruction, provided that the iterative optimization is initialized with the solution to the convex formulation in (16). Last but not least, notice that an additional benefit of the formulation in (18) is that, by analyzing $\hat{\mathbf{S}}_{\mathcal{OH}}$, information of the potential links between nodes in $\mathcal{O}$ and the hidden nodes in $\mathcal{H}$ is obtained. While network-tomography schemes [2] go beyond the scope of this paper, the results in this section can be used as a first step towards that goal.

Remark 2 (Computational complexity): The computational complexity required to solve the optimization problems proposed in this paper scales polynomially with the size of the graph. More specifically, since (AS1) guarantees that $H \ll O$, for the convex formulations, the complexity scales as $\mathsf{O}(O^7)$, which is an order similar to that of the "plain vanilla" LVGL in (9), but considerably larger than the order $\mathsf{O}(MO^2)$ for correlation networks. Regarding the complexity to solve the non-convex formulation in (18) using Algorithm 1, each of the steps (19)-(21) entails solving a convex problem, so the complexity scales as $\mathsf{O}(TO^7)$, with T denoting the number of iterations. In practice, our simulations show that the number of iterations required to converge in all tested scenarios is fairly low (with T taking values between 3-6), which is a behavior also observed in other applications of the BSUM algorithm to sparsity-promoting biconvex problems. As a result, the complexity to solve the non-convex problem in (18) is expected to scale similarly to that required to solve the non-iterative convex formulations presented in the previous sections. While this complexity is associated with the fact of considering challenging operating conditions, the aforementioned levels hinder the application of the proposed algorithms to large graphs. An approach to mitigate this issue is to exploit the structure of the problems at hand, developing tailored block-coordinate algorithms that solve for each variable separately and exploit the sparsity of the GSO. Indeed, some algorithmic alternatives such as projected gradient and the alternating direction method of multipliers (ADMM) could be applied to implement more scalable and efficient models [15], [21], [45]. Although interesting, the development of efficient algorithms is beyond the scope of this paper so it is left as future work.

Algorithm 1: BSUM graph-learning method for stationary signals with hidden variables (BSUM-GSHV)

Input: $\hat{\mathbf{C}}_{\mathcal{O}}$

Outputs: $\hat{\mathbf{S}}_{\mathcal{O}}$, $\hat{\mathbf{C}}_{\mathcal{OH}}$, and $\hat{\mathbf{S}}_{\mathcal{OH}}$

Initialize $\hat{\mathbf{S}}_{\mathcal{O}}^{(0)}$ and $\hat{\mathbf{K}}$ by solving (16)

Initialize $\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(0)}$ following (22)

for $t = 0$ **to** $T - 1$ **do**

  Update $\mathbf{W}_{\mathcal{O}}^{(t)} = \left(\left| \hat{\mathbf{S}}_{\mathcal{O}}^{(t)} \right| + \delta \right)^{-1}$ and $\mathbf{W}_{\mathcal{OH}}^{(t)} = \left(\left| \hat{\mathbf{S}}_{\mathcal{OH}}^{(t)} \right| + \delta \right)^{-1}$

  Update $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ by solving (19) using $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$

  Update $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}$ by solving (20) using $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$

  Update $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t+1)}$ by solving (21) using $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}$

end

$\hat{\mathbf{S}}_{\mathcal{O}} = \hat{\mathbf{S}}_{\mathcal{O}}^{(T)}$, $\hat{\mathbf{S}}_{\mathcal{OH}} = \hat{\mathbf{S}}_{\mathcal{OH}}^{(T)}$, $\hat{\mathbf{C}}_{\mathcal{OH}} = \hat{\mathbf{C}}_{\mathcal{OH}}^{(T)}$

Remark 3 (Graph stationary vis-à-vis graph smoothness): Suppose that we are given two datasets $\mathbf{X}_{\mathcal{O}}$ and $\mathbf{X}'_{\mathcal{O}}$, both with the same number of signals. Moreover, suppose that we also know that the observed signals $\mathbf{X}_{\mathcal{O}}$ are smooth on an unknown graph, that $\mathbf{X}'_{\mathcal{O}}$ are stationary on an unknown graph, and that our goal is to identify the underlying graphs. Based on that information, we run the algorithms in Section IV for the dataset $\mathbf{X}_{\mathcal{O}}$ and those in this section for the dataset $\mathbf{X}'_{\mathcal{O}}$. An interesting question is which one yields a better recovery result. While the exact answer depends on all the particularities of each of the setups, from a general point of view stationary schemes are expected to achieve better results. The reason is that stationarity strongly limits the degrees of

freedom of the GSO, while smoothness is a more lenient assumption, an intuition that will be validated in Section VII. Equally relevant, there can be situations where the data is both stationary and smooth. That is the case, for example, if the covariance matrix shares the eigenvectors with the graph Laplacian and its power spectral density is low pass. In such a setup, one could combine both network-recovery approaches, leading to a better recovery performance. This is precisely the subject of the ensuing section.

VI. Topology inference from stationary and smooth graph signals with hidden variables

In this section, we address Problem 1 by assuming that the graph signals $\mathbf{X}$ are both smooth and stationary on the unknown graph $\mathcal{G}$. These two assumptions can be jointly considered to design optimization problems with additional structure to enhance the estimation of $\mathbf{S}_{\mathcal{O}}$. To that end, we consider the smoothness-based inference problem described in (13) and incorporate the robust commutativity constraint accounting for stationarity [cf. (14)], resulting in

$$\begin{aligned}\min_{\tilde{\mathbf{L}}_{\mathcal{O}},\tilde{\mathbf{K}},r}\ & \operatorname{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}})+2\operatorname{tr}(\tilde{\mathbf{K}})+r+\alpha\|\tilde{\mathbf{L}}_{\mathcal{O}}\|_{F,off}^2 \\ & -\beta\log(\operatorname{diag}(\tilde{\mathbf{L}}_{\mathcal{O}}))+\gamma_*\|\tilde{\mathbf{K}}\|_*+\gamma_{2,1}\|\tilde{\mathbf{K}}\|_{2,1} \\ \text{s. t.}\ & \operatorname{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}})+2\operatorname{tr}(\tilde{\mathbf{K}})+r\geq 0, \\ & \tilde{\mathbf{L}}_{\mathcal{O}}\in\mathcal{L}, \\ & \|\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}+\tilde{\mathbf{K}}-\tilde{\mathbf{L}}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}}-\tilde{\mathbf{K}}^\top\|_F^2\leq\epsilon.\end{aligned} \tag{23}$$

Since the smooth formulation involves the Laplacian matrix, note that we adopted the Laplacian of the observed adjacency matrix as the GSO. Regarding the stationarity constraint, as discussed for (16), the value of ϵ should be selected based on the number of available signals M and the observation noise. It is also worth noting that the matrix $\tilde{\mathbf{K}}$ is inconspicuously absorbing the error derived from the presence of the hidden variables and from using $\tilde{\mathbf{L}}_{\mathcal{O}}$ instead of $\mathbf{L}_{\mathcal{O}}$ in both the smoothness penalty and the commutativity constraint. Regarding matrix $\tilde{\mathbf{K}}$, two different regularizers are considered: the nuclear norm (to promote solutions with a low rank) and the $\ell_{2,1}$ norm (to promote column sparsity). Since having solutions with columns that are zero also reduces the rank, it is prudent to tune the value of the hyperparameters γ_* and $\gamma_{2,1}$ jointly, so that the (joint) dependence between the rank and the column sparsity is kept under control.

We close the section by noting that the formulation in (23) is convex so that its globally optimal solution can be found efficiently. However, non-convex versions of (23) that leverage the re-weighted $\ell_{2,1}$ norm to promote column sparsity and factorization approaches for the low-rank penalty (similar to those used in Section V) could be developed here as well.

VII. Numerical experiments

This section runs numerical experiments to gain insights on the proposed schemes and evaluate their recovery performance. First, we test the smooth-based approaches with synthetic data and compare the results with existent algorithms from the literature. Secondly, we assess the performance of the stationary-based schemes proposed in Section V, comparing them with those in Section VI and the classical LVGL. Lastly, we apply the proposed algorithms to two real-world datasets and compare the obtained results with those of existing alternatives.[3]

A. Synthetic experiments based on smooth signals

We start by defining the default setup for the experiments in this section. With $\mathbf{L}=\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ denoting the eigendecomposition of the graph Laplacian, the smooth signals $\mathbf{X}$ are generated as $\mathbf{X}=\mathbf{V}\mathbf{Z}$, where the columns of $\mathbf{Z}\in\mathbb{R}^{N\times M}$ are independent realizations of a multivariate Gaussian distribution $\mathbf{Z}\sim\mathcal{N}(\mathbf{0},\,\mathbf{\Lambda}^\dagger)$. Note that this model, which is oftentimes referred to as factor analysis [23], [46], [47], assigns more energy to the low-frequency components, promoting smoothness on the generated graph signals. Unless otherwise stated, the number of signals is set to $M=100$ and the number of nodes to $N=20$. Moreover, to measure the recovery performance of the algorithms, in this section we focus on unweighted graphs and employ the F_{score}, which is defined as

$$F_{\text{score}}=2\cdot\frac{precision\cdot recall}{precision+recall}, \tag{24}$$

where $precision$ indicates the percentage of estimated edges that are edges of the ground-truth graph and $recall$ the percentage of existing edges that were correctly estimated.

Influence of hidden nodes. The results in Figure 1.a show the variation of the F_{score}, as the number of hidden variables H increases, for different recovery algorithms. Graphs are randomly generated using the model in [23], where nodes are placed in the unit square uniformly at random and edges are computed with a Gaussian radial basis function (RBF) as $A_{ij}=\exp(-d(i,j)^2/2\sigma^2)$, with $d(i,j)$ being the euclidean distance between two vertices and $\sigma=0.5$. Edges with weights smaller than 0.75 are removed and the surviving ones are set to 1. The hidden nodes are chosen uniformly at random among all the nodes in the graph. The algorithms considered in this experiment are the following: (i) GL-SigRep refers to the algorithm presented in [23]; (ii) GSm is a modified version of GL-SigRep that incorporates the logarithmic penalty and relies on the sample covariance matrix $\hat{\mathbf{C}}$ for the smoothness term in the objective function; (iii) GSm-LR represents the low-rank regularized algorithm proposed in (13), with $\gamma_{2,1}=0$; and (iv) GSm-GL denotes the algorithm described in (13), with $\gamma_*=0$, where column-sparsity is promoted in $\tilde{\mathbf{K}}$ via group Lasso. Comparing GL-SigRep with GSm allows us to quantify the improvement obtained exclusively from including hidden variables in the formulation, providing a fairer analysis of the proposed algorithms. The results in Figure 1.a indicate that, although the performance of all the algorithms deteriorates when the number of hidden variables increases, the algorithms GSm-LR and GSm-GL that account for the presence of hidden variables, outperform the alternatives. Moreover, their performance drops slowly as H increases, demonstrating the importance of taking into account the presence of hidden variables. The overall decay was expected since a higher number of hidden variables renders the topology inference problem more challenging and ill-posed, confirming the importance

[3]The MATLAB scripts for running all the numerical experiments presented in this section as well as additional related test cases can be found in https://github.com/andreibuciulea/topoIDhidden

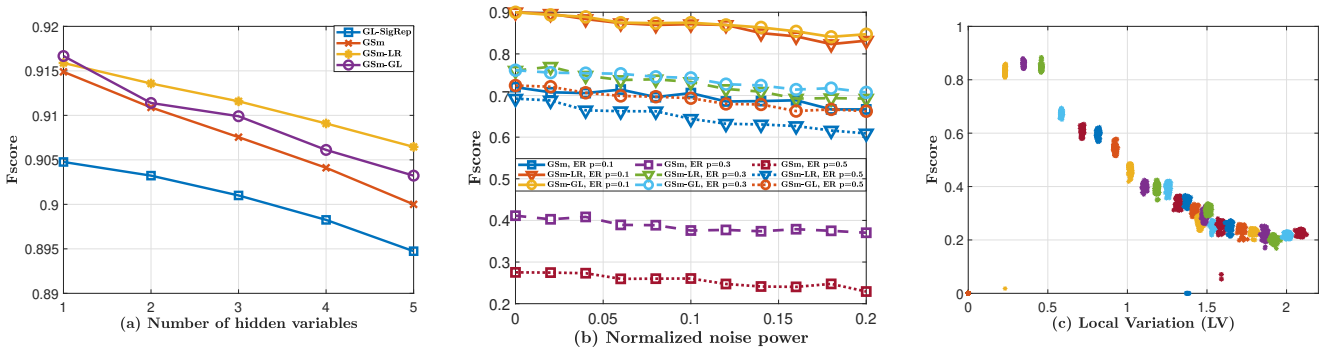


Fig. 1. Median F_{score} for the algorithms based on smooth graph signals with $N = 20$ and $M = 100$. The different panels assess the impact of varying (a) the number of hidden variables H for different algorithms when using RBF graphs, (b) the noise level present in the observations $\mathbf{X}$ when using Erdős Rényi graphs with different link probabilities $p = \{0.1, 0.3, 0.5\}$, and (c) the average level of LV of the observations $\mathbf{X}$ for a GSm-LR algorithm when using RBF graphs.

of (AS1). Comparing GSm-LR with GSm-GL, we observe that their performance is similar since the generated graphs are sufficiently sparse. It is also worth mentioning that the GSm scheme clearly outperforms GL-SigRep, illustrating the benefits of replacing the formulation introduced in [23] with the one presented in this paper, which relies on the matrix $\hat{\mathbf{C}}$ and the logarithmic barrier.

Noisy smooth observations. The second experiment assumes that the observations $\mathbf{X}_{\mathcal{O}}$ correspond to the ground-truth signals corrupted by additive white Gaussian noise (AWGN). For that setup, we evaluate the link-identification performance upon evaluating the F_{score} achieved by GSm-LR and GSm-GL schemes, comparing them with GSm, as the power of the AWGN increases, for graphs with different sparsity levels. In the experiments, we use Erdős Rényi (ER) graphs with edge probability values of $p = \{0.1, 0.3, 0.5\}$ and set the number of hidden variables to $H = 1$. The results, shown in Figure 1.b, reveal that the performance of the algorithms deteriorates not only when the noise increases but also for higher values of p. This behavior is consistent with the discussion provided in Section IV, since the formulation assumes that sparsity exists and, as a result, promotes solutions where several of the columns of $\tilde{\mathbf{K}}$ are zero. Furthermore, we observe that GSm-LR and GSm-GL have similar performance for lower values of p, but when the graphs become denser GSm-GL outperforms GSm-LR. This illustrates the fact that the low-rank regularization $\|\tilde{\mathbf{K}}\|_*$ is more sensitive to the sparsity of the graph than the group Lasso penalty $\|\tilde{\mathbf{K}}\|_{2,1}$. It is also worth noting that, even though the proposed schemes were not designed to specifically account for noisy observations, the rate at which F_{score} decays is smaller than the rate at which the noise power increases, showcasing the "natural" robustness to noise of the proposed schemes. Finally, note that GSm-LR and GSm-GL outperform GSm for the different values of p, which reinforces the importance of considering the presence of hidden variables.

Influence of the LV level. Next, we assess the relevance of the smoothness prior to the performance of the GSm-LR scheme. To that end, Figure 1.c depicts the F_{score} obtained with this scheme for different values of LV. Note that as we move to the right on the x-axis, the observed signals exhibit a larger variation (higher frequency) and, as a result, are less smooth. To control the LV level, the signals are generated combining K successive eigenvectors as $\mathbf{X} = \mathbf{V}_K\mathbf{Z}$, with $\mathbf{V} \in \mathbb{R}^{O\times K}$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^{K\times M}$. The smoothest signals are obtained by selecting the first K eigenvectors since they are associated with the low-frequency components. In contrast, activating the K last eigenvectors maximizes the local variation of the graph signals. For this experiment, we set $H = 1$, $K = 5$, and $N = 30$. The first generated signal is associated with eigenvectors $k = 1, ..., 5$, the second one with eigenvectors $k = 2, ..., 6$, and the last (26th) one with eigenvectors $k = 26, ..., 30$. The link-identification performance for those 26 types of signals are shown in Figure 1.c, where the vertical axis represents the F_{score} and the horizontal axis the average LV $\text{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})/M$. Each color represents a different set of active frequencies and, for each set, 128 realizations of $\mathbf{Z}$ have been generated (corresponding to the cloud of points shown in the figure). The results highlight the importance of the low values of LV when assuming smooth signals on the graph since the link identification performance decays noticeably as the signal becomes high-pass.

B. Synthetic experiments based on stationary signals

In these experiments, we focus on signals that are stationary on the sought GSO $\mathbf{S}$. To facilitate comparisons with GL, two different signal models are considered: (i) $\mathbf{C}_{poly}$ and (ii) $\mathbf{C}_{MRF}$. For the first one, the covariance of the observed signals is generated as a random polynomial of the GSO of the form $\mathbf{C}_{poly} = \mathbf{H}^2$ with $\mathbf{H} = \sum_{l=0}^{L} h_l\mathbf{S}^l$, where h_l are random coefficients following a normalized zero-mean Gaussian distribution. Note that this generative model guarantees that the covariance is PSD and a polynomial (of degree $2L$) of the GSO. In the second model, the covariance is generated as $\mathbf{C}_{MRF} = (\sigma\mathbf{I}+\delta\mathbf{S})^{-1}$, where σ is some positive number large enough to guarantee that $\mathbf{C}_{MRF}^{-1}$ is PSD and δ is some positive random number. As in the previous case, this generation guarantees the covariance matrix to be PSD and a polynomial of the GSO. Moreover, it also guarantees that the sparsity pattern of $\mathbf{C}_{MRF}^{-1}$ coincides with that of the GSO $\mathbf{S}$, which is the model assumed by GL. Regarding the metric used to evaluate the performance, rather than using

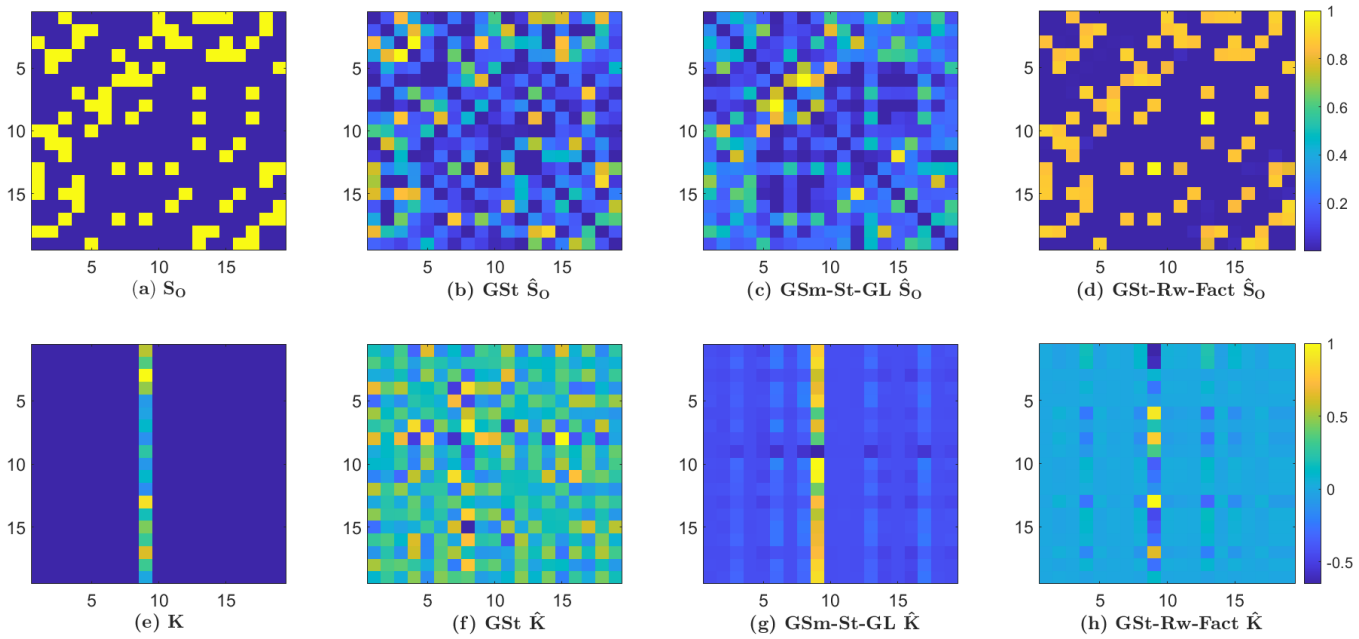


Fig. 2. Graphical representation of the estimates of matrices $\mathbf{S}_{\mathcal{O}}$ (top row) and $\mathbf{K} = \mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{O}\mathcal{H}}^{\top}$ (bottom row) for different algorithms that assume the observed signals to be stationary on the graph, with $N = 20$ and $H = 1$. The ground-truth matrices $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{K}$ are represented in the first column [cf. panels (a) and (e)]. Analogously, the estimates $\hat{\mathbf{S}}_{\mathcal{O}}$ and $\hat{\mathbf{K}}$ generated by GSt are represented in panels (b) and (f), those generated by GSm-St-GL in panels (c) and (g), and those generated by GSt-Rw-Fact in (d) and (h).

the F_{score}, we will generate multiple graphs and report the ratio of graphs that have been perfectly recovered (i.e., those graphs for which *all* the entries of the associated $\mathbf{S}_{\mathcal{O}}$ are estimated correctly). The reason for using this metric is that the incorporation of the stationary constraints boosts the ability of the algorithm to identify the topology, so that the value of F_{score} will be very close to one for all tested schemes, rendering the comparison more difficult. Differently, reporting the ratio of graphs perfectly recovered helps us to better assess the differences between the tested algorithms.

Leveraging the structure of $\mathbf{K}$. While the ultimate goal of this work is to recover $\mathbf{S}_{\mathcal{O}}$, the properties of matrix $\mathbf{K}$ played a key role in developing several of our topology-inference algorithms. For that reason, the goal of this experiment is to illustrate the recovered (estimated) $\hat{\mathbf{S}}_{\mathcal{O}}$ and $\hat{\mathbf{K}}$, so that we can gain insights on the effectiveness of the different approaches considered in the manuscript and their influence in recovering the graph. The results are shown in Figure 2, where the first row represents the GSOs and the second row the matrices $\mathbf{K}$. The first column corresponds to the ground-truth values, and the second, third and fourth columns present the estimates obtained with the low-rank scheme GSt [cf. (16)], the group Lasso scheme GSm-St-GL [cf. (23) with $\gamma_* = 0$], and the factorized scheme GSt-Rw-Fact [cf. (19)-(21)], respectively. First, focusing on $\hat{\mathbf{K}}$, it is apparent that for the depicted example the low-rank scheme GSt is not capable of recovering the column-sparse structure of the original matrix $\mathbf{K}$. Differently, when using either the group Lasso regularization (Figure 2.g) or the factorized approach (Figure 2.h), the estimated $\hat{\mathbf{K}}$ exhibits a row-sparsity pattern that is close to that of the ground truth. More importantly, when looking at the estimated $\hat{\mathbf{S}}_{\mathcal{O}}$ we observe that, as desired, the more accurate estimation of $\mathbf{K}$ translates into a superior estimation of the network topology, with GSm-St-GL yielding better estimates than GSt and GSt-Rw-Fact outperforming GSm-St-GL due to the replacement of the ℓ_1 norm with the linearized version of the logarithmic penalty. Overall, we believe that this simple experiment provides further intuition and strengthens the discussion about the different regularizers presented in Sections IV and V. The next step is to test the stationary-based schemes in a more systematic way, which is the goal of the following subsections.

Number of hidden variables. This experiment investigates the effect of the hidden nodes on the ability of our algorithms to recover the true graph topology. To that end, we vary the number of hidden variables H. We consider both the $\mathbf{C}_{poly}$ and $\mathbf{C}_{MRF}$ models for the observations, assume that the covariance matrices can be perfectly estimated, and select the set of hidden nodes as those with the minimum degree. The results are shown in Figure 3, where the x-axis represents the number of hidden variables and the y-axis the proportion of graphs successfully recovered. The results in Figure 3.a confirm that larger values of H render the inference problem more challenging, leading to a worse ratio of recovered graphs. We also observe that for the $\mathbf{C}_{MRF}$ model, LVGL achieves the best performance, especially when H increases. This is not surprising since the LVGL is tailored for this specific type of signal generation. On the other hand, LVGL fails to recover any graph when the observed signals follow the more general $\mathbf{C}_{poly}$ model. This contrasts with the GSt and GSt-Rw-Fact methods proposed in this paper, which recover the graphs in both settings. For both $\mathbf{C}_{MRF}$ and $\mathbf{C}_{poly}$ models, the proposed algorithms outperform GSt-Rw-nh, which solves the same problem as GSt-Rw-Fact but ignores the presence of hidden variables. This behavior of GSt-Rw-nh was expected since, as the number of hidden variables increases, their influence is

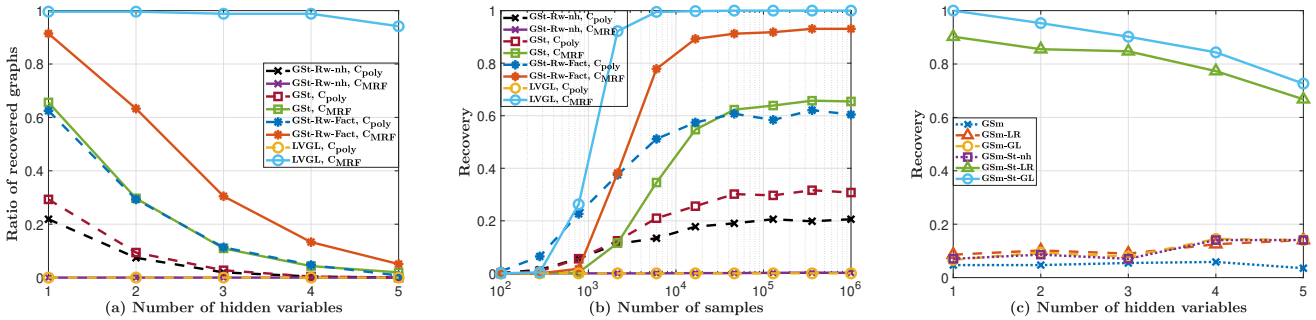


Fig. 3. The ratio of recovered graphs averaged over 200 realizations of random graphs with $N = 20$ and stationary observations. The different panels assess the impact of increasing (a) the number of hidden variables H for a scenario with perfectly-known covariance matrices; (b) the number of signal observations M when using the sample covariance matrix; (c) the number of hidden variables H when the inputs are not only stationary but also smooth signals.

more significant and the stationarity constraint becomes less accurate. It is also worth noting that the results obtained in Figure 3.a outperform those presented in Figure 1.a. This is due to the fact that graph-stationarity imposes more structure on the observed signals than graph-smoothness, at the expense of needing more observations to accurately estimate the covariance matrices.

Sample covariance matrix. The next step is to assess the effect of replacing the true covariance matrix with its sampled estimate $\hat{\mathbf{C}}_{\mathcal{O}} = \frac{1}{M}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^{\top}$. The number of hidden variables is set to $H = 1$, both $\mathbf{C}_{MRF}$ and $\mathbf{C}_{MRF}$ generative models are tested, the signals are assumed to be Gaussian and zero mean, and all other parameters are set as in the default test-case scenario. Figure 3.b illustrates the ratio of recovered graphs as the number of samples M varies. Clearly, the larger the value of M the better the estimate of $\hat{\mathbf{C}}_{\mathcal{O}}$. Analyzing the results in Figure 3.b, we observe that, when using $\mathbf{C}_{MRF}$, LVGL obtains the best performance and needs the least number of samples to achieve its best ratio of recovered graphs. As noted in the previous experiment, we also observe that LVGL is incapable of recovering graphs when the observations are generated using the $\mathbf{C}_{poly}$ model. On the other hand, GSt and GSt-Rw-Fact achieve a good performance for both covariance models, even though they need a higher number of samples.

If we focus on the covariance model $\mathbf{C}_{MRF}$, GSt-Rw-Fact achieves a performance close to that of LVGL. This behavior is consistent with the one observed in scenarios where all nodes were observed, and latent variables did not exist [13]. Upon comparing the results achieved by GSt and GSt-Rw-Fact, the experiments reveal that GSt: i) needs a higher value of M than GSt-Rw-Fact to achieve the same performance; and ii) converges to a worse ratio of recovered graphs. To conclude, as mentioned in Figure 3.a, ignoring the presence of hidden variables fails to capture the true structure of the inference problem, impacting the recovery ratio of GSt-Rw-nh for both covariance models. This is consistent with the results shown in previous experiments and, once again, illustrates the benefits of incorporating additional structure and using more sophisticated regularizers.

Leveraging graph stationarity and smoothness. To close the experiments based on synthetic data, we consider here the case where the observed signals are simultaneously smooth and stationary on the unknown graph and evaluate the schemes proposed in Section VI. As done in the smooth-based experiments, we create the graph signals as $\mathbf{X} = \mathbf{V}\mathbf{Z}$, with $\mathbf{Z}$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{\dagger})$. We note that the covariance of $\mathbf{X}$ is given by $\mathbf{C} = (\mathbf{L}^{\dagger})^2$, which is certainly a polynomial of the GSO provided that we set $\mathbf{S} = \mathbf{L}$. In other words, while the signals generated in Section VII-A were already stationary on the graph, none of the algorithms leveraged that existing structure. Hence, the goal here is to assess the benefits of incorporating that underlying structure into the recovery algorithms. To that end, we compare the schemes GSm-LR and GSm-GL, which only assume that the signals are smooth on the graph, with GSm-St-LR, which corresponds to (23) with $\gamma_{2,1} = 0$, and GSm-St-GL, which corresponds to the (23) with $\gamma_* = 0$. Additionally, we compare the aforementioned algorithms with two schemes that do not consider the presence of hidden variables, GSm and GSm-St-nh, with the latter assuming that the signals are both smooth and stationary on the graph. Note that GSm-St-LR and GSm-St-GL are, respectively, versions of GSm-LR and GSm-GL that account for the stationarity of the signals. Figure 3.c shows the ratio of recovered graphs as the number of hidden variables increases for the different algorithms. The advantages of including the stationarity assumption are clear, since, even for $H = 3$, the stationary-aware algorithms are able to perfectly recover more than 60% of the generated graphs. In contrast, the algorithms that ignore stationarity and account only for smoothness recover correctly less than 20% of the graphs. As expected, including additional information about the observed signals endows the optimization problem with more structure and results in better estimates. Focusing on the importance of considering the presence of hidden variables, we observe that i) GSm-St-LR and GSm-St-GL outperform GSm-St-nh; and ii) GSm-LR and GSm-GL outperform GSm. As expected, not considering the presence of hidden variables leads to an inaccurate estimation of the GSO, decreasing the percentage of recovered graphs by GSm and GSm-St-nh. If, as in Section VII-A, the recovery performance is measured using the F_{score} associated with individual links, then the differences narrow, with GSm-LR and GSm-GL achieving a (median) F_{score} of around 0.95 and GSm-St-LR and GSm-St-GL a F_{score} that is basically 1.

C. Learning graph structure from real datasets

We close this section by evaluating our algorithms and comparing their recovery performance with existing alternatives in

the literature using two real-world datasets.

Learning meteorological graph from temperature data. We start by considering the average monthly temperature collected at 88 measuring stations in Switzerland during the period between 1981 and 2010 [48]. This leads to a set of signals $\mathbf{X} \in \mathbb{R}^{88\times 12}$, with 12 signals that represent the monthly average temperatures measured at the 88 weather stations. The goal of the experiment is to use these observations to infer a graph where stations with similar temperature patterns across the year are connected. While using the geographical graph based on physical distances between the stations can be a more natural (non-data-based) solution to the problem at hand, one must note that Switzerland is a steep terrain. As a result, two nearby stations do not necessarily record similar temperatures across the year, since, for instance, their difference in altitude is large. Motivated by this and, as also done in [23], we build the "ground-truth" graph upon considering the similarity between stations in terms of their altitude. More specifically, in this experiment, we consider that two stations are connected with a unitary weight if their altitude difference is smaller than 300 meters. As we want to infer the best-represented graph from the available smooth signals and also take into account the presence of hidden variables, we are going to assume that $\mathcal{O} = \{1, ..., 20\}$, so that only the 20 first stations are observed, with our goal being inferring the connections between those stations.

We leverage the schemes developed in Section IV (GSm-LR and GSm-GL) and Section VI (GSm-St-LR and GSm-St-GL) to learn the graph associated with the observed nodes from the temperature measurements. To facilitate comparisons, the evaluation metrics used here are the same as those in [23], namely F_{score}, $precision$, $recall$, and normalized mutual information (NMI); in addition, the GL-SigRep algorithm from [23] is used as a baseline. The results achieved by the optimal setting of the regularization constants for each of the algorithms are listed in Table I. The main observation is that the explicit consideration of hidden variables when inferring the graph structure leads to better performance. Furthermore, we also observe that GSm-LR outperforms both GL-Sig-Rep and GSm-GL. It is also worth noticing that GSm-St-LR and GSm-St-GL obtain the same performance as GSm-LR, revealing that assuming stationarity for this dataset does not seem to further enhance the recovery results. Although this contrasts with the results from the synthetic experiments, it is not surprising since the number of available samples ($M = 12$) is smaller than the number of nodes, which leads to a rank-deficient $\hat{\mathbf{C}}_{\mathcal{O}}$ and renders the commutativity constrain inefficient. Indeed, the fact of the covariance being rank-deficient was the reason for not testing the algorithms developed in Section VI in this experiment.

Learning structural properties of proteins. In this case, our goal is to identify the structural properties of proteins from a mutual information graph of the co-variation of amino-acid residues simulating the presence of hidden variables. We have access to the mutual information matrix of protein BPT1 BOVIN and also to the binary ground-truth contact network built by medical experts, see [49] and [50] for details. The original dimension of both matrices is 53×53, but in our hidden-variable setup, we consider that we can only observe a submatrix of size 41×41 and leave the other 12 nodes as hidden. The y-axis in Figure 4 represents the fraction of the real contact edges recovered for several schemes and the x-axis represents the number of top-edge predictions. This way, a fraction of recovered edges of 0.6 indicates that if we consider the estimated 100 links with the highest weight, 60 of them match the ground-truth links. Five different algorithms are considered: GSt-Rw-Fact (Section VI); GSt no hidden (which approaches the topology-identification problem with stationarity assumptions but ignoring the presence of hidden variables [31]); LVGL; network deconvolution [49]; and mutual information, with the last two being baselines that have been advocated for this particular dataset. The best performance is achieved by the scheme GSt-Rw-Fact that is accounting for the presence of hidden variables, showcasing the benefits of a more robust formulation. Interestingly, we also observe that even though LVGL accounts for hidden variables, it leads to the worst recovery performance, illustrating the relevance of using topology-inference algorithms that go beyond classical graphical models when dealing with real datasets.

TABLE I
PERFORMANCE ACHIEVED BY THE SCHEMES GL-SIGREP ([23]), GSM-LR (SECTION IV), GSM-GL (SECTION IV), GSM-ST-LR (SECTION VI) AND GSM-ST-GL (SECTION VI) WHEN LEARNING A METEOROLOGICAL GRAPH.

Algorithms	F_{score}	Precision	Recall	NMI
GL-SigRep	0.8800	0.9016	0.8594	0.5746
GSm-GL	0.9118	0.8611	0.9688	0.6647
GSm-LR	0.9130	0.8514	0.9844	0.6806
GSm-St-LR	0.9130	0.8514	0.9844	0.6806
GSm-St-GL	0.9130	0.8514	0.9844	0.6806

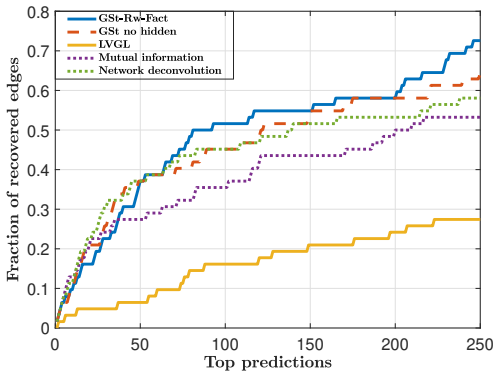
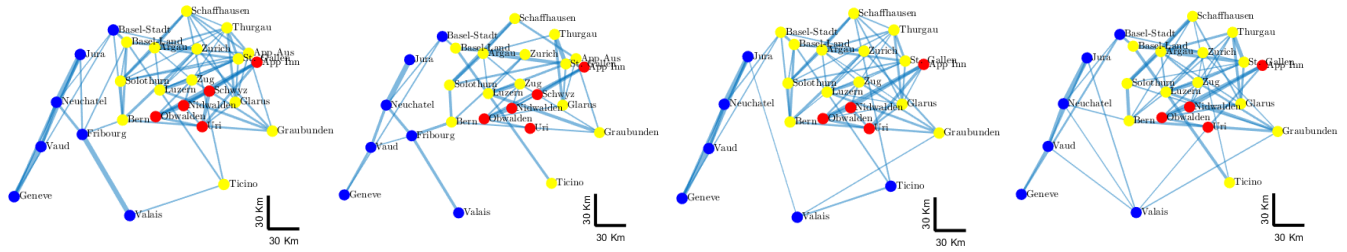


Fig. 4. Fraction of the real contact edges between amino-acids [50] recovered for each method as a function of the number of edges considered.

Learning graph from voting data. In this final real-data experiment, our goal is to learn a political graph from voting data [51]. More specifically, the 26 cantons of Switzerland are considered as nodes and the percentage of votes of each canton for 37 related initiatives (submitted to the voters between 2008 and 2012) are considered as graph signals. To validate the estimated graphs, we require a ground truth that reflects the level of association between the political preferences of the cantons. Since defining such a ground-truth graph is not evident, our first experiment is to compare the graph estimated by GSm-St-LR with the one estimated by GL-SigRep, with the latter being equivalent to the solution implemented in [23]. Once the two graphs are estimated, we apply spectral



(a) (b) (c) (d)

Fig. 5. Political associations among the 26 cantons of Switzerland estimated from electoral data. The colors blue, yellow, and red denote the 3 main clusters identified using spectral clustering and represent if a canton is against, supports, or strongly supports the initiatives, respectively. The two left-most graphs correspond to the political association networks estimated by (a) GL-SigRep and (b) GSm-St-LR when the voting data of all 26 cantons is considered. While the graphs are slightly different, the way in which cantons are clustered is the same. The two right-most graphs, which have 23 nodes each, represent the association networks estimated by (c) GL-SigRep and (d) GSm-St-LR when the voting data of 3 cantons (one belonging to each of the clusters) is removed. We observe that, in this case, the clusters in (c) and (d) are not the same and that (c) is less robust to the presence of hidden data.

clustering to obtain 3 clusters that group the 26 cantons according to their voting patterns.

Figures 5.a and 5.b show the graphs estimated by GL-SigRep and GSm-St-LR respectively, along with the 3 clusters. To identify the cluster each node belongs to, we used 3 colors (blue, red, and yellow). Figures 5.a and 5.b reveal that, while GSm-St-LR estimates a sparser graph (in general less edges for each node) than GL-SigRep, the 3-cluster partition is the same for both graphs. Equally important, the clusters convey meaningful information about the electoral preferences of the cantons, implicitly validating the obtained graphs. More specifically: i) the cantons that are against the initiatives correspond to the blue cluster; ii) the cantons that strongly support the initiatives correspond to the red cluster; and iii) the cantons that moderately support the initiatives correspond to the yellow cluster.

The next step is to assess the influence of (and robustness to) hidden variables. To that end, we randomly remove the voting data associated with one canton from each cluster and re-estimate the two graphs. The estimation results for GL-SigRep and GSm-St-LR in the presence of 3 hidden variables are reported in Figures 5.c and 5.d, respectively, with the particular realization shown corresponding to the removal of Fribourg, Appenzell Ausserrhoden, and Schwyz. Focusing first on the GSm-St-LR algorithm, the comparison of Figures 5.b and 5.d reveals that, while the graphs change slightly (new weak links appear in Figure 5.d to account for 2-hop relations that were broken after dropping the hidden nodes), the assignment of cantons to clusters does not change. On the other hand, for the GL-SigRep-based graphs (Figures 5.a and 5.c), we observe that while the links barely change, two of the nodes (Basel-Stadt and Ticino) are assigned to a different cluster.

In other words, the results of Figures 5.c and 5.d confirm that GSm-St-LR is more robust than GL-SigRep to the presence of hidden variables since it is able to maintain the same clustering pattern even when hidden nodes are present.[4]

VIII. Conclusions

This paper analyzed the problem of inferring the topology of a network from nodal signal observations in the presence of hidden (latent) nodes. To approach this ill-conditioned network-inference task, we considered that the observed signals were (i) smooth on the sought graph; (ii) stationary on the graph; and (iii) a combination of the two previous assumptions. To render the problem tractable, we further assumed that the number of hidden variables was much smaller than the number of observed nodes and formulated constrained optimization problems that accounted for the topological and signal constraints. The key to handle the presence of hidden nodes was to consider block-matrix factorization approaches that led to sparse and low-rank constrained optimizations. Since several of the resulting formulations were non-convex, novel judicious convex relaxations were designed. The performance of the developed algorithms was evaluated in several synthetic and real-world datasets and the results were compared with alternatives from the literature.

[4]For conciseness, only one experiment with three missing nodes is presented, but the difference in terms of robustness is also observed if the hidden nodes at hand change or if the value of H is either 1 or 2. We refer interested readers to the repository provided in footnote 3, which allows them to run the experiments for any desired configuration.

Appendix A
Proof of Proposition 1

Key to our proof are the results from [44], which guarantee convergence of BSUM algorithms to a stationary point.

We aim to show that our proposed algorithm satisfies the conditions specified in [44, Th. 1b]. To that end, let $f(\mathbf{y})$ represent the objective function in (18), with $\mathbf{y} := [\mathbf{y}_1^\top, \mathbf{y}_2^\top, \mathbf{y}_3^\top]^\top$ and $\mathbf{y}_1 := \text{vec}(\mathbf{S}_{\mathcal{O}})$, $\mathbf{y}_2 := \text{vec}(\mathbf{S}_{\mathcal{OH}})$, $\mathbf{y}_3 := \text{vec}(\mathbf{C}_{\mathcal{OH}})$ denoting the 3 blocks of variables considered in our algorithm. For each of the $B = 3$ block of variables $\mathbf{y}_b$, we approximate $f(\mathbf{y})$ by defining the functions $u_1(\mathbf{y}_1)$, $u_2(\mathbf{y}_2)$, and $u_3(\mathbf{y}_3)$, corresponding to the objective functions in (19), (20) and (21), respectively. Also, recall that $\mathcal{Y}^*$ denotes the set of stationary points of $f(\mathbf{y})$ and that $\mathbf{y}^{(t)} := [(\mathbf{y}_1^{(t)})^\top, (\mathbf{y}_2^{(t)})^\top, (\mathbf{y}_3^{(t)})^\top]^\top$ is the solution obtained after running t iterations of our algorithm.

With the previous definitions in place, the assumptions required to ensure converge of our algorithm are the following.
(AS A) The approximation functions $u_b(\mathbf{y}_b)$ must be a global upper bound of $f(\mathbf{y})$ and the first order behavior of $u_b(\mathbf{y}_b)$ and $f(\mathbf{y})$ must be the same.
(AS B) *The function $f(\mathbf{y})$ must be regular (cf. [44]) at every point in $\mathcal{Y}^*$.*
(AS C) *The level set $\mathcal{Y}^{(0)} = \{\mathbf{y} \mid f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})\}$ is compact.*
(AS D) *The problems in* (19)-(21) *must have a unique solution for any point $\mathbf{y}^{(t)} \in \mathcal{Y}^*$ for at least two of the blocks.*

We address each of the four assumptions separately, proving that our approach satisfies all of them.

Assumption (***AS A***) requires the surrogate functions $u_b(\mathbf{y}_b)$ to be global upper bounds of $f(\mathbf{y})$. For the first block ($b = 1$), we approximate $f(\mathbf{y})$ with the Taylor series of order 1 of the logarithmic penalty, given by

$$\tilde{u}_1(\mathbf{y}_1) = \sum_{i=1}^{O^2} \log\left(|[\mathbf{y}_1^{(t)}]_i| + \delta\right) + \sum_{i=1}^{O^2} \frac{\text{sign}([\mathbf{y}_1^{(t)}]_i)}{|[\mathbf{y}_1^{(t)}]_i| + \delta}\left([\mathbf{y}_1]_i - [\mathbf{y}_1^{(t)}]_i\right) + \rho f_c(\mathbf{y}_1), \tag{25}$$

where f_c denotes the commutativity penalty in (19). Since the entries of $\mathbf{y}_1^{(t)}$ are always either positive or negative [cf. (7) and (8)], we have that $\text{sign}([\mathbf{y}_1^{(t)}]_i)[\mathbf{y}_1]_i = |[\mathbf{y}_1]_i|$. After dropping the constant terms, we obtain

$$u_1(\mathbf{y}_1) = \sum_{i=1}^{O^2} \frac{|[\mathbf{y}_1]_i|}{|[\mathbf{y}_1^{(t)}]_i| + \delta} + \rho f_c(\mathbf{y}_1), \tag{26}$$

which is the objective function in (19). Because the log is a concave differentiable function it follows that its Taylor series of order one constitutes a global upper bound. Therefore, u_1 satisfies (***AS A***). The proof for u_2 is equivalent to the proof for u_1 so it is omitted for brevity. Lastly, $u_3(\mathbf{y}_3) = f(\mathbf{y})$ when the blocks $\mathbf{y}_1$ and $\mathbf{y}_2$ remain constant, so it also satisfies the requirements, and hence, (***AS A***) is fulfilled.

To proof (***AS B***), according to the definition of regular functions presented in [44], it suffices to show that the non-smooth parts of $f(\mathbf{y})$ are separable across the different blocks of variables. To that end, we recall that $\mathbf{y}_1 := \text{vec}(\mathbf{S}_\mathcal{O})$, $\mathbf{y}_2 := \text{vec}(\mathbf{S}_{\mathcal{O}\mathcal{H}})$ and $\mathbf{y}_3 := \text{vec}(\mathbf{C}_{\mathcal{O}\mathcal{H}})$, and decompose f as $f = g_A + g_B + g_C$, with functions g_A, g_B and g_C being defined as

- $g_A(\mathbf{S}_\mathcal{O}, \mathbf{S}_{\mathcal{O}\mathcal{H}}, \mathbf{C}_{\mathcal{O}\mathcal{H}}) = \eta\|\mathbf{S}_{\mathcal{O}\mathcal{H}}\|_F^2 + \eta\|\mathbf{C}_{\mathcal{O}\mathcal{H}}\|_F^2 + \rho\|\hat{\mathbf{C}}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{O}\mathcal{H}}^\top - \mathbf{S}_\mathcal{O}\hat{\mathbf{C}}_\mathcal{O} - \mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{C}_{\mathcal{O}\mathcal{H}}^\top\|_F^2$, where g_A is a smooth function,
- $g_B(\mathbf{S}_\mathcal{O}) = \sum_{i,j=1}^{O} \log(|[\mathbf{S}_\mathcal{O}]_{ij}| + \delta)$, where g_B is a non-smooth function,
- $g_C(\mathbf{S}_{\mathcal{O}\mathcal{H}}) = \sum_{i,j=1}^{O,H} \log(|[\mathbf{S}_{\mathcal{O}\mathcal{H}}]_{ij}| + \delta)$, where g_C is a non-smooth function.

Since the non-smooth terms appear in $g_B(\mathbf{S}_\mathcal{O})$, which only involves variables of the first block $\mathbf{y}_1 = \text{vec}(\mathbf{S}_\mathcal{O})$, and $g_C(\mathbf{S}_{\mathcal{O}\mathcal{H}})$, which only involves variables of the second block $\mathbf{y}_2 = \text{vec}(\mathbf{S}_{\mathcal{O}\mathcal{H}})$, it follows that the function $f(\mathbf{y})$ is regular for all feasible points.

Next, we show that the level set $\mathcal{Y}^{(0)} = \{\mathbf{y} \mid f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})\}$ is compact as required by (***AS C***). First, note that the entries of $\mathbf{S}_\mathcal{O}$ and $\mathbf{S}_{\mathcal{O}\mathcal{H}}$ are continuous subsets of $\mathbb{R}$ (e.g., $[\mathbf{S}_\mathcal{O}]_{ij}, [\mathbf{S}_{\mathcal{O}\mathcal{H}}]_{ij} \in \mathbb{R}_+$ when $\mathcal{S} = \mathcal{A}$), and that $\mathbf{C}_{\mathcal{O}\mathcal{H}} \in \mathbb{R}^{O\times H}$, so $f(\mathbf{y})$ is continuous. Moreover, since we have that $f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})$, this implies that the continuous functions $\log(|[\mathbf{S}_\mathcal{O}]_{ij}| + \delta)$, $\log(|[\mathbf{S}_{\mathcal{O}\mathcal{H}}]_{ij}| + \delta)$, and $\|\mathbf{C}_{\mathcal{O}\mathcal{H}}\|_F^2$ are all bounded, rendering the domain of $f(\mathbf{y})$ bounded. Therefore, it follows that the level set $\mathcal{Y}^{(0)}$ is compact.

Finally, since the optimization problems in (20) and (21) are strictly convex, two of the three problems have unique solutions, satisfying (***AS D***) and concluding the proof.

REFERENCES

[1] A. Buciulea, S. Rey, C. Cabrera, and A. G Marques, "Network reconstruction from graph-stationary signals with hidden variables," in *Asilomar Conf. Signals, Syst., Comput.* IEEE, 2019, pp. 56–60.

[2] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, New York, NY, 2009.

[3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[4] O. Sporns, *Discovering the Human Connectome*, MIT Press, Boston, MA, 2012.

[5] 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.

[6] S. Rey, F. J. I. Garcia, 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.

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

[8] 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.

[9] P. Djuric and C. Richard, *Cooperative and Graph Signal Processing: Principles and Applications*, Academic Press, 2018.

[10] A. G. Marques, N. Kiyavash, J. M. F. Moura, D. Van De Ville, and R. Willett, "Graph signal processing: Foundations and emerging directions (editorial)," *IEEE Signal Process. Mag.*, vol. 37, Nov. 2020.

[11] V. Kalofolias, "How to learn a graph from smooth signals," in *Intl. Conf. Artif. Intel. Statist. (AISTATS)*. J Mach. Learn. Res., 2016, pp. 920–929.

[12] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 6350–6354, Mar. 2016.

[13] 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, Sep. 2017.

[14] S. Segarra, A. G. Marques, M. Goyal, and S. Rey, "Network topology inference from input-output diffusion pairs," in *IEEE Wrkshp. Statistical Signal Process. (SSP)*. IEEE, 2018, pp. 508–512.

[15] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[16] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on sparsifying transform learning," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1712–1727, 2019.

[17] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Statist.*, vol. 34, pp. 1436–1462, 2006.

[18] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[19] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graph," *Annu. Cognitive Sc. Conf.*, pp. 778 – 783, 2010.

[20] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Sparse structural equation modeling for inference of gene regulatory networks exploiting genetic perturbations," *PLoS, Comput. Biology*, vol. 9, no. 5, pp. 1–13, May 2013.

[21] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 563–575, Aug. 2014.

[22] J. Mei and J. M. F. Moura, "Signal processing on graphs: Estimating the structure of a graph," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 5495–5499, 2015.

[23] 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, Dec. 2016.

[24] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 6315–6319, Mar. 2016.

[25] Y. Shen, B. Baingana, and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[26] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent variable graphical model selection via convex optimization," *Annu. Allerton Conf. Commun., Control, Comput.*, vol. 40, no. 4, pp. 1935–1967, 2012.

[27] X. Yang, M. Sheng, Y. Yuan, and T. Q. S. Quek, "Network topology inference from heterogeneous incomplete graph signals," *IEEE Trans. Signal Process.*, vol. 69, pp. 314–327, 2020.

[28] A. Anandkumar, D. Hsu, A. Javanmard, and S. Kakade, "Learning linear bayesian networks with latent variables," in *Intl. Conf. Machine Learning (ICML)*, 2013, pp. 249–257.

[29] J. Mei and J. M. F. Moura, "Silvar: Single index latent variable models," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2790–2803, 2018.

[30] A. Chang, T. Yao, and G. I. Allen, "Graphical models and dynamic latent factors for modeling functional brain connectivity," *2019 IEEE Data Science Wrksp. (DSW)*, pp. 57–63, 2019.

[31] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[32] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017.

[33] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," *Intl. Conf. on Learning Representations (ICLR)*, 2018.

[34] X. Wang, C. Yao, H. Lei, and A. M. C. So, "An efficient alternating direction method for graph learning from smooth signals," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 5380–5384, 2021.

[35] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, vol. 13, no. 9, pp. 228, 2020.

[36] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *arXiv preprint arXiv:2010.08120*, 2020.

[37] B. Girault, "Stationary graph signals using an isometric graph translation," in *Eur. Signal Process. Conf. (EUSIPCO)*, Aug 2015, pp. 1516–1520.

[38] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[39] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graphical Statist.*, vol. 22, no. 2, pp. 231–245, 2013.

[40] S. Segarra, Y. Wang, C. Uhler, and A. G Marques, "Joint inference of networks from stationary graph signals," *Asilomar Conf. Signals, Syst., Comput.*, pp. 975–979, 2017.

[41] 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.

[42] 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.

[43] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Intl. Conf. Comp. Learning Theory*. Springer, 2005, pp. 545–560.

[44] 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, 2016.

[45] J. V. de M. Cardoso, J. Ying, and D. P. Palomar, "Algorithms for learning graphs in financial markets," *arXiv preprint arXiv:2012.15410*, 2020.

[46] A. Basilevsky, *Statistical factor analysis and related methods*, Wiley, 1994.

[47] D. J. Bartholomew, M. Knott, and I. Moustaki, *Latent variable models and factor analysis: A unified approach*, Wiley, 2011.

[48] "Meteoswiss, historic measured meteorological data," *[Online]. https://www.meteoswiss.admin.ch/home/measurement-and-forecasting-systems/datenmanagement/historic-measured-meteorological-data.html*.

[49] S. Feizi, D. Marbach, M. Medard, and M. Kellis, "Network deconvolution as a general method to distinguish direct dependencies in networks," *Nat Biotech*, vol. 31, no. 8, pp. 726–733, 2013.

[50] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, "Protein 3d structure computed from evolutionary sequence variation," *PLoS ONE*, vol. 6, no. 12, pp. 1–20, 2011.

[51] "Database on swiss popular votes," *[Online]. http://www.swissvotes.ch*.

Andrei Buciulea (Student Member, IEEE) received the Telecommunications Engineering degree and Master's degree in Telecommunications Engineering from King Juan Carlos University of Madrid, Spain, in 2017 and 2019, respectively. He is currently working towards the Ph. D. degree with the Department of Signal Theory and Communications, King Juan Carlos University.

His research interests include data science, network science, graph signal processing, network topology inference, and machine learning.

In 2020, he has been awarded with a predoctoral grant at King Juan Carlos University.

Samuel Rey (Student Member, IEEE) received the degree in Telecommunication Engineering in 2016 and the M.Sc. in Telecommunications Engineering in 2018, both with highest honors, from King Juan Carlos University (URJC), Madrid, Spain. He is currently working towards his Ph. D. thesis with the Department of Signal Theory and Communications of King Juan Carlos University.

His current research focuses are graph signal processing, graph neural networks, non-convex optimization, and data science over networks.

He received the "Best Young Investigator Award" across all M. Sc. students at URJC in 2018. He was awarded with the Spanish Federal FPU Scholarship for Ph. D. studies in 2018, and with the Mobility Grant for Ph. D. FPU students in 2021.

Antonio G. Marques (Senior Member, IEEE) received the Telecommunications Engineering degree and the Doctorate degree, both with highest honors, from Carlos III University of Madrid, Spain, in 2002 and 2007, respectively. In 2007, he became a faculty of the Department of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain, where he currently develops his research and teaching activities as a full professor. From 2005 to 2015, he held different visiting positions at the University of Minnesota, Minneapolis. In 2015, 2016 and 2017 he was a visitor scholar at the University of Pennsylvania, Philadelphia.

His current research focuses on signal processing, machine learning, data science and artificial intelligence over graphs, and nonlinear and stochastic optimization of wireless, power and transportation networks.

Dr. Marques has served the IEEE in a number of posts, including as an associate editor and the technical / general chair of different conferences, and, currently, he is a member of the IEEE Signal Process. Theory and Methods Tech. Comm. His work has been awarded in several journals, conferences and workshops, with recent ones including IEEE SSP 2016, IEEE SAM 2016, IEEE SPS IEEE Y.A. Best Paper Award 2020, and CIT 2021. He is the recipient of the "2020 EURASIP Early Career Award" and a member of IEEE, EURASIP and the ELLIS society.