

RESEARCH

Open Access

# Stationary time-vertex signal processing



Andreas Loukas<sup>1\*†</sup> and Nathanaël Perraudin<sup>2†</sup>

## Abstract

This paper considers regression tasks involving high-dimensional multivariate processes whose structure is dependent on some known graph topology. We put forth a new definition of time-vertex wide-sense stationarity, or *joint stationarity* for short, that goes beyond product graphs. Joint stationarity helps by reducing the estimation variance and recovery complexity. In particular, for any jointly stationary process (a) one reliably learns the covariance structure from as little as a single realization of the process and (b) solves MMSE recovery problems, such as interpolation and denoising, in computational time nearly linear on the number of edges and timesteps. Experiments with three datasets suggest that joint stationarity can yield accuracy improvements in the recovery of high-dimensional processes evolving over a graph, even when the latter is only approximately known, or the process is not strictly stationary.

**Keywords:** Stationarity, Multivariate time-vertex processes, Harmonic analysis, Graph signal processing, PSD estimation

## 1 Introduction

One of the main challenges when modeling multivariate processes is to decouple the estimation variance from the problem size. Consider an  $N$ -variate process unfolding over  $T$  timesteps. If only mild assumptions are made, then the number of realizations needed to reliably estimate the first two moments is up to a logarithmic factor proportional to  $O(NT)$ , i.e., the data size [1]. Assuming that the process is time wide-sense stationarity (TWSS) makes the length  $T$  of the process inconsequential. This is ideal for the univariate setting as it enables us to make relevant predictions even based on a single realization. If one additionally assumes that the signal autocorrelation is compactly supported, such that most data dependencies take place within a short time horizon, then the estimation variance can be reduced further by (roughly) splitting the observations into parts and considering each as an independent realization. This approach suffices when  $N$  is relatively small. For high-dimensional processes, however, one needs to incorporate additional assumptions to obtain meaningful predictions [2–5].

In this spirit, this paper focuses on high-dimensional processes that are supported on the vertex set and are

statistically dependent on the edge set of some known graph topology. Whether examining epidemic spreading [6], how traffic evolves in the roads of a city [7], or neuronal activation patterns present in the brain [8], many of the high-dimensional processes one encounters are inherently constrained by some underlying network. This realization has been the driving force behind recent efforts to re-invent classical models by taking into account the graph structure, with advances in many problems, such as denoising [9] and semi-supervised learning [10, 11], among others.

Yet, standard models for processes (evolving) on graphs often fail to produce useful results when applied to real datasets. One of the main reasons for this shortcoming is that they model only a limited set of spatiotemporal behaviors. The well-used graph Tikhonov and total variation priors, for instance, assume that the signal varies slowly or in a piece-wise constant manner over edges, without specifying any precise relations [12–14]. Similarly, assuming that the graph Laplacian encodes the conditional correlations of variables, as is done with Gaussian Markov random fields [15], becomes a rigid model when the graph is known [16]. To capture the behavior of complex networked systems, such as transportation and biological networks, it is crucial to train expressive models, being able to reproduce a wide range of graph and temporal behaviors.

\*Correspondence: [andreas.loukas@epfl.ch](mailto:andreas.loukas@epfl.ch)

<sup>†</sup>Andreas Loukas and Nathanaël Perraudin contributed equally to this work.

<sup>1</sup>Laboratoire de Traitement des Signaux 2, École Polytechnique Fédérale Lausanne, 1015 Lausanne, Switzerland

Full list of author information is available at the end of the article

## 1.1 Contributions

This paper considers the statistical modeling of processes evolving on graphs. In particular, we investigate the relationship between two different hypotheses: TWSS and VWSS [17–19], which are individually helpful in reducing the variance of covariance estimation for time series and graph signals, respectively. We propose a combined multivariate hypothesis that we refer to as time-vertex wide-sense stationarity, or *joint stationarity* for short. The necessary first step of our analysis consists of reformulating the standard properties of stationarity (such as the relation of the covariance matrix and power spectral density to an appropriate Fourier transform) from the lens of time-vertex analysis [20, 21]. This analysis is purposeful, yet not trivial as joint stationarity is more complicated than assuming stationarity on the product of two graphs<sup>1</sup>.

We use the hypothesis of joint stationarity to control variance and computational complexity in estimation and recovery tasks. Similar to [17], also here, one may reliably estimate the model parameters from few observations (e.g., see Fig. 4) and solve MMSE recovery problems in time linear on the number of edges and timesteps (e.g., see Fig. 3). Complimenting previous work, we also provide an analysis of the power spectral density (PSD) estimation, which brings insight into the inherent trade-off between bias and variance. In addition, we experimentally demonstrate that assuming joint stationarity aids in recovery even when only an approximation of the graph is known, or the process is only approximately jointly stationary. These experiments corroborate that the joint stationarity hypothesis is a useful assumption, particularly in situations when the problem features a large number of variables but only a limited number of observations.

To test the utility of joint stationarity, we apply our methods on three diverse datasets: (a) a meteorological dataset containing the hourly temperature of 32 weather stations over 1 month in Molene, France [18], (b) a traffic dataset depicting high-resolution daily vehicle flow of 4 weekdays in the highways of Sacramento, and (c) simulated SIRS-type epidemics over Europe. Our experiments confirm that for high-dimensional processes evolving over graphs, assuming joint stationarity yields an improvement in recovery performance as compared to time- or vertex-based stationarity methods, even when the graph is only approximately known and the data violate the strict conditions of our definition.

## 1.2 Related work

There exists an extensive literature on multivariate stationary processes, developing the original work of Wiener

et al. [22, 23]. The reader may find interesting Bloomfield's book [24] focusing on spectral relations. We focus on two main approaches that relate to our work, graphical models and signal processing on graphs.

### 1.2.1 Graphical models

In the context of graphical models, multivariate stationarity has been used jointly with a graph in the work of [25, 26]. Though relevant, we note that there is a key difference of these models with our approach: we assume that the graph is given, whereas in graphical models, the graph structure (or more precisely the precision matrix) is learned from the data. Knowing the graph allows us to search for more involved relations between the variables. As such, we are not restricted to the case that the conditional dependencies are given by the graph (and therefore that they are sparse), but allow non-adjacent variables to be conditionally dependent, modeling a broader set of behaviors. We also note that our approach is eventually more scalable. We refer to [16] for elements of connections between graphical models and graph signal processing.

### 1.2.2 Graph signal processing

The idea of studying the stationarity of a random vector w.r.t. a graph was first introduced in [18, 27] and then in [17, 19]. While these contributions have different starting points, they both roughly propose the same definition. Another more recent contribution relating to stationarity on graphs in the context of PSD estimation is [28]. Despite the relevance of these works, it is important to stress that the current paper is the first to consider a stationary hypothesis over graph signals varying in time. Moreover, the new results are non-trivial as they cannot be obtained by applying previous definitions on a product graph. In addition, some of the analysis presented here (particularly that of Section 4) is novel and can also be employed for the previously studied case of stationary graph signals. To make the connection with previous works transparent, in the following, every technical result (e.g., Lemma, Theorem, Proposition) that emerges as a generalization of [18, 19, 27] contains a reference in its heading pointing to the former claim.

The forecasting of time-evolving signals on graphs was also considered in [29–32]. Nonetheless, there are several differences with these works, with the most important being that we define joint stationarity and that we are not restricted to the causal case (where a process is reconstructed only from its past). Finally, it should be noted that some preliminary results of this work appeared in a conference paper [33]. This work extends the conference paper in many directions. We refine the definition of joint stationarity and explore how it relates to other well-known stationarity hypotheses. We propose a new PSD estimator

<sup>1</sup>As it will be discussed in Section 3.2, joint stationarity is strictly more general than vertex stationarity on the product of the two graphs (first proposed in [14] in the deterministic setting), as the latter can only model processes with specific PSD.

and provide a theoretical analysis of the bias and variance of the old and new PSD estimators. Additionally, we study the complexity of the proposed solution and evaluate its merit w.r.t. two new datasets.

## 2 Preliminaries

### 2.1 General notation

We use boldface symbols for matrices and vectors (e.g.,  $\mathbf{A}$  and  $\mathbf{a}$  respectively) and calligraphic symbols for sets (e.g.,  $\mathcal{V}$  and  $\mathcal{E}$ ). Symbol  $j$  denotes the imaginary unit,  $\mathbf{I}_N$  is the  $N \times N$  identity matrix, and  $\mathbf{1}_N$  is the all-ones vector of size  $N$ . We use brackets to index matrix elements and subscripts for matrix blocks: if  $\mathbf{A}$  is of size  $n_1 \times n_2$ , then  $\mathbf{A}[n_1, n_2]$  is the element at the  $n_1$ th row and  $n_2$ th column and  $\mathbf{A}_{n_1, n_2}$  is a (block) matrix. Vector  $\mathbf{a} = \text{vec}(\mathbf{A})$  (with-out subscript) is the vectorized representation of  $\mathbf{A}$  and  $\mathbf{a}_n$  is its  $n$ th column. Moreover,  $\mathbf{A}^\top$  is its transpose and  $\mathbf{A}^*$  is its transposed complex conjugate (meaning that  $(\mathbf{A}_T^*)^\top$  is the complex conjugate). If  $\mathbf{A}$  is  $N \times N$  Hermitian, its eigendecomposition is generically written as  $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^*$ , where  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N]$  is a matrix having eigenvectors as columns and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$  is the diagonal matrix of eigenvalues. Symbols  $h(\cdot)$ ,  $f(\cdot)$ , and  $g(\cdot)$  are reserved for scalar/matrix functions. A matrix function with a single argument takes as an input a symmetric matrix  $\mathbf{A}$  and outputs  $h(\mathbf{A}) = \mathbf{U}\text{diag}(h(\lambda_1), \dots, h(\lambda_N))\mathbf{U}^\top$ . The operator  $\otimes$  denotes the Kronecker product. The Kronecker sum  $\oplus$  can be defined in terms of the Kronecker product as  $\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_M + \mathbf{I}_N \otimes \mathbf{B}$ , where matrix  $\mathbf{B}$  has size  $M \times M$ .

### 2.2 Harmonic time-vertex analysis

We consider signals supported on the vertices  $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$  of a weighted undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W}_G)$ , with  $\mathcal{E}$  the set of edges of cardinality  $E = |\mathcal{E}|$  and  $\mathbf{W}_G$  the weighted adjacency matrix. Suppose that signal  $\mathbf{x}_t$  is sampled at  $T$  successive regular intervals of unit length. A real time-vertex signal  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$  is then the matrix having graph signal  $\mathbf{x}_t$  as its  $t$ th column.

The frequency representation of a time-vertex signal  $\mathbf{X}$  is given by the joint Fourier transform [14, 21] (or JFT for short)

$$\hat{\mathbf{X}} = \text{JFT}\{\mathbf{X}\} \triangleq \text{GFT}\{\text{DFT}\{\mathbf{X}\}\} = \mathbf{U}_G^* \mathbf{X} (\mathbf{U}_T^*)^\top, \quad (1)$$

with  $\mathbf{U}_G$  and  $\mathbf{U}_T$  being, respectively, the unitary graph Fourier transform (GFT) and discrete Fourier transform (DFT) matrices, whereas  $(\mathbf{U}_T^*)^\top$  is the complex conjugate of  $\mathbf{U}_T$ . In vector form, we have that  $\hat{\mathbf{x}} = \text{JFT}\{\mathbf{x}\} \triangleq \mathbf{U}_J^* \mathbf{x}$ , where  $\mathbf{U}_J = \mathbf{U}_T \otimes \mathbf{U}_G$ . As is often the case, we choose  $\mathbf{U}_G$  to be the eigenvector matrix of the combinatorial<sup>2</sup> graph

Laplacian matrix  $\mathbf{L}_G = \text{diag}(\mathbf{W}_G \mathbf{1}_N) - \mathbf{W}_G$ , where  $\mathbf{1}_N$  is the all-ones vector of size  $N$ , and  $\text{diag}(\mathbf{W}_G \mathbf{1}_N)$  is the diagonal degree matrix. Matrix  $\mathbf{U}_T$  is the eigenvector matrix of the Laplacian  $\mathbf{L}_T$  of a cyclic graph  $\mathcal{T}$ :

$$\mathbf{U}_T^*[\tau, t] = \frac{e^{-j\omega_\tau t}}{\sqrt{T}}, \quad (2)$$

$$\text{with } \omega_\tau = \frac{2\pi(\tau - 1)}{T} \quad \text{for } t, \tau = 1, 2, \dots, T.$$

With this in place,  $\hat{\mathbf{X}}[n, \tau]$  can be seen as the Fourier coefficient associated with the joint frequency  $[\lambda_n, \omega_\tau]$ , where  $\lambda_n$  denotes the  $n$ th graph eigenvalue and  $\omega_\tau$  the  $\tau$ th angular frequency.

The JFT maintains a close connection with the product graph  $\mathcal{J}$  [14, 21]. The latter is the graph whose adjacency matrix is  $\mathbf{W}_J = \mathbf{W}_T \oplus \mathbf{W}_G$  (this amounts to a Cartesian product between  $\mathcal{G}$  and the ring graph  $\mathcal{T}$ ). The connection is revealed if one realizes that the Laplacian  $\mathbf{L}_J = \mathbf{L}_T \oplus \mathbf{L}_G$  of  $\mathcal{J}$  carries the eigendecomposition  $\mathbf{L}_J = \mathbf{U}_J(\Lambda_T \oplus \Lambda_G)\mathbf{U}_J$ . It follows that computing the JFT (in vector form) is the same as computing the GFT of  $\mathbf{x}$  w.r.t. graph  $\mathcal{J}$ . The main issue with any<sup>3</sup> product graph interpretation is that it imposes a strict dependence between the eigenvalues of  $\mathbf{L}_G$  and  $\mathbf{L}_T$  (since the eigenvalues of  $\mathbf{L}_J$  are given by  $\Lambda_T \oplus \Lambda_G$ ). As we will see in the next paragraph, to attain full generality, one needs to abandon the product graph. For an in-depth discussion of JFT and its properties, we refer the reader to [34].

### 2.3 Joint time-vertex filtering

Filtering a time-vertex signal  $\mathbf{x}$  with a *joint filter*  $h(\mathbf{L}_G, \mathbf{L}_T)$  corresponds to element-wise multiplication in the joint frequency domain  $[\lambda, \omega]$  by a function  $h : [0, \lambda_{\max}] \times [-1, 1] \mapsto \mathbb{R}$  [21, 34–36]. When a joint filter  $h(\mathbf{L}_G, \mathbf{L}_T)$  is applied to  $\mathbf{x}$ , the output is

$$h(\mathbf{L}_G, \mathbf{L}_T) \mathbf{x} = \mathbf{U}_J h(\Lambda_G, \Omega) \mathbf{U}_J^* \mathbf{x}, \quad (3)$$

where  $\Lambda_G \in \mathbb{R}^{N \times N}$  and  $\Omega \in \mathbb{R}^{T \times T}$  are diagonal matrices with  $\Lambda_G[n, n] = \lambda_n$  and  $\Omega[\tau, \tau] = \omega_\tau$ , whereas  $h(\Lambda_G, \Omega)$  is a diagonal  $NT \times NT$  matrix defined as

$$h(\Lambda_G, \Omega) = \text{diag} \left( \text{vec} \left( \begin{bmatrix} h(\lambda_1, \omega_1) & \cdots & h(\lambda_1, \omega_T) \\ \vdots & \ddots & \vdots \\ h(\lambda_N, \omega_1) & \cdots & h(\lambda_N, \omega_T) \end{bmatrix} \right) \right)$$

and  $\text{diag}(\text{vec}(\mathbf{A}))$  creates a matrix with diagonal elements the vectorized form of  $\mathbf{A}$ . The bi-variate notation  $h(\cdot, \cdot)$  is meant to illustrate that joint filters operate *independently*

<sup>2</sup>Though we use the combinatorial Laplacian in our presentation, our results can be adapted to alternative positive semi-definite matrix definitions of a graph Laplacian, such as the normalized Laplacian.

<sup>3</sup>The same limitation holds for product graph constructions that do not rely on the Cartesian product [14].

on the two domains, something impossible<sup>4</sup> in the product graph framework [14, 21]. For convenience, we will often overload notation and write  $h(\theta_{n,\tau})$  to refer to the bivariate function  $h(\lambda_n, \omega_\tau)$ . Furthermore, we say that a joint filter is *separable*, if its joint frequency response  $h$  can be written as the product of a frequency response  $h_1$  defined solely in the vertex domain and one  $h_2$  in the time domain, i.e.,  $h(\theta) = h_1(\lambda) \cdot h_2(\omega)$ .

### 3 Joint time-vertex stationarity

Let  $\mathbf{X} \in \mathbb{R}^{N \times T}$  be a real discrete periodic multivariate stochastic process with a finite number of timesteps  $T$  that is indexed by the vertex  $v_i$  of graph  $\mathcal{G}$  and time  $t$ . We refer to such processes as time-vertex processes, or *joint processes* for short.

Our objective is to provide a definition of stationarity that captures statistical invariance of the first two moments of a joint process  $\mathbf{x} = \text{vec}(\mathbf{X}) \sim \mathcal{D}(\bar{\mathbf{x}}, \Sigma)$ , i.e., the mean  $\bar{\mathbf{x}} = \mathbf{E}[\mathbf{x}]$  and the covariance  $\Sigma = \mathbf{E}[\mathbf{x}\mathbf{x}^T] - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$ . Crucially, the definition should do so in a manner that is faithful to the graph and temporal structure.

#### 3.1 Definition

Typically, wide-sense stationarity is thought of as an invariance of the two first moments of a process w.r.t. translation. For the first moment, things are straightforward: stationarity implies a constant mean  $\mathbf{E}[\mathbf{x}] = c\mathbf{1}$ , independently of the domain of interest. The second moment, however, is more complicated as it depends on the exact form translation takes in the particular domain. Unfortunately, for graphs, translation is a non-trivial operation and three alternative translation operators exist: the generalized translation [37], the graph shift [13], and the isometric graph translation [27]. Due to this challenge, there are currently three alternative (though akin) definitions of stationarity appropriate for graphs [17–19].

The ambiguity associated with translation on graphs urges us to seek an alternative starting point for our definition. Fortunately, there exists an interpretation which holds promise: *up to its constant mean, a wide-sense stationary process corresponds to a white process filtered linearly on the underlying space*. This “filtering interpretation” of stationarity is well known classically<sup>5</sup> as well as in the graph setting [19] and is equivalent to asserting that the second moment can be expressed as  $\Sigma = h(\mathbf{L}_T)$ , where  $h(\mathbf{L}_T)$  is a linear filter. Thankfully, not only filtering is elegantly and uniquely defined for graphs [37], but

also stating that a process is graph wide-sense stationary if  $\mathbf{E}[\mathbf{x}] = c\mathbf{1}_N$  and  $\Sigma = h(\mathbf{L}_G)$ , is a graph filter, is generally consistent<sup>6</sup> with current definitions [17–19].

This motivates us to also express the definition of stationarity for joint processes in terms of joint filtering:

**Definition 1 (JWSS)** *A joint process  $\mathbf{x} = \text{vec}(\mathbf{X})$  is called jointly wide-sense stationary (JWSS), if and only if*

- (a) *The first moment of the process is constant*  
 $\mathbf{E}[\mathbf{x}] = c\mathbf{1}_{NT}$ .
- (b) *The covariance matrix of the process is a joint filter*  
 $\Sigma = h(\mathbf{L}_G, \mathbf{L}_T)$ , where  $h(\cdot, \cdot)$  is a non-negative real function referred to as joint power spectral density (JPSD).

Let us examine Definition 1 in detail.

*First moment condition.* As in the classical case, the first moment of a JWSS process has to be constant over the time and the vertex sets, i.e.,  $\bar{\mathbf{X}}[i, t] = c$  for every  $i = 1, 2, \dots, N$  and  $t = 1, 2, \dots, T$ . For alternative choices of the graph Laplacian with a null-space not spanned by the constant vector, the first moment condition should be modified to requiring that the expected value of a JWSS process is in the null space of the matrix  $\mathbf{L}_T \oplus \mathbf{L}_G$  (see Remark 2 [19] for a similar observation on stochastic graph signals).

*Second moment condition.* According to the definition, the covariance matrix of a JWSS process takes the form of a joint filter  $h(\mathbf{L}_G, \mathbf{L}_T)$ , and is therefore diagonalizable by the JFT matrix  $\mathbf{U}_J$ . It may also be interesting to notice that the matrix  $h(\mathbf{L}_G, \mathbf{L}_T)$  can be expressed as follows

$$\Sigma = h(\mathbf{L}_G, \mathbf{L}_T) = \begin{pmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & \cdots & \mathbf{H}_{1,T} \\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & & \mathbf{H}_{2,T} \\ \vdots & & \ddots & \vdots \\ \mathbf{H}_{T,1} & \mathbf{H}_{T,2} & \cdots & \mathbf{H}_{T,T} \end{pmatrix}, \quad (4)$$

where each block  $\mathbf{H}_{t_1, t_2}$  of  $\Sigma$  is an  $N \times N$  matrix defined as:

$$\mathbf{H}_{t_1, t_2} = \frac{1}{T} \sum_{\tau=1}^T h_{\omega_\tau}(\mathbf{L}_G) e^{j\omega_\tau(t_1 - t_2 + 1)} \quad (5)$$

and  $h_{\omega_\tau}(\mathbf{L}_G)$  is the graph filter with frequency response  $h_{\omega_\tau} = h(\lambda, \omega_\tau)$ . Being a covariance matrix,  $h(\mathbf{L}_G, \mathbf{L}_T)$  must necessarily be positive-semidefinite; thus,  $h(\cdot, \cdot)$  is real (the eigenvalues of every Hermitian matrix are real) and non-negative. Also, equivalently, every zero mean JWSS process  $\mathbf{x} = \text{vec}(\mathbf{X})$  can be generated by joint filtering  $\mathbf{x} = h(\mathbf{L}_G, \mathbf{L}_T)^{1/2} \epsilon$  a white process  $\epsilon$  with zero mean and identity covariance. The following proposition

<sup>4</sup>Defining joint filters in terms of a product graph would imply that there is a fixed relation between angular and graph frequencies (determined by the product graph construction). As a result, in the product graph, framework filters are univariate functions.

<sup>5</sup>As the correlation between two instants  $t_1$  and  $t_2$  depends only on the difference between these two instants  $\mathbf{E}[\mathbf{x}[t_1] \mathbf{x}[t_2]] - \mathbf{E}[\mathbf{x}[t_1]] \mathbf{E}[\mathbf{x}[t_2]] = \gamma[t_1 - t_2]$ , the covariance matrix has to be circulant, a property that is shared by linear filters.

<sup>6</sup>The only exception: for graphs with repeated eigenvalues, the conditions  $\mathbf{E}[\mathbf{x}] = c\mathbf{1}$  and  $\Sigma = h(\mathbf{L}_G)$  are sufficient but not necessary for the graph stationarity definition based on isometric graph translation [18].



exploits these facts to provide an interpretation of JWSS processes in the joint frequency domain.

**Proposition 1** (Generalizes Theorem 1 [17] and Proposition 1 [18, 19]) *A joint process  $\mathbf{X}$  over a connected graph  $\mathcal{G}$  is jointly wide-sense stationary (JWSS) if and only if:*

- (a) *The joint spectral modes are in expectation zero*  

$$\mathbf{E} \left[ \hat{\mathbf{X}}[n, \tau] \right] = 0 \quad \text{if } \lambda_n \neq 0 \text{ and } \omega_\tau \neq 0.$$
- (b) *The product graph spectral modes are uncorrelated*  

$$\mathbf{E} \left[ \hat{\mathbf{X}}[n_1, \tau_1] \hat{\mathbf{X}}[n_2, \tau_2] \right] = 0, \text{ whenever}$$

$$n_1 \neq n_2 \text{ or } \tau_1 \neq \tau_2.$$
- (c) *There exists a non-negative function  $h(\cdot, \cdot)$ , referred to as joint power spectral density (JPSD), such that*

$$\mathbf{E} \left[ \left| \hat{\mathbf{X}}[n, \tau] \right|^2 \right] - \left| \mathbf{E} \left[ \hat{\mathbf{X}}[n, \tau] \right] \right|^2 = h(\lambda_n, \omega_\tau),$$

for every  $n = 1, 2, \dots, N$  and  $\tau = 1, 2, \dots, T$ .

(For clarity, this and other proofs of the paper have been moved to the “Appendix”.)

We briefly present a few additional properties of JWSS processes that will be useful in the rest of the paper.

**Property 1** (Generalizes Example 1 [17–19]) *White centered i.i.d. noise  $\mathbf{w} \in \mathbb{R}^{NT} \sim \mathcal{D}(\mathbf{0}_{NT}, \mathbf{I}_{NT})$  is JWSS with constant JPSD for any graph.*

The proof follows easily by noting that the covariance of  $\mathbf{w}$  is diagonalized by the joint Fourier basis of any graph  $\Sigma_{\mathbf{w}} = \mathbf{I} = \mathbf{U}_j \mathbf{I} \mathbf{U}_j^*$ . This last equation tells us that the JPSD is constant, which implies that similar to the classical case, the energy of white noise is evenly spread across all joint frequencies.

A second interesting property of JWSS processes is that stationarity is preserved through a filtering operation.

**Property 2** (Generalizes Theorem 2 [17], Property 1 [19]) *When a joint filter  $f(\mathbf{L}_G, \mathbf{L}_T)$  is applied to a JWSS process  $\mathbf{X}$  with JPSD  $h$ , the result  $\mathbf{Y}$  remains JWSS with mean  $cf(0,0)\mathbf{1}_{NT}$ , where  $c$  is the mean of  $\mathbf{X}$ , and JPSD  $f^2(\lambda, \omega) h(\lambda, \omega)$ .*

Finally, we notice that for real processes  $\mathbf{X}$ , which are the focus of this paper, the function  $h$  forming the joint filter should be symmetric w.r.t.  $\omega$ , meaning that  $h(\lambda, \omega) = h(\lambda, -\omega)$ . This property can be easily derived from the definition of the Fourier transform.

### 3.2 Relations to classical definitions

We next provide an in-depth examination of the relations between joint wide-sense stationarity, time and vertex stationarity, as well as their multivariate equivalents. For clarity, we order the rows/columns of the covariance matrix

$\Sigma$  such that each  $\Sigma_{t_1, t_2}$  block of size  $N \times N$  measures the covariance between  $\mathbf{x}_{t_1}$  and  $\mathbf{x}_{t_2}$  (see (4)).

#### 3.2.1 Standard definitions

As we discuss below, known definitions of stationarity in time/vertex domains are particular cases of joint stationarity.

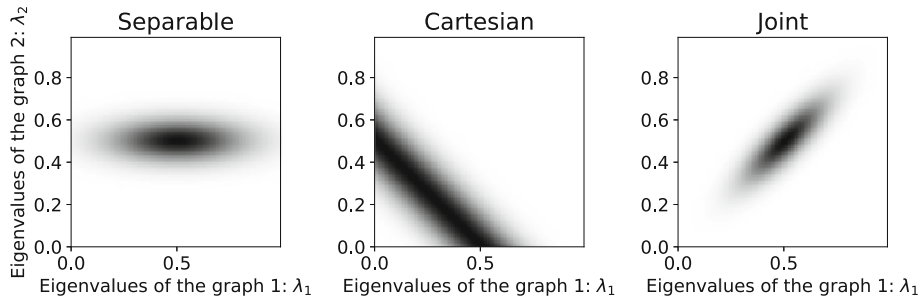
$TWSS \cap VWSS \subset JWSS$ . The known versions of stationarity (TWSS, VWSS) are oblivious to any structure along one of the two dimensions of  $\mathbf{X}$ . In this manner, assuming that  $\mathbf{X}$  is TWSS amounts to interpreting each of the  $N$  time series as a separate realization of the same process with TPSD  $h_T(\omega)$ . Similarly, if  $\mathbf{X}$  is VWSS, then each graph signal  $\mathbf{x}_t$  is taken as a separate realization of a single stochastic graph signal with VPSD  $h_G(\lambda)$  [17, 19]. It is a simple consequence that, different from the JWSS hypothesis, assuming that  $\mathbf{X}$  is both TWSS and VWSS is equivalent to limiting our scope to separable JPSD defined as the product of two univariate functions  $h(\lambda, \omega) = h_G(\lambda)h_T(\omega)$ —see also Fig. 1.

#### 3.2.2 Definitions based on the product graph

As explained in Section 2, the JFT can be interpreted as a graph Fourier transform taken over a product graph whose Laplacian is  $\mathbf{L}_J = \mathbf{L}_G \oplus \mathbf{L}_T$ . This construction can give rise to two additional definitions for joint stationarity:

**VWSS on a product graph.** The first is obtained by applying the VWSS definition of [17, 19] on the graph associated with  $\mathbf{L}_J$ . The resulting model is not sufficiently general in order to generate the full spectrum of JWSS processes. The reason is that, whereas the JPSD  $h(\lambda, \omega)$  can be any two-dimensional non-negative function, the JPSD of any VWSS process on  $\mathbf{L}_J$  is necessarily one-dimensional (the eigenvalues of  $\mathbf{L}_J$  are the sums of all combinations of the eigenvalues of  $\mathbf{L}_G$  and  $\mathbf{L}_T$ )—see Fig. 1 for a pictorial demonstration and “Appendix: Univariate vs multivariate JPSD” for examples from real data. The same reasoning also holds for alternative products between graphs, such as the strong and Kronecker products [14].

**Covariance diagonalized by the product graph Fourier transform.** The second definition, which we refer to as JWSS-alternate, entails asserting that the covariance matrix  $\Sigma$  can be diagonalized by the JFT, i.e., the eigenbasis of  $\mathbf{L}_J$ . This can be seen to differ from the JWSS definition only in case of graph Laplacian eigenvalue multiplicities: whenever the graph Laplacian features repeated eigenvalues, for Definition 1, the degrees of freedom of the JPSD  $h$  decrease, as necessarily  $h(\lambda_1, \omega) = h(\lambda_2, \omega)$  when  $\lambda_1 = \lambda_2$ . This restriction is motivated by the following observation: for an eigenspace with multiplicity



**Fig. 1** The joint stationarity hypothesis is more general than assuming either (standard) VWSS and TWSS or VWSS on a (Cartesian) product graph. The figure presents three examples of PSDs plotted as 2-dimensional function of  $\lambda_1, \lambda_2$  that, for simplicity, corresponds to the eigenvalues of two graphs. The second graph (time) is a ring. In the separable case (left), the PSD has to satisfy  $h(\lambda_1, \lambda_2) = h_1(\lambda_1)h_2(\lambda_2)$ , making it unable to capture any dependencies between  $\lambda_1$  and  $\lambda_2$ . Using VWSS (middle) limits the PSD to  $h(\lambda_1, \lambda_1) = h(\lambda_1 + \lambda_2)$  leading to constant values along the diagonal line  $\lambda_1 + \lambda_2 = c$ . Joint stationarity (right) can encode any PSD  $h(\lambda_1, \lambda_2)$ , as exemplified here

greater than one, there exists an infinite number of possible eigenvectors corresponding to the different rotations in the space, and the JPSD is in general ill-defined. The condition  $h(\lambda_1, \omega) = h(\lambda_2, \omega)$  when  $\lambda_1 = \lambda_2$  deals with this ambiguity, as it ensures that the JPSD is the same independently of the choice of eigenvectors. On the contrary, with JWSS-alternate, one should construct an arbitrary basis of each eigenspace with multiplicity and set<sup>7</sup>  $h(\lambda_1, \omega) \neq h(\lambda_2, \omega)$ . This approach, which was followed in [38], features more degrees of freedom at the expense of the loss of filtering interpretation and higher computational complexity: one may not anymore use filters to estimate the JPSD (without reverting to Definition 1), whereas using the JFT to diagonalize the covariance scales like  $O(N^3 + N^2T + NT \log(T))$ . On the contrary, in our setting, the PSD estimation complexity can be reduced to be close to linear in the number of edges  $E$  and timesteps  $T$  (see “Appendix: Implementation details of the JPSD estimator”).

Nevertheless, we should mention that the differences mentioned above are mostly academic. Eigenvalue multiplicities occur mainly when graph automorphisms exist. In the absence of such symmetries (e.g., in the graphs used in our experiments), the two definitions yield the same outcome.

### 3.2.3 Multivariate definitions

On the other hand, joint stationarity can itself be derived as the combination of two multivariate versions of time/vertex stationarity, which we refer to respectively as MTWSS (see [25]) and MVWSS. Before formally defining them in Definitions 2 and 3, let us state our result formally:

**Theorem 1** *A joint process  $\mathbf{X}$  is JWSS if and only if it is MTWSS and MVWSS.*

To put this in context, we examine the two multivariate definitions independently.

(a) *JWSS  $\subset$  MTWSS.* The covariance matrix of a JWSS process has a block circulant structure, as  $\Sigma_{t_1, t_2} = \Sigma_{\delta, 1} = \Gamma_{\delta}$ , where  $\delta = t_1 - t_2 + 1$ . Hence,  $\Sigma$  can be written as

$$\Sigma_{\mathbf{x}} = \begin{pmatrix} \Gamma_1 & \Gamma_2 & \cdots & \Gamma_T \\ \Gamma_T & \Gamma_1 & & \Gamma_{T-1} \\ \vdots & & \ddots & \vdots \\ \Gamma_2 & \Gamma_3 & \cdots & \Gamma_1 \end{pmatrix},$$

implying that correlations only depend on  $\delta$  and not on any time localization. This property is shared by multivariate time wide-sense stationary processes:

**Definition 2** (MTWSS [25]) *A joint process  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$  is multivariate time wide-sense stationary (MTWSS), if and only if the following two properties hold:*

- (a) *The expected value is constant as  $\mathbf{E}[\mathbf{x}_t] = c\mathbf{1}$  for all  $t$ .*
- (b) *For all  $t_1, t_2$ , the second moment satisfies  $\Sigma_{t_1, t_2} = \Sigma_{\delta, 1} = \Gamma_{\delta}$ , where  $\delta = t_1 - t_2 + 1$ .*

Similarly to the univariate case, the time power spectral density (TPSD) is defined to encode the statistics of the process in the spectral domain

$$\hat{\Gamma}_{\tau} = \sum_{\delta=1}^T \Gamma_{\delta} e^{-j\omega_{\tau} \delta}. \tag{6}$$

We then obtain the TPSD of a JWSS process by constructing a graph filter from  $h$  while fixing  $\omega$ . Setting  $h_{\omega_{\tau}}(\lambda) = h(\lambda, \omega_{\tau})$ , the TPSD of a JWSS process is  $\hat{\Gamma}_{\tau} = h_{\omega_{\tau}}(\mathbf{L}_G)$ .

<sup>7</sup>More generally, in an analogy to [18], the JPSD could be block diagonal with each block being of size equal to the multiplicity.

(b)  $JWSS \subset MVWSS$ . For a JWSS process, each block of  $\Sigma$  has to be a linear graph filter, i.e.,  $\Sigma_{t_1, t_2} = \gamma_{t_1, t_2}(\mathbf{L}_G)$ , meaning that

$$\Sigma = \begin{pmatrix} \gamma_{1,1}(\mathbf{L}_G) & \gamma_{1,2}(\mathbf{L}_G) & \cdots & \gamma_{1,T}(\mathbf{L}_G) \\ \gamma_{2,1}(\mathbf{L}_G) & \gamma_{2,2}(\mathbf{L}_G) & & \\ \vdots & & \ddots & \vdots \\ \gamma_{T,1}(\mathbf{L}_G) & & \cdots & \gamma_{T,T}(\mathbf{L}_G) \end{pmatrix}.$$

This is perhaps better understood when compared to the multivariate version of vertex stationarity defined below:

**Definition 3** (MVWSS) *A joint process  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$  is called multivariate vertex wide-sense stationary (MVWSS), if and only if the following two properties hold independently:*

- (a) *The expected value of each signal  $\mathbf{x}_t$  is constant  $\mathbf{E}[\mathbf{x}_t] = c_t \mathbf{1}$  for all  $t$ .*
- (b) *For all  $t_1$  and  $t_2$ , there exist a kernel  $\gamma_{t_1, t_2}$  such that  $\Sigma_{t_1, t_2} = \gamma_{t_1, t_2}(\mathbf{L}_G)$ .*

It can be seen that every JWSS process must also be MVWSS, or equivalently  $JWSS \subset MVWSS$ .

#### 4 Joint power spectral density estimation

The joint stationarity assumption can be useful in overcoming the challenges associated with dimensionality. The main reason is that for JWSS processes, the estimation variance is decoupled from the problem size. Concretely, suppose that we want to estimate the covariance matrix  $\Sigma$  of a joint process  $\mathbf{x} = \text{vec}(\mathbf{X})$  from  $K$  samples  $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(K)}$ . As we show in the following, if the process is JWSS such that  $\Sigma = h(\mathbf{L}_G, \mathbf{L}_T)$ , the JPSD estimation variance is  $O(1)$ . This is a sharp decrease from the classical and MTWSS settings, for which  $K \approx NT$  and  $K \approx N$  realizations are necessary<sup>8</sup>, respectively.

This section presents two JPSD estimators. The first provides unbiased estimates at a complexity that is  $O(N^3 T \log(T))$ . The second estimator decreases further the estimation variance at the cost of a bounded bias and is approximated with (close to) linear complexity.

##### 4.1 Sample JPSD estimator

We define the *sample JPSD estimator* for every graph frequency  $\lambda_n$  and angular frequency  $\omega_\tau$  as the estimate

$$\hat{h}(\lambda_n, \omega_\tau) \triangleq \sum_{k=1}^K \frac{|\text{JFT}\{\mathbf{X}_{(k)}\}[n, \tau]|^2}{K}. \quad (7)$$

In case the process does not have zero mean, it should be centered by subtracting the constant signal  $c \mathbf{1}_N \mathbf{1}_T^*$ , where

$c = \sum_{k,i,t} \mathbf{X}_{(k)}[i, t]/(KNT)$ . In that case, the unbiased estimator should involve division by  $K - 1$ , instead of  $K$  as we have in (7).

##### 4.1.1 Analysis

For simplicity, in the following, we suppose that the process is correctly centered. As the Theorem 2 claims, the sample JPSD estimator is unbiased, and its variance decreases linearly with the number of samples  $K$ .

**Theorem 2** *For every distribution with bounded second and fourth order moments, the sample JPSD estimator  $\hat{h}(\theta)$*

- (a) *is unbiased, i.e.,  $\mathbf{E}[\hat{h}(\theta)] = h(\theta)$ , and*
- (b) *has variance  $\mathbf{Var}[\hat{h}(\theta)] = h^2(\theta) \frac{\gamma - 1}{K}$ ,*

where constant  $\gamma$  depends only on the distribution of  $\mathbf{x}$ .

*Proof* For any  $\theta = [\lambda, \omega]$ , the sample estimate is

$$\hat{h}(\theta) = h(\theta) \sum_{k=1}^K \frac{\hat{\epsilon}_{(k)} \hat{\epsilon}_{(k)}^*}{K}, \quad (8)$$

with  $\hat{\epsilon}_{(k)}$  being independent realizations of  $\hat{\epsilon}$ , a zero mean complex random variable with unit variance. To see this, write  $\mathbf{x} = h(\mathbf{L}_G, \mathbf{L}_T)^{1/2} \epsilon$ , where the random vector  $\epsilon$  has zero mean and identity covariance. Then, the complex random variable  $\hat{\epsilon}$  is the JFT coefficient of  $\epsilon$  corresponding to frequencies  $\lambda$  and  $\omega$ . The bias follows by noting that  $\mathbf{E}[\hat{\epsilon}_{(k)} \hat{\epsilon}_{(k)}^*] = 1$ , for every  $k$ . The variance is computed similarly by exploiting the fact that different terms in the sum are independent as they correspond to distinct realizations and setting  $\gamma = \mathbf{E}[|\hat{\epsilon}|^4]$ .  $\square$

For the standard case of a Gaussian joint process, we provide an exact characterization of the distribution.

**Corollary 1** *For every Gaussian JWSS process, the sample JPSD estimate follows a Gamma distribution with shape  $K/2$  and scale  $2h(\theta)/K$ . The estimation error variance is equal to  $\mathbf{Var}[\hat{h}(\theta)] = 2h^2(\theta)/K$ .*

*Proof* We continue in the context of the proof of Theorem 2. For a Gaussian distribution,  $\hat{\epsilon}$  is centered and scaled Gaussian and thus  $\hat{\epsilon}^2$  is a chi-squared random variable with 1 degree of freedom. Our estimate is, therefore, a scaled sum of i.i.d. chi-squared variables and corresponds to a Gamma distribution. The corollary then follows directly.  $\square$

Observe that the variance depends linearly on the fourth-order moment of  $|\hat{\epsilon}|$  (see proof of Theorem 2) and is inversely proportional to the number of samples, but it

<sup>8</sup>The number of realizations needed for obtaining a good sample covariance matrix of an  $n$ -dimensional process is  $O(n \log n)$  [1, 39].

is independent of  $N$  and  $T$ . This implies that  $\|\Sigma - \hat{\Sigma}_2\|$  can be made arbitrarily small using  $K = O(1)$  samples. In the following, we discuss how to achieve an even smaller variance by exploiting the properties of  $h(\theta)$ .

### 4.2 Convolutional JPSD estimator

When the number of available realizations  $K$  is small (even 1), one may make use of additional assumptions on to obtain reasonable estimates. To this end, we next present a parametric JPSD estimator that allows us to trade off bias for variance.

Before delving into JWSS processes, it is helpful to consider the purely temporal case. For a TWSS process, it is customary to assume that the autocorrelation function has support  $L$  that is a few times smaller than  $T$ . Then, cutting the signal into  $\frac{T}{L}$  smaller parts and computing the average estimate reduces the variance (by a factor of  $\frac{T}{L}$ ), without sacrificing frequency resolution. This basic idea stems from two established methods used to estimate the PSD of a temporal signal, namely Bartlett's and Welch's method [40, 41]. Averaging across different windows is equivalent to smoothing the TSPD by convolving it with a window in the frequency domain: this results in attenuation of the correlation for long delays, enforcing localization in the time domain.

#### 4.2.1 Estimator

Armed with this interpretation, we proceed by smoothing the JPSD with a user-specified bi-variate window  $g$ , such as a Gaussian or a disc window. The convolutional JPSD estimator computes the JPSD at joint frequency  $\theta = (\lambda, \omega)$  as:

$$\ddot{h}(\theta) \triangleq \frac{1}{c_g(\theta)} \sum_{\substack{n=1 \\ \tau=1}}^{N,T} g(\theta - \theta_{n,\tau})^2 \sum_{k=1}^K \frac{|\text{JFTX}_{(k)}[n, \tau]|^2}{K}, \quad (9)$$

where  $c_g(\theta) \triangleq \sum_{n,\tau} g(\theta - \theta_{n,\tau})^2$  is a normalization factor. For implementation specifics, including a discussion on the choice of the bivariate kernel  $g$ , we refer the reader to "Appendix: Implementation details of the JPSD estimator".

The convolutional JPSD estimator is related to known PSD estimators for TWSS and VWSS processes. The Dirac function is denoted by  $\phi$ . We have that (a) for  $g(\theta) = \phi(\lambda) \cdot g_T(\omega)$ , we recover the classical TSPD estimator, applied independently for each  $\lambda$ . (b) For  $g(\theta) = g_G(\lambda) \cdot \phi(\omega)$ , we recover the VPSD estimator from [17] applied independently for each  $\omega$ . Similar to the latter, the estimator can be closely approximated at a complexity that is linear w.r.t. the number of graph edges/nodes, and up to a logarithmic factor linear to the number of timesteps (see "Appendix: Implementation details of the JPSD estimator").

#### 4.2.2 Analysis

To provide a meaningful bias analysis, we introduce a Lipschitz continuity assumption on the JPSD, matching the intuition that localized phenomena tend to have a smooth representation in the frequency domain.

**Theorem 3** *At  $\theta$ , the convolutional JPSD estimator  $\ddot{h}(\theta)$*

(a) *has bias*

$$\left| \mathbf{E} \left[ \ddot{h}(\theta) - h(\theta) \right] \right| \leq \frac{\epsilon}{c_g(\theta)} \sum_{n,\tau=1}^{T,N} g(\theta - \theta_{n,\tau})^2 \|\theta - \theta_{n,\tau}\|_2,$$

where  $\epsilon$  is the Lipschitz constant of  $h(\theta)$ , and

(b) *when the entries of  $\hat{\mathbf{X}}$  are independent random variables, its variance is*

$$\mathbf{Var} \left[ \ddot{h}(\theta) \right] = \sum_{n,\tau} \frac{g(\theta - \theta_{n,\tau})^4}{c_g(\theta)^2} \mathbf{Var} \left[ \dot{h}(\theta_{n,\tau}) \right],$$

where  $\mathbf{Var} \left[ \dot{h}(\theta_{n,\tau}) \right]$  is the variance of the sample JPSD estimator at  $\theta_{n,\tau}$ .

The derivations of the bias and variance are given in Lemmas 1 and 2, respectively.

We note two corner cases of interest. In the most convenient case, the JPSD is constant, and our estimator is unbiased (the Lipschitz constant  $\epsilon$  is zero). On the other hand, if the JPSD fluctuates rapidly, the bias of the estimate will be significant unless  $g$  is close to a Dirac. Here, the sample estimator should be preferred.

We further consider as a theoretical example the case of a Gaussian JWSS process and a (spectral) disc window with bandwidth  $B$ , i.e.,  $g_B(\theta) = 1$  if  $\|\theta\|_2 \leq \frac{B}{2}$  and 0 otherwise. Though perhaps not the most practical choice from a computational perspective, we consider here a disc window because it leads to simple and intuitive estimates.

**Corollary 2** *For every  $\epsilon$ -Lipschitz Gaussian JWSS process and disc window  $g_B(\theta)$ , the convolutional estimate has*

$$\left| \mathbf{E} \left[ \ddot{h}(\theta) - h(\theta) \right] \right| \leq \frac{\epsilon B}{2} \quad \text{and} \quad \mathbf{Var} \left[ \ddot{h}(\theta) \right] = \frac{2 h_S^2}{K |\mathcal{S}_\theta|}, \quad (10)$$

with set  $\mathcal{S}_\theta = \{\theta_{n,\tau} \mid \|\theta_{n,\tau} - \theta\|_2 \leq B/2\}$  and  $h_S^2 = \sum_{\theta_{n,\tau} \in \mathcal{S}} h(\theta_{n,\tau})^2$ .

*Proof* The results follow from Theorem 3 and Corollary 1 by noting that when a disc window is used, (a)  $c_g(\theta) = |\mathcal{S}_\theta|$  and (b)  $g(\theta - \theta_{n,\tau})^2 = 1$  for all  $n, \tau$  in the window (there are  $|\mathcal{S}_\theta|$  in total) and zero otherwise. The independence condition required by the variance clause of the theorem is satisfied since  $\hat{\mathbf{x}}$  is Gaussian (as a rotation  $\hat{\mathbf{x}} = \mathbf{U}_j^* \mathbf{x}$  of a Gaussian vector) with diagonal covariance.  $\square$



The above result suggests that by selecting our window (bandwidth), we can trade off bias for variance. The trade-off is particularly beneficial as long as (a) the JPSD is smooth relatively to the disc size ( $\epsilon B \ll 1$ ) and (b) the graph eigenvalues are clustered ( $|S_\theta| \gg 1$  when  $h(\theta) \gg 0$ ).

## 5 Recovery of JWSS Processes

This section considers the MMSE problem of recovering a JWSS process  $\mathbf{x} = \text{vec}(\mathbf{X})$  from linear measurements  $\mathbf{y}$  corrupted by a zero-mean JWSS process  $\mathbf{w}$ :

$$\begin{aligned} \min_{f: \mathbb{R}^{N'} \rightarrow \mathbb{R}^N} \quad & \mathbb{E} \|f(\mathbf{y}) - \mathbf{x}\|_2^2 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \end{aligned} \quad (\text{P0})$$

where the function  $f$  is linear on  $\mathbf{y}$ , i.e., there exists a matrix  $\mathbf{W}$  and a vector  $\mathbf{b}$  such that  $f(\mathbf{y}) = \mathbf{W}\mathbf{y} + \mathbf{b}$ . We remark that (a) for  $\mathbf{A}$  binary diagonal and  $\mathbf{w} = \mathbf{0}$ , (P0) is an *interpolation* problem, (b) for  $\mathbf{A} = \mathbf{I}$  and  $\mathbf{w}$  white noise (P0) is a *denoising* problem, and (c) for  $\mathbf{A}$  diagonal with  $A_{ii} = 1$  if  $i \leq Nt$  and zero otherwise and  $\mathbf{w} = \mathbf{0}$  it corresponds to *forecasting*. We mainly consider the former two problems since, for forecasting, it is more computationally efficient to utilize autoregressive models [29].

The *minimum mean-squared linear estimate* is known to be

$$\hat{\mathbf{x}} = \Sigma_{\mathbf{xy}} \Sigma_{\mathbf{y}}^{-1} (\mathbf{y} - \bar{\mathbf{y}}) + \bar{\mathbf{x}}, \quad (11)$$

with the definitions  $\Sigma_{\mathbf{y}} = \mathbf{A}\Sigma\mathbf{A}^* + \Sigma_{\mathbf{w}}$  and  $\Sigma_{\mathbf{xy}} = \Sigma\mathbf{A}^*$ . Obtaining  $\hat{\mathbf{x}}$  therefore entails solving a linear system in matrix  $\Sigma_{\mathbf{y}}$  that—naively approached—has  $O(N^2 T^2)$  complexity. In addition, the condition number of  $\Sigma_{\mathbf{y}}$  can be large, rendering direct inversion unstable. For instance, this may happen when one attempts to reverse any smoothing operation  $\mathbf{A}$  that severely attenuates part of the signal's spectrum.

We next discuss how to deal with these issues:

### 5.1 Decreasing the complexity

Thankfully, even if  $\Sigma_{\mathbf{y}}$  is not always sparse, we can approximate its multiplication by a vector without actually computing it as (a)  $\mathbf{A}$  is, for many applications (denoising, prediction, forecasting), sparse, and (b) per our assumption,  $\Sigma$  and  $\Sigma_{\mathbf{w}}$  are joint filters, and therefore, they can be implemented at complexity that is (up to logarithmic factors) linear to the number of edges  $E$  and timesteps  $T$  [20, 34, 36]. Therefore, if we employ an iterative method such as the (preconditioned) conjugate gradient to compute the solution, the complexity of each iteration will be linear on the problem size.

### 5.2 Singular or badly conditioned $\Sigma_{\mathbf{y}}$

We choose the solution with the minimal residual by substituting the inverse  $\Sigma_{\mathbf{y}}^{-1}$  in (11) with the pseudo-inverse  $\Sigma_{\mathbf{y}}^+$ . However, instead of solving the normal equations  $\hat{\mathbf{x}} =$

$\Sigma_{\mathbf{xy}} \left( \Sigma_{\mathbf{y}}^2 \right)^{-1} \Sigma_{\mathbf{y}} (\mathbf{y} - \bar{\mathbf{y}}) + \bar{\mathbf{x}}$ , which has the effect of significantly increasing the condition number of our matrix, we suggest to employ the minimal residual conjugate gradient method for symmetric matrices [42]. For badly conditioned covariance matrices, an alternative solution is to rewrite the problem as a regularized least squares problem

$$\min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 + \|h_{\mathbf{w}}(\mathbf{L}_G, \mathbf{L}_T)^{1/2} h_{\mathbf{x}}(\mathbf{L}_G, \mathbf{L}_T)^{-1/2} (\mathbf{z} - \bar{\mathbf{x}})\|_2^2 \quad (12)$$

and solve it using the generalization of the fast iterative shrinkage-thresholding algorithm (FISTA) scheme [43–45]. This problem was shown to converge to the correct solution when  $\mathbf{w}$  is white noise. More details about the optimization procedures can be found in [17]. Similarly, in the noiseless case one removes term  $\|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2$  in (12) and introduces instead the constraint  $\mathbf{A}\mathbf{z} = \mathbf{y}$ . The resulting optimization problem can be solved using a Douglas-Rachford scheme [46].

## 6 Experiments

### 6.1 Joint power spectral density estimation

The first step in our evaluation is to analyze the efficiency of JPSD estimation. Our objective is dual. First, we aim to study the role of the different method parameters into the estimation accuracy and computational complexity, essentially providing practical guidelines for their usage. In addition, we wish to illustrate the usefulness of the joint stationarity assumption, even when the graph is only approximately known.

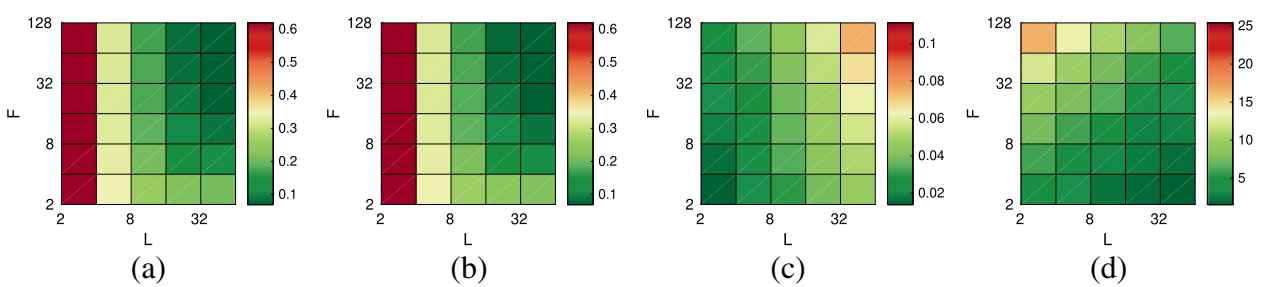
#### 6.1.1 Variance-bias-complexity tradeoffs

To validate the analysis of Section 4 for the computational and accuracy trade-offs inherent to our JPSD estimation method, we performed numerical experiments with random geometric graphs of  $N = 256$  vertices (we build a 10-nearest neighbor graph, weighted by a radial basis function kernel tuned so th at the average weighted degree is slightly above 7) and JWSS processes ( $T = 128$  timesteps). Though our approach works with any JPSD, including high frequency ones, in this experiment, we consider a stochastic process generated by the discrete damped wave equation with a non-separable JPSD  $h(\lambda, \omega) = \exp(-|\omega|/2) \cos(\omega \arccos(1 - \lambda))$

#### 6.1.2 Variance-bias

First, we examine the relation between the real JPSD  $h$  and the convolutional estimate  $\hat{h}$  obtained using the “fast” method described in “Appendix: Implementation details of the JPSD estimator”. We use the following metrics:

$$\left. \begin{array}{l} \text{error} \\ \frac{\tilde{\mathbb{E}}[\|\hat{\mathbf{H}} - \mathbf{H}\|_F]}{\|\mathbf{H}\|_F} \end{array} \right| \left. \begin{array}{l} \text{bias} \\ \frac{\|\tilde{\mathbb{E}}[\hat{\mathbf{H}} - \mathbf{H}]\|_F}{\|\mathbf{H}\|_F} \end{array} \right| \left. \begin{array}{l} \text{standard deviation} \\ \frac{\tilde{\mathbb{E}}[\|\hat{\mathbf{H}} - \tilde{\mathbb{E}}\hat{\mathbf{H}}\|_F]}{\|\mathbf{H}\|_F} \end{array} \right|,$$



**Fig. 2** Influence of the parameters (window size  $L$  and number of graph filters  $F$ ) on the **a** estimation error, **b** bias, **c** normalized standard deviation, and **d** execution time. For improved visibility, the scale of **c** has been changed

where  $\mathbf{H} = h(\Lambda_G, \Omega)$ ,  $\tilde{H} = \tilde{h}(\Lambda_G, \Omega)$ , and  $\tilde{\mathbf{E}}[\cdot]$  is the empirical expectation computed over 20 independent experiments. We remind the reader that there are two parameters influencing the performance of the convolutional JPSD estimator (see “Appendix: Implementation details of the JPSD estimator”: the window size  $L$  corresponding to our assumption for the support length of the autocorrelation in time, and the number of graph filters  $F$  used to capture power density in the graph spectral dimension. As discussed in Theorem 3, the bias will be small as long as the JPSD is a smooth function (it has a small Lipschitz constant  $\epsilon$ ), in which case one may opt for small  $L$  and  $F$ . Figure 2a–d report four key metrics for an exhaustive search of  $L, F$  combinations. We observe that large values of  $F$  and  $L$  generally reduce the estimation error (Fig. 2a) because they result in reduced bias (Fig. 2b). Nevertheless, setting the parameters to their maximum values is not suggested as the variance is increased (Fig. 2c).

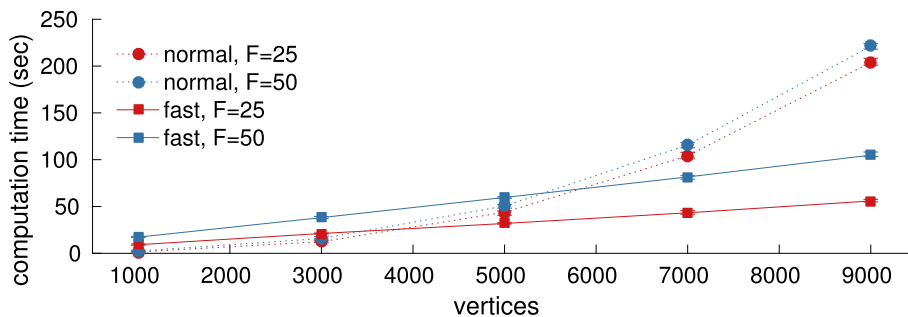
**6.1.3 Complexity**

In Fig. 2d, we see that utilizing a large number of filters (i.e., large  $F$ ) increases the average execution time. Figure 3 delves further into the issue of scalability. In particular, we vary the number of vertices from 1000 to 9000 and focus on a process with JPSD  $h(\theta) = e^{-\lambda/\lambda_{\max}} e^{-5\omega^2}$ .

We then examine the min/median/max execution time of the convolutional JPSD estimator for a for increasing problem sizes when ran in a desktop computer and repeated 10 times. We compare two implementations. The first, which naively performs the convolution in the spectral domain, uses the eigenvalue decomposition and therefore scales quadratically with the number of vertices. Due to its optimized code and simplicity, this should be the method of choice when  $N$  is small. For larger problems, we suggest using the fast implementation. As shown in the figure, this scales linearly with  $N$  (here  $E = O(N)$ ) when the number of filters  $F$  and timesteps  $T$  are held constant. In this experiment, we set  $L$  to 64.

**6.1.4 How to choose  $L$  and  $F$ ?**

Having no computational constrains, one should choose the parameter combination that minimized the Akaike information criterion (AIC) score  $AIC = 2FL - 2\ln(\tilde{\ell})$ , where  $\tilde{\ell}$  is the distribution dependent estimated likelihood  $\tilde{\ell} = \mathbf{P}(\mathbf{x}|\tilde{\Sigma})$  and  $\tilde{\Sigma}$  is the estimated covariance based on the convolutional JPSD estimator with parameters  $L$  and  $F$  [47]. This procedure is often unfeasible as it is based on computing each model’s log-likelihood and thus entails estimating one JPSD for each parameterization in consideration (as well as knowing the distribution



**Fig. 3** Scalability of the convolutional JPSD estimator in seconds (vertical axis) w.r.t. the number of vertices (horizontal axis). The fast implementation should be favored when the graph is composed of more than a few thousand vertices. The approximation error of the fast implementation was negligible in our experiments

type). We have found experimentally that setting  $F = \min(N, 50)$  provides a good trade-off between computational complexity and error. On the other hand, we suggest setting  $L$  to an upper bound of the autocorrelation support.

**6.1.5 Learning from few realizations and a noisy graph**

Figure 4 illustrates the benefit of a joint stationarity prior as compared to (a) an empirical covariance estimator which makes no assumptions about the data and (b) the MTWSS process estimator with optimal bandwidth [22]. As expected, accurate estimation is challenging when the number of realizations is much smaller than the number of problem variables ( $NT$ ), returning errors above one for the empirical estimator. Introducing stationarity priors regularizes the estimation resulting in more stable estimates.

What is perhaps surprising is that even when the graph (and  $U_G$ ) is known only approximately, estimating the second order moment of the distribution using the joint stationarity assumption is beneficial. To portray this phenomenon, we also plot the estimation error when using a noisy graph (we corrupted the weighted adjacency matrix by Gaussian noise, resulting in an SNR of 10 dB). Undoubtedly, introducing noise to the graph

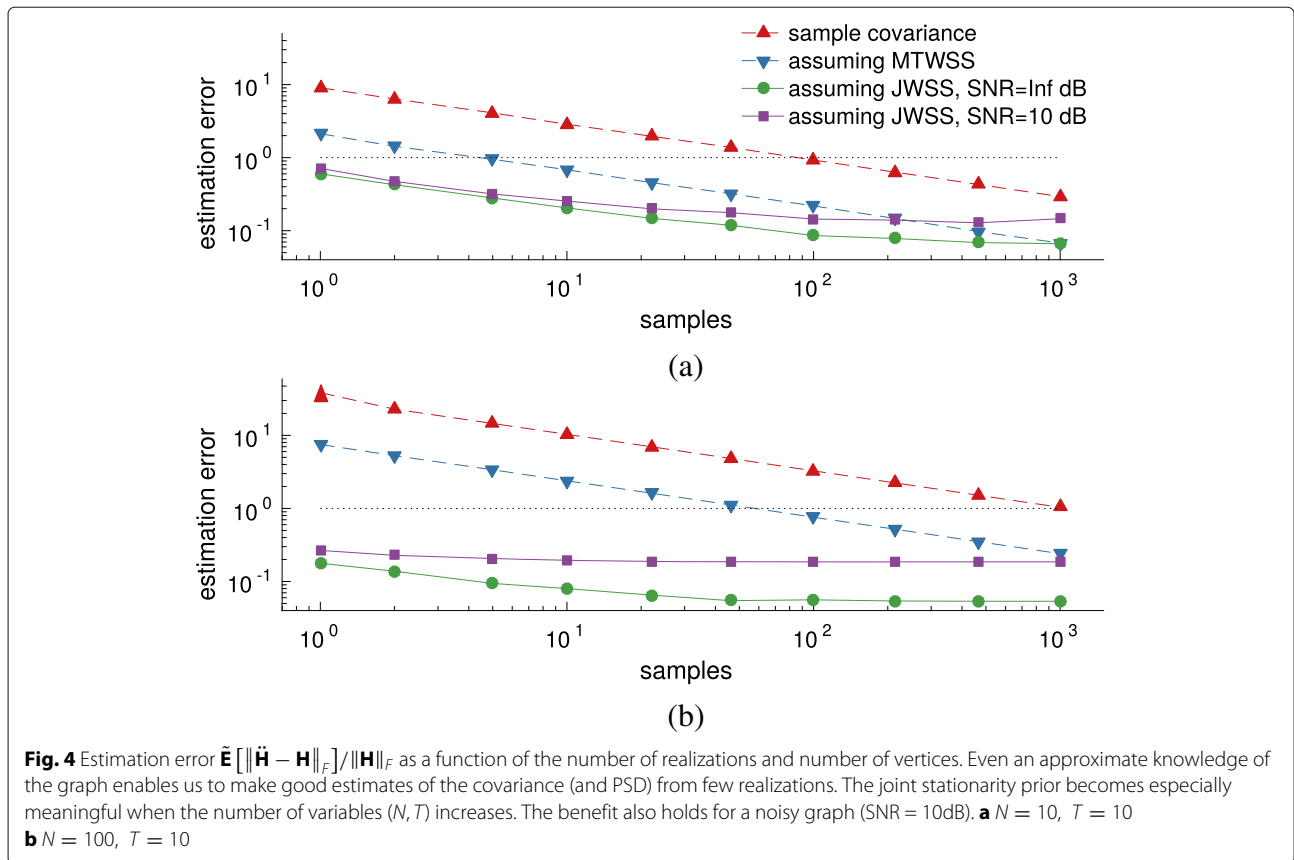
edges negatively affects estimation by introducing bias. Still, even with noise, the proposed method significantly outperforms purely time-based methods when less than  $NT$  realizations are available.

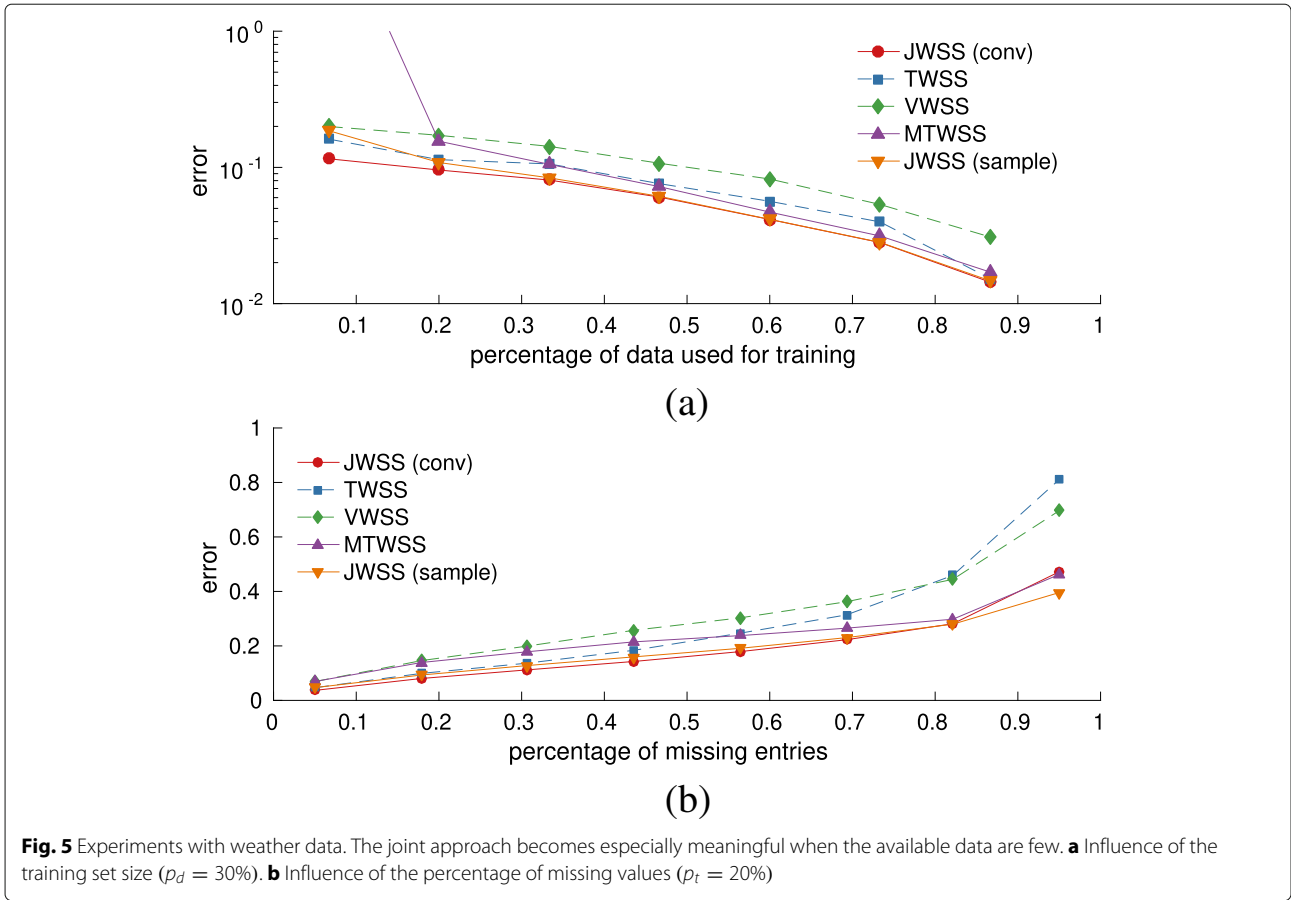
**6.2 Recovery performance on three datasets**

We apply our methods on three diverse datasets featuring multivariate processes evolving over graphs: (a) a weather dataset depicting the temperature of 32 weather stations over 1 month, (b) a traffic dataset depicting high-resolution daily vehicle flow of 4 weekdays, and (c) SIRS-type epidemics in Europe. Our experiments aim to show that joint stationarity is a useful model, even in datasets which may violate the strict conditions of our definition, and that it can yield a significant improvement in recovery performance, as compared to time- or vertex-based stationarity methods.

**6.2.1 Experimental setup**

We split the  $K$  realizations of each dataset into a *training set* of size  $p_t K$  and a *test set* of size  $(1 - p_t)K$ , respectively. The training set is used to estimate the JPSD. Then, in the first two experiments, we attempt to recover the values of  $p_d NT$  variables randomly discarded from the test set. This corresponds to  $A$  being a binary diagonal matrix and





$\mathbf{w} = 0$  in Problem P0, for which the solution is not given by a Wiener filter. In the third experiment, we instead consider a denoising problem with  $\mathbf{A} = \mathbf{I}$  and  $\mathbf{w}$  being a random Gaussian vector. In each case, we report the RMSE for the recovered signal normalized by the  $\ell_2$ -norm of the original signal. We compare our joint method with the sample and convolutional JPSD estimators to *univariate time/vertex stationarity* [17]. These methods solve the statistical recovery problem under the assumption that signals are stationary in the time/vertex domains, but considering different vertices/timesteps as independent. These methods are known to outperform non-model based methods, such as Tikhonov regularization (ridge regression) and total-variation regularization (lasso) over the time or graph dimensions [12, 13]. We also compare to the more involved *MTWSS model* [25] where the values at different vertices are correlated and the covariance is block circulant of size  $NT \times NT$  (see Definition 2). The latter is only shown for the weather dataset as the large number of variables present in the other datasets (e.g.,  $\approx 10^8$  parameters for the traffic dataset) prohibited computation. We remark that the graph Laplacians we considered did not possess eigenvalue multiplicities, meaning that the results obtained using the JWSS-alternate definition

are identical to that with JWSS using a sample JPSD estimator—thus, we do not include JWSS-alternate in our comparison.

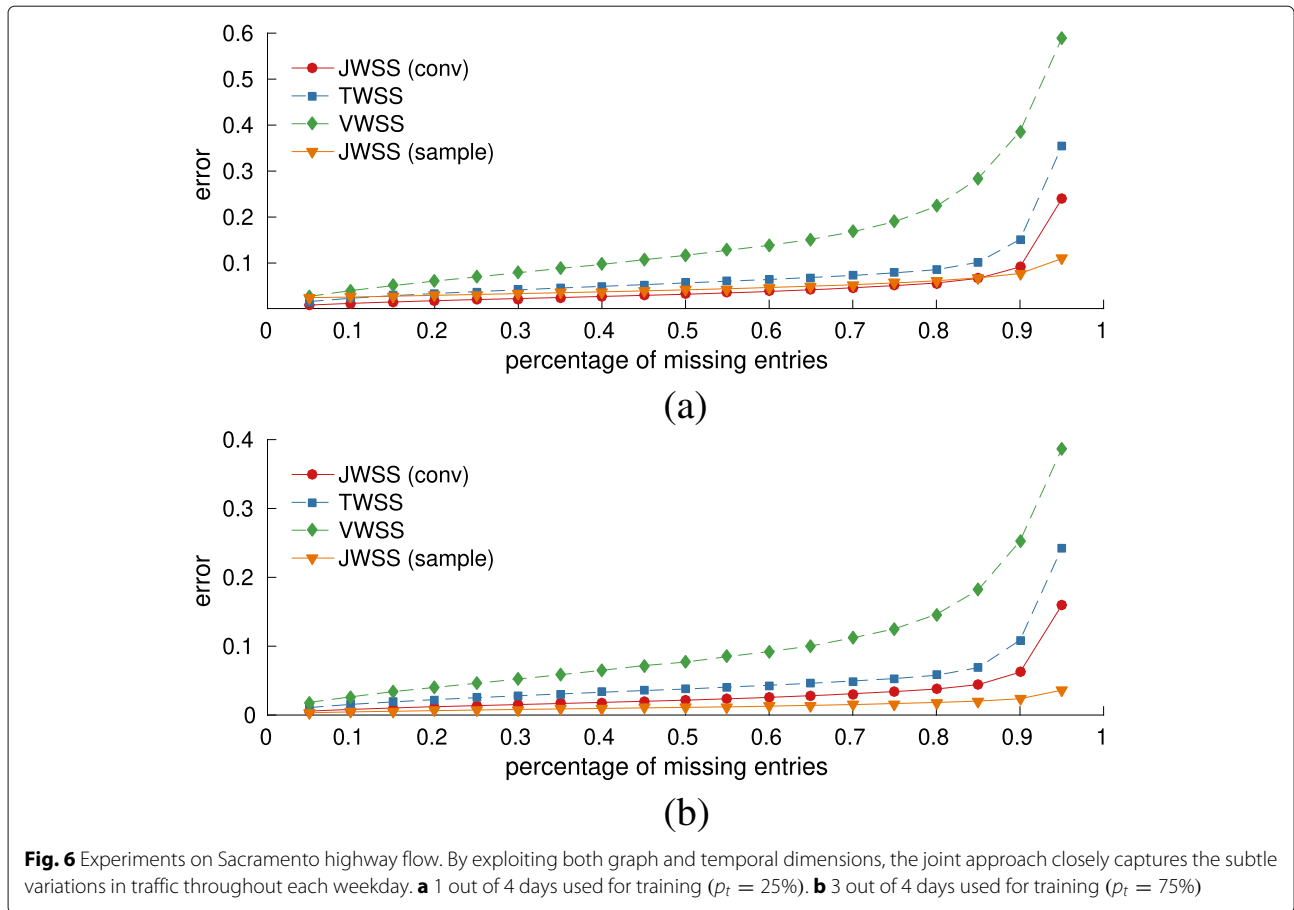
### 6.2.2 Molene dataset

The French national meteorological service has published in open access a dataset<sup>9</sup> with hourly weather observations collected during the month of January 2014 in the region of Brest (France) [18]. The graph was built from the coordinates of the weather stations by connecting all the neighbors in a given radius with a weight function  $\mathbf{W}_G[i_1, i_2] = \exp(-k d(i_1, i_2)^2)$ , where  $d(i_1, i_2)$  is the Euclidean distance between the stations  $i_1$  and  $i_2$ . Parameter  $k$  was adjusted to obtain an average degree around 5 ( $k$  however is not a sensitive parameter). We split the data in  $K = 15$  consecutive periods of  $T = 48$  h each. As sole pre-processing, we removed the mean (over time and stations) of the temperature<sup>10</sup>.

<sup>9</sup> Access to the raw data is possible directly from [https://donneespubliques.meteofrance.fr/donnees\\_libres/Hackathon/RADOMEH.tar.gz](https://donneespubliques.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz)

<sup>10</sup> Though computing separate means (one for each data chunk) yields slightly better performance, to be consistent with the proposed model, we computed a single mean over all training data.





We first test the influence of training set size  $p_t$ , while discarding  $p_d = 30\%$  of the test variables. As seen in Fig. 5a, due to its large sample complexity, the MTWSS approach provides good recovery estimates when the number of realizations is large, approaching that of joint stationarity, but suffers for small training sets (though not shown in the figure, the relative mean error was 9.8 when only  $p_t = 10\%$  of the data was used for training). Due to their stricter modeling assumptions, univariate stationarity methods returned relevant estimates when trained from few realizations but exhibited larger bias. The convolutional JPSD estimator can be seen to improve upon the sample estimator when the amount of data used for JPSD estimation is small (less than 20%). For bigger training sets, the two estimators yield similar accuracy. Figure 5b reports the achieved errors for recovery problems with progressively larger percentage  $5\% \leq p_d \leq 95\%$  of discarded entries for a training percentage of  $p_t = 20\%$ . We can observe that the error trends are consistent across all cases.

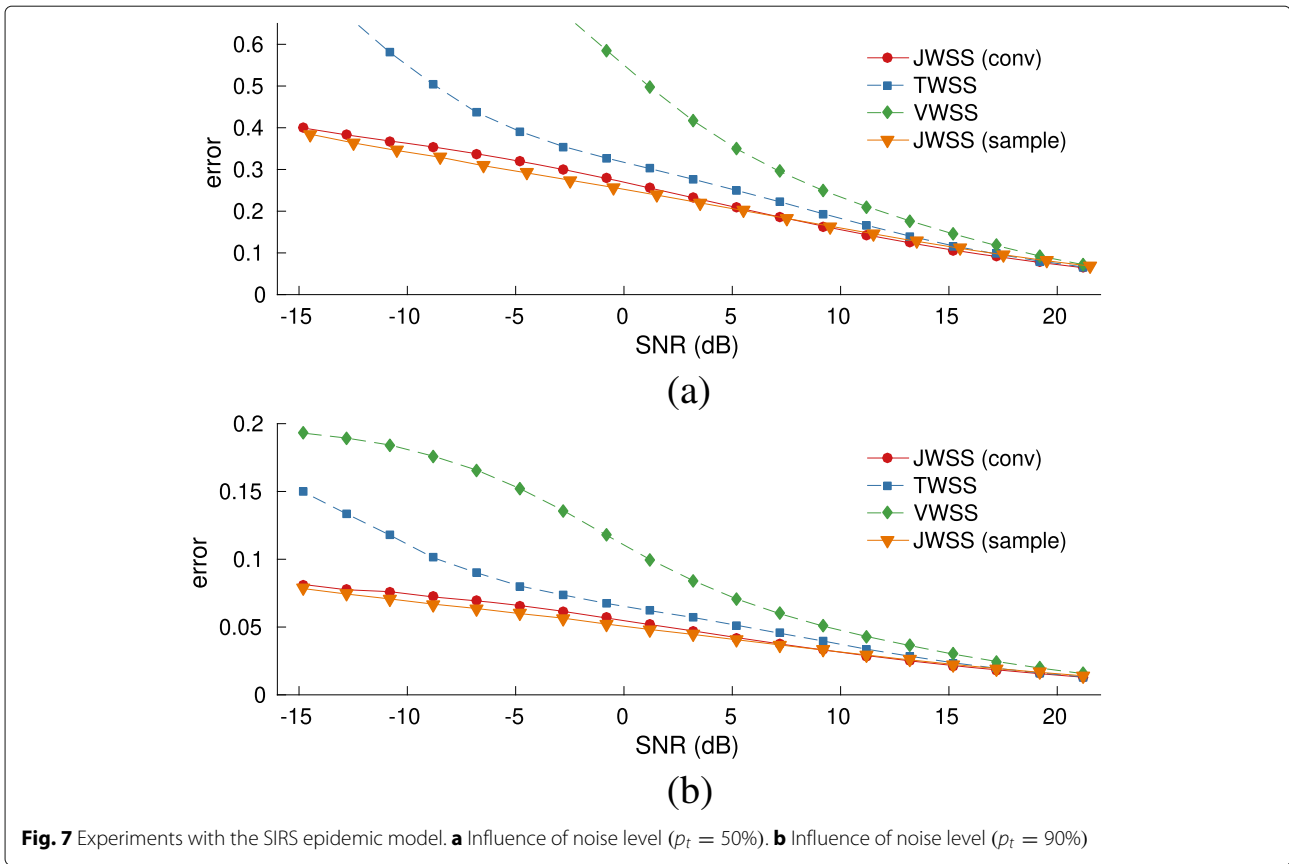
### 6.2.3 Traffic dataset

The California department of transportation publishes high-resolution traffic flow measurements (number of

vehicles per unit interval) from stations deployed in the highways of Sacramento<sup>11</sup>. We focused on 727 stations over four weekdays in the period 01–06 April 2016. Starting from the road connectivity network obtained by the OpenStreetMap.org, we constructed one time series for each highway segment by setting the flow over it to be a weighted average of all nearby stations, while abiding to traffic direction. This resulted in a graph of  $N = 710$  vertices and a total of  $T = 24 \times 12$  measurements per day for  $K = 4$  days. We used the convolutional JPSD estimator with parameters  $L = T/2$  and  $F = 75$ , which were experimentally found to give good performance in the training set.

Figure 6a and b depict the mean recovery errors when the training sets were 1 ( $p_t = 25\%$ ) and 3 days ( $p_t = 75\%$ ) respectively. The strong temporal correlations present in highway traffic were useful in recovering missing values. Considering both the temporal and spatial dimensions of the problem resulted in accurate estimates, with less than 0.04 error when  $p_d = 50\%$  of the data were removed and the PSD was estimated from 1 day. As expected, the

<sup>11</sup>The data correspond to the 3rd district of California and can be downloaded from <http://pems.dot.ca.gov/>



convolutional estimator is efficient in the case when the training set is small (1 out of 4 days used for training): assuming that the JPSD is smooth helps to reduce estimation variance and computational complexity but can lead to a slight decrease in accuracy when a large amount of training data is available.

### 6.2.4 SIRS epidemic

Our third experiment simulates the spread of an infectious disease over  $N = 200$  major cities of Europe, as predicted by the susceptible-infected-recovered-susceptible (SIRS) model, one of the standard models used to study epidemics. We intend to examine the predictive power of the considered methods when dealing with different realizations of a non-linear and probabilistic process over a graph (the data are fictitious). We parameterized SIRS as follows: length of infection period, 2 days; length of immunity period, 10 days; probability of contagion across neighboring cities per day, 0.005; and total period,  $T = 180$  days. We generated a total of  $K = 10$  infections, all having the same starting point.

In contrast to the previous experiments, here, we attempt to recover the data after they have been corrupted with additive Gaussian noise. Figure 7a and b depict the mean recovery error as a function of the input signal-to-

noise ratio (SNR), respectively, when  $p_t = 50\%$  and  $p_t = 90\%$  of the data were used for training. As in previous experiments, the joint stationarity attains better recovery. The difference becomes clearer for low SNR, in which case the error is decreased (roughly) by a factor of two w.r.t. the best alternative.

### 6.2.5 Code

We remark that our simulations were done using the GSP-BOX [48], the UNLocBoX [49], and the LTFAT [50]. The code reproducing our experiments is available at <https://lts2.epfl.ch/stationary-time-vertex-signal-processing/>.

## 7 Conclusion

This paper proposed a new definition of wide-sense stationarity appropriate for multivariate processes supported on the vertices of a graph.

Our model presents two key benefits. *First, the estimation and recovery of JWSS processes is efficient, both in terms of estimation variance and computational complexity.* In particular, the JPSD of a JWSS process can be estimated from few observations at a complexity that is roughly linear to the number of graph edges and timesteps. After the PSD has been estimated, the linear MMSE recovery problems of interpolation and denoising

can be solved in the same asymptotic complexity. *Second, joint stationarity is a volatile model, which is able to capture non-trivial statistical relations in the temporal and vertex domains.* Our experiments suggested that we can model real spatiotemporal processes as jointly stationary without significant loss. Specifically, the JWSS prior was found more expressive than (univariate) TWSS and VWSS priors and improved upon the multivariate time stationarity prior when the dimensionality was large, but the model estimation was based on few observations of the process.

## Appendix

### Implementation details of the JPSD estimator

A straightforward implementation requires  $O(N^3)$  operations for computing the eigenbasis of our graph,  $O(N^2 \times KT)$  for performing  $KT$  independent GFT,  $O(T \log(T) \times KN)$  for  $KN$  independent FFT, and  $O(N^2 T^2)$  for the convolution.

This section describes how to approximate a convolutional estimate using a number of operations that is linear to  $ET$ . Before describing the exact algorithm, we note two helpful properties of the estimator. First, we can compute  $\ddot{h}(\theta)$  by obtaining estimates for each  $\mathbf{X}_{(k)}$  independently and then averaging over  $k$ :

$$\dot{h}(\theta) = \frac{1}{K c_g(\theta)} \sum_k \sum_{n,\tau} g(\theta - \theta_{n,\tau})^2 |\text{JFT}\{\mathbf{X}_{(k)}\}[n, \tau]|^2$$

As we will see in the following, the terms inside the outer sum can be approximated efficiently, avoiding the need for an expensive JFT. In addition, when the convolution window is separable, i.e.,  $g(\theta) = g_G(\lambda) \cdot g_T(\omega)$ , as is assumed in this contribution, the joint convolution can be performed successively (and at any order) in the time and vertex domains

$$\ddot{h}(\theta) = \sum_{\tau} \frac{g_T(\omega - \omega_{\tau})^2}{c_{g_T}(\omega)} \left( \sum_n \frac{g_G(\lambda - \lambda_n)^2}{c_{g_G}(\lambda)} \dot{h}(\theta_{n,\tau}) \right),$$

where  $c_g(\theta) = c_{g_T}(\omega) \cdot c_{g_G}(\lambda)$ . Exploiting this property, we treat the implementation of the two convolutions separately and the presented algorithms can be combined in any order.

**Fast time convolution.** This is the textbook case of TSPSD estimation that is solved by the Welch's method [41]. The method entails splitting each time series into equally sized overlapping segments and averaging over segments the squared amplitude of the Fourier coefficients. The procedure is equivalent to an averaging (over time) of the squared coefficients of a short-time Fourier transform (STFT), with half-overlapping windows  $w_T$  defined such that  $\text{DFT}w_T(t) = g_T(\omega)$  [51, 52]. Let  $L$  be the support of the autocorrelation or equivalently the number

of frequency bands. We suggest using the iterated sine window

$$w_T(t) \triangleq \begin{cases} \sin(0.5\pi \cos(\pi t/L)^2) & \text{if } t \in [-L/2, L/2] \\ 0 & \text{otherwise,} \end{cases}$$

as it turns the STFT into a tight operator. In order to get an estimate of  $\dot{h}$  at unknown frequencies, we interpolate between the  $L$  known points using splines [53].

**Fast graph convolution.** Inspired by the technique of [17], we perform the graph convolution using an approximated graph filtering operation [54] that scales linearly to the number of graph edges  $E$ . In particular,

$$\sum_{n=1}^N \frac{g_G(\lambda - \lambda_n)^2}{c_{g_G}(\lambda)} \dot{h}(\theta_{n,\tau}) = \frac{\mathbf{E}[\|g_G(\lambda \mathbf{I}_N - \mathbf{L}_G) \mathbf{x}_{\tau}\|_2^2]}{c_{g_G}(\lambda)}. \quad (13)$$

We suggest using the Gaussian window

$$g_G(\lambda - \lambda_n) \triangleq e^{-(\lambda - \lambda_n)^2 / \sigma^2}, \quad (14)$$

with  $\sigma^2 = 2(F+1)\lambda_{\max}/F^2$ . As we did before, we only compute the above for  $F = O(1)$  different values of  $\lambda$  and approximate the rest using splines. As the eigenvalues are not known, we need a stable way to estimate  $c_{g_G}(\lambda)$ . We obtain an unbiased estimate by filtering  $Q = O(1)$  random Gaussian signals on the graph  $\epsilon \in \mathbb{R}^N \sim \mathcal{N}(0, \mathbf{I}_N)$ , such that

$$c_{g_G}(\lambda) = \mathbf{E} \left[ \sum_{q=1}^Q \|g_G(\lambda \mathbf{I}_N - \mathbf{L}_G) \epsilon_{(q)}\|_2^2 \right], \quad (15)$$

with variance equal to  $2 \sum_{n=1}^N g^4(\lambda - \lambda_n)/Q$ . We omit the analysis, as it is similar to that in Theorem 2. According to our numerical evaluation, the approximation error introduced by the latter estimator and spectral filtering is almost negligible for smooth JPSD.

$O(TKF \times E + QF \times E) = O((TK + Q)EF)$  for the fast graph convolutions. Here, the  $TK$  and  $Q$  convolutions are performed in order to estimate the quantities at (13) and (15) for  $F$  different values of  $\lambda$ . (b)  $O(NK \times T \log(L))$  for the fast time convolution, corresponding to  $NK$  STFT. Thus, in total the complexity of the fast convolutional JPSD estimator is  $O(TKFE + QEF + NKT \log(L))$ . Furthermore, when  $Q, F, K, L$  are constants, the complexity simplifies to  $O(TE)$ . We remark that, though asymptotically superior, the fast implementation can be significantly slower when the number of variables is small. Our experiments demonstrate that it should be preferred for  $N$  larger than a few thousands (see Fig. 3).

### Univariate vs multivariate JPSD

As discussed in Section 3.2, one could potentially pose a VWSS hypothesis on a product graph to define joint stationarity, but the direct effect of such a choice is that the spectral domain becomes 1-dimensional instead of 2-dimensional. To see why this is problematic, in Fig. 8, we plot the two different representations of the JPSD for the three datasets featured in our experiments. It can be seen that the 2D representation (corresponding to the JWSS hypothesis) is more structured than its 1D counterpart. More importantly, a JWSS hypothesis leads to a smoother JPSD: this is what our convolutional JPSD estimator employs to decrease the estimation variance.

### Deferred proofs

*Proof of Proposition 1* In order to simplify the notation in the next proof, we define the unravel function  $u_r : \mathbb{Z}^2 \rightarrow \mathbb{Z}$  that transforms the double indexes  $n, \tau$  of the matrix indexing of  $\mathbf{X}$  into its vector index of  $u(n, \tau) = (\tau - 1)N + n$ , i.e.,  $\mathbf{X}[n, \tau] = \text{vec}(\mathbf{X})[u(n, \tau)]$ .

By construction of the JFT basis,  $\hat{\mathbf{X}}[0, 0]$  captures the DC-offset of a signal, and condition (a) is equivalent to stating that  $\mathbf{E}[\mathbf{x}] = c\mathbf{1}_{NT}$ . Moreover, if the graph is connected and (a) holds, at least one of  $\mathbf{E}[\hat{\mathbf{X}}[n_1, \tau_1]]$  and  $\mathbf{E}[\hat{\mathbf{X}}[n_2, \tau_2]]$  must be zero when  $n_1 \neq n_2$  or  $\tau_1 \neq \tau_2$  and

$$\begin{aligned} \mathbf{E}[\hat{\mathbf{X}}[n_1, \tau_1] \hat{\mathbf{X}}[n_2, \tau_2]] &= \mathbf{E}[\hat{\mathbf{X}}[n_1, \tau_1] \hat{\mathbf{X}}[n_2, \tau_2]] \\ &\quad - \mathbf{E}[\hat{\mathbf{X}}[n_1, \tau_1]] \mathbf{E}[\hat{\mathbf{X}}[n_2, \tau_2]] \\ &= (\mathbf{U}_J^* \Sigma \mathbf{U}_J)[u(n_1, \tau_1), u(n_2, \tau_2)]. \end{aligned}$$

Therefore, condition (b) is equivalent to stating that  $\Sigma = \mathbf{U}_J \mathbf{D} \mathbf{U}_J^*$  for some diagonal matrix  $\mathbf{D}$ . In addition, (c) asserts that  $\mathbf{D}[u(n, \tau), u(n, \tau)] = h(\lambda_n, \omega_\tau)$  for every  $n, \tau$ . Thus, taken together, (b) and (c) state that  $\Sigma = \mathbf{U}_J \mathbf{D} \mathbf{U}_J^* = \mathbf{U}_J h(\Lambda_G, \Lambda_T) \mathbf{U}_J^* = h(\mathbf{L}_G, \mathbf{L}_T)$ , which is the second moment condition of a JWSS process.  $\square$

*Proof of Theorem 1* For the first moment, it is straightforward to see that  $\mathbf{E}[\mathbf{X}[n, t]] = c$  if and only if both  $\mathbf{E}[\mathbf{X}[n, t]] = c_t$  and  $\mathbf{E}[\mathbf{X}[n, t]] = c_n \forall n, t$ .

For the second moment, the covariance matrix of a JWSS process is by definition the linear operator associated to a joint filter  $\Sigma = h(\mathbf{L}_G, \mathbf{L}_T)$ . Using (5),  $\Sigma_{t_1, t_2}$  can be written as

$$\Sigma_{t_1, t_2} = \mathbf{U}_G \gamma_\delta(\Lambda) \mathbf{U}_G^* = \gamma_\delta(\mathbf{L}_G), \quad (16)$$

where  $\delta = t_1 - t_2 + 1$  and

$$\gamma_\delta(\lambda) = \frac{1}{T} \sum_{\tau=1}^T h(\lambda, \omega_\tau) e^{j\omega_\tau \delta}. \quad (17)$$

Hence, the process satisfies the (b) statement of Definition 2 (TWSS) and 3 (VWSS). Conversely, if a process is TWSS and VWSS, we have  $\Sigma_{t_1, t_2} = \gamma_{t_1, t_2}(\mathbf{L}_G) = \gamma_\delta(\mathbf{L}_G)$  with the same  $\delta$  as before. As a result, using (5), its covariance matrix can be written as a joint filter  $h(\mathbf{L}_G, \mathbf{L}_T)$ , where

$$h(\lambda_n, \omega_\tau) = \sum_{\delta=1}^T \gamma_\delta(\lambda_n) e^{j\omega_\tau \delta}, \quad (18)$$

and hence also satisfies the property of the second moment of JWSS processes.  $\square$

*Proof of Property 2* The output of a filter  $f(\mathbf{L}_J)$  can be written in vector form as  $\mathbf{y} = f(\mathbf{L}_J)$ . If the input signal  $\mathbf{x}$  is JWSS, we can confirm that the first moment of the filter output is  $\mathbf{E}[f(\mathbf{L}_J)\mathbf{x}] = f(\mathbf{L}_J)\mathbf{E}[\mathbf{x}] = f(0, 0)\mathbf{E}[\mathbf{x}]$ , which remains constant as  $\mathbf{E}[\mathbf{x}]$  is constant by hypothesis. The computation of the second moment gives

$$\begin{aligned} \Sigma_y &= \mathbf{E}[f(\mathbf{L}_J)\mathbf{x} (f(\mathbf{L}_J)\mathbf{x})^*] - \mathbf{E}[f(\mathbf{L}_J)\mathbf{x}] \mathbf{E}[f(\mathbf{L}_J)\mathbf{x}]^* \\ &= f(\mathbf{L}_J) \mathbf{E}[\mathbf{x}\mathbf{x}^*] f(\mathbf{L}_J) - f(\mathbf{L}_J) \mathbf{E}[\mathbf{x}] \mathbf{E}[\mathbf{x}]^* f(\mathbf{L}_J)^* \\ &= f(\mathbf{L}_J) \Sigma_x f(\mathbf{L}_J)^* = \mathbf{U}_J (f^2(\Theta) h_X(\Theta)) \mathbf{U}_J^*, \end{aligned}$$

which satisfies the second moment condition of JWSS processes. Above,  $f^2(\Theta)$  is a diagonal  $NT \times NT$  matrix, whose diagonal is obtained by applying the bivariate function  $f^2(\cdot, \cdot)$  on  $[\lambda_n, \omega_\tau]$  for all  $n, \tau$  ( $f$  can be interpreted as the frequency response of a joint filter). Matrix  $h_X(\Theta)$  is similarly defined.  $\square$

**Lemma 1** *If function  $h(\theta)$  is  $\epsilon$ -Lipschitz, then the bias is bounded by*

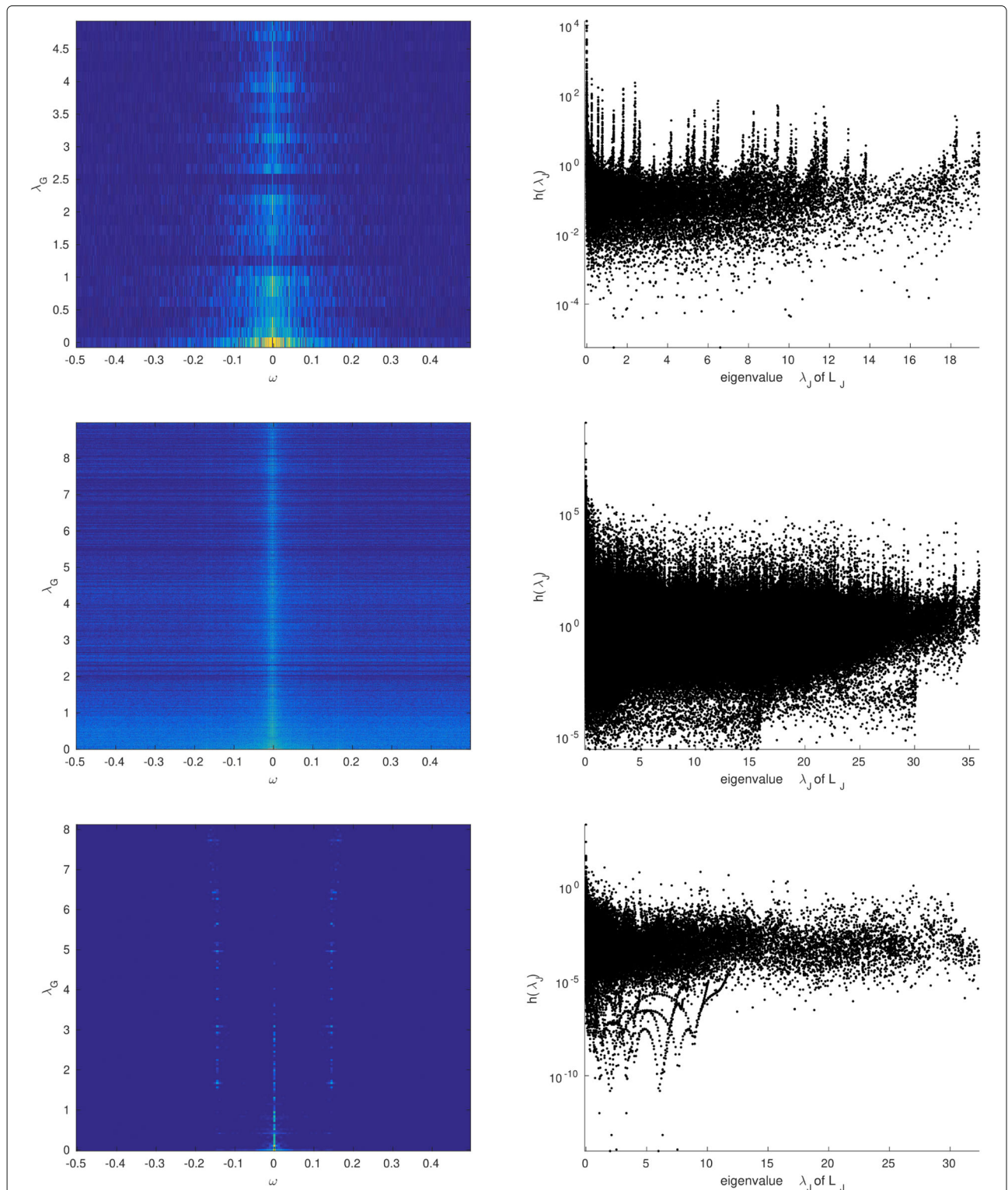
$$\left| \mathbf{E}[\ddot{h}(\theta) - h(\theta)] \right| \leq \frac{\epsilon}{c_g(\theta)} \sum_{n=1, \tau=1}^{T, N} g(\theta - \theta_{n, \tau})^2 \|\theta - \theta_{n, \tau}\|.$$

*Proof* Since  $h(\theta)$  is  $\epsilon$  Lipschitz, we have  $|h(\theta) - h(\theta_{n, \tau})| \leq \epsilon \|\theta - \theta_{n, \tau}\|$ . Hence, we write

$$\begin{aligned} \left| \mathbf{E}[\ddot{h}(\theta) - h(\theta)] \right| &= \left| \left( \sum_{n, \tau=1}^{NT} \frac{g(\theta - \theta_{n, \tau})^2}{c_g(\theta)} h(\theta_{n, \tau}) \right) - h(\theta) \right| \\ &= \left| \sum_{n, \tau=1}^{NT} \frac{g(\theta - \theta_{n, \tau})^2}{c_g(\theta)} (h(\theta_{n, \tau}) - h(\theta)) \right| \\ &\leq \frac{\epsilon}{c_g(\theta)} \sum_{n, \tau=1}^{NT} g(\theta - \theta_{n, \tau})^2 \|\theta - \theta_{n, \tau}\|, \end{aligned}$$

where the second equality stems from  $\sum_{n, \tau} g^2(\theta - \theta_{n, \tau}) = c_g(\theta)$ .  $\square$





**Fig. 8** The JPSD of the three datasets used in our experiments interpreted under a JWSS hypothesis (left) and a VWSS on a product graph hypothesis (right). In both left and right sub-figures, the JPSDs are plotted in logarithmic scale. Interpreting the JPSD as a 2-dimensional function leads to a smoother and more structured representation

**Lemma 2** If  $\mathbf{X}$  is a JWSS process such that the entries of  $\hat{\mathbf{X}}$  are independent random variables, the convolutional JPSD estimate at  $\theta$  has variance

$$\mathbf{Var} \left[ \ddot{h}(\theta) \right] = \sum_{n,\tau} \frac{g(\theta - \theta_{n,\tau})^4}{c_g(\theta)^2} \mathbf{Var} \left[ \dot{h}(\theta_{n,\tau}) \right], \quad (19)$$

where  $\mathbf{Var} \left[ \dot{h}(\theta_{n,\tau}) \right]$  is the variance of the sample JPSD estimator at  $\theta_{n,\tau}$ .

*Proof Set*

$$\alpha_{n,\tau} = g(\theta - \theta_{n,\tau})^2 h(\theta_{n,\tau}) / c_g(\theta)$$

and  $\hat{\mathbf{E}}_{(k)} = \text{mat}(\hat{\epsilon}_{(k)}) = \text{mat}(h(\mathbf{\Lambda}_G, \mathbf{\Omega})^{+1/2} \hat{\mathbf{x}}_{(k)})$ , where  $+$  denotes the pseudo-inverse,  $\hat{\epsilon}_{(k)}$  is white, and  $\text{mat}(\cdot)$  is the matricization operator. The centered random variable

$$\begin{aligned} \ddot{h}(\theta) - \mathbf{E} \left[ \ddot{h}(\theta) \right] &= \sum_{n,\tau} \frac{g(\theta - \theta_{n,\tau})^2}{c_g(\theta)} (\dot{h}(\theta_{n,\tau}) - h(\theta_{n,\tau})) \\ &= \sum_{n,\tau} \alpha_{n,\tau} \left( \sum_k \frac{\hat{\mathbf{E}}_{(k)}[n, \tau] \hat{\mathbf{E}}_{(k)}^*[n, \tau] - 1}{K} \right) \\ &= \sum_{n,\tau} \alpha_{n,\tau} z_{n,\tau} \end{aligned}$$

is a weighted sum of centered, identically distributed random variables  $z_{n,\tau}$ . Moreover, when the elements of  $\hat{\mathbf{E}}_{(k)}$  are independent, so are the variables  $z_{n,\tau}$ . It follows that

$$\begin{aligned} \mathbf{Var} \left[ \ddot{h}(\theta) \right] &= \sum_{n,\tau} \alpha_{n,\tau}^2 \mathbf{Var} \left[ z_{n,\tau}^2 \right] \\ &= \sum_{n,\tau} \frac{g(\theta - \theta_{n,\tau})^4}{c_g(\theta)^2} \mathbf{Var} \left[ \dot{h}(\theta_{n,\tau}) \right], \end{aligned}$$

which matches our claim.  $\square$

#### Abbreviations

AIC: Akaike information criterion; DFT: Discrete Fourier transform; GFT: Graph Fourier transform; JFT: Joint Fourier transform; JPSD: Joint power spectral density; JWSS: Jointly wide-sense stationary; MTWSS: Multivariate time wide-sense stationary; MVWSS: Multivariate vertex wide-sense stationary; PSD: Power spectral density; TPSD: Time power spectral density; TWSS: Time wide-sense stationary; VPSD: Vertex power spectral density; VWSS: Vertex wide-sense stationary; WSS: Wide-sense stationary

#### Acknowledgements

We thank Francesco Grassi for his help with the code.

#### Authors' contributions

The two authors contributed equally both for the experiments and for the writing of the paper. Both authors read and approved the final manuscript.

#### Funding

This work has been supported by the Swiss National Science Foundation research project *Towards Signal Processing on Graphs* (grant number: 2000\_21/154350/1) and research project *Deep Learning for Graph-Structured Data* (grant number: PZ00P2 179981).

#### Availability of data and materials

The code to reproduce the results is available at <https://lts2.epfl.ch/stationary-time-vertex-signal-processing/>.

- Access to the raw Molene dataset is possible directly from [https://donneespubliques.meteofrance.fr/donnees\\_libres/Hackathon/RADOMEH.tar.gz](https://donneespubliques.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz)
- The traffic data corresponding to the 3rd district of California and can be downloaded from <http://pems.dot.ca.gov/>.
- The epidemic dataset is synthetically generated from a SIR model. The network used for the model can be downloaded from <https://www.visualizing.org/global-flights-network/>.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>Laboratoire de Traitement des Signaux 2, École Polytechnique Fédérale Lausanne, 1015 Lausanne, Switzerland. <sup>2</sup>Swiss Data Science Center, Eidgenössische Technische Hochschule Zürich, Universitätstrasse 25, 8006 Zürich, Switzerland.

Received: 29 November 2018 Accepted: 2 July 2019

Published online: 20 August 2019

#### References

1. M. Rudelson, Random vectors in the isotropic position. *J. Funct. Anal.* **164**(1), 60–72 (1999)
2. H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Sci. Bus. Media (2005)
3. O. Ledoit, M. Wolf, A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivar. Anal.* **88**(2), 365–411 (2004)
4. C. Lam, Q. Yao, et al., Factor modeling for high-dimensional time series: inference for the number of factors. *Ann. Stat.* **40**(2), 694–726 (2012)
5. G. Connor, The three types of factor models: a comparison of their explanatory power. *Financ. Anal. J.* **51**(3), 42–46 (1995)
6. M. J. Keeling, K. T. Eames, Networks and epidemic models. *J. R. Soc. Interface.* **2**(4), 295–307 (2005)
7. P. Mohan, V. N. Padmanabhan, R. Ramjee, in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. Nericell: rich monitoring of road and traffic conditions using mobile smartphones (ACM, 2008), pp. 323–336
8. W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, A. Ribeiro, Graph frequency analysis of brain signals. *IEEE J. Sel. Top. Signal Proc.* **10**(7), 1189–1203 (2016)
9. F. Zhang, E. R. Hancock, Graph spectral image smoothing using the heat kernel. *Pattern Recog.* **41**(11), 3328–3342 (2008)
10. A. J. Smola, R. Kondor, Kernels and regularization on graphs. *Learning theory and kernel machines*, 144–158 (2003)
11. M. Belkin, P. Niyogi, Semi-supervised learning on riemannian manifolds. *Mach. Learn.* **56**(1–3), 209–239 (2004)
12. D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Process. Mag. IEEE.* **30**(3), 83–98 (2013)
13. A. Sandryhaila, J. M. Moura, Discrete signal processing on graphs. *IEEE Trans. Signal Process.* **61**, 1644–1656 (2013)
14. A. Sandryhaila, J. M. Moura, Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.* **31**(5), 80–90 (2014)
15. A. Gadde, A. Ortega, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. A probabilistic interpretation of sampling theory of graph signals (IEEE, 2015), pp. 3257–3261
16. C. Zhang, D. Florêncio, P. A. Chou, Graph signal processing—a probabilistic framework. *Microsoft Res.* (2015). Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31
17. N. Perraudin, P. Vandergheynst, Stationary signal processing on graphs. *IEEE Trans. Signal Process.* **65**(13), 3462–3477 (2017). <https://doi.org/10.1109/TSP.2017.2690388>
18. B. Girault, in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. Stationary graph signals using an isometric graph translation (IEEE, 2015), pp. 1516–1520
19. A. G. Marques, S. Segarra, G. Leus, A. Ribeiro, Stationary graph processes and spectral estimation. *IEEE Trans. Signal Process.* **65**(22), 5911–5926 (2016)

20. E. Isufi, A. Loukas, A. Simonetto, G. Leus, Autoregressive moving average graph filtering. *IEEE Trans. Signal Process.* **65**(2), 274–288 (2017)
21. A. Loukas, D. Foucard, in *Global Conference on Signal and Information Processing (GlobalSIP)*. Frequency analysis of time-varying graph signals (IEEE, 2016), pp. 346–350
22. N. Wiener, P. Masani, The prediction theory of multivariate stochastic processes. *Acta Math.* **98**(1), 111–150 (1957)
23. N. Wiener, P. Masani, The prediction theory of multivariate stochastic processes, ii. *Acta Math.* **99**(1), 93–137 (1958)
24. P. Bloomfield, *Fourier analysis of time series: an introduction*. (Wiley, 2004)
25. F. R. Bach, M. I. Jordan, Learning graphical models for stationary time series. *IEEE Trans. Signal Process.* **52**(8), 2189–2199 (2004)
26. R. Dahlhaus, M. Eichler, Causality and graphical models in time series analysis. *Oxf. Stat. Sci. Ser.*, 115–137 (2003)
27. B. Girault, Signal processing on graphs-contributions to an emerging field. PhD thesis, Ecole normale supérieure de Lyon (2015)
28. S. P. Chepuri, G. Leus, in *Sensor Array and Multichannel Signal Processing Workshop (SAM)*. Subsampling for graph power spectrum estimation (IEEE, 2016), pp. 1–5
29. A. Loukas, E. Isufi, N. Perraudin, in *Asilomar Conference on Signals, Systems, and Computers*. Predicting the evolution of stationary graph signals (IEEE, 2017), pp. 60–64
30. J. Mei, J. M. Moura, Signal processing on graphs: causal modeling of unstructured data. *IEEE Trans. Signal Process.* **65**(8), 2077–2092 (2017)
31. V. N. Ioannidis, D. Romero, G. B. Giannakis, Inference of spatio-temporal functions over graphs via multikernel kriged kalman filtering. *Trans. Signal Process.* **66**(12), 3228–3239 (2017)
32. P. D. Lorenzo, E. Isufi, P. Banelli, S. Barbarossa, G. Leus, in *2017 25th European Signal Processing Conference (EUSIPCO)*. Distributed recursive least squares strategies for adaptive reconstruction of graph signals, (2017), pp. 2289–2293. <https://doi.org/10.23919/EUSIPCO.2017.8081618>
33. N. Perraudin, A. Loukas, F. Grassi, P. Vandergheynst, Towards stationary time-vertex signal processing. *Int. Conf. Acoust. IEEE Speech Signal Process. (ICASSP)* (2017)
34. F. Grassi, A. Loukas, N. Perraudin, B. Ricaud, A time-vertex signal processing framework. *IEEE Trans. Signal Process.* **66**(3), 817–829 (2017)
35. A. Loukas, A. Simonetto, G. Leus, Distributed autoregressive moving average graph filters. *IEEE Signal Process. Lett.* **22**(11), 1931–1935 (2015)
36. E. Isufi, A. Loukas, A. Simonetto, G. Leus, in *Signal Processing Conference (EUSIPCO), 2016 24th European*. Separable autoregressive moving average graph-temporal filters (IEEE, 2016), pp. 200–204
37. D. I. Shuman, B. Ricaud, P. Vandergheynst, Vertex-frequency analysis on graphs. *Appl. Comput. Harmon. Anal.* **40**(2), 260–291 (2016)
38. S. Segarra, S. P. Chepuri, A. G. Marques, G. Leus, Statistical graph signal processing: Stationarity and spectral estimation, 325–347 (2018)
39. R. Vershynin, How close is the sample covariance matrix to the actual covariance matrix? *J. Theor. Probab.* **25**(3), 655–686 (2012)
40. M. S. Bartlett, Periodogram analysis and continuous spectra. *Biometrika.* **37**(1/2), 1–16 (1950)
41. P. Welch, The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *Trans. Audio Electroacoustics IEEE.* **15**(2), 70–73 (1967)
42. O. Axelsson, Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations. *Linear Algebra Appl.* **29**, 1–16 (1980)
43. P. L. Combettes, V. R. Wajs, Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **4**(4), 1168–1200 (2005)
44. P. L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*, 185–212 (2011)
45. N. Komodakis, J.-C. Pesquet, Playing with duality: an overview of recent primal/dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.* **32**(6), 31–54 (2015)
46. P. L. Combettes, J.-C. Pesquet, A douglas-rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Sel. Top. Signal Process.* **1**(4), 564–574 (2007)
47. H. Akaike, A new look at the statistical model identification. *IEEE Trans. Autom. Control.* **19**(6), 716–723 (1974)
48. N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, D. K. Hammond, GSPBOX: A toolbox for signal processing on graphs. *ArXiv e-prints* (2014). [1408.5781](https://arxiv.org/abs/1408.5781)
49. N. Perraudin, D. Shuman, G. Puy, P. Vandergheynst, UNLocBoX A matlab convex optimization toolbox using proximal splitting methods. *ArXiv e-prints* (2014). [1402.0779](https://arxiv.org/abs/1402.0779)
50. Z. Prusa, P. L. Sondergaard, N. Holighaus, C. Wiesmeyr, P. Balazs, The large time-frequency analysis toolbox 2.0. *Sound, Music, and Motion: 10th International Symposium.* **8905**, 419–442 (2014)
51. K. Gröchenig, *Foundations of time-frequency analysis*. Springer Sci. Bus. Media (2013)
52. H. G. Feichtinger, T. Strohmer, *Gabor analysis and algorithms: theory and applications*. Springer Sci. Bus. Media (2012)
53. C. De Boor, *A practical guide to splines*, vol. 27. (Springer-Verlag, New York, 1978)
54. A. Susnjara, N. Perraudin, D. Kressner, P. Vandergheynst, Accelerated filtering on graphs using Lanczos method. *arXiv preprint* (2015). [arXiv:1509.04537](https://arxiv.org/abs/1509.04537)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---