Forecasting Time Series with VARMA Recursions on Graphs

Elvin Isufi *Student Member, IEEE,*, Andreas Loukas, Nathanael Perraudin and Geert Leus *Fellow, IEEE,*

Abstract—Graph-based techniques emerged as a choice to deal with the dimensionality issues in modeling multivariate time series. However, there is yet no complete understanding of how the underlying structure could be exploited to ease this task. This work provides contributions in this direction by considering the forecasting of a process evolving over a graph. We make use of the (approximate) time-vertex stationarity assumption, i.e., time-varying graph signals whose first and second order statistical moments are invariant over time and correlated to a known graph topology. The latter is combined with VAR and VARMA models to tackle the dimensionality issues present in predicting the temporal evolution of multivariate time series.

We find out that by projecting the data to the graph spectral domain: (*i*) the multivariate model estimation reduces to that of fitting a number of uncorrelated univariate ARMA models and (*ii*) an optimal low-rank data representation can be exploited so as to further reduce the estimation costs. In the case that the multivariate process can be observed at a subset of nodes, the proposed models extend naturally to Kalman filtering on graphs allowing for optimal tracking. Numerical experiments with both synthetic and real data validate the proposed approach and highlight its benefits over state-of-the-art alternatives.

Index Terms—Multivariate time series, prediction, forecasting, graph signal processing, joint stationarity, time-vertex graph signals, Kalman filter, ARMA models, VARMA models.

I. INTRODUCTION

Forecasting multivariate processes is a central topic in signal processing [2], [3]. Especially in the high-dimensional setting, the forecasting of *unstructured data* is considered a complex task, both from an estimation and computational perspective –indeed, most works in this direction stay within the limit of a few dozen time series. To cope with the dimensionality challenge, particularly when the number of observations is smaller than the number of time series, typical solutions involve factor models [4], [5], shrinkage estimators [6], [7], and low-rank data representations [3].

Recently, forecasting has been considered for *graph-structured* processes, i.e., time-varying signals living on the nodes of graphs [8]–[12]. Here, given a graph and a set of time series, the aim is to exploit the underlying structure for improving the prediction accuracy of the considered model. The core idea is that the additional information, carried by the graph, can enhance our ability to make relevant predictions. The graph basically aids to restrict the degrees of freedom (DoFs) of the model in a manner faithful to the problem at

hand. Depending on the scenario, these works exploit different signal priors w.r.t. the graph to formulate the prediction task.

The first to consider prediction in the graph context is the work of [8], which used the vector autoregressive (VAR) recursions to learn the underlying topology and then used this structure to forecast future values. Their findings rely on a temporal interpretation of edges. More specifically, it is assumed that the information requires a unit time step to reach adjacent nodes. Coupling the duration of graph interactions and the fixed temporal resolution in such a manner may be beneficial for graph learning purposes, but negatively affects forecasting since it ignores hidden correlations between nodes.

Relevant work has also been done in *tracking* time-varying signals on graphs[1]. In the adaptive approaches [9], [10], the prediction/tracking of time-varying graph signals comes as a byproduct of the presented adaptive estimation strategies. Here, the authors exploit a low-rank representation of the signal, due to the bandlimitedness in the graph spectral domain, to track slow signal variations over time.

A different approach for network process tracking is followed in [11], [12]. Here, a virtual graph extrapolated to the temporal dimension is considered and the multivariate time series is treated as a single time-invariant (extended) graph signal on this larger graph. Then, under priors such as diffusion or smoothness, the tracking task is rephrased as an interpolation problem. To ease the excessive computational burden, gradient descent algorithms [11] or Kalman-based recursions [12] are invoked.

Almost all the above works treat the time series as the evolution of a noise-corrupted deterministic process. However, in many practical situations including opinion blogs, temperature sensor networks, financial markets, or brain networks, the time series has a strong stochastic component. The latter requires models able to capture the stochasticity along both the graph and time dimension and allow the short time prediction from few training samples. Some earlier contributions have considered stochasticity in time, see e.g., [8], [10], yet leaving unexplored the stochasticity along the graph dimension.

This work takes one step further and treats time series as wide-sense stationary (for short stationary) statistical processes w.r.t. both the graph and temporal domains. While temporal stationarity preserves the process statistics w.r.t. translations in time, the considered joint stationarity hypothesis also preserves these statistics w.r.t. graph localization. A typical example comprises temperature measurements in sensor networks. The

[1]For the sake of clarity, the reader should distinguish between forecasting and tracking a time-varying signal. In forecasting, x_t is predicted based on historical data $\{x_0, \ldots, x_{t-1}\}$, while in tracking noisy measurements at time t are also used.

graph stationarity implies that neighboring sensors yield in expectation a constant value. The joint stationary, on the other hand, considers the expected constant value to be also preserved in consecutive time instances[2].

Continuing our prior works [16], [17], we exploit the (approximate) time-vertex stationarity of graph time series and extend classical VAR and vector autoregressive moving average (VARMA) recursions for modeling and predicting time-varying processes on graphs. Specifically, the contributions of this work are:

1) We propose VAR and VARMA models for forecasting time series on graphs. These recursions extend the non-causal models of [16] to causal ones and generalize the approach of [8] to (i) VAR recursions that do not impose a temporal interpretation of the edges, and (ii) VARMA recursions where the MA part allows to model time series with a more complete structure.

2) We propose a model fitting method that makes use of the data structure and stationarity. Specifically, we separate the multivariate process into uncorrelated univariate time series (one per graph frequency), hence tackling the dimensionality issue present in VARMA modeling. The latter enables us to use well-established univariate techniques for estimating the model coefficients from limited training data. We further introduce an optimal low-rank data representation, which intrinsically makes use of the graph to further reduce the computational overhead of the model estimation.

3) We introduce a sub-graph tracker to estimate the time series on a subset of nodes. The latter makes use the prediction at time t along with graph sub-sampled measurements to estimate the attributes on the nodes of interest. We show a direct connection between the proposed approach and Kalman filtering (KF) on graphs [18] that allows a direct characterization of the sub-graph tracker performance.

The paper makes an additional minor contribution. It generalizes the separable two-dimensional time-vertex ARMA filter of [17] to a non-separable version and shows that the latter encompasses other state-of-the-art time-vertex recursions on graphs as the ones presented in [8], [19] and can be used for distributed predictors.

To validate the joint stationarity hypothesis and the proposed models, we consider three scenarios: (i) the Molene weather data set[3]; the NOAA U.S. temperature data set [20] (ii) a dynamic mesh data set containing a walking dog; and (iii) an epidemic spreading scenario over a flight transportation network. The obtained results corroborate the relevance of graph VARMA models for forecasting stochastic time series supported on graphs and show that the proposed approaches outperform both classical univariate and multivariate ARMA techniques and state-of-the-art graph-based techniques.

[2]We remark that although perfect stationarity either in the time or graph domain is difficult to be observed, several real data sets [13]–[16] have shown some degree of graph/time stationarity that can be exploited for prediction.

[3]Data publicly available at https://donneespubliques. meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz.

Notation. The scalar a_i indicates the ith entry of vector $\boldsymbol{a}$, and similarly the scalar $A_{i,j}$ indicates the (i, j)th entry of matrix $\boldsymbol{A}$. When needed these entries will also be denoted as $[\boldsymbol{a}]_i$, or $[\boldsymbol{A}]_{ij}$. $\boldsymbol{A}^{\top}$, $\boldsymbol{A}^*$ and $\boldsymbol{A}^{\mathsf{H}}$ are respectively the transpose, the element-wise conjugate, and the Hermitian of $\boldsymbol{A}$. $\|\boldsymbol{a}\|_p$ and $\|\boldsymbol{A}\|_p$ indicate respectively the p-th norm of $\boldsymbol{a}$ and $\boldsymbol{A}$, whereas $\|\boldsymbol{A}\|_F$ is the Frobenius norm of matrix $\boldsymbol{A}$. The trace of $\boldsymbol{A}$ is denoted as $\mathrm{tr}(\boldsymbol{A})$ and $\boldsymbol{A} \otimes \boldsymbol{B}$ is the Kronecker product of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. $\boldsymbol{a} = \mathrm{diag}(\boldsymbol{A})$ is the operation that stores the diagonal elements of $\boldsymbol{A}$ into the vector $\boldsymbol{a}$, and $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{a})$ is a diagonal matrix containing the vector $\boldsymbol{a}$ in the main diagonal. The operation $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A})$ concatenates the columns of $\boldsymbol{A}$ in $\boldsymbol{a}$ and $\mathrm{vec}^{-1}(\cdot)$ is the inverse operation. The $N \times N$ identity matrix is denoted as $\boldsymbol{I}_N$, the $N \times 1$ all ones vector as $\boldsymbol{1}_N$, and the $N \times 1$ all zeros vector as $\boldsymbol{0}_N$. For a set $\mathcal{S} \subseteq \{1, \ldots, N\}$, $|\mathcal{S}|$ is the set cardinality, and $\boldsymbol{D}_{\mathcal{S}}$ is an $N \times N$ diagonal matrix with $[\boldsymbol{D}_{\mathcal{S}}]_{ii} = 1$ if $i \in \mathcal{S}$ and zero otherwise. To avoid confusion, we refer to *causality* in the classical sense [3], i.e., an event happening at time t is influenced only by events prior to t independently of the graph dimension. We will refer to the notion of causality used by [8] as *restricted causality*, being a subclass of the former.

Paper organization. Section II contains the background information. Section III formulates the prediction problem in the graph signal processing (GSP) perspective and develops the optimal predictor. Section IV focuses on the model parameter design. The subgraph tracker is introduced and analyzed in Section V. Section VI contains the numerical results and the paper conclusions are drawn in Section VII.

II. PRELIMINARIES

This section starts with the fundamentals of graph and time-vertex signal processing. Then, it proceeds with the two-dimensional time-vertex ARMA models and the definition of joint stationarity that will be used in the rest of the paper.

A. Graph Signal Processing

Consider a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W}_{\mathcal{G}})$, where $\mathcal{V}$ is the set of N nodes (vertices), $\mathcal{E}$ is the edge set, and $\boldsymbol{W}_{\mathcal{G}}$ is the weighted adjacency matrix with $[\boldsymbol{W}_{\mathcal{G}}]_{ij} = [\boldsymbol{W}_{\mathcal{G}}]_{ji} > 0$ if $(i, j) \in \mathcal{E}$ and zero otherwise. A graph signal is a mapping from the vertex set to the set of complex numbers, i.e., $\mathrm{x} : \mathcal{V} \rightarrow \mathbb{C}$. For convenience we represent a graph signal by the vector $\boldsymbol{x} = [x_1, \ldots, x_N]^{\top}$, where x_i is the value related to node i. The graph Fourier transform (GFT) of $\boldsymbol{x}$ is

$$\mathrm{GFT}\{\boldsymbol{x}\} = \boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}} \boldsymbol{x}, \qquad (1)$$

where $\boldsymbol{U}_{\mathcal{G}}$ is the unitary eigenvector matrix of the discrete Laplacian matrix $\boldsymbol{L}_{\mathcal{G}} = \mathrm{diag}(\boldsymbol{W}_{\mathcal{G}} \boldsymbol{1}_N) - \boldsymbol{W}_{\mathcal{G}} = \boldsymbol{U}_{\mathcal{G}} \boldsymbol{\Lambda}_{\mathcal{G}} \boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$. The diagonal matrix $\boldsymbol{\Lambda}_{\mathcal{G}}$ contains the eigenvalues on its main diagonal, often referred to as the graph frequencies [21]. The GFT allows filtering on graphs in the graph frequency domain. The filtered version $\boldsymbol{y}$ of $\boldsymbol{x}$ with the graph filter $h(\boldsymbol{L}_{\mathcal{G}}) \in \mathbb{R}^{N \times N}$ corresponds to an element-wise multiplication in the spectral domain, i.e.,

$$\boldsymbol{y} = h(\boldsymbol{L}_{\mathcal{G}}) \boldsymbol{x} \triangleq \boldsymbol{U}_{\mathcal{G}} h(\boldsymbol{\Lambda}_{\mathcal{G}}) \boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}} \boldsymbol{x}, \qquad (2)$$

with matrix $h(\boldsymbol{L}_{\mathcal{G}})$ obtained by applying the scalar function $h : [0, \lambda_{max}] \to \mathbb{R}$ on the eigenvalues of $\boldsymbol{L}_{\mathcal{G}}$. Here, $h(\boldsymbol{\Lambda}_{\mathcal{G}})$ is diagonal and is referred to as the graph filter frequency response. See [21]–[23] for more details on GSP and [24]–[27] for graph filters.

B. Time-Vertex Signal Processing

For a time-varying graph signal $\boldsymbol{x}_t$, let $\boldsymbol{X}_{1:T} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{N \times T}$ be the $N \times T$ matrix that collects T successive realizations. Let also $\boldsymbol{x}_{1:T} = \text{vec}(\boldsymbol{X}_{1:T})$ denote the vectorized form of $\boldsymbol{X}_{1:T}$. From [28], the joint (time-vertex) Fourier transform (JFT) of $\boldsymbol{X}_{1:T}$ is

$$\text{JFT}\{\boldsymbol{X}_{1:T}\} = \boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}} \boldsymbol{X}_{1:T} \boldsymbol{U}_{\mathcal{T}}^{*}, \qquad (3)$$

where, once again $\boldsymbol{U}_{\mathcal{G}}$ is the graph Laplacian eigenvector matrix, while $\boldsymbol{U}_{\mathcal{T}}^{*} \in \mathbb{C}^{T \times T}$ is the complex conjugate of the discrete Fourier transform (DFT) matrix. $\boldsymbol{U}_{\mathcal{T}}$ can also be interpreted as the unitary eigenvector matrix of the circulant time-shift operator $\boldsymbol{L}_{\mathcal{T}} = \boldsymbol{U}_{\mathcal{T}} \boldsymbol{\Lambda}_{\mathcal{T}} \boldsymbol{U}_{\mathcal{T}}^{\mathsf{H}}$

$$[\boldsymbol{\Lambda}_{\mathcal{T}}]_{tt} = e^{-\jmath \omega_t}, \quad \text{and} \quad \omega_t = \frac{2\pi(t-1)}{T} \qquad (4)$$

for $\jmath = \sqrt{-1}$ and $t = 1, \ldots, T$. In a vectorized form, (3) becomes $\text{JFT}\{\boldsymbol{x}_{1:T}\} = \boldsymbol{U}_{\mathcal{J}}^{\mathsf{H}} \boldsymbol{x}$, with $\boldsymbol{U}_{\mathcal{J}} = \boldsymbol{U}_{\mathcal{T}} \otimes \boldsymbol{U}_{\mathcal{G}}$ being a unitary matrix.

Similar to the GFT, the JFT allows a *joint* time-vertex filtering of $\boldsymbol{x}_{1:T}$ in the joint time-vertex frequency domain. The latter is given by the eigendecomposition of the extended Laplacian operator $\boldsymbol{L}_{\mathcal{J}} = \boldsymbol{I}_N \otimes \boldsymbol{L}_{\mathcal{G}} + \boldsymbol{L}_{\mathcal{T}} \otimes \boldsymbol{I}_N$ [28]. The joint filtered version $\boldsymbol{y}_{1:T}$ of $\boldsymbol{x}_{1:T}$ with a joint filter $h(\boldsymbol{L}_{\mathcal{J}})$ corresponds again to an element-wise multiplication, but now in the joint frequency representation, i.e.,

$$\boldsymbol{y}_{1:T} = h(\boldsymbol{L}_{\mathcal{J}})\boldsymbol{x}_{1:T} \triangleq \boldsymbol{U}_{\mathcal{J}} \, h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}}) \, \boldsymbol{U}_{\mathcal{J}}^{\mathsf{H}} \boldsymbol{x}_{1:T}. \qquad (5)$$

Here, $h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})$ is an $NT \times NT$ diagonal matrix called the joint time-vertex frequency response with kth diagonal element $[h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})]_{kk} = h(\lambda_n, e^{\jmath \omega_t})$ and $k = N(t-1) + n$ for $n = 1, \ldots, N$ and $t = 1, \ldots, T$. By means of inverse vectorization, we write $\boldsymbol{Y}_{1:T} = \text{vec}^{-1}(\boldsymbol{y}_{1:T})$ and the tth column of $\boldsymbol{Y}_{1:T}$, $\boldsymbol{y}_t$, is the filtered signal at time t.

Time-vertex ARMA filters. An efficient way to (distributively) implement (5) is by extending the two-dimensional ARMA filters [17] to the more general form

$$\boldsymbol{y}_t + \sum_{p=1}^{P} \sum_{l=0}^{L_p} \psi_{l,p} \boldsymbol{L}_{\mathcal{G}}^{l} \boldsymbol{y}_{t-p} = \sum_{q=0}^{Q} \sum_{k=0}^{K_q} \varphi_{k,q} \boldsymbol{L}_{\mathcal{G}}^{k} \boldsymbol{x}_{t-q}, \quad (6)$$

where P, L_p, Q, K_q are positive scalars and $\psi_{l,p}, \varphi_{k,q}$ denote complex coefficients. In computing the filter output $\boldsymbol{y}_t$, recursion (6) considers a linear combination of the current and past graph signal realizations $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t-Q}\}$ and of the previous outputs $\{\boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-P}\}$. This temporal shifting of the filter input-output provides a filtering in the temporal domain. The graph filtering is provided by shifting each temporal realization $\boldsymbol{x}_{t-\tau}$ $(\boldsymbol{y}_{t-\tau})$ in the graph dimension using powers of $\boldsymbol{L}_{\mathcal{G}}$.

To see more formally the two-dimensional filtering effect of (6), we first apply the DFT and then the GFT on both sides of (6) to obtain the per time-vertex frequency relation

$$\left(1 + \sum_{p=1}^{P} \sum_{l=0}^{L_p} \psi_{l,p} \lambda_n^{l} e^{-\jmath \omega_t p}\right) \hat{y}_n(e^{\jmath \omega_t}) =$$
$$\sum_{q=0}^{Q} \sum_{k=0}^{K_q} \varphi_{k,q} \lambda_n^{l} e^{-\jmath \omega_t q} \hat{x}_n(e^{\jmath \omega_t}), \quad (7)$$

where $\hat{y}_n(e^{\jmath \omega_t})$ $(\hat{x}_n(e^{\jmath \omega_t}))$ is the nth entry of $\text{GFT}\{\text{DFT}\{\boldsymbol{y}_t\}\}$ $(\text{GFT}\{\text{DFT}\{\boldsymbol{x}_t\}\})$. Then, for stable filter coefficients $\psi_{l,p}$, i.e., coefficients that abide to

$$1 + \sum_{p=1}^{P} \sum_{l=0}^{L_p} \psi_{l,p} \lambda_n^{l} e^{-\jmath \omega_t p} \neq 0, \quad \forall \, n = 1, \ldots, N, \quad (8)$$

(6) implements the two-dimensional time-vertex frequency response

$$h(\lambda_n, e^{\jmath \omega_t}) = \frac{\sum_{q=0}^{Q} \sum_{k=0}^{K_q} \varphi_{k,q} \lambda_n^{k} e^{-\jmath \omega_t q}}{1 + \sum_{p=1}^{P} \sum_{l=0}^{L_p} \psi_{l,p} \lambda_n^{l} e^{-\jmath \omega_t p}}. \qquad (9)$$

It is easy to see that the above frequency response is a joint time-vertex filter. Similarly to its separable counterpart [17], recursion (6) considers polynomial shifting in the graph dimension. Thus, from the locality of $\boldsymbol{L}_{\mathcal{G}}$ [21], and since $\boldsymbol{L}_{\mathcal{G}}^{k} \boldsymbol{x}_{t-q} = \boldsymbol{L}_{\mathcal{G}}(\boldsymbol{L}_{\mathcal{G}}^{k-1} \boldsymbol{x}_{t-q})$ [24], [26], recursion (6) enjoys the same distributed implementation as [17].

We conclude this section with the following observation.

Remark 1 (Model generality). *Recursion (6) encompasses some of the state-of-the-art two-dimensional time-vertex models. Specifically:*
- *for $\psi_{l,p} = \psi_l a_p$ and $\varphi_{p,q} = \varphi_p b_q$ it specializes to the separable case [17];*
- *for $\psi_{l,p} = 0 \; \forall l, p$, it boils down to the two-dimensional finite impulse response time-vertex filter [19];*
- *for $L_p = p$, $\sum_{k=0}^{K_0} \varphi_{k,0} \boldsymbol{L}_{\mathcal{G}}^{k} = \boldsymbol{I}_N$ and $\varphi_{k,q} = 0$ otherwise, it reduces to the model of [8].*

In Section III-B we exploit recursion (6) to model network processes for the task of prediction. This will allow us to exploit its polynomial implementation, and thus provide a distributed predictor.

C. Stationarity on Networks

Stationarity is another useful property that extends to graph signals [13]–[15] and allows performing graph Wiener filtering/interpolation and spectral estimation. From the analogies with the concept of stationarity in time signals, graph stationarity requires that the signal first and second order moments are preserved along the graph dimension. Limiting ourselves to zero-mean processes, one way to define graph wide-sense stationarity (GWSS) is:

Definition 1 (GWSS). *A random graph process $\boldsymbol{x}$ is wide-sense stationary, if and only if its covariance matrix $\boldsymbol{\Sigma}_x = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{H}}]$ is diagonalizable by the GFT basis $\boldsymbol{U}_{\mathcal{G}}$, i.e., $\boldsymbol{\Sigma}_x = \boldsymbol{U}_{\mathcal{G}}\text{diag}(\boldsymbol{p}_x)\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$. The $N \times 1$ vector $\boldsymbol{p}_x$ is the GFT of the*

autocorrelation function of $\boldsymbol{x}$ and is referred to as the graph power spectral density.

This condition imposes the invariance of the second order statistics w.r.t. a translation using the shift operator $\boldsymbol{L}_{\mathcal{G}}$. That is, as for time signals the second order moment is invariant to temporal translations, here, the invariance of the graph signal statistics should also account for the underlying structure. Moreover, similar to classical temporal stationarity, a useful property of GWSS signals is that $\boldsymbol{x}$ can be driven by a graph filter applied to a zero-mean white noise. The graph filter $h(\boldsymbol{\Lambda}_{\mathcal{G}})$ should then satisfy $|h(\boldsymbol{\Lambda}_{\mathcal{G}})|^2 = \mathrm{diag}(\boldsymbol{p}_x)$.

In analogy with the definition of the JFT, stationarity can be extended to time-varying graph processes [16]. This new concept of stationarity requires now that the first and second order moments are jointly preserved along the graph and temporal dimension. Limiting ourselves again to zero-mean processes, joint wide-sense stationarity (JWSS) can be defined as follows:

Definition 2 (JWSS). *A random time-vertex process $\boldsymbol{x}_{1:T} = \mathrm{vec}(\boldsymbol{X}_{1:T})$ is called jointly (or time-vertex) wide-sense stationary, if and only if its covariance matrix $\boldsymbol{\Sigma}_{x_{1:T}} = \mathbb{E}[\boldsymbol{x}_{1:T}\boldsymbol{x}_{1:T}^{\mathsf{H}}]$ is diagonalizable by the JFT basis $\boldsymbol{U}_{\mathcal{J}}$, i.e., $\boldsymbol{\Sigma}_{1:T} = \boldsymbol{U}_{\mathcal{J}}\mathrm{diag}(\boldsymbol{p}_{1:T})\boldsymbol{U}_{\mathcal{J}}^{\mathsf{H}}$. The $NT \times 1$ vector $\boldsymbol{p}_{1:T}$ is the JFT of the autocorrelation function of the process $\boldsymbol{x}_{1:T}$ and is referred to as joint power spectral density (JPSD).*

With this definition, one now assumes simultaneously wide-sense stationarity w.r.t. both the *time* and *vertex* domains. This condition requires now invariance w.r.t. translation using the joint time-vertex shift operator. Further, for a JWSS time-vertex process the covariance is now related to a joint filter $h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})$. Specifically, $\boldsymbol{x}_{1:T}$ can be driven by time-vertex filtering zero-mean white noise, where $h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})$ must now satisfy $|h(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})|^2 = \mathrm{diag}(\boldsymbol{p}_{1:T})$.

III. MODELING OF NETWORK PROCESSES

This section formulates the problem of modeling the temporal evolution of a JWSS network process. We start with a joint non-causal model and highlight its implementation issues. Then, we follow the classical literature of VARMA recursions [3] to propose joint causal models for graph-time series. Finally, the optimal MSE predictor is derived.

A. Joint Non-Causal Models on Graphs

A JWSS process $\boldsymbol{x}_{1:T}$ can be obtained through filtering of a random white process $\boldsymbol{\varepsilon}_{1:T} = \mathrm{vec}(\boldsymbol{E}_{1:T})$ with a joint time-vertex filter (5) (recall Definition 2). Specifically, we can model $\boldsymbol{x}_{1:T}$ with the *joint non-causal model*

$$a(\boldsymbol{L}_{\mathcal{J}})\boldsymbol{x}_{1:T} = b(\boldsymbol{L}_{\mathcal{J}})\boldsymbol{\varepsilon}_{1:T}, \tag{10}$$

where $\boldsymbol{\varepsilon}_{1:T}$ is the innovation vector, i.e., a random vector with zero-mean and identity covariance matrix. The $NT \times NT$ matrices $a(\boldsymbol{L}_{\mathcal{J}}) = \boldsymbol{U}_{\mathcal{J}}a(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})\boldsymbol{U}_{\mathcal{J}}^{\mathsf{H}}$ and $b(\boldsymbol{L}_{\mathcal{J}}) = \boldsymbol{U}_{\mathcal{J}}b(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})\boldsymbol{U}_{\mathcal{J}}^{\mathsf{H}}$ represent arbitrary joint filters. From (10) and the statistics of $\boldsymbol{\varepsilon}_{1:T}$, the JPSD of $\boldsymbol{x}_{1:T}$ is

$$\boldsymbol{p}_{1:T} = \mathrm{diag}\Big(|a(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})|^{-2}|b(\boldsymbol{\Lambda}_{\mathcal{G}}, \boldsymbol{\Lambda}_{\mathcal{T}})|^2\Big) \tag{11}$$

for an invertible matrix model $a(\boldsymbol{L}_{\mathcal{J}})$.

Despite its generality, model (10) presents several issues for the task of prediction. First, its computational complexity results in a heavy task even for moderate values of N and T. Second, (10) is not causal. This is problematic for prediction where one needs to forecast the future in a timely manner from the process history. To accomplish these challenges, next we introduce *joint causal models*, which ease the computational costs and model the JWSS graph process in a causal manner.

B. Joint Causal Models on Graphs

In the following, we introduce two models for forecasting time series on graphs.

Graph-VARMA model. Similar to the multivariate case [3], a joint causal model on graphs has the form

$$\boldsymbol{x}_t = -\sum_{p=1}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\,\boldsymbol{x}_{t-p} + \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\,\boldsymbol{\varepsilon}_{t-q}, \tag{12}$$

where the $N \times N$ matrices $a_p(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}}a_p(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$, $b_0(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{I}_N$, and $b_q(\boldsymbol{L}_{\mathcal{G}}) = \boldsymbol{U}_{\mathcal{G}}b_q(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ are graph filters and $\boldsymbol{\varepsilon}_t$ is a random vector with zero-mean and covariance matrix $\boldsymbol{\Sigma}_{\varepsilon}$.

Following the convention of time series analysis, we call the above model a graph-VARMA (G-VARMA) model. Nevertheless, we also refer to it as a joint causal model because (for sufficiently large P and Q) (12) can describe all causal JWSS processes.

Remark 2 (G-VARMA complexity). *The per-iteration complexity of the G-VARMA model is $O((P + Q)N^2)$ amounting to $P + Q$ matrix-vector multiplications.*

In the sequel, we specialize model (12) with graph filters $a_p(\boldsymbol{L}_{\mathcal{G}})$ and $b_q(\boldsymbol{L}_{\mathcal{G}})$ that have a polynomial structure over the graph. With the aim to implement a distributed predictor, we drop the MA part of (12) and consider graph polynomial-VAR (GP-VAR) models. The rationale beyond the latter choice is that a graph polynomial VARMA model leads in practice to unstable predictors, or in best cases to predictors with larger MSE than GP-VAR. To some degree, this observation is not surprising and goes in line with the conclusions in multivariate modeling [3, Chapter 11], [29, Chapter 2]. Differently, this behaviour is not consistently observed for model (12) due to its fitting procedure in the graph Fourier domain (see Proposition 2 in Section IV-A).

Graph polynomial-VAR model. By exploiting the proposed joint time-vertex filter (6), we write the matrices $a_p(\boldsymbol{L}_{\mathcal{G}})$ and $b_q(\boldsymbol{L}_{\mathcal{G}})$ as polynomials of the graph Laplacian $\boldsymbol{L}_{\mathcal{G}}$ and get the GP-VAR model[4]

$$\boldsymbol{x}_t = -\sum_{p=1}^{P}\sum_{l=0}^{L_p} \psi_{l,p}\boldsymbol{L}_{\mathcal{G}}^l\boldsymbol{x}_{t-p} + \boldsymbol{\varepsilon}_t \tag{13}$$

which is obtained from (12) by setting $a_p(\boldsymbol{L}_{\mathcal{G}}) = \sum_{l=0}^{L_p} \psi_{l,p}\boldsymbol{L}_{\mathcal{G}}^l$ for $p = 1, \ldots, P$ and $Q = 0$. Further, note

---

[4]We denote the approach of [8] as restricted graph polynomial-VAR (RGP-VAR) model.

that the order L_p of the polynomials varies based on the time index.

Model (13) preserves the recursive implementation of (12), thus it reduces further the computational complexity. The subsequent remark quantifies this cost.

Remark 3 (GP-VAR complexity). *The Laplacian polynomial form of (13) allows it to be implemented distributively in the vertex domain with a complexity similar to [17]. By setting the maximum order of the Laplacian polynomial as $L_{max} = max\{L_1, \ldots, L_p\}$, the per-iteration complexity of the GP-VAR model (13) is $O(L_{max}|\mathcal{E}|)$.*

Since in general $|\mathcal{E}| \ll N^2$, the (distributed) per-iteration computational complexity of the GP-VAR model (13) is much smaller than that of the G-VARMA model (12). Compared to the non-causal model (10), if we arrest the recursion after T iterations, the computational complexity of the GP-VAR is $O(L_{\max}|\mathcal{E}|T)$, while that of the G-VARMA is $O((P + Q)N^2T)$. Moreover, w.r.t. the RGP-VAR model of [8], (13) allows the polynomial order in the graph domain L_p to differ for each p and not to be restricted as $L_p \leq p$. The latter allows capturing more hidden interactions between nodes and, therefore, yields better predictors. In Section IV-B, we will see that this polynomial implementation simplifies the model design, to that of finding scalar coefficients rather than matrices as in (12).

C. Optimal predictor

Let $\boldsymbol{X}_{1:T}$ be the output of the more general G-VARMA model (12). We consider the realizations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ are observed, and the goal is to estimate $\boldsymbol{x}_t$ from these values. Following the conventional approach [3], the value of $\boldsymbol{x}_t$ is estimated as the conditional expectation

$$
\begin{aligned}
\tilde{\boldsymbol{x}}_t = \mathbb{E}[\boldsymbol{x}_{t|t-1}] &\triangleq \mathbb{E}\Big[\boldsymbol{x}_t | \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}\Big] \\
&= -\sum_{p=1}^{P} a_p(\boldsymbol{L}_\mathcal{G})\,\boldsymbol{x}_{t-p} + \sum_{q=0}^{Q} b_q(\boldsymbol{L}_\mathcal{G})\mathbb{E}[\boldsymbol{\varepsilon}_{t-q|t-1}],
\end{aligned}
\tag{14}
$$

i.e., the expected value of $\boldsymbol{x}_t$ given the past realizations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$. Similarly, $\mathbb{E}[\boldsymbol{\varepsilon}_{t-q|t-1}]$ is the conditional expectation of $\boldsymbol{\varepsilon}_{t-q}$ given $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$. Note that, since the past realizations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ follow model (12), the knowledge of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}\}$ allows computing the value of the *specific realizations* for the past innovations $\{\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_{t-1}\}$ deterministically as $\boldsymbol{\varepsilon}_{t-q|t-1} = \boldsymbol{x}_{t-q|t-1} - \tilde{\boldsymbol{x}}_{t-q}$ for $q = 1, \ldots, Q$ (recall $b_0(\boldsymbol{L}_\mathcal{G}) = \boldsymbol{I}_N$). So, we can rewrite (14) as

$$
\begin{aligned}
\tilde{\boldsymbol{x}}_t = &-\sum_{p=1}^{P} a_p(\boldsymbol{L}_\mathcal{G})\boldsymbol{x}_{t-p} + \boldsymbol{b}_0(\boldsymbol{L}_\mathcal{G})\mathbb{E}[\boldsymbol{\varepsilon}_t] \\
&+ \sum_{q=1}^{Q} b_q(\boldsymbol{L}_\mathcal{G})\left(\boldsymbol{x}_{t-q|t-1} - \tilde{\boldsymbol{x}}_{t-q}\right) \\
= &-\sum_{p=1}^{P} a_p(\boldsymbol{L}_\mathcal{G})\,\boldsymbol{x}_{t-p} + \sum_{q=1}^{Q} b_q(\boldsymbol{L}_\mathcal{G})\left(\boldsymbol{x}_{t-q|t-1} - \tilde{\boldsymbol{x}}_{t-q}\right).
\end{aligned}
\tag{15}
$$

where in the last equality we used $\mathbb{E}[\boldsymbol{\varepsilon}_t] = \boldsymbol{0}_N$. We refer to $\tilde{\boldsymbol{x}}_t$ as the *one-step ahead predictor*. The k−step ahead predictor can be obtained by repeating the above computation k times.

The one-step ahead predictor for the GP-VAR model (13) can be obtained from (15) by setting $a_p(\boldsymbol{L}_\mathcal{G}) = \sum_{l=0}^{L_p} \psi_{l,p}\boldsymbol{L}_\mathcal{G}^l$ for $p = 1, \ldots, P$ and $Q = 0$. In the sequel, we show that (15) yields the optimal decision from an MSE perspective.

D. Mean Square Error Analysis

We now illustrate the impact of the predictor parameters $a_p(\boldsymbol{L}_\mathcal{G})$ and $b_q(\boldsymbol{L}_\mathcal{G})$ on the MSE of the one-step ahead predictor (15).

The main result is summarized in the following proposition.

Proposition 1. *Let $\boldsymbol{x}_t$ be the output of the joint causal model (12) and let $a_p(\boldsymbol{L}_\mathcal{G})$ and $b_q(\boldsymbol{L}_\mathcal{G})$ be a given set of model parameters. Then, the MSE of the one-step ahead predictor (15) is*

$$
MSE = \mathbb{E}\left[\|b_0(\boldsymbol{L}_\mathcal{G})\boldsymbol{\varepsilon}_t\|_2^2\right] = tr\left(\boldsymbol{\Sigma}_\varepsilon\right),
\tag{16}
$$

which corresponds to the smallest achievable MSE.

(The proof is given in the Appendix.)

IV. MODEL FITTING: PARAMETER DESIGN

In this section, we focus on identifying the model parameters, i.e., the kernel matrices $a_p(\boldsymbol{L}_\mathcal{G})$ and $b_q(\boldsymbol{L}_\mathcal{G})$ for the G-VARMA, or the scalars $\psi_{l,p}$ for the GP-VAR, that fit process $\boldsymbol{X}_{1:T}$. We start with the more complex G-VARMA case and introduce a GFT-based decoupling approach to ease the computational burden. To further reduce the parameter estimation costs, we introduce a low-rank model which trades prediction accuracy with computational complexity. Then, we consider the GP-VAR predictor and show that the scalar parameters can be estimated from the graph process statistics. The section is concluded with the notion of joint frequency smoothing and its application to parameter design.

A. G-VARMA fitting

Given the realizations $\boldsymbol{X}_{1:T}$, the canonical way for estimating the kernel matrices $a_p(\boldsymbol{L}_\mathcal{G})$ and $b_q(\boldsymbol{L}_\mathcal{G})$ is by solving

$$
\underset{a_p(\boldsymbol{L}_\mathcal{G}),b_q(\boldsymbol{L}_\mathcal{G})}{\text{minimize}} \sum_{\tau=\max\{P,Q\}}^{T-1} \left\| \boldsymbol{x}_\tau - \tilde{\boldsymbol{x}}_\tau[a_p(\boldsymbol{L}_\mathcal{G}), b_q(\boldsymbol{L}_\mathcal{G})] \right\|_2^2.
\tag{17}
$$

Problem (17) involves a non-linear system of $N \times (T - 1 - \max\{P,Q\})$ equations with $(P + Q)N$ unknowns whose computational complexity can become prohibitive even for moderate values of N and T. To alleviate these costs, the following proposition establishes a decoupling approach that breaks down problem (17) into N uncorrelated (yet not equivalent) well-studied problems with smaller complexity.

Proposition 2 (Decoupling). *Consider the joint causal model (12)*

$$
\boldsymbol{x}_t = -\sum_{p=1}^{P} a_p(\boldsymbol{L}_\mathcal{G})\,\boldsymbol{x}_{t-p} + \sum_{q=0}^{Q} b_q(\boldsymbol{L}_\mathcal{G})\,\boldsymbol{\varepsilon}_{t-q},
$$

and denote with $\hat{\varepsilon}_{t,n} = [\boldsymbol{U}_\mathcal{G}^\mathsf{H} \boldsymbol{\varepsilon}_t]_n$ and $\hat{x}_{t,n} = [\boldsymbol{U}_\mathcal{G}^\mathsf{H} \boldsymbol{x}_t]_n$ the nth GFT coefficient of $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{x}_t$, respectively. Then, the input-output relation between $\hat{\varepsilon}_{t,n}$ and $\hat{x}_{t,n}$ is given by the ARMA recursion

$$\hat{x}_{t,n} = -\sum_{p=1}^P [a_p(\boldsymbol{\Lambda}_\mathcal{G})]_{n,n}\, \hat{x}_{t-p,n} + \sum_{q=0}^Q [b_q(\boldsymbol{\Lambda}_\mathcal{G})]_{n,n}\, \hat{\varepsilon}_{t-q,n}, \tag{18}$$

with $[b_0(\boldsymbol{\Lambda}_\mathcal{G})]_{n,n} = 1 \; \forall n$ and $n = 1, \ldots, N$.

(The proof is given in the Appendix.)

Although simple in its derivation, Proposition 2 relates different aspects of the graph and the time series with the model fitting. First, it uses the GFT eigenbasis to formulate (12) in the graph Laplacian eigenspace[5]. Second, the model fitting is performed in the graph frequency domain. Here, the parameter estimation is split into N uncorrelated problems of a smaller complexity involving T equations and $P + Q$ unknowns. Despite being non-linear problems, the estimation for each graph frequency time series results in fitting a univariate temporal ARMA. So, we can use well-studied methods to solve it, such as the Gauss-Newton approach [30].

Remark 4 (Estimation cost). *In our analysis, the eigendecomposition of the graph Laplacian $\boldsymbol{L}_\mathcal{G}$ is crucial to estimate the G-VARMA parameters. Therefore, the proposed framework suits better small to medium-sized graphs, where the eigenvalue decomposition cost (inherent in joint models), is overshadowed by that of the model estimation. Nevertheless, next, we introduce a low-rank model estimation that reduces the estimation complexity at the expense of fitting accuracy.*

Low-rank models. To further reduce the model estimation cost, we can consider estimating $a_p(\boldsymbol{L}_\mathcal{G})$ and $b_q(\boldsymbol{L}_\mathcal{G})$ only from a subset of graph frequencies, say $K \leq N$. The latter incurs considerable savings in terms of estimation complexity since (i) there is no need to compute the full eigendecomposition of $\boldsymbol{L}_\mathcal{G}$, but only the eigenvectors relative to the chosen K graph frequencies; and (ii) we only need to fit $K \leq N$ temporal ARMA time series to the data. For this purpose, we use the following definition of low-rank matrix approximation.

Definition 3 (Low-rank approximation). *Let $\mathcal{S}$ be an index set of cardinality $K = |\mathcal{S}|$, with indicator matrix $\boldsymbol{D}_\mathcal{S}$. Let also $\boldsymbol{U}$ be a unitary (rotation) matrix. The $\{\boldsymbol{U}, \mathcal{S}\}$ low-rank approximation of a matrix $\boldsymbol{X}_{1:T}$ is*

$$\tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}} = \boldsymbol{U}\boldsymbol{D}_\mathcal{S}\boldsymbol{U}^\mathsf{H}\boldsymbol{X}_{1:T}. \tag{19}$$

For prediction, which is an online task, the above low-rank approximation should be done only w.r.t. the graph dimension. The following theorem provides a constructive way to design the set $\mathcal{S}$ that achieves the *optimal low-rank* approximation for a JWSS process.

[5]Recall that for JWSS graph processes the eigenvectors $\boldsymbol{U}_\mathcal{G}$ of $\boldsymbol{L}_\mathcal{G}$ coincide with the eigenvectors of $\boldsymbol{\Sigma}_x$.

Theorem 1. *Let $\boldsymbol{X}_{1:T}$ be a zero-mean JWSS process. The optimal K-rank approximation of $\boldsymbol{X}_{1:T}$ is given by*

$$\{\boldsymbol{U}_\mathcal{G}, \mathcal{S}^\star\} = \arg\min_{\boldsymbol{U},\mathcal{S}} \mathbb{E}\left[\left\|\boldsymbol{X} - \tilde{\boldsymbol{X}}_{\boldsymbol{U},\mathcal{S}}\right\|_F^2\right] \quad s.t. \quad |\mathcal{S}| = K,$$

where $\mathcal{S}^\star$ contains the indices of the K largest diagonal elements of $\boldsymbol{U}_\mathcal{G}^\mathsf{H}\boldsymbol{\Sigma}_x\boldsymbol{U}_\mathcal{G}$.

(The proof is given in the Appendix.)

This result suggests that the best rank$-K$ approximation of $\boldsymbol{X}_{1:T}$ is obtained by rotating the JWSS process by the graph Laplacian eigenvectors. This finding is beneficial since the first step in the model estimation consists of decoupling the time series using the GFT. It further allows us to claim the following:

Corollary 1. *Let $\boldsymbol{U}_{\mathcal{G},K}$ denote the $N \times K$ matrix containing the K columns of $\boldsymbol{U}_\mathcal{G}$ relative to the K highest eigenvalues of $\boldsymbol{\Sigma}_x$. Then, $\boldsymbol{U}_{\mathcal{G},K}^\mathsf{H}\boldsymbol{\Sigma}_x\boldsymbol{U}_{\mathcal{G},K}$ leads to the same result as selecting the highest K diagonal elements of $\boldsymbol{U}_\mathcal{G}^\mathsf{H}\boldsymbol{\Sigma}_x\boldsymbol{U}_\mathcal{G}$.*

An important outcome of the last two results is that the full eigendecomposition of $\boldsymbol{L}_\mathcal{G}$ is not needed. Indeed, only the eigenvectors relative to the K largest eigenvalues of $\boldsymbol{\Sigma}_x$ are needed (recall that $\boldsymbol{U}_\mathcal{G}$ jointly diagonalizes both $\boldsymbol{L}_\mathcal{G}$ and $\boldsymbol{\Sigma}_x$), which can be obtained with a lower cost. Therefore, the low-rank representation can be combined with the model estimation, by only modeling the time series for the graph frequencies in $\mathcal{S}$. In this way, we attain a reduction in the model estimation cost of N/K allowing the G-VARMA to cope well with largely-sized graphs. Later in Section VI-D, we will see that real data enjoy this low-rank representation with $K \ll N$.

B. GP-VAR fitting

The parameter estimation for the GP-VAR model (13) consists of estimating the scalars $\psi_{l,p}$. Since this problem has a lower complexity, we find the $\psi_{l,p}$ coefficients by directly minimizing the prediction MSE

$$\text{MSE} = \mathbb{E}\left[\left\|\boldsymbol{x}_t + \sum_{p=1}^P \boldsymbol{\Psi}_p\boldsymbol{x}_{t-p}\right\|_2^2\right], \tag{21}$$

with $\boldsymbol{\Psi}_p = \sum_{l=0}^{L_p} \psi_{l,p}\boldsymbol{L}_\mathcal{G}^l$. From $\|\boldsymbol{x}\|_2^2 = \text{tr}(\boldsymbol{x}\boldsymbol{x}^\mathsf{H})$ and the linearity of the trace and the expectation, (21) becomes

$$\text{MSE} = \text{tr}\left(\sum_{p=1}^P \boldsymbol{\Psi}_p\mathbb{E}\left[\boldsymbol{x}_{t-p}\boldsymbol{x}_t^\mathsf{H}\right]\right) + \text{tr}\left(\sum_{p=1}^P \mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t-p}^\mathsf{H}\right]\boldsymbol{\Psi}_p^\mathsf{H}\right)$$
$$+ \text{tr}\left(\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^\mathsf{H}\right]\right) + \text{tr}\left(\sum_{p_1=1}^P \sum_{p_2=1}^P \boldsymbol{\Psi}_{p_1}\mathbb{E}\left[\boldsymbol{x}_{t-p_1}\boldsymbol{x}_{t-p_2}^\mathsf{H}\right]\boldsymbol{\Psi}_{p_2}^\mathsf{H}\right). \tag{22}$$

Then, since $\boldsymbol{R}_x(i) = \mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_{t-i}]$ is the autocorrelation of the process at lag i, the GP-VAR coefficients can be found by solving the convex problem (20).

From (21)-(20), the graph topology imposes a structure on the model coefficients. In fact, this, somehow small, modification renders the model parameter estimation a computationally

easier task compared to the classical VAR model. Thus, more robust to cope with larger multivariate dimensions. The GP-VAR model inherits several benefits from the classical literature of multivariate VAR analysis [3]. In (20), we rely on the autocorrelation matrix to estimate the coefficients, which must be estimated from the training data $\boldsymbol{X}_{1:T}$, i.e., $\boldsymbol{R}_x(i) = (T-i)^{-1}\sum_{\tau=0}^{T-i}\boldsymbol{x}_\tau\boldsymbol{x}_{\tau+i}^{\mathsf{H}}$. Differently from [9]–[12] that do not exploit stationary assumptions, in the GP-VAR model we can incorporate well-established techniques for estimating $\boldsymbol{R}_x(i)$, such as the shrinkage estimators [7] and the random matrix theory-based estimator [31] in low-samples regime, or the Kullback-Leiber divergence [32] and Tyler's estimator [33] to deal with heavy tail issues in non-Gaussian scenarios.

Remark 5 (Yule-Walker estimation). *An alternative to* (20) *for estimating $\psi_{l,p}$ is to follow a Yule-Walker approach [34]. That is, minimize in a least squares sense the function $f(\psi_{l,p}) = \boldsymbol{R}_x(0) + \sum_{p=1}^{P}\boldsymbol{\Psi}_p\boldsymbol{R}_x(p)$ with $\boldsymbol{\Psi}_p = \sum_{l=0}^{L_p}\psi_{l,p}\boldsymbol{L}_{\mathcal{G}}^l$. Besides being a computationally lighter problem to solve, the Yule-Walker strategy can be combined with the approach in [35] and, thus, estimate the $\psi_{l,p}$ from subsampled measurements.*

C. Spectral Smoothing

We can further improve the model fitting by using the spectral smoothing heuristic [16, Section IV. B]. This involves the convolution of the JPSD with a smoothly decaying windowed function in the spectral domain. Specifically, if $\boldsymbol{P}_{1:T} = |\mathrm{JFT}\{\boldsymbol{X}_{1:T}\}|^2$ is the JPSD in matrix form, we fit the models to the process having a final JPSD

$$[\tilde{\boldsymbol{P}}_{1:T}]_{\lambda,e^{-\jmath\omega}} = \sum_{n,t} g_{\mathcal{G}}(\lambda-\lambda_n)^2 g_{\mathcal{T}}(e^{-\jmath\omega}-e^{-\jmath\omega_t})^2[\boldsymbol{P}_{1:T}]_{\lambda_n,e^{-\jmath\omega_t}},$$
(23)

where $g_{\mathcal{G}}$ and $g_{\mathcal{T}}$ are smoothly decaying functions in the graph and temporal and frequency domains, respectively. $g_{\mathcal{G}}$ and $g_{\mathcal{T}}$ are scaled Gaussian functions centered at zero and normalized to sum to one. The Gaussian width is a free parameter that controls the estimator's bias-variance trade-off [16].

The idea of spectral smoothing is common in time series analysis. In fact, for stationary time series, spectral smoothing (i.e., the convolution with $g_{\mathcal{T}}$ along each node) is a technique encountered in the literature (see [36], [37]). For stationary graph signals, spectral smoothing w.r.t. the graph Fourier domain (i.e., the convolution with $g_{\mathcal{G}}$ along each t) is used to better estimate stationary graph signals [15], [38]. This heuristic also relates to the spectral leakage phenomenon in eigenvalue estimation [39], exploiting the locality of spectral leaking to better estimate the eigenvalues of the covariance matrix from few samples.
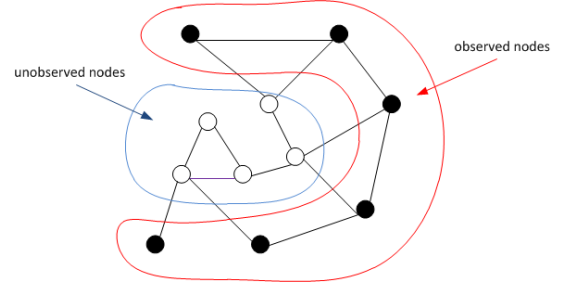

Fig. 1: Illustration of the sampling set and its complementary set. The black nodes are observed at time t, while the white nodes are not accessed. At previous time instants, all nodes are observed.

With respect to this work, (23) relates the joint stationarity of the signal with the challenge of estimating the model parameters in a low sample regime. This heuristic exploits the fact that the signal JPSD is likely to have similar values in adjacent frequencies and that abrupt changes are more likely to occur due to noise and finite sample estimation inaccuracies. Therefore, by exploiting the latter, a model fitting to data with a JPSD as in (23) will often lead to a lower estimation error.

V. TRACKING ON A SUBSET OF NODES

This section looks at estimating $\boldsymbol{x}_t$ by using both the historical data and measurements at time t collected on a subset $\mathcal{S}_t \subseteq \mathcal{V}$ of nodes. In what follows, we consider the G-VARMA model to address this task. At the end of this section, we remark how the tracking can be addressed by other models.

A possible application involving this scenario comprises a survey outcome in a social network. Here, it is likely that a (possibly significant) number of users do not take part in the survey after a certain time instant. By using the developed predictors and the observations on $\mathcal{S}_t$, we are interested to estimate the survey outcome at time t on another set $\bar{\mathcal{S}}_t$. The set $\bar{\mathcal{S}}_t$ might then be: (i) $\bar{\mathcal{S}}_t = \mathcal{V}$, i.e., improve the estimation on all nodes; (ii) $\bar{\mathcal{S}}_t = \mathcal{V}\backslash\mathcal{S}_t$, i.e., the complementary set of $\mathcal{S}_t$; or (iii) a combination on the two. Fig. 1 illustrates one such example.

Let $\boldsymbol{D}_t \in \{0,1\}^{|\mathcal{S}_t|\times N}$ be a binary matrix containing the non-zero rows of $\boldsymbol{D}_{\mathcal{S}_t} = \mathrm{diag}(\boldsymbol{1}_{\mathcal{S}_t})$. Note that by construction $\boldsymbol{D}_t\boldsymbol{D}_t^{\top} = \boldsymbol{I}_{|\mathcal{S}_t|}$ and $\boldsymbol{D}_t^{\top}\boldsymbol{D}_t = \boldsymbol{D}_{\mathcal{S}_t}$. Similarly, let $\bar{\boldsymbol{D}}_t$ be the $|\bar{\mathcal{S}}_t|\times N$ binary matrix relative to the set $\bar{\mathcal{S}}_t$. Let also $\boldsymbol{z}_t = \boldsymbol{D}_t(\boldsymbol{x}_t + \boldsymbol{w}_t)$ be the $|\mathcal{S}_t|\times 1$ vector of measurements at time t with $\boldsymbol{w}_t$ the zero-mean measurement noise with covariance matrix $\boldsymbol{\Sigma}_w$ independent from $\boldsymbol{\varepsilon}_t$. Then, for $\boldsymbol{x}_t$ following the G-VARMA model (12), we have

$$\boldsymbol{z}_t = \boldsymbol{D}_t\boldsymbol{\varepsilon}_t + \boldsymbol{D}_t\tilde{\boldsymbol{x}}_t + \boldsymbol{D}_t\boldsymbol{w}_t,$$
(24)

$$\begin{aligned}
\underset{\{\psi_{l,p}\}}{\text{minimize}} \quad & \mathrm{tr}\left(\boldsymbol{R}_x(0) + \sum_{p=1}^{P}\boldsymbol{\Psi}_p\boldsymbol{R}_x(p) + \sum_{p=1}^{P}\boldsymbol{R}_x(p)\boldsymbol{\Psi}_p^{\mathsf{H}} + \sum_{p_1=1}^{P}\sum_{p_2=1}^{P}\boldsymbol{\Psi}_{p_1}\boldsymbol{R}_x(p_2-p_1)\boldsymbol{\Psi}_{p_2}^{\mathsf{H}}\right) \\
\text{subject to} \quad & \boldsymbol{\Psi}_p = \sum_{l=0}^{L_p}\psi_{l,p}\boldsymbol{L}_{\mathcal{G}}^l.
\end{aligned}$$
(20)

with $\tilde{\boldsymbol{x}}_t$ from (15). The linear minimum mean square estimate (LMMSE) for $\boldsymbol{\varepsilon}_t$ is

$$\tilde{\boldsymbol{\varepsilon}}_t = \boldsymbol{\Sigma}_\varepsilon \boldsymbol{D}_t^\top \boldsymbol{H}_t^{-1}(\boldsymbol{z}_t - \boldsymbol{D}_t \tilde{\boldsymbol{x}}_t), \qquad (25)$$

where $\boldsymbol{H}_t = \boldsymbol{D}_t\left(\boldsymbol{\Sigma}_\varepsilon + \boldsymbol{\Sigma}_w\right)\boldsymbol{D}_t^\top$ [40]. Substituting $\tilde{\boldsymbol{\varepsilon}}_t$ into the signal evolution over the set $\bar{\mathcal{S}}_t$, we get

$$\tilde{\boldsymbol{x}}_{\bar{\mathcal{S}},t} = \bar{\boldsymbol{D}}_t\Big(\tilde{\boldsymbol{x}}_t + \boldsymbol{\Sigma}_\varepsilon \boldsymbol{D}_t^\top \boldsymbol{H}_t^{-1}(\boldsymbol{z}_t - \boldsymbol{D}_t \tilde{\boldsymbol{x}}_t)\Big). \qquad (26)$$

This is the estimate of $\boldsymbol{x}_t$ on $\bar{\mathcal{S}}_t$ which considers the G-VARMA prediction $\tilde{\boldsymbol{x}}_t$ and the measurements collected over $\mathcal{S}_t$. Estimator (26) considers, in addition to its past realizations, i.e., $\bar{\boldsymbol{D}}_t\tilde{\boldsymbol{x}}_t$, an inverse filtering on the observed vertices and then a spreading onto the set $\bar{\mathcal{S}}_t$.

Following the KF convention [41], $\tilde{\boldsymbol{x}}_{\bar{\mathcal{S}},t}$ in (26) is the *a posteriori* estimate of $\boldsymbol{x}_t$ on the set of interest $\bar{\mathcal{S}}_t$ and $\tilde{\boldsymbol{x}}_t$ is the *a priori* estimate of $\boldsymbol{x}_t$. The latter allows then to easily quantify in closed form the MSE of estimator (26). In fact, given the *a priori* error covariance matrix over $\mathcal{V}$, $\boldsymbol{P}_t^- = \boldsymbol{\Sigma}_\varepsilon$, and the *a posteriori* error covariance matrix over $\bar{\mathcal{S}}_t$

$$\boldsymbol{P}_{\bar{\mathcal{S}},t}^+ = \bar{\boldsymbol{D}}_t(\boldsymbol{I}_N - \boldsymbol{K}_t\boldsymbol{D}_t)\boldsymbol{P}_t^-\bar{\boldsymbol{D}}_t^\top, \qquad (27)$$

with Kalman gain matrix

$$\boldsymbol{K}_t = \boldsymbol{P}_t^- \boldsymbol{D}_t^\top \boldsymbol{H}_t^{-1}, \qquad (28)$$

the MSE of (26) is

$$\begin{aligned} MSE &= \mathbb{E}\Big[\|\boldsymbol{x}_{\bar{\mathcal{S}},t} - \bar{\boldsymbol{D}}_t\boldsymbol{x}_t\|_2^2\Big] \\ &= \mathrm{tr}\Big(\bar{\boldsymbol{D}}_t(\boldsymbol{I}_N - \boldsymbol{\Sigma}_\varepsilon\boldsymbol{D}_t^\top\boldsymbol{H}_t^{-1}\boldsymbol{D}_t)\boldsymbol{\Sigma}_\varepsilon\bar{\boldsymbol{D}}_t^\top\Big), \end{aligned} \qquad (29)$$

with $\boldsymbol{H}_t$ defined in (25).

The MSE (29) shows the dependence of the reconstruction performance on the sampling set (through $\boldsymbol{D}_t$) and the G-VARMA model (through the residual error $\boldsymbol{\Sigma}_\varepsilon$). To select the sampling set $\mathcal{S}_t$, we can exploit the above KF reformulation of (26) and use the sparse-sensing sampling strategies [42] adopted in [18] for tracking diffusion processes on graphs. However, as found in the latter work, the MSE would improve by 1dB-2dB w.r.t. a random uniform sampling. While the latter can be used (at the expense of a higher complexity), in this work we show the potential of the proposed tracker by building $\mathcal{S}_t$ uniformly at random.

Remark 6 (Other models). *As the KF interpretation high-lighted, estimator (26) uses $\boldsymbol{P}_t^- = \boldsymbol{\Sigma}_\varepsilon$ of G-VARMA to connect the prediction step with the tracking from $\boldsymbol{z}_t$. Therefore, if we are interested to use another predictive model, the shown KF reformulation can be adopted to incorporate the a priori predictor of $\boldsymbol{x}_t$ into the measurements $\boldsymbol{z}_t$. For the GP-VAR, for instance, the $\boldsymbol{P}_t^-$ can be derived from (22).*

VI. NUMERICAL RESULTS

This section tests the proposed models in four different scenarios: the Molene weather data set, the NOAA U. S. temperature data set, a walking dog dynamic mesh, and a simulated susceptible-infected (SI) epidemic spreading over a flight network.
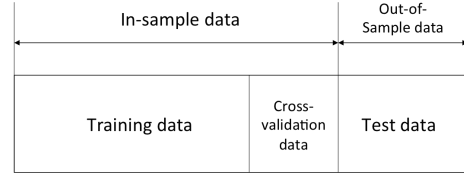

Fig. 2: Model fitting process [43].

The predictors in Section III-B are compared with: (i) the disjoint univariate ARMA that predicts each time series per node; (ii) the standard VAR [3]; (iii) the RGP-VAR from [8]; (iv) the LMS on graphs algorithm from [9]; and (v) the RLS on graphs algorithm from [10]. For the LMS and RLS on graphs algorithms, we considered samples collected on all nodes. The tracking performance of the approach introduced in Section V is compared with the Wiener inpainting [15] (applied at each time instant) and the kernel Kalman filter (KKF) [12]. For the KKF, we considered subsampled measurements only at time t.

For all models, we find the respective parameters by cross-validation, as discussed in [43]. With reference to Fig. 2, the time series is split along the temporal dimension into two parts: the in-sample data and the out-of-sample (or testing) data. The in-sample data are further split into a training and a validation set. The estimation procedure starts with first fitting each model with different model parameters (e.g., different values for P and Q in the G-VARMA predictor) to the training data and measures their performance on the validation set. Then, the selected parameters are the one that yield the lowest root normalized MSE (rNMSE) defined as

$$\mathrm{rNMSE} = \sqrt{\frac{\sum_{t=1}^\tau \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\|_2^2}{\sum_{t=1}^\tau \|\boldsymbol{\theta}_t\|_2^2}}, \qquad (30)$$

for some unknowns $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau$ and respective estimates $\tilde{\boldsymbol{\theta}}_1, \ldots, \tilde{\boldsymbol{\theta}}_\tau$. Finally, the model is refitted with the selected parameters to the entire in-sample data and tested on the out-of-sample data.

We indicate with σ_{g} the Gaussian width used to perform the spectral smoothing along the graph dimension [cf. (23)] and with γ the regularization parameter that determines a bias-variance trade-off in finding the ARMA coefficients along the temporal dimension[6] [44], [45]. In the simulations, we made use of the GSP toolbox [46].

A. Considered scenarios and data pre-processing

In the following, we discuss the four datasets used in our evaluation and provide details for graph construction and preprocessing.

Molene data set. The Molene data set contains hourly temperature measurements of $N = 32$ weather stations near Brest (France) for $T = 744$ hours. We consider a geometric graph (illustrated in Fig. 3 (a)) built from the node coordinates using the default nearest neighbor approach of [46]. This graph

---

[6]This is a regularization parameter used by the Matlab function armax.m that penalizes the ℓ_2 norm of the ARMA coefficients.
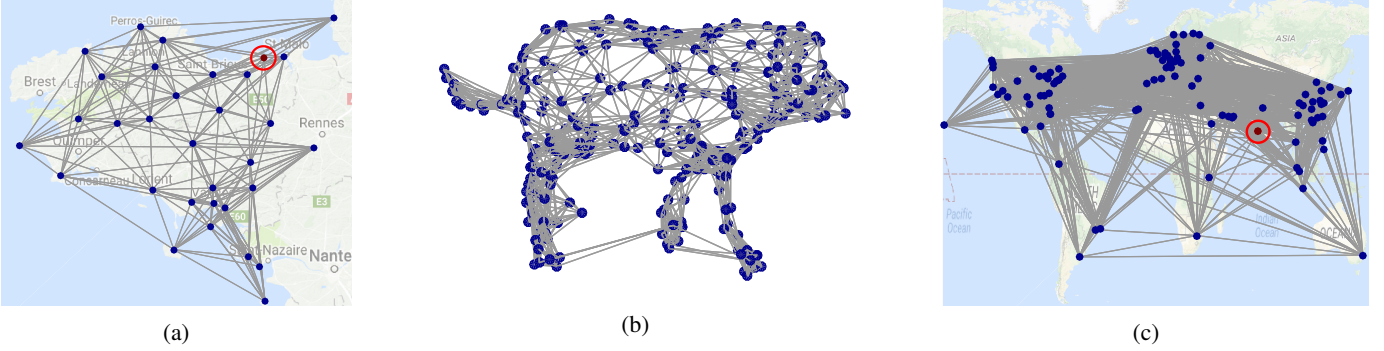
Fig. 3: Graph topologies for the considered scenarios. (a) Molene graph. The vertex in red is the one selected to illustrate the prediction performance. (b) Dog mesh graph. In building the graph, we considered unknown the manifold structure. Thus, we can see edges connecting nodes in different legs that are close by each-other (in the time step on which the graph was constructed). (c) Airline connectivity graph. The nodes in blue are in susceptible (S) state, while the random picked node in red (Mumbai) is in the infected (I) state.

follows a $10-$NN construction and the weight of the edge (i, j) follows the Gaussian kernel

$$[\boldsymbol{W}_{\mathcal{G}}]_{ij} = \exp(-\operatorname{dist}(i, j)/\overline{\operatorname{dist}})$$

where $\operatorname{dist}(i, j)$ is the Euclidean distance between stations i and j and $\overline{\operatorname{dist}}$ is the average Euclidean distance of all sensors. The red station is used later on to illustrate the temperature prediction. The Laplacian matrix $\boldsymbol{L}_{\mathcal{G}}$ is normalized to have a unitary spectral norm. Before the fitting process, the in-sample mean is subtracted from the raw data.

NOAA data set. This is another temperature data set and comprises of hourly temperature recordings at $N = 109$ stations across the United States in 2010 [20] for a total of $T = 8759$ hours. This data set has been used by four graph-based alternative techniques that we compare with, respectively the RGP-VAR [8], the LMS and RLS on graphs [10], and by the KKF [12]. We use the same graph structure as [10] and [12], which is built following the approach of [8] and relies on the $7-$NN geographical distances. The combinatorial Laplacian $\boldsymbol{L}_{\mathcal{G}}$ is used to represent the graph connectivity and the in-sample mean is subtracted from the raw data before preprocessing.

Walking dog mesh. We start from the 3D coordinates of a mesh depicting a walking dog [47]. We aim at predicting the average point position along the three coordinates. The mesh has $N = 251$ points (nodes) over $T = 59$ time steps. The graph follows a 10-NN construction built from the coordinates at $t = 1$ and is illustrated in Fig. 3 (b). This mesh is sparser than the one in [28] (where this data set is used to remove noise) to reduce the computational time. As a preprocessing step, we again subtract the in-sample mean from the data and, as normalization, we divide by the largest absolute value.

SI epidemic diffusion. This scenario considers an epidemic diffusion following the susceptible-infected model [48] over $N = 100$ international airports. The graph, depicted in Fig. 3 (c), captures the airline connections with edge weights $[\boldsymbol{W}_{\mathcal{G}}]_{i,j} = [\boldsymbol{W}_{\mathcal{G}}]_{j,i} = 1$ if there is a flight connection between two airports and zero otherwise. The graph has 3565 edges and an average degree of 35.4. $\boldsymbol{L}_{\mathcal{G}}$ is normalized to have a

unitary spectral norm and the in-sample mean is subtracted from the data.

The time-varying process of interest is the daily evolution of the infection chance of each node (airport). We select an infection rate of 10^{-3} and consider a small fixed population occupying each airport (60 targets). The infected targets are assumed to return to the susceptible state after 12 days and the overall epidemic diffusion is analyzed for $T = 122$ days. In the initial state, all targets at the blue nodes are in the susceptible-state (S-state), while all targets at the red node are in the infected-state (I-state).

B. Prediction performance

We first compare the predictive power of the different algorithms. For all scenarios, the training, cross-validation, and test data consist respectively of 35%, 15%, and 50%. The parameter fitting criterion is the average of the rNMSE (30) over the five-step ahead prediction. The results for the weather data sets (Molene and NOAA) are shown in Fig. 4, while those of the waling dog and epidemic diffusion in Fig. 5.

Molene data set. Fig. 4a shows the prediction rNMSE up to five future steps (hours) for all methods. We see that the proposed G-VARMA and GP-VAR models achieve the best performance over other approaches, although the error is small only for predictions of up to three hours. The GP-VAR falls sightly behind the G-VARMA in terms of accuracy due its more constrained form (localized graph filters).

Fig. 4b depicts the true out-of-sample signal and the one and three-steps ahead prediction of the G-VARMA model at the red sensor in Fig. 3 (a). We see that the one-step ahead prediction matches well the true signal with a deviation lower than 0.5 degrees, while the three-steps ahead prediction is characterized by a larger deviation (up to 3 degrees Celsius).

NOAA data set. Fig. 4c shows the prediction power of the different algorithms in this scenario. We first remark that the training of standard VAR crashed continuously due to the high number of parameters involved. The graph-based techniques on the other hand, lead to trainable models due to the reduced number of parameters. These results show that the proposed
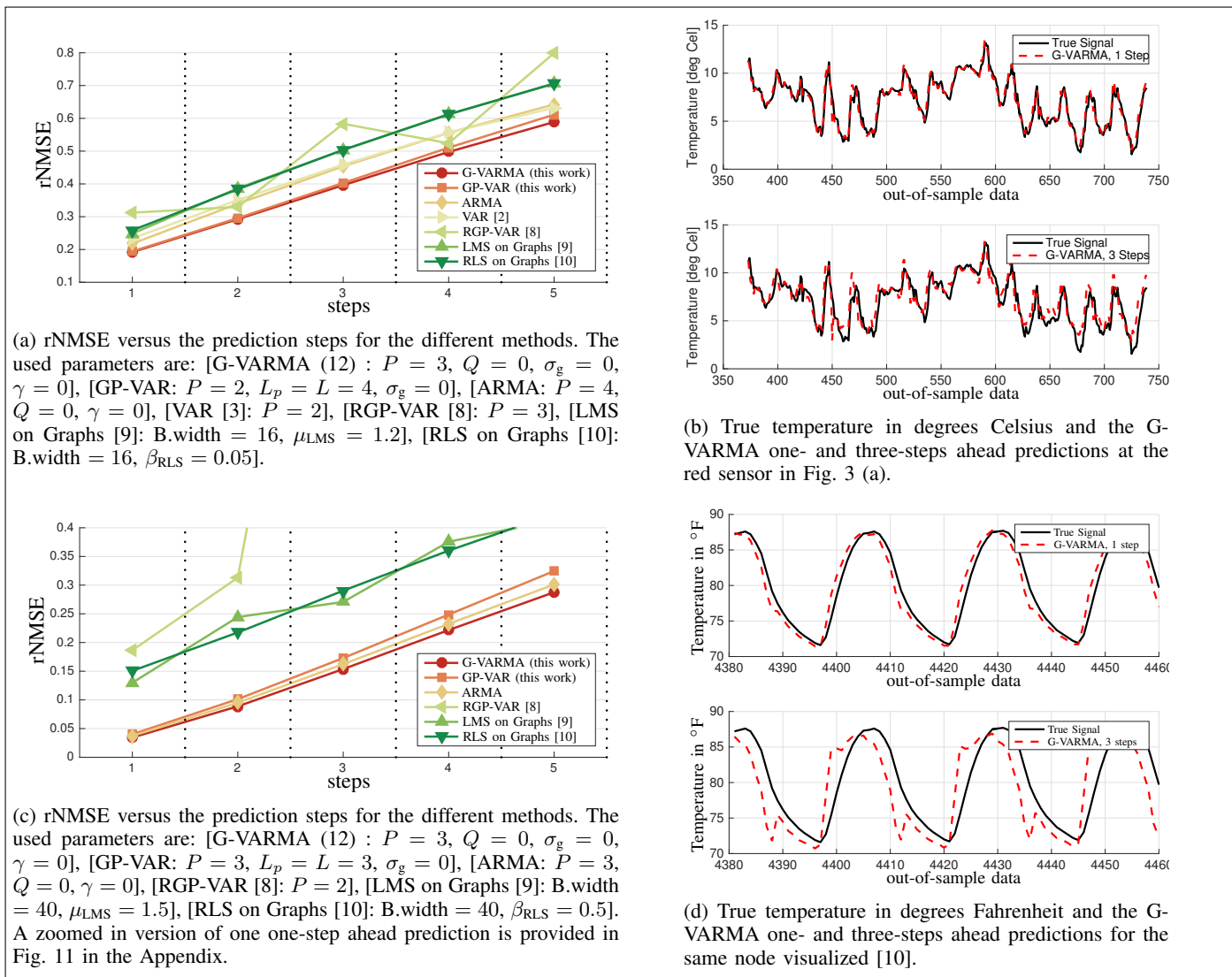
(a) rNMSE versus the prediction steps for the different methods. The used parameters are: [G-VARMA (12) : $P = 3$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [GP-VAR: $P = 2$, $L_p = L = 4$, $\sigma_g = 0$], [ARMA: $P = 4$, $Q = 0$, $\gamma = 0$], [VAR [3]: $P = 2$], [RGP-VAR [8]: $P = 3$], [LMS on Graphs [9]: B.width $= 16$, $\mu_{\text{LMS}} = 1.2$], [RLS on Graphs [10]: B.width $= 16$, $\beta_{\text{RLS}} = 0.05$].

(b) True temperature in degrees Celsius and the G-VARMA one- and three-steps ahead predictions at the red sensor in Fig. 3 (a).

(c) rNMSE versus the prediction steps for the different methods. The used parameters are: [G-VARMA (12) : $P = 3$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [GP-VAR: $P = 3$, $L_p = L = 3$, $\sigma_g = 0$], [ARMA: $P = 3$, $Q = 0$, $\gamma = 0$], [RGP-VAR [8]: $P = 2$], [LMS on Graphs [9]: B.width $= 40$, $\mu_{\text{LMS}} = 1.5$], [RLS on Graphs [10]: B.width $= 40$, $\beta_{\text{RLS}} = 0.5$]. A zoomed in version of one one-step ahead prediction is provided in Fig. 11 in the Appendix.

(d) True temperature in degrees Fahrenheit and the G-VARMA one- and three-steps ahead predictions for the same node visualized [10].

Fig. 4: Comparison of the predictive power of the different algorithms in the weather datasets. (a)-(b) Molene data set. (c)-(d) NOAA data set. The results in plots (b) and (d) are w.r.t. the out-of-sample data.

methods exploit better the graph parameterization and yield the minimum performance.

The disjoint ARMA compares well with the proposed algorithms and yields values closer to minimum achieved by the G-VARMA. This is not entirely surprising for two main reasons. First, the time series present a strong temporal stationarity which is known to be well exploited by ARMA models. Second, we fixed the graph as in [10] and [12] for a direct comparison with these methods. This choice penalizes the stationarity assumptions. Therefore, we believe that the performance of the G-VARMA and the GP-VAR can be further improved by optimizing the graph topology.

As a final remark, note that we used the parameters of the LMS ($\mu_{\text{LMS}} = 1.5$) and RLS ($\beta_{\text{RLS}} = 0.5$) on graphs algorithms as fixed[7] by [10]. By cross-validation, we found that the optimal values are $\mu_{\text{LMS}} = 1.75$ and $\beta_{\text{RLS}} = 0.05$. These cross-validated values lead to a slightly lower rNMSE,

[7]This is done to ensure a fair comparison with these methods.

but still significantly worse when compared with the proposed methods and to the disjoint ARMA.

Fig. 4d shows the one- and the three-steps ahead predictions at node 25, which is the same used to display the tracking results of LMS, RLS, and KKF in [10]. We observe that the one-step ahead prediction matches relatively well the true signal value, while a bigger error is observed for the three-step ahead prediction especially closer to the transition points.

Walking dog mesh. The rNMSE vs number of step ahead results are shown in Fig. 5a. The classic multivariate VAR is not shown as it yielded unstable predictions (as the number of time series is large). Incorporating the graph in the predictive models seems to avoid these issues. In fact, all graph-based approaches give more stable predictors. Additionally, the RGP-VAR [8] yields the worst prediction due to fewer DoF. A comparison with its generalized version, i.e., the GP-VAR, shows an rNMSE close to the best performance. We note that the rNMSE gap between the proposed methods and other
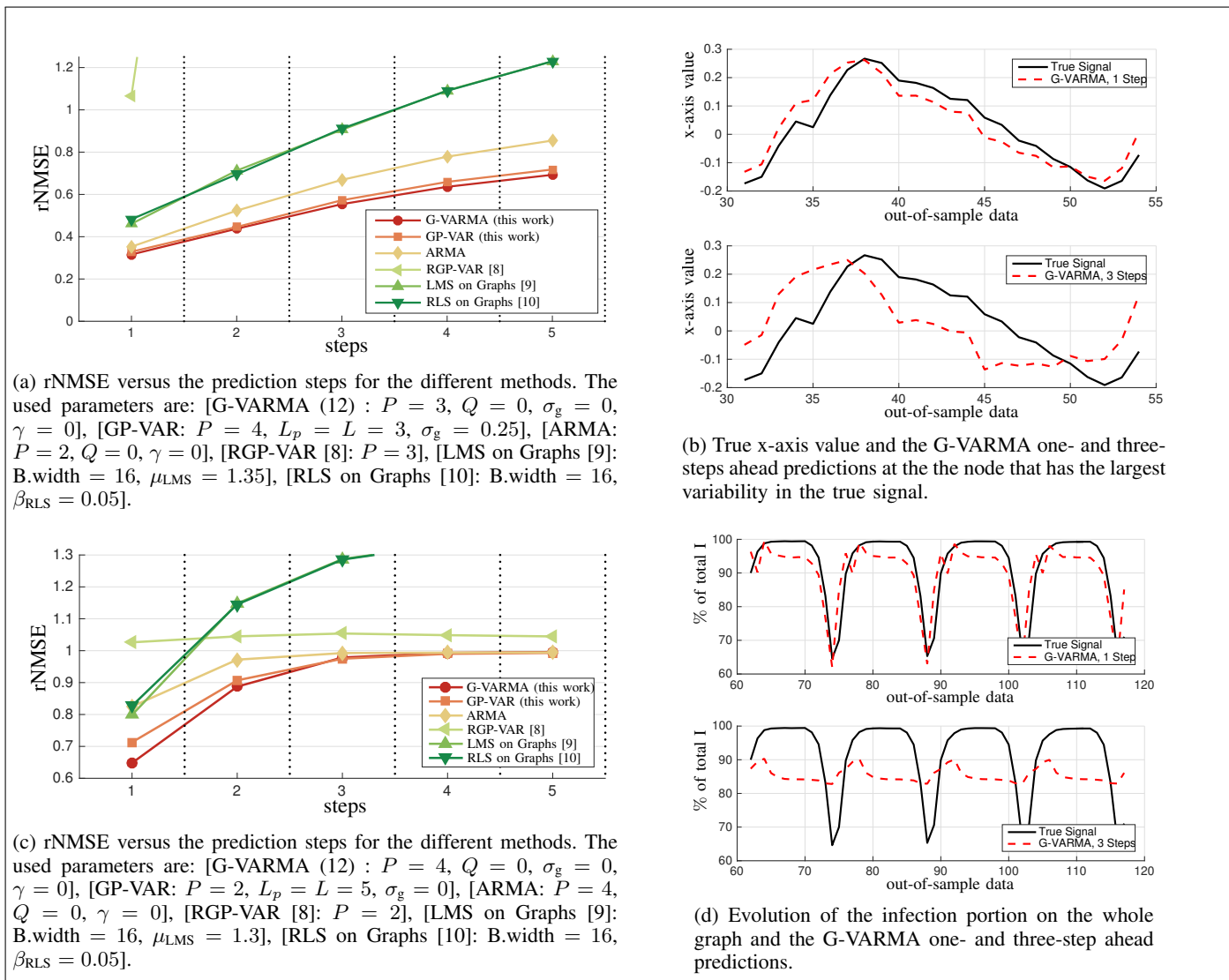
(a) rNMSE versus the prediction steps for the different methods. The used parameters are: [G-VARMA (12) : $P = 3$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [GP-VAR: $P = 4$, $L_p = L = 3$, $\sigma_g = 0.25$], [ARMA: $P = 2$, $Q = 0$, $\gamma = 0$], [RGP-VAR [8]: $P = 3$], [LMS on Graphs [9]: B.width $= 16$, $\mu_{LMS} = 1.35$], [RLS on Graphs [10]: B.width $= 16$, $\beta_{RLS} = 0.05$].

(b) True x-axis value and the G-VARMA one- and three-steps ahead predictions at the the node that has the largest variability in the true signal.

(c) rNMSE versus the prediction steps for the different methods. The used parameters are: [G-VARMA (12) : $P = 4$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [GP-VAR: $P = 2$, $L_p = L = 5$, $\sigma_g = 0$], [ARMA: $P = 4$, $Q = 0$, $\gamma = 0$], [RGP-VAR [8]: $P = 2$], [LMS on Graphs [9]: B.width $= 16$, $\mu_{LMS} = 1.3$], [RLS on Graphs [10]: B.width $= 16$, $\beta_{RLS} = 0.05$].

(d) Evolution of the infection portion on the whole graph and the G-VARMA one- and three-step ahead predictions.

Fig. 5: Comparison of the predictive power of the different algorithms in the three scenarios. (a)-(b) Walking dog mesh. (c)-(d) SI epidemic diffusion. The results in plots (b) and (d) are w.r.t. the out-of-sample data.

graph-based alternatives is larger w.r.t. the Molene data set. The spectral smoothing plays a role here for the GP-VAR since the signal evolution (i.e., the point position) is more regular along the graph dimension than temperature. This is reflected in a higher value of σ_g.

Fig. 5b illustrates the true signal evolution along with its G-VARMA one- and three-step ahead predictions. The selected node is the one that presents the largest peak-to-peak variability in the true signal. Compared to the Molene data set, we note a worse three-step ahead prediction, while the one-step ahead prediction is still comparable. Fig. 6 further shows the true dog position in four causal time steps and the three step ahead G-VARMA prediction. We can observe that the proposed model matches well the dog position in most of the nodes. Larger errors are observed around the body parts that move the most, e.g., legs and tail.

SI epidemic diffusion. The results for this scenario are shown in Figs. 5c-5d. Similarly to the walking dog mesh, the VAR

model yields unstable results. From the results in Fig. 5c, all graph-based methods offer a similar rNMSE for the one-step ahead prediction[8]. The latter highlights the importance of the underlying topology in building prediction models, since the univariate ARMA yields a larger rNMSE than the proposed models. For more than one-step ahead predictions, we see that the proposed approaches exploit better the graph and keep yielding a smaller rNMSE than the adaptive algorithms. Nevertheless, starting from three-steps ahead, the univariate ARMA offers a similar rNMSE as the proposed methods.

We make the subsequent remarks. First, the rNMSE is higher in this data set w.r.t. the previous two cases. We attribute this degradation to the abrupt transitions in the patient recovery instances (i.e., every 12 days). Second, also in this data set we see that the G-VARMA does not have coefficients in the MA

---

[8]An exception is the RGP-VAR recursion [8], which offers a worse performance. However, the focus of this approach is to learn a topology that facilitates prediction and it might suffer when the fixed graph structure differs from the learned one.
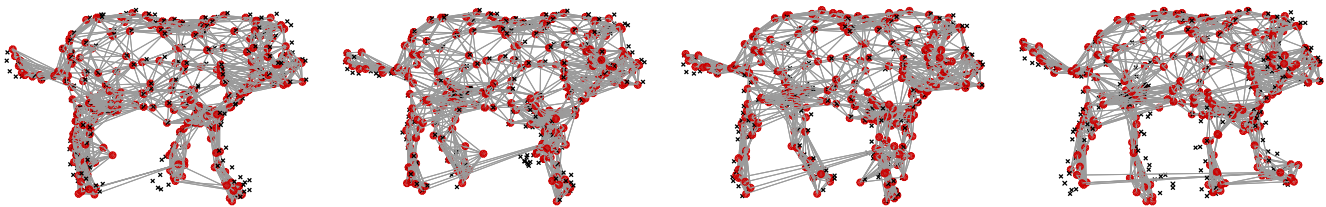
Fig. 6: Four test time steps of the three step ahead G-VARMA prediction in the walking dog experiment. The red points correspond to the prediction and the black to the ground-truth. The four time steps are causal with a delay of 2 steps.
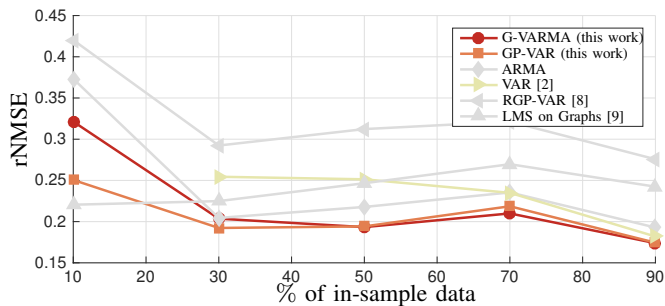

Fig. 7: One-step ahead prediction rNMSE versus the percentage of in-sample-data for the different algorithms in the Molene data set. The RLS on graphs [10] is not shown here and offers a similar performance as the LMS on graphs algorithm. Note that the standard VAR could not be trained for 10% in-sample data.
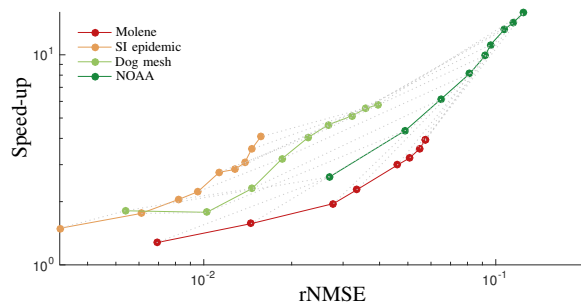

Fig. 8: Gained speed-up versus the loss in prediction rNMSE for the G-VARMA model estimated using the low-rank representation of the process. Both axes are in log scale to improve the visibility. For the NOAA data set, the maximum rNMSE loos is 0.13 with a speed-up of 16.

part. We have found that the value of Q varies depending on the amount of training data and cross-validation criterion. For instance, the G-VARMA model with $P = 1$, $Q = 1$, $\sigma_g = 0.5$, and $\gamma = 0.44$ yields a better performance in the test data than the one shown in Fig. 5c, though in the cross-validation data its performance is suboptimal. Finally, we have observed that the chosen models are characterized by a bounded memory in time. That is, only a few past realizations are useful for the prediction task. The GP-VAR model shows that there is a larger graph influence (i.e., information from further nodes) in the model. Indeed, information from nodes up to four hops away is exploited in Fig. 5c and in Fig. 4a.

To conclude, Fig. 5d illustrates the evolution of the I-state on the whole graph and the respective G-VARMA predictions. The proposed model follows relatively well the I-state evolution for one-step ahead prediction with an error up to 4%, yet its performance degrades in the three-step ahead prediction[9]. The decreased amplitude (w.r.t the ground truth) is mainly attributed to a couple outlier nodes where the I-state evolution is not predicted correctly, and, therefore, influence the overall result.

C. In-sample data effect

We now analyze the effect of the in-sample data size on prediction accuracy. For consistency with the earlier results, we kept the cross-validation data to be 30% of the in-sample

[9]This may be improved by using other model parameters.

data, whereas the model parameters are selected to minimize the one-step ahead rNMSE.

Fig. 7 shows the one-step ahead prediction rNMSE as a function of the in-sample data percentage for the Molene data set. We first highlight that the multivariate VAR [3] is the one that suffers the most in low in-sample data regimes. For this model, we see a clear trend of rNMSE reduction when more data are used for estimating the coefficients. All graph-based methods are more robust to low in-sample data sizes since they have fewer parameters. The latter along with the spectral smoothing seems to play a role when the amount of training data is limited. The slight variation in the rNMSE for more training data is within the experimental variance. When the percentage of in-sample data is very small (10%), the GP-VAR improves upon G-VARMA due to its fewer DoF.

We conclude that the proposed methods make a better use of the underlying topology and provide a good trade-off between the model complexity and the amount of training data. As a final remark, we do not show these results for the other scenarios since the VAR [3] recursion (which is the one to compare with) does not lead to stable predictors.

D. Low-rank model effect

We now test the effects of the low-rank G-VARMA coefficient estimation described in Theorem 1. Under the settings of Fig. 4, we estimate the models by ignoring different portions of the process JPSD. We considered nine ignoring percentages log-spaced in the interval $[0.74\%, 5\%]$ and evaluate the rNMSE loss and the code speed-up. The code running time is
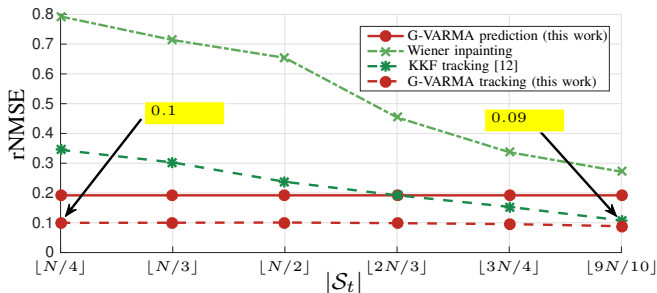
Fig. 9: rNMSE versus the cardinality of sampling set for different tracking algorithms in the Molene data set. The used parameters are: [G-VARMA (12) : $P = 3$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [Wiener inpainting, $w_t \sim \mathcal{N}(\mathbf{0}_N, 0.5\mathbf{I}_N)$], [KKF [12]: $\mu_{\text{KKF}} = 10^{-7}$, $\sigma_{\text{KKF}} = 1.25$, $b_{\text{KKF}} = 0.09$].
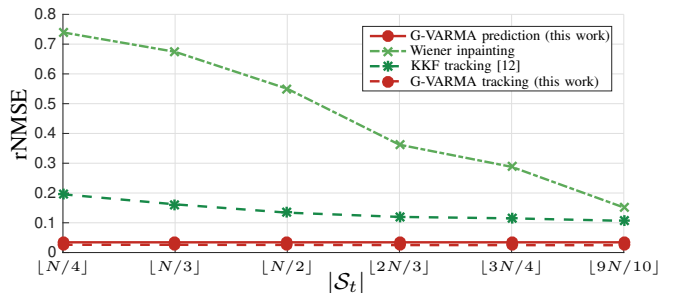

(a) Comparison of the different methods.

Fig. 10: rNMSE versus the cardinality of sampling set for different tracking algorithms in the Molene data set. The used parameters are: [G-VARMA (12) : $P = 3$, $Q = 0$, $\sigma_g = 0$, $\gamma = 0$], [Wiener inpainting, $w_t \sim \mathcal{N}(\mathbf{0}_N, 0.5\mathbf{I}_N)$], [KKF [12]: $\mu_{\text{KKF}} = 10^{-7}$, $\sigma_{\text{KKF}} = 1.8$, $b_{\text{KKF}} = 0.01$]. A zoomed in version of the G-VARMA tracking algorithm is provided in the Appendix.

averaged over 100 iterations and the speed-up achieved using the low-rank procedure is compared w.r.t. the case where the entire JPSD is used.

Fig. 8 shows the gained speed-up as a function of the rNMSE degradation in the one-step ahead prediction. Each marker corresponds to an increasing data ignorance percentage. The obtained results show a trade-off between prediction accuracy and computational time. We see that the most sensitive scenario to data ignorance is the NOAA data set as its NMSE loss increases the fastest, while the least sensitive scenario is the SI epidemic one. This much higher sensitivity of the NOAA data set is due to the misalignment between the used graph and the stationarity assumptions. Better results are expected in this data set with a topology that imposes a stronger joint stationarity. These observations highlight, nevertheless, the sparsity in the process JPSD which can be exploited to ease the model estimation costs.

E. Tracking performance

We here evaluate the tracking performance of the G-VARMA model on the Molene and NOAA data sets. The G-VARMA parameters are the same as the one used in Section VI-B with coefficients estimated following the low-rank model and ignoring 1.5% of the process JPSD. The noise power for the Wiener inpainting is cross-validated following the procedure in Fig. 2 using all the 50% in-sample data. For the KKF, we cross-validated the different parameters, i.e., the KKF kernel weight (μ_{KKF}), the diffusion coefficient (σ_{KKF}), and the edge propagation weight (b_{KKF}) using all the 50% in-sample data in the Molene data set. Instead, for the NOAA data set we used the parameters proposed in [12]. During the training process, 75% of the nodes are sampled uniformly at random and the considered score is the tracking rNMSE. To the G-VARMA tracker, a small artificial noise is added to the out-of-sample data to avoid numerical issues when inverting H_t. In the test phase, we average the rNMSE over 100 random node selections and consider $\bar{\mathcal{S}}_t = \mathcal{V}$.

Molene data set. Fig. 9 depicts the average tracking rNMSE as a function of the size $|\mathcal{S}_t|$ of the observation set. The proposed method is seen to improve substantially

the accuracy of the G-VARMA predictor and outperforms the other approaches even with $N/4 = 8$ collected samples at time t. Wiener inpainting provides suboptimal predictions as it does not exploit the historical data. These results show also the benefits of the G-VARMA predictors over the Wiener inpainting and the KKF approach.

NOAA data set. The tracking results for the NOAA data set are shown in Fig. 10a. The latter enforces the behavior of the different methods observed for the Molene data set. The Wiener inpainting still offers the worse performance as it does not explore the temporal dimension. Similarly, both the G-VARMA prediction and tracking offer a lower rNMSE than the KKF [12]. The performance of the KKF can be slightly improved by using cross-validation (validated parameters $\mu_{\text{KKF}} = 10^{-6}$, $\sigma_{\text{KKF}} = 1.5$, $b_{\text{KKF}} = 0.15$).

In Fig. 12 (see the Appendix), we zoomed in the performance of the proposed G-VARMA tracking methods. A larger cardinality of the sampling set leads to small reduction of the rNMSE. We mainly attribute this minimal improvement to the mismatch between the data behavior and the G-VARMA model. Further reduction of the rNMSE can be obtained by selecting the $|\mathcal{S}_t|$ in a sparse sensing fashion [42] and by introducing seasonality into the G-VARMA model [49].

VII. CONCLUSION

The purpose of this work was to determine multivariate models for forecasting the temporal evolution of time series on graphs. By leveraging the (approximate) joint time-vertex stationary of the time series, we reduced the DoFs for the VAR and VARMA models by designing their coefficients in the graph spectral domain. The latter decomposition allowed us to fit the model parameters with well-known univariate techniques by tackling both the computational and stability issues present in classical VAR and VARMA models. To further reduce the estimation costs, we proposed an optimal low-rank representation for the time series. Finally, a sub-

graph tracker was introduced when measurements form a subset of nodes become available.

The findings clearly indicated that the graph structure plays a major role in easing the computational complexity of high dimensional multivariate models. This role is further enhanced when the amount of training data is limited. Several numerical experiments showed the benefits of the proposed methods over the classical multivariate forecasting techniques and other the state-of-the-art graph-based approaches.

Overall, this work strengthens the idea that practical benefits could be achieved in the time series forecasting if the underlying data connections are exploited. Further work is needed in two main directions. First, it is important to understand the implications of the stationary assumption for the task of prediction. Second, it is useful to analyze the effects of the graph structure in the prediction accuracy.

APPENDIX
PROOF OF PROPOSITION 1

By substituting the expression of $\tilde{\boldsymbol{x}}_t$ (15) and $\boldsymbol{x}_t$ (12) into the the one-step ahead predictor MSE $= \mathbb{E}\left[\|\tilde{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2^2\right]$, we have

$$\text{MSE} = \mathbb{E}\left[\left\|\sum_{q=1}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\left(\boldsymbol{x}_{t-q} - \tilde{\boldsymbol{x}}_{t-q}\right) - \sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{\varepsilon}_{t-q}\right\|_2^2\right]. \tag{31}$$

Then, by substituting $\boldsymbol{\varepsilon}_{t-q} = \boldsymbol{x}_{t-q} - \tilde{\boldsymbol{x}}_{t-q}$ in (31) we obtain

$$\text{MSE} = \mathbb{E}\left[\|b_0(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{\varepsilon}_t\|_2^2\right], \tag{32}$$

which concludes the derivation of (16). The MSE (32) depends only on the unknown innovation at time t, which corresponds to the smallest achievable MSE for the given set of coefficients $\{a_p(\boldsymbol{L}_{\mathcal{G}}), b_q(\boldsymbol{L}_{\mathcal{G}})\}$. This concludes the proof. ∎

PROOF OF PROPOSITION 2

By regrouping the terms containing $\boldsymbol{x}_\tau$ on the left hand-side of (12), the nth element rotated by the unitary matrix $\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ is

$$\left[\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\sum_{p=0}^{P} a_p(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{x}_{t-p}\right]_n = \left[\sum_{p=0}^{P} [a_p(\boldsymbol{\Lambda}_{\mathcal{G}})]_{nn}\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\boldsymbol{x}_{t-p}\right]_n$$
$$= \sum_{p=0}^{P} [a_p(\boldsymbol{\Lambda}_{\mathcal{G}})]_{nn}\,\hat{x}_{t-p,n}, \tag{33}$$

where $a_0(\boldsymbol{L}_{\mathcal{G}}) = a_p(\boldsymbol{\Lambda}_{\mathcal{G}}) = \boldsymbol{I}_N$ Similarly, the n-th element of the remaining right hand-side of (12) rotated by $\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ is

$$\left[\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\sum_{q=0}^{Q} b_q(\boldsymbol{L}_{\mathcal{G}})\boldsymbol{\varepsilon}_{t-q}\right]_n = \left[\sum_{q=0}^{Q} b_q(n)\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}\boldsymbol{\varepsilon}_{t-q}\right]_n$$
$$= \sum_{q=0}^{Q} [b_q(\boldsymbol{\Lambda}_{\mathcal{G}})]_{nn}\,\hat{\varepsilon}_{t-q,n}. \tag{34}$$

Then, the claim (18) is obtained by direct substitution of (35)-(37) into (12). ∎

PROOF OF THEOREM 1

Let us define $\boldsymbol{A} = \boldsymbol{U}(\boldsymbol{I}_N - \boldsymbol{D}_{\mathcal{S}})\boldsymbol{U}^{\mathsf{H}}$. Following the Eckart–Young–Mirsky Theorem [50], [51], the expected approximation error is

$$\mathbb{E}\left[\left\|\boldsymbol{X}_{1:T} - \tilde{\boldsymbol{X}}_{U,\mathcal{S}}\right\|_F^2\right] = \mathbb{E}\left[\|\boldsymbol{A}\boldsymbol{X}_{1:T}\|_F^2\right] = \text{tr}\left(\boldsymbol{A}\mathbb{E}[\boldsymbol{X}_{1:T}\boldsymbol{X}_{1:T}^{\mathsf{H}}]\boldsymbol{A}^{\mathsf{H}}\right). \tag{35}$$

Then, from Theorem 2 of [52], where for each time instant t, the graph signal $\boldsymbol{x}_t$ is JWSS stationary with covariance $\boldsymbol{\Sigma}_t = \mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathsf{H}}] = \boldsymbol{U}_{\mathcal{G}}g_t^2(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}}$ implying that

$$\boldsymbol{\Sigma}_{\mathcal{G}} = \mathbb{E}[\boldsymbol{X}_{1:T}\boldsymbol{X}_{1:T}^{\mathsf{H}}] = \sum_{t=1}^{T}\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathsf{H}}] = \sum_{t=1}^{T}\boldsymbol{\Sigma}_x$$
$$= \boldsymbol{U}_{\mathcal{G}}g^2(\boldsymbol{\Lambda}_{\mathcal{G}})\boldsymbol{U}_{\mathcal{G}}^{\mathsf{H}} \tag{36}$$

where $g^2(\boldsymbol{\Lambda}_{\mathcal{G}}) = \sum_{t=1}^{T} g_t^2(\boldsymbol{\Lambda}_{\mathcal{G}})$ are reordered such that $g(\lambda_1) \geq g(\lambda_2) \geq \ldots \geq g(\lambda_N)$. By substituting of $\boldsymbol{\Sigma}_{\mathcal{G}}$ into (35) we get

$$\mathbb{E}\left[\left\|\boldsymbol{X}_{1:T} - \tilde{\boldsymbol{X}}_{U,\mathcal{S}}\right\|_F^2\right] = \text{tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{\mathcal{G}}\boldsymbol{A}^{\mathsf{H}}\right) = \left\|\boldsymbol{\Sigma}_{\mathcal{G}}^{1/2} - \boldsymbol{U}\boldsymbol{D}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{\Sigma}_{\mathcal{G}}^{1/2}\right\|_F^2$$
$$= \left\|g(\boldsymbol{\Lambda}_{\mathcal{G}}) - \boldsymbol{U}\boldsymbol{D}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}g(\boldsymbol{\Lambda}_{\mathcal{G}})\right\|_F^2. \tag{37}$$

Setting then $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{D}_{\mathcal{S}}\boldsymbol{U}^{\mathsf{H}}$, (37) becomes

$$\mathbb{E}\left[\left\|\boldsymbol{X}_{1:T} - \tilde{\boldsymbol{X}}_{U,\mathcal{S}}\right\|_F^2\right] = \sum_{i=1}^{N}|g(\lambda_i) - [\boldsymbol{B}]_{ii}g(\lambda_i)|^2$$
$$+ \sum_{i \neq j}|[\boldsymbol{B}]_{ij}g(\lambda_i)|^2 \geq \sum_{i=1}^{N}|g(\lambda_i)|^2\,|1 - [\boldsymbol{B}]_{ii}|^2$$
$$\overset{(a)}{\geq} \sum_{i=K+1}^{N}|g(\lambda_i)|^2 = \mathbb{E}\left[\left\|\boldsymbol{X}_{1:T} - \tilde{\boldsymbol{X}}_{U_{\mathcal{G}},\mathcal{S}^\star}\right\|_F^2\right], \tag{38}$$

where in the third step we use the fact that $[\boldsymbol{B}]_{ii} \leq 1$ and is exactly 1 at most K times. The last expression shows that the global minimum is achieved for $\tilde{\boldsymbol{X}}_{U_{\mathcal{G}},\mathcal{S}^\star}$ with $\mathcal{S}$ containing the K largest components of $g(\lambda)$. In fact, (a) is independent on the choice of $\boldsymbol{B}$, while the last equality holds only for $\boldsymbol{U} = \boldsymbol{U}_{\mathcal{G}}$. Any other rotation will lead to a different $\boldsymbol{B}$ thus a larger error. ∎

ZOOMED IN FIGURES

This section presents the zoomed in versions of Fig. 4c in Fig. 11 and Fig. 10 in Fig. 12.

REFERENCES

[1] A. Loukas, E. Isufi, and N. Perraudin, "Predicting the evolution of stationary graph signals," in *51st Asilomar Conference on Signals, Systems, and Computers*, 2017.

[2] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[3] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[4] G. Connor, "The three types of factor models: A comparison of their explanatory power," *Financial Analysts Journal*, vol. 51, no. 3, pp. 42–46, 1995.
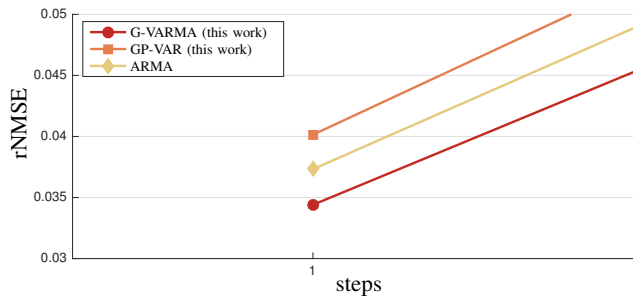
Fig. 11: Zoomed in version of Fig. 4c that shows the minimal differences between the proposed G-VARMA, GP-VAR, and the disjoint ARMA approach. We can see that the G-VARMA model slightly improves the disjoint ARMA approach and this gap increases for more steps ahead predition.
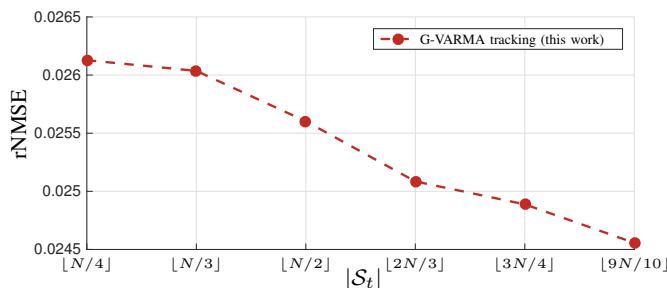

Fig. 12: Zoomed in version of Fig. 10 that shows the reduction of the rNMSE for the G-VARMA tracking algorithm when more nodes are sampled at time t.

[5] C. Lam, Q. Yao, and N. Bathia, "Factor modeling for high dimensional time series," in *Recent Advances in Functional Data Analysis and Related Topics*. Springer, 2011, pp. 203–207.

[6] A. Meucci, *Risk and asset allocation*. Springer Science & Business Media, 2009.

[7] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[8] J. Mei and J. M. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, 2017.

[9] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 555–568, 2016.

[10] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Transactions on Signal Processing*, 2018.

[11] K. Qiu, X. Mao, X. Shen, X. Wang, T. Li, and Y. Gu, "Time-Varying Graph Signal Reconstruction," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 870–883, 2017.

[12] D. Romero, V. N. Ioannidis, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions on dynamic graphs," *IEEE Transactions on Signal Processing*, vol. 11, no. 6, pp. 856 – 869, 2017.

[13] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911 – 5926, 2017.

[14] B. Girault, "Stationary graph signals using an isometric graph translation," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1516–1520.

[15] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462 – 3477, 2017.

[16] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *arXiv preprint arXiv:1611.00255*, 2016.

[17] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable Autoregressive Moving Average Graph-Temporal Filters," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 200–204.

[18] E. Isufi, P. Banelli, P. Di Lorenzo, and G. Leus, "Observing and tracking bandlimited graph processes," *arXiv preprint arXiv:1712.00404*, 2017.

[19] E. Isufi, G. Leus, and P. Banelli, "2-Dimensional Finite Impulse Response Graph-Temporal Filters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington D.C, USA, 2016.

[20] A. Arguez, I. Durre, A. Applequist, R. S. Vose, M. F. Squires, X. Yin, R. R. Heim Jr., and T. W. Owen, "NOAA'S 1981-2020 U. S. Climate Normals: An Overview," in *Bulletin of the American Meteorological Society*, 2012, pp. 1687–1697.

[21] 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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[22] G. Taubin, "Geometric signal processing on polygonal meshes," *Eurographics STAR - State of The Art Report*, 2000.

[23] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, pp. 1644–1656, 2013.

[24] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev Polynomial Approximation for Distributed Signal Processing," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*. IEEE, 2011, pp. 1–8.

[25] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite Impulse Response Graph Filters in Wireless Sensor Networks," *IEEE Signal Processing Letters*, Jan 2015.

[26] S. Segarra, A. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, 2017.

[27] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[28] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A Time-Vertex Signal Processing Framework: Scalable Processing and Meaningful Representations for Time-Series on Graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817 – 829, 2018.

[29] Y. Feng, D. P. Palomar *et al.*, "A signal processing perspective on financial engineering," *Foundations and Trends® in Signal Processing*, vol. 9, no. 1–2, pp. 1–231, 2016.

[30] A. Wills and B. Ninness, "On gradient-based search for multivariable system estimates," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 298–306, 2008.

[31] A. M. Tulino, S. Verdú *et al.*, "Random matrix theory and wireless communications," *Foundations and Trends® in Communications and Information Theory*, vol. 1, no. 1, pp. 1–182, 2004.

[32] Y. Sun, P. Babu, and D. P. Palomar, "Regularized tyler's scatter estimator: Existence, uniqueness, and algorithms," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5143–5156, 2014.

[33] D. E. Tyler, "A distribution-free m-estimator of multivariate scatter," *The Annals of Statistics*, pp. 234–251, 1987.

[34] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 2009.

[35] S. P. Chepuri and G. Leus, "Graph sampling for covariance estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 451–466, 2017.

[36] M. S. Bartlett, "Periodogram analysis and continuous spectra," *Biometrika*, pp. 1–16, 1950.

[37] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, pp. 70–73, 1967.

[38] B. Girault, S. S. Narayanan, and A. Ortega, "Towards a definition of local stationarity for graph signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4139–4143.

[39] A. Loukas, "How close are the eigenvectors of the sample and actual covariance matrices?" in *International Conference on Machine Learning*, 2017, pp. 2228–2237.

[40] S. M. Kay, "Fundamentals of statistical signal processing, volume i: estimation theory," 1993.

[41] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[42] S. P. Chepuri and G. Leus, "Sparse sensing for statistical inference," *Foundations and Trends® in Signal Processing*, vol. 9, no. 3–4, pp. 233–368, 2016.

[43] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[44] L. Ljung, "System identification," in *Signal Analysis and Prediction*. Springer, 1998, pp. 163–173.

[45] A. Wills, B. Ninness, and S. Gibson, "On gradient-based search for multivariable system estimates," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 832–837, 2005.

[46] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[47] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1746–1753.

[48] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Systems*, vol. 36, no. 1, pp. 26–46, 2016.

[49] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*. Springer, 2002, vol. 2.

[50] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[51] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, 2008.

[52] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3914 – 3918.