

Reconstruction of Time-varying Graph Signals via Sobolev Smoothness

Jhony H. Giraldo, Arif Mahmood, Belmar Garcia-Garcia, Dorina Thanou, and Thierry Bouwmans

Abstract—Graph Signal Processing (GSP) is an emerging research field that extends the concepts of digital signal processing to graphs. GSP has numerous applications in different areas such as sensor networks, machine learning, and image processing. The sampling and reconstruction of static graph signals have played a central role in GSP. However, many real-world graph signals are inherently time-varying and the smoothness of the temporal differences of such graph signals may be used as a prior assumption. In the current work, we assume that the temporal differences of graph signals are smooth, and we introduce a novel algorithm based on the extension of a Sobolev smoothness function for the reconstruction of time-varying graph signals from discrete samples. We explore some theoretical aspects of the convergence rate of our Time-varying Graph signal Reconstruction via Sobolev Smoothness (GraphTRSS) algorithm by studying the condition number of the Hessian associated with our optimization problem. Our algorithm has the advantage of converging faster than other methods that are based on Laplacian operators without requiring expensive eigenvalue decomposition or matrix inversions. The proposed GraphTRSS is evaluated on several datasets including two COVID-19 datasets and it has outperformed many existing state-of-the-art methods for time-varying graph signal reconstruction. GraphTRSS has also shown excellent performance on two environmental datasets for the recovery of particulate matter and sea surface temperature signals.

Index Terms—Graph signal processing, time-varying graph signals, Sobolev smoothness, signal reconstruction, COVID-19.

I. INTRODUCTION

Graphs provide the ability to model interactions of data residing on irregular and complex structures. Social, financial, and sensor networks are examples of data that can be modeled on graphs. Graph Signal Processing (GSP) is an emerging field that aims to extend the concepts of classical digital signal processing to graphs [1]–[4]. GSP has been used in several applications such as computer vision [5], [6], forecasting [7], sensor networks [8], biological networks [9], image, and 3D point cloud processing [10]–[12]. Similarly, the study of machine learning on graphs [13] has benefited profoundly from GSP; some examples include semi-supervised learning

Jhony H. Giraldo, Belmar Garcia-Garcia, and Thierry Bouwmans are with the laboratoire MIA, Mathématiques, Image et Applications, La Rochelle Université, 17000 La Rochelle, France e-mail: jgiral01@univ-lr.fr, belmar_2g@hotmail.com, tbouwman@univ-lr.fr.

Arif Mahmood is with the Department of Computer Science, Information Technology University, 48730 Lahore, Pakistan e-mail: arif.mahmood@itu.edu.pk.

Dorina Thanou is with the Center for Intelligent Systems, EPFL, 1015 Lausanne, Switzerland e-mail: dorina.thanou@epfl.ch.

The authors thank the support provided by CONACYT Mexico under the grant number 770750.

[14], [15], graph learning [16]–[18], and the development of filters for graph convolutional networks [19], [20].

The sampling and recovery of graph signals are fundamental tasks in GSP that have recently received considerable attention. Naturally, the mathematics of sampling theory and spectral graph theory have been combined, leading to generalized Nyquist sampling principles for graphs [21]–[24], where most of these previous works have focused on static graph signals. The reconstruction of time-varying graph signals is an important problem that has not been well explored¹. The reconstruction of time-varying graph signals from discrete samples has several real world applications, such as the estimation of new cases of infectious diseases [25], or the recovery of the sea surface temperature for the study of the earth's climate dynamics [26].

Some recent methods for the reconstruction of time-varying graph signals have also utilized the assumption that temporal differences of graph signals are smooth. For example, Qiu *et al.* [26] extended the definition of smooth signals from static to time-varying graph signals, and solved an optimization problem for reconstruction. However, Qiu's method may have slow convergence rate because the eigenvalue spread of the Hessian associated with their optimization problem may be large. The techniques presented in [27]–[29] may be used to circumvent the problem of slow convergence. However, these techniques require eigenvalue decomposition or matrix inversion, which are computationally expensive for large graphs.

In the current work, we build on our previous work [25], and we propose a new algorithm to reconstruct time-varying graph signals from samples. Our algorithm is based on the extension of the Sobolev norm defined in GSP for time-varying graph signals [6], [30]. Therefore, we dub our algorithm as Time-varying Graph signal Reconstruction via Sobolev Smoothness (GraphTRSS). Our algorithm uses a Sobolev smoothness function to formulate an optimization problem for the reconstruction of a time-varying graphs signal from its samples, where the samples are obtained with a random sampling strategy. The graph is constructed with a k -Nearest Neighbors (k -NN) method. The optimization problem of GraphTRSS is solved with the conjugate gradient method. We analyze the convergence rate of our algorithm by studying the condition number of the Hessian associated with our optimization function, and we conclude that our algorithm converges faster than its closest competitor [26] under certain conditions. Moreover, GraphTRSS does not require expensive

¹One can think of the reconstruction of time-varying graph signals as a matrix completion problem where each row (or column) is associated with a node, and each column (or row) is associated with time.

eigenvalue decomposition or matrix inversion. Finally, we evaluate our algorithm on synthetic data, two COVID-19, and two environmental datasets, where our algorithm outperforms several state-of-the-art methods.

The main contributions of this work are summarized as follows:

- We use the concept of Sobolev norms from static graph signals to define a smoothness function for time-varying graph signals, and then we use this new conception to introduce an algorithm for reconstruction.
- We provide several mathematical insights of GraphTRSS. Specifically, we analyze the convergence rate of our algorithm by studying the condition number of the Hessian associated with our problem.
- GraphTRSS improves convergence speed without relying on expensive matrix inversion or eigenvalue decomposition.
- Concepts of reconstruction of graph signals from GSP are introduced in the mathematical modeling of infectious diseases for COVID-19, as well as environmental data.

The rest of the paper is organized as follows. Section II shows related work in time-varying graph signals reconstruction. Section III presents the mathematical notations and theoretical background of this paper. Section IV explains the details of GraphTRSS. Section V introduces the experimental framework. Finally, Sections VI and VII show the results and conclusions, respectively.

II. RELATED WORK

The problems of sampling and reconstruction of graph signals have been widely explored in GSP [23], [24], [31]–[36]. Pesenson [37] introduced concepts of Paley-Wiener spaces in graphs, where a graph signal can be determined by its samples in a set of nodes called uniqueness set. One can say that a set of nodes is a uniqueness set of a certain graph if the fact that two graph signals in the Paley-Wiener space of the graph coincide in the uniqueness set implies that they coincide in the whole set of nodes. As a result, a bandlimited graph signal can exactly be reconstructed from its samples if the graph signal is sampled according to its uniqueness set. However, the bandlimitedness assumption is not realistic, *i.e.*, the graph signals in real-world datasets tend to be approximately bandlimited instead of strictly bandlimited. Therefore, several researchers have proposed reconstruction algorithms based on the smoothness assumption of graph signals [38]–[40], where the smoothness is measured with a Laplacian function. Similarly, other studies have used the Total Variation of graph signals [41], or extensions of the concept of stationarity on graph signals [42], [43] for reconstruction.

For time-varying graph signals, scientists have developed notions of joint harmonic analysis linking together the time-domain signal processing techniques with GSP [44], while other researchers have proposed reconstruction algorithms assuming bandlimitedness of the signals at each instant [41], [45]. Most of these methods do not fully exploit the underlying temporal correlations of the time-varying graph signals. Qiu *et al.* [26] proposed an approach where the temporal correlations are captured with a temporal difference operator in the time-varying graph signal. However, Qiu's method may have slow

convergence because the optimization problem depends on the Laplacian matrix. In particular, the eigenvalue spread of the Hessian associated with their problem may be large, leading to poor condition numbers. Other studies have reported the undesirable effects in convergence due to the large eigenvalue spread when dealing with shift operators derived from the Laplacian or the adjacency matrix. For example, Hua *et al.* [28] used a method of second-order moments to solve the problem of slow convergence speed, and this approach showed improvement in performance at the expense of additional computational cost by expensive matrix inversion operations. We can also circumvent the problem of slow convergence by using energy-preserving shift operators as in [27]. For example, Xia *et al.* [29] exploited an energy-preserving shift operator for distributed learning problems over graphs. However, the energy-preserving shift operators require the eigendecomposition of the adjacency (or Laplacian) matrix, which is computational prohibitive for large graphs.

In the current work, we assume smoothness in the temporal differences of graph signals as in [26]. However, our algorithm improves the condition number of the Hessian associated with our problem without requiring expensive eigenvalue decompositions or matrix inversions. We show that GraphTRSS converges faster than Qiu's method [26] under some conditions.

III. TIME VARYING GRAPH SIGNALS

A. Notation

In this paper, calligraphic letters such as \mathcal{V} denote sets, and $|\mathcal{V}|$ represents its cardinality. Uppercase boldface letters such as \mathbf{W} represent matrices, and lowercase boldface letters such as \mathbf{y} denote vectors. \mathbf{I} is the identity matrix, and $\mathbf{1}$ is a vector of ones with appropriate dimensions. The Hadamard and Kronecker products between matrices are denoted by \circ and \otimes , respectively. $(\cdot)^T$ represents transposition. The vectorization of matrix \mathbf{A} is denoted as $\text{vec}(\mathbf{A})$, and $\text{diag}(\mathbf{x})$ is the diagonal matrix with entries $\{x_1, x_2, \dots, x_N\}$ as its diagonal elements. The ℓ_2 -norm of a vector is written as $\|\cdot\|_2$. $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ represent the maximum and minimum eigenvalues of matrix \mathbf{A} , respectively. Finally, $\kappa(\cdot)$ is the condition number and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

B. Preliminaries

A graph can be represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ is the set of nodes, and $\mathcal{E} = \{(i, j)\}$ is the set of edges. Each element in \mathcal{E} represents a connection between the vertices i and j . The adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ represents the structure of the graph, where $\mathbf{W}(i, j) > 0 \forall (i, j) \in \mathcal{E}$ represents the weight of the connection between the vertices i and j . In the current work, we consider G as a connected, undirected, and weighted graphs. The adjacency matrix is symmetric when we have undirected graphs, *i.e.*, the weights between (i, j) and (j, i) are the same. The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix such that $\mathbf{D} = \text{diag}(\mathbf{W}\mathbf{1})$, *i.e.*, each element $\mathbf{D}(i, i)$ of the diagonal is the sum of all edge weights incident at the i th node. In this work, we use the combinatorial Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Since \mathbf{L} is a positive semi-definite matrix, it has eigenvalues

$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ and corresponding eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$. Finally, a graph signal is a function that maps a set of nodes to the real values $x : \mathcal{V} \rightarrow \mathbb{R}$, and therefore we can represent a static graph signal as $\mathbf{x} \in \mathbb{R}^N$ where $\mathbf{x}(i)$ is the value of the graph signal at the i th node.

The graph Fourier operator is defined by the eigenvalue decomposition of the Laplacian matrix $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top$ [4], where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]$ and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$. One can associate λ_i of \mathbf{L} with the frequency of the i th eigenvalue [4]. The Graph Fourier Transform (GFT) $\hat{\mathbf{x}}$ of the graph signal \mathbf{x} can be defined as $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$, and similarly, the inverse GFT is given by $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$ [4]. A graph signal \mathbf{x} is called bandlimited if we can express it with a few spectral components, *i.e.*, \mathbf{x} is bandlimited if $\exists \rho \in \{1, 2, \dots, N-1\}$ such that its GFT satisfies $\hat{\mathbf{x}}(i) = 0 \forall i > \rho$. Several authors have proved that we can perfectly reconstruct a graph signal from at least ρ samples if \mathbf{x} is bandlimited [23], [37]. However, real-world graph signals tend to be approximately bandlimited, and therefore researchers have focused on using approximately smooth Laplacian functions for the reconstruction of static and time-varying graph signals.

C. Reconstruction of Time-Varying Graph Signals

The sampling and reconstruction of graph signals have played a central role in the study of GSP [23], [24], [32], [35]. Several studies have used the smoothness assumption in graphs to solve problems of reconstruction and sampling of graph signals. Formally, notions of smoothness in \mathbf{x} have been introduced with concepts of local variation [2]. One can define the local variation of a graph signal \mathbf{x} at node i as:

$$\|\nabla_i \mathbf{x}\|_2 \triangleq \left[\sum_{j \in \mathcal{N}_i} \mathbf{W}(i, j)[\mathbf{x}(j) - \mathbf{x}(i)]^2 \right]^{\frac{1}{2}}, \quad (1)$$

where \mathcal{N}_i is the set of neighbors of node i . Moreover, one can introduce the discrete p -Dirichlet form for notions of global smoothness as $S_p(\mathbf{x}) \triangleq \frac{1}{p} \sum_{i \in \mathcal{V}} \|\nabla_i \mathbf{x}\|_2^p$, then:

$$S_p(\mathbf{x}) = \frac{1}{p} \sum_{i \in \mathcal{V}} \left[\sum_{j \in \mathcal{N}_i} \mathbf{W}(i, j)[\mathbf{x}(j) - \mathbf{x}(i)]^2 \right]^{\frac{p}{2}}. \quad (2)$$

For example, when $p = 2$, we have $S_2(\mathbf{x})$ which is known as the graph Laplacian quadratic form [2]:

$$S_2(\mathbf{x}) = \sum_{(i, j) \in \mathcal{E}} \mathbf{W}(i, j)[\mathbf{x}(j) - \mathbf{x}(i)]^2 = \mathbf{x}^\top \mathbf{L} \mathbf{x}. \quad (3)$$

Notice that $S_2(\mathbf{x}) = 0 \iff \mathbf{x} = \tau \mathbf{1}$, with τ a constant. The Laplacian quadratic form $S_2(\mathbf{x})$ is small when we have slight variations of the graph signal across the nodes in the graph, *i.e.*, a smooth graph signal. Some researchers have used the graph Laplacian quadratic form as a regularization term to solve problems of reconstruction of graph signals [46]. For time-varying graph signals, an extension of the graph Laplacian quadratic form has also been used for reconstruction [26].

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$ be a time-varying graph signal, where $\mathbf{x}_i \in \mathbb{R}^N$ is a graph signal in G at time i . One can

extend the concept of graph Laplacian quadratic form to time-varying graph signals by summing the Laplacian quadratic form of each graph signal \mathbf{x}_i . Then, we have:

$$S_2(\mathbf{X}) = \sum_{i=1}^M \mathbf{x}_i^\top \mathbf{L} \mathbf{x}_i = \text{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}). \quad (4)$$

However, notice that (4) does not have temporal relationships between graph signals at different times i . To include time information, Qiu *et al.* [26] introduced the temporal difference operator $\mathbf{D}_h \in \mathbb{R}^{M \times (M-1)}$ defined as follows:

$$\mathbf{D}_h = \begin{bmatrix} -1 & & & \\ 1 & -1 & & \\ & 1 & \ddots & \\ & & \ddots & -1 \\ & & & 1 \end{bmatrix} \in \mathbb{R}^{M \times (M-1)}. \quad (5)$$

As a consequence, the temporal difference graph signal is such that:

$$\mathbf{X} \mathbf{D}_h = [\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_2, \dots, \mathbf{x}_M - \mathbf{x}_{M-1}]. \quad (6)$$

Qiu *et al.* [26] noted that $S_2(\mathbf{X} \mathbf{D}_h)$ exhibits better smoothness properties compared to $S_2(\mathbf{X})$ in real-world datasets for time-varying graph signals, *i.e.*, the difference signal $\mathbf{x}_i - \mathbf{x}_{i-1}$ exhibits smoothness in the graph even if the signal \mathbf{x}_i is not smooth across the graph. They defined the structure of time-varying graph signals as follows [26]:

Definition 1. *The α -structured set $\mathcal{B}_\alpha(G)$ composed of smoothly evolving graph signals is defined as:*

$$\mathcal{B}_\alpha(G) = \{\mathbf{X} : \text{tr}((\mathbf{X} \mathbf{D}_h)^\top \mathbf{L} \mathbf{X} \mathbf{D}_h) \leq (M-1)\alpha\}, \quad (7)$$

where α indicates the smoothness level of the time-varying graph signal.

Qiu *et al.* [26] used Definition 1 to introduce a Time-varying Graph Signal Reconstruction (TGSR) method as follows:

$$\min_{\tilde{\mathbf{X}}} \frac{1}{2} \|\mathbf{J} \circ \tilde{\mathbf{X}} - \mathbf{Y}\|_F^2 + \frac{v}{2} \text{tr}((\tilde{\mathbf{X}} \mathbf{D}_h)^\top \mathbf{L} \tilde{\mathbf{X}} \mathbf{D}_h), \quad (8)$$

where $\mathbf{J} \in \{0, 1\}^{N \times M}$ is the sampling matrix of \mathbf{X} , v is a regularization parameter, and $\mathbf{Y} \in \mathbb{R}^{N \times M}$ is the matrix of signals that we know (the observed values). The sampling matrix \mathbf{J} is defined as follows:

$$\mathbf{J}(i, j) = \begin{cases} 1 & \text{if } i \in \mathcal{S}_j, \\ 0 & \text{if } i \notin \mathcal{S}_j, \end{cases} \quad (9)$$

where \mathcal{S}_j is the set of sampled nodes at column j .

Theorem 1 (Qiu *et al.* [26]). *The solution of (8) is unique when the following conditions are satisfied by the sampling matrix \mathbf{J} :*

- 1) For any $n \in \{1, \dots, N\}$, $\exists m \in \{1, \dots, M\}$ such that $\mathbf{J}(n, m) = 1$.
- 2) There is a fiducial time $m_0 \in \{1, \dots, M\}$, such that for any $m \in \{1, \dots, M\}$, with $m \neq m_0$, there exist a node $n_m \in \{1, \dots, N\}$ satisfying that $\mathbf{J}(n_m, m_0) = \mathbf{J}(n_m, m) = 1$.

Proof: see [26].

Theorem 1 provides some properties that \mathbf{J} should satisfy to obtain a unique solution for (8). Notice that deterministic sampling methods [23], [24], [47], [48] or sampling on product graphs [49] do not satisfy the first condition of Theorem 1, so it is not suitable for these problems. In contrast, in the current work we propose a random sampling strategy as in [26].

Eqn. (8) reconstructs a time-varying graph signal $\tilde{\mathbf{X}}$ with 1) a small error $\|\mathbf{J} \circ \tilde{\mathbf{X}} - \mathbf{Y}\|_F^2$, and 2) a small value of the temporal difference graph signal smoothness $\text{tr}((\tilde{\mathbf{X}}\mathbf{D}_h)^T \mathbf{L} \tilde{\mathbf{X}}\mathbf{D}_h)$. Finally, the parameter v in (8) weights the importance between the error and smoothness terms. This parameter v is usually tuned experimentally.

In the current work, we explore two additional temporal difference operators: a two-time steps operator and a three-time steps operator. The mathematical definitions of these operators are given in Section I of the supplementary material.

IV. SOBOLEV SMOOTHNESS OF TIME-VARYING GRAPH SIGNALS

In the current work, we extend the definition of Sobolev norms in GSP [6], [30] from static graph signals to a smoothness function for time-varying graph signals, and then we formulate a new reconstruction algorithm. The Sobolev norm was defined by Pesenson [30] as follows:

Definition 2 (Sobolev norm). *Let \mathbf{L} and \mathbf{x} be the Laplacian matrix and graph signal, respectively. For fixed parameters $\epsilon \geq 0$ and $\beta \in \mathbb{R}^+$, the Sobolev norm is given as follows:*

$$\|\mathbf{x}\|_{\beta, \epsilon} \triangleq \|(\mathbf{L} + \epsilon \mathbf{I})^{\beta/2} \mathbf{x}\|_2. \quad (10)$$

When \mathbf{L} is symmetric (as is the case in this work), we have that:

$$\|\mathbf{x}\|_{\beta, \epsilon}^2 = \mathbf{x}^T (\mathbf{L} + \epsilon \mathbf{I})^\beta \mathbf{x}. \quad (11)$$

Notice the Sobolev norm in (11) is equal to the Laplacian quadratic form in (3) when $\epsilon = 0$ and $\beta = 1$. We use the Sobolev norm to define a new smoothness function for time-varying graph signals as follows:

Definition 3 (Sobolev smoothness of time-varying graph signals). *Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$ be a time-varying graph signal, let \mathbf{D}_h be the temporal difference operator, and let \mathbf{L} be the combinatorial Laplacian matrix of a graph G . For fixed parameters $\epsilon \geq 0$ and $\beta \in \mathbb{R}^+$, the Sobolev smoothness of \mathbf{X} is given as follows:*

$$\begin{aligned} S_{\beta, \epsilon}(\mathbf{X}) &\triangleq \sum_{i=2}^M (\mathbf{x}_i - \mathbf{x}_{i-1})^T (\mathbf{L} + \epsilon \mathbf{I})^\beta (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \text{tr} ((\mathbf{X}\mathbf{D}_h)^T (\mathbf{L} + \epsilon \mathbf{I})^\beta (\mathbf{X}\mathbf{D}_h)). \end{aligned} \quad (12)$$

A. Sobolev Reconstruction

We use the Sobolev smoothness in Definition 3 to formulate two new reconstruction algorithms. The first algorithm solves the problem for the noiseless case, while the second algorithm solves the noisy case.

In the noiseless case, we assume that the sampling mechanism does not add noise to the problem. As a consequence, the observed graph signal is given by $\mathbf{Y} = \mathbf{J} \circ \tilde{\mathbf{X}}$. One

can formulate an optimization problem to get an approximate reconstruction of the time-varying graph signal as follows:

$$\min_{\tilde{\mathbf{X}}} \frac{1}{2} \text{tr} ((\tilde{\mathbf{X}}\mathbf{D}_h)^T (\mathbf{L} + \epsilon \mathbf{I})^\beta \tilde{\mathbf{X}}\mathbf{D}_h) \text{ s.t. } \mathbf{J} \circ \tilde{\mathbf{X}} = \mathbf{Y}. \quad (13)$$

The optimization function in (13) reconstructs a smooth spatiotemporal graph signal $\tilde{\mathbf{X}}$ given the constraint $\mathbf{Y} = \mathbf{J} \circ \tilde{\mathbf{X}}$. The noiseless case can be solved by a gradient projection algorithm. The iterative update is such that:

$$\tilde{\mathbf{X}}^{t+1} = \left(\tilde{\mathbf{X}}^t - \xi \nabla_{\tilde{\mathbf{X}}} f_s(\tilde{\mathbf{X}}^t) \right)^+, \quad (14)$$

where $f_s(\tilde{\mathbf{X}}^t) = \frac{1}{2} \text{tr} ((\tilde{\mathbf{X}}^t\mathbf{D}_h)^T (\mathbf{L} + \epsilon \mathbf{I})^\beta \tilde{\mathbf{X}}^t\mathbf{D}_h)$, ξ is the step size, $\nabla_{\tilde{\mathbf{X}}} f_s(\tilde{\mathbf{X}}^t)$ is the gradient of function $f_s(\tilde{\mathbf{X}}^t)$ given as:

$$\nabla_{\tilde{\mathbf{X}}} f_s(\tilde{\mathbf{X}}^t) = (\mathbf{L} + \epsilon \mathbf{I})^\beta \tilde{\mathbf{X}}^t \mathbf{D}_h \mathbf{D}_h^T, \quad (15)$$

and $(\mathbf{V})^+$ is the projection of signal \mathbf{V} to the signal space $\mathbf{Y} = \mathbf{J} \circ \tilde{\mathbf{X}}$ given as follows [26]:

$$(\mathbf{V})^+ = \mathbf{Y} + \mathbf{V} - \mathbf{J} \circ \mathbf{V}. \quad (16)$$

The above formulation can be extended to the noisy case as follows. We take into account the noise by relaxing the constraint in (13):

$$\min_{\tilde{\mathbf{X}}} \frac{1}{2} \|\mathbf{J} \circ \tilde{\mathbf{X}} - \mathbf{Y}\|_F^2 + \frac{v}{2} \text{tr} ((\tilde{\mathbf{X}}\mathbf{D}_h)^T (\mathbf{L} + \epsilon \mathbf{I})^\beta \tilde{\mathbf{X}}\mathbf{D}_h). \quad (17)$$

The optimization problem in (17) reconstructs a time-varying graph signal with a small error (given by the first term) and a small value for the Sobolev smoothness of its temporal difference signal. Here, we assume that the difference of graph signals $\mathbf{X}\mathbf{D}_h$ presents better smoothness properties than \mathbf{X} alone.

In the current work, we solve (8) and (17) using the conjugate gradient method as in [26]. Therefore, we update the search direction $\Delta \tilde{\mathbf{X}}^t$ and the step size μ on each iteration t as follows:

$$\mu = -\frac{\langle \Delta \tilde{\mathbf{X}}^t, \nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}^t) \rangle}{\langle \Delta \tilde{\mathbf{X}}^t, \nabla_{\tilde{\mathbf{X}}} f_u(\Delta \tilde{\mathbf{X}}^t) + \mathbf{Y} \rangle}, \quad (18)$$

where $\nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}) = \mathbf{J} \circ \tilde{\mathbf{X}} - \mathbf{Y} + v \mathbf{L} \tilde{\mathbf{X}} \mathbf{D}_h \mathbf{D}_h^T$ to solve (8), or $\nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}) = \mathbf{J} \circ \tilde{\mathbf{X}} - \mathbf{Y} + v (\mathbf{L} + \epsilon \mathbf{I})^\beta \tilde{\mathbf{X}} \mathbf{D}_h \mathbf{D}_h^T$ to solve (17). Using μ , we have that $\tilde{\mathbf{X}}^{t+1} = \tilde{\mathbf{X}}^t + \mu \Delta \tilde{\mathbf{X}}^t$, where $\Delta \tilde{\mathbf{X}}^t = -\nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}^t) + \gamma \Delta \tilde{\mathbf{X}}^{t-1}$ and $\gamma = \frac{\|\nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}^t)\|_F^2}{\|\nabla_{\tilde{\mathbf{X}}} f_u(\tilde{\mathbf{X}}^{t-1})\|_F^2}$. The stopping condition is given either by achieving a maximum number of iterations or when:

$$\|\Delta \tilde{\mathbf{X}}^t\|_F \leq \delta, \quad (19)$$

where $\delta = 10^{-6}$ in the experiments for both (8) and (17).

Fig. 1 shows the pipeline of our algorithm applied to a graph with the regions in the world with confirmed cases of COVID-19 by November 18, 2020. The advantage of the Sobolev minimization in (17) is its fast convergence with respect to the Laplacian minimization in (8). In the following section, we analyze some properties of the convergence rate of the optimization problems described in (8) and (17).

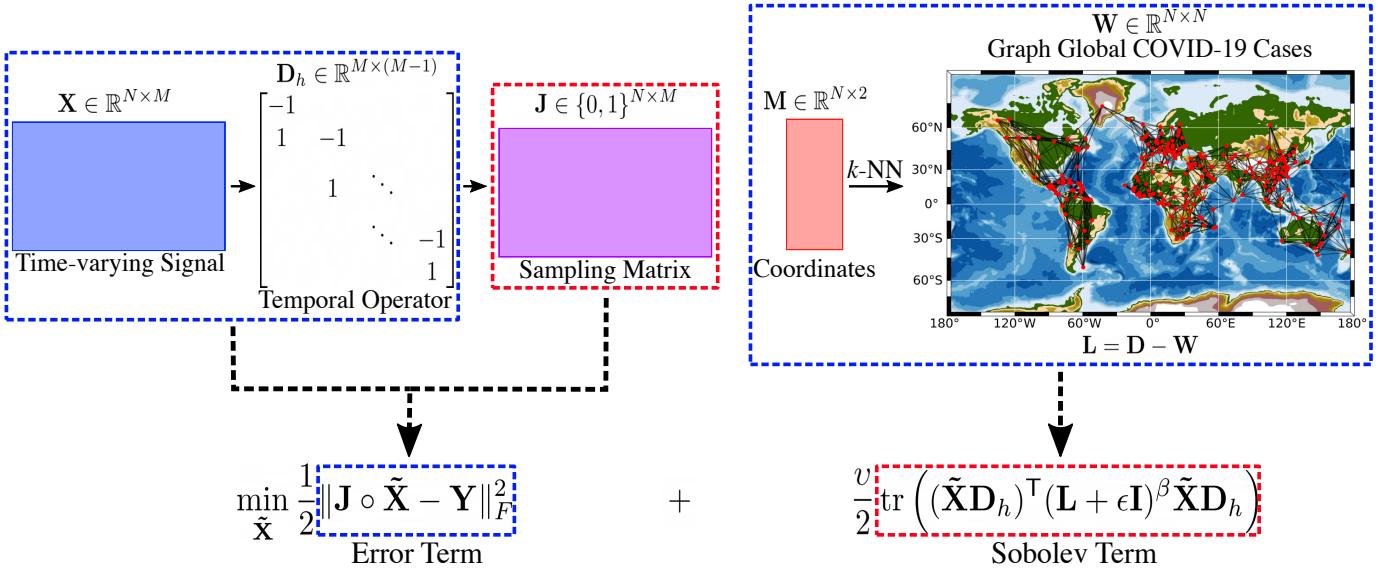


Fig. 1: The framework of our algorithm (GraphTRSS) using a matrix of coordinates $\mathbf{M} \in \mathbb{R}^{N \times 2}$ to construct a graph with N regions in the world with confirmed cases of COVID-19 by November 18, 2020. The graph is constructed with a k -Nearest Neighbors (k -NN) method. GraphTRSS uses the operator $\mathbf{D}_h \in \mathbb{R}^{M \times (M-1)}$ to capture temporal information in the time-varying signal $\mathbf{X} \in \mathbb{R}^{N \times M}$ with M temporal snapshots, and it also uses different sampling strategies $\mathbf{J} \in \{0, 1\}^{N \times M}$ according to the desired output (reconstruction or forecasting). Finally, the optimization function, which includes the error and Sobolev terms, reconstructs or predicts the missing values, *i.e.*, the indexes of \mathbf{X} where \mathbf{J} has values zero.

B. Convergence Rate

Intuitively, the convergence of a gradient descent method is faster when we have well-conditioned optimization problems. Interested readers are referred to Section II in the supplementary material to see a toy example. Formally, the rate of convergence of a gradient descent method is at best linear. We can accelerate this rate if we reduce the condition number of the Hessian associated with the objective function of the problem [50].

Theorem 2. Let $\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}) = \mathbf{Q} + [v(\mathbf{D}_h \mathbf{D}_h^T) \otimes (\mathbf{L} + \epsilon \mathbf{I})^\beta]$ and $\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}) = \mathbf{Q} + [v(\mathbf{D}_h \mathbf{D}_h^T) \otimes \mathbf{L}]$ be the Hessian associated with the Sobolev and Laplacian time-varying graph signal reconstruction problems in (17) and (8), respectively, where $\mathbf{Q} = \text{diag}(\text{vec}(\mathbf{J})) \in \mathbb{R}^{MN \times MN}$ and $\mathbf{J} \neq \mathbf{0}$. We have that:

- 1) If $\lambda_N, \lambda_{(D)N} \geq 1$ then $0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \leq (\lambda_N + \epsilon)^\beta \lambda_{(D)N}$ and $0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \leq \lambda_N \lambda_{(D)N}$, where $\lambda_{(D)N} = \lambda_{\max}(\mathbf{D}_h \mathbf{D}_h^T)$.
- 2) When $v \rightarrow \infty$, $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$ and $\kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \rightarrow \infty$.
- 3) When $\epsilon \rightarrow \infty$ and $\beta > 0$, or when $\beta \rightarrow \infty$ and $\lambda_N + \epsilon > 1$, $\lambda_{\max}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$ and so $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$.

Proof: See Appendix A.

The main result of Theorem 2 is that the upper bound of the minimum eigenvalue for the Hessian associated with GraphTRSS is looser than the corresponding upper bound of TGSR in [26] for $\epsilon > 0$, $\beta > 1$, and $\lambda_N + \epsilon \geq 1$, which favors better condition numbers for the Sobolev problem. In practice, there is a range of values of ϵ and β where $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) < \kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}))$, and then the GraphTRSS converges faster than the problem described in (8). We compute the condition

numbers of the Hessian of both GraphTRSS and TGSR in four real-world datasets in Section VI, Fig. 10 to show that indeed $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) < \kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}))$ in a large range of ϵ values. Theorem 2 also shows some intuitions on how to choose the parameters for GraphTRSS and TGSR regarding convergence rate in results 2) and 3). For example, we should keep small values of v , ϵ , and β to avoid harming the convergence rate of these algorithms. Similarly, we should maintain small values of ϵ and β to get benefits in the convergence rate of GraphTRSS.

The final question is about the influence of parameter β in the Sobolev reconstruction algorithm in (17). Intuitively, the parameter β is changing the shape of the frequencies of \mathbf{L} . For simplicity, let us assume $\epsilon = 0$, and let us consider only the time-varying graph signal \mathbf{X} without the temporal difference operator \mathbf{D}_h in (17). Since the matrix of eigenvectors of \mathbf{L} is an orthogonal matrix ($\mathbf{U}^T \mathbf{U} = \mathbf{I}$), we have:

$$\text{tr}(\mathbf{X}^T \mathbf{L}^\beta \mathbf{X}) = \text{tr}(\hat{\mathbf{X}}^T \Lambda^\beta \hat{\mathbf{X}}) = \sum_{t=1}^M \sum_{i=1}^N \hat{x}_t^2(i) \lambda_i^\beta, \quad (20)$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_M] = \mathbf{U}^T \mathbf{X}$. Eqn. (20) is penalizing each Fourier term of each $\hat{\mathbf{x}}_t$ with powers of the eigenvalues of \mathbf{L} . For example, when $\beta = 1$, we penalize the higher frequencies stronger than the lower frequencies of each $\hat{\mathbf{x}}_t$, *i.e.*, a smooth function in G . For $\beta > 1$, we also obtain a smooth function, but the penalization in the higher frequencies is more prominent, while for $\beta < 1$, we get more penalization in the lower frequencies. Fig. 2 shows the eigenvalues of \mathbf{L}^β for several values of β , where we normalized the eigenvalues such that $(\lambda_i / \lambda_N)^\beta \forall i \in \mathcal{V}$ for visualization purposes.

There are two important aspects for choosing the parameters ϵ and β when using (17). From a graph topological point of

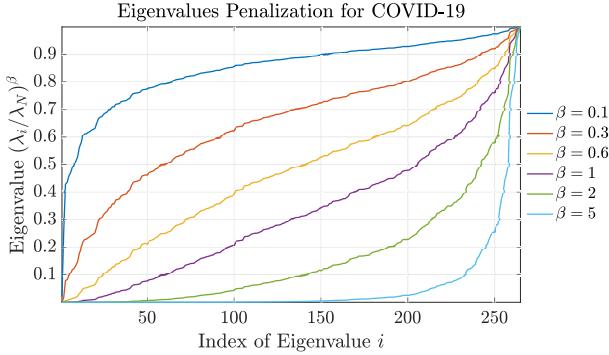


Fig. 2: Eigenvalue penalization of the Laplacian matrix for different values of β from the dataset of COVID-19.

view, we notice that adding $\epsilon\mathbf{I}$ to the Laplacian adds self-loops at the vertices in the graph with weights ϵ , and from Theorem 2, we notice that the value of ϵ should be small to avoid harming $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$. Therefore, it is reasonable to maintain a small value of ϵ so that the graph topology is not heavily modified and also to avoid harming $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$. Secondly, we also know from Theorem 2 that choosing a large value of β harms $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$. In practice, different datasets can benefit from specific penalization shapes like the ones in Fig. 2. As a result, we set $\beta = 1$ in the experiments to maintain a good $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$ and be as fair as possible on the comparison with Qiu's method [26]. However, we do additional experiments where we see that different values of β can improve the performance of (17) in different datasets, *i.e.*, different datasets require different assumptions. Finally, we can conclude that TGSR by Qiu *et al.* [26] is a specific case of GraphTRSS when $\epsilon = 0$ and $\beta = 1$.

V. EXPERIMENTAL FRAMEWORK

This section presents the datasets used in this work and the experimental framework details. We divide our experiments into three parts: 1) synthetic dataset, 2) COVID-19 datasets, and 3) environmental datasets. The graph G can be constructed based on the coordinate locations of the nodes in each dataset with k -NN algorithm. Let $\mathbf{M} \in \mathbb{R}^{N \times 2}$ be the matrix of coordinates of all nodes such that $\mathbf{M} = [\mathbf{m}_1, \dots, \mathbf{m}_N]^T$, where $\mathbf{m}_i \in \mathbb{R}^2$ is the vector with the latitude and the longitude of node i . The weights of each edge (i, j) are given by the Gaussian kernel:

$$\mathbf{W}(i, j) = \exp\left(-\frac{\|\mathbf{m}_i - \mathbf{m}_j\|_2^2}{\sigma^2}\right), \quad (21)$$

where $\|\mathbf{m}_i - \mathbf{m}_j\|_2^2$ is the Euclidean distance between the vertices i and j , and σ is the standard deviation given by $\sigma = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{m}_i - \mathbf{m}_j\|_2$. The Gaussian kernel in (21) assigns higher weights to geographically-close locations and vice versa. The construction of the graph depends on the specific application, and GraphTRSS is not sensitive to the graph construction methods.

A. Datasets

Our algorithm is evaluated on one synthetic dataset and four real-world datasets including, 1) a synthetic graph, 2) global

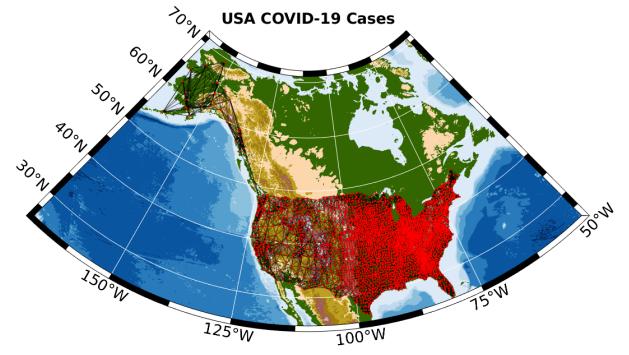


Fig. 3: Graph with the places in the United States in the Johns Hopkins University dataset [51]. The graph was constructed with a k -NN with $k = 10$.

COVID-19 cases [51], 3) USA COVID-19 cases [51], 4) mean concentration of Particulate Matter (PM) 2.5 [26], and 5) sea surface temperature [26].

1) Synthetic Graph and Signals: We use the synthetic dataset created by Qiu *et al.* [26], where 100 nodes are generated randomly from the uniform distribution in a 100×100 square area. Therefore, Qiu *et al.* [26] used a k -NN to construct the graph. The time-varying graph signal is generated with the recursive function $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{L}^{-\frac{1}{2}} \mathbf{f}_t$, where: 1) \mathbf{x}_1 is a low-frequency graph signal with energy 10^4 , 2) $\mathbf{L}^{-\frac{1}{2}} = \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^T$, where $\Lambda^{-\frac{1}{2}} = \text{diag}(0, \lambda_2^{-\frac{1}{2}}, \dots, \lambda_N^{-\frac{1}{2}})$, and 3) \mathbf{f}_t is an i.i.d. white Gaussian signal such that $\|\mathbf{f}_t\|_2 = \alpha$. As a result, the synthetic time-varying graph signal satisfies Definition 1.

2) Global COVID-19: We use the global COVID-19 dataset provided by the Johns Hopkins University [51]. This dataset contains the cumulative number of daily COVID-19 cases for 265 locations in the world (Fig. 1 shows the locations in the dataset), and we use the data between January 22, 2020, and November 18, 2020; *i.e.*, 302 days. The graph is constructed with the k -NN method with $k = 10$ using the coordinates of each place [4].

3) USA COVID-19: We also use the USA COVID-19 dataset provided by the Johns Hopkins University [51]. Fig. 3 shows the map with the points of the graph in the USA COVID-19 dataset. This dataset is more fine-grained than the global set since the data contain 3232 localities in the US. We use the same temporal window and k -NN method as in the global dataset. The experiments related to the US can give us a different view of our algorithm since several travel restrictions were applied between countries in the pandemic, while inside countries, there was slightly more freedom to move around.

4) Particulate Matter 2.5: In this work, we use the daily mean PM 2.5 concentration dataset from California provided by the US Environmental Protection Agency². We use the data captured daily from 93 sensors in California for the first 220 days in 2015.

5) Sea Surface Temperature: We use the sea surface temperature captured monthly and provided by the NOAA Phys-

²<https://www.epa.gov/outdoor-air-quality-data>

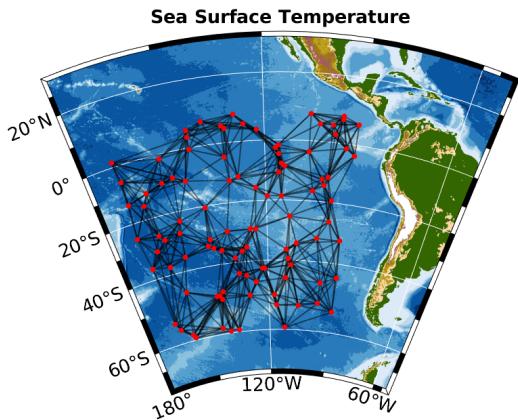


Fig. 4: Graph with the spots in the sea for the dataset of temperature. The graph was constructed with a k -NN with $k = 10$.

ical Sciences Laboratory (PSL) from their website³. In this work, we use the same experimental framework as in [26] for comparison purposes, *i.e.*, we use a subset of 100 points on the Pacific Ocean within a time frame of 600 months. Fig. 4 shows the locations of the spots in the sea for this dataset.

B. Evaluation Metrics

We use the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE) to compare GraphTRSS with the methods of the literature. $\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_x} (\hat{\mathbf{x}}_i - \mathbf{x}_i^*)^2}{N_x}}$, $\text{MAE} = \frac{\sum_{i=1}^{N_x} |\hat{\mathbf{x}}_i - \mathbf{x}_i^*|}{N_x}$, and $\text{MAPE} = \frac{1}{N_x} \sum_{i=1}^{N_x} \left| \frac{\mathbf{x}_i^* - \hat{\mathbf{x}}_i}{\mathbf{x}_i^*} \right|$, where $\hat{\mathbf{x}}$ is the recovered signal, \mathbf{x}^* is the ground truth signal, and N_x is the length of the signal.

C. Experiments

For the synthetic dataset, we perform experiments analyzing several sampling densities, Signal-to-Noise Ratio (SNR) levels, smoothness α of the synthetic time-varying graph signal, and the condition number of $\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})$ and $\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})$. For the real dataset, we perform several experiments for reconstruction and forecasting. In the same way, we study the convergence, the ϵ and β parameters, and the temporal difference operators for GraphTRSS. We compare GraphTRSS with Natural Neighbor Interpolation (NNI) by Sibson [52], Graph Regularization (GR) by Narang *et al.* [39], Tikhonov regularization by Perraudeau *et al.* [43], [53], Time-varying Graph Signal Reconstruction (TGSR) by Qiu *et al.* [26], and Random Sampling and Decoder (RSD) by Puy *et al.* [46]. NNI [52] is based on Voronoi tessellation, where the interpolation is given by a linear combination of the neighbor points of the element to be interpolated in the Voronoi partition. GR [39] can be viewed as the solution of (8) without the time component. Tikhonov regularization [53] is based on joint stationarity, where the solution is constrained to be smooth in the graph and in time. RSD method [46] solves an optimization

problem where we have smoothness in the graph and a probability distribution function for the sampling. The role of RSD and GR in the experiments is to see how these static graph signal methods compare to time-varying graph signal algorithms. Similarly, since RSD and GR cannot reconstruct graph signals where we do not have any sample at a specific time t , they are not included in some of the experiments as in forecasting. GR, Tikhonov, TGSR, and GraphTRSS have a maximum number of 20000 iterations in the optimization algorithm for a fair comparison. We optimize the parameters of each method in all experiments for a fair comparison.

In the reconstruction experiments, we adopt two sampling strategies. The first reconstruction experiment performs a random sampling on each temporal snapshots for the number of nodes N , *i.e.*, we randomly select a percentage of nodes on each time step as in [26]. For RSD by Puy *et al.* [46], we use their optimal sampling procedure because their reconstruction algorithm depends on their specific sampling distribution. The interested readers are referred to Section III in the supplementary material for further details about RSD. The second reconstruction experiment performs a random sampling of entire time-snapshots for the total number of graph signals M . Both experiments compute the error metrics for each method on the non-sampled nodes for a set of sampling densities \mathcal{M} . The set \mathcal{M} was chosen according to each dataset, for example, $\mathcal{M} = \{0.5, 0.6, \dots, 0.9, 0.995\}$ for COVID-19 datasets. Section IV in the supplementary material shows a small analysis of the bandwidths of \mathbf{XD}_h for COVID-19 global. We evaluate each method with a Monte Carlo cross-validation with 100 repetitions for each sampling density (50 repetitions for the COVID-19 USA dataset). Deterministic sampling methods do not work well with these reconstruction methods as stated in Section III-C. However, the interested readers are referred to Section V in the supplementary material for a small experiment with the deterministic sampling methods in [15], [23], [47].

In the forecasting experiment, we compute the error metrics for several temporal snapshots t in the set $\{1, 2, \dots, 10\}$. For example, since COVID-19 datasets are sampled daily, we will predict the new COVID-19 cases in the last day, in the two last days, and so on until the last ten days. However, for the sea surface temperature, we will predict up to ten months since this dataset is sampled monthly, *i.e.*, we are interpolating multiple time steps ahead in all datasets.

We perform some studies to analyze the parameters of GraphTRSS. The first of these studies computes the average RMSE (in \mathcal{M}) and the average number of iterations required to satisfy the stopping condition with variations in ϵ . The second study performs reconstruction with the three temporal difference operators (one, two, and three steps). The last study computes reconstruction with several values of β . These studies are evaluated with a Monte Carlo cross-validation with ten repetitions.

We also compare the loss function vs the iteration number for TGSR and GraphTRSS in reconstruction, where the loss is the evaluation of (8) and (17) at each iteration. In this case, 50 repetitions for each sampling densities in \mathcal{M} are performed, where each repetition is computed with the best parameters

³<https://psl.noaa.gov/>

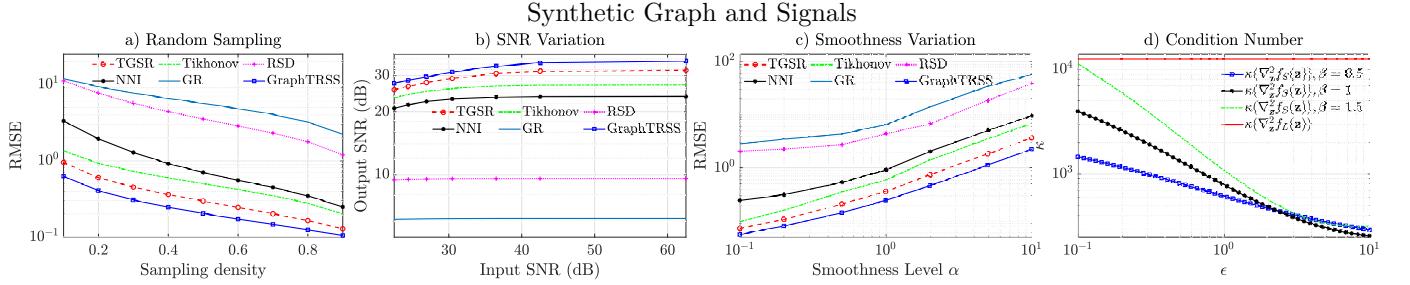


Fig. 5: Comparison of GraphTRSS with several methods in the literature on synthetic data for four experiments on: a) reconstruction with several sampling densities, b) variation of the SNR, c) variation of the smoothness level in (7), and d) $\kappa(\nabla_z^2 f_S(z))$ and $\kappa(\nabla_z^2 f_L(z))$ with several values of ϵ and β . Our algorithm is compared with Natural Neighbor Interpolation (NNI) [52], Graph Regularization (GR) [39], Tikhonov regularization [43], [53], Time-varying Graph Signal Reconstruction (TGSR) [26], and Random Sampling and Decoder (RSD) [46].

TABLE I: Summary of the average error metrics in the real datasets for several sampling schemes. The best and second-best performing methods on each category are shown in **red** and **blue**, respectively.

Method	Random Sampling											
	COVID-19 Global			COVID-19 USA			Sea Surface Temperature			PM 2.5 Concentration		
RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	
NNI (Sibson [52])	1555.14	189.15	4.23	54.92	10.90	2.17	0.77	0.56	0.07	4.95	2.96	0.59
GR (Narang <i>et al.</i> [39])	4744.25	1119.07	90.95	65.49	13.94	3.47	2.43	1.76	0.40	5.37	3.32	0.67
Tikhonov (Perraudin <i>et al.</i> [53])	1253.71	18.60	6.07	37.00	6.03	1.03	0.95	0.70	0.12	4.32	2.66	0.55
TGSR (Qiu <i>et al.</i> [26])	1136.39	213.96	11.30	33.51	6.14	1.19	0.36	0.26	0.03	3.90	2.28	0.39
RSD (Puy <i>et al.</i> [46])	2045.37	506.30	36.17	58.59	12.05	2.50	5.56	4.62	0.97	5.43	3.46	0.72
GraphTRSS (ours)	1134.15	152.76	2.41	33.47	5.96	1.10	0.36	0.26	0.03	3.83	2.21	0.38
Method	Entire Snapshots Sampling											
	COVID-19 Global			COVID-19 USA			Sea Surface Temperature			PM 2.5 Concentration		
RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	
NNI (Sibson [52])	1342.08	170.64	3.40	56.69	11.71	2.60	0.98	0.68	0.08	10.87	4.44	0.85
Tikhonov (Perraudin <i>et al.</i> [53])	1255.05	182.97	5.94	36.96	6.12	1.04	1.17	0.86	0.13	4.54	2.81	0.56
TGSR (Qiu <i>et al.</i> [26])	1459.45	788.10	100.06	36.38	13.75	4.74	19.19	19.15	2.13	10.94	9.31	1.49
GraphTRSS (ours)	1114.16	143.69	1.49	32.96	5.45	0.88	0.94	0.67	0.07	4.42	2.70	0.48
Method	Forecasting											
	COVID-19 Global			COVID-19 USA			Sea Surface Temperature			PM 2.5 Concentration		
RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	
NNI (Sibson [52])	16722.30	2371.30	6.43	904.23	283.23	19.54	3.04	2.23	0.23	30.71	17.15	2.76
Tikhonov (Perraudin <i>et al.</i> [53])	3612.06	1190.20	30.48	99.88	31.2	2.03	1.94	1.53	0.30	4.80	3.23	0.67
TGSR (Qiu <i>et al.</i> [26])	3313.50	2472.08	238.65	91.5	54.36	6.88	18.72	18.69	2.39	10.09	9.11	1.27
GraphTRSS (ours)	2416.30	583.36	4.80	75.33	20.65	0.86	1.20	0.98	0.13	4.36	2.83	0.54

of each method. For a fair comparison, GraphTRSS and Qiu's method use the same sampling matrix \mathbf{J} . We also compute the running times for several sampling densities. Finally, we compute $\kappa(\nabla_z^2 f_S(z))$ and $\kappa(\nabla_z^2 f_L(z))$ for $\beta = 1$ and several values of ϵ in all real datasets. All experiments were executed in MATLAB R2017b on a 2.3GHz MacBook Pro with 16GB memory. The code of GraphTRSS has been made available⁴.

VI. RESULTS AND DISCUSSION

This section presents the results of the experiments on the selected datasets. GR and RSD methods cannot reconstruct entire temporal snapshots because these algorithms only use spatial information. Therefore, GR and RSD are not included in the experiments of entire snapshots sampling and forecasting. A detailed analysis for each dataset is provided as follows.

A. Synthetic Graph and Signals

Fig. 5 shows the results in the synthetic dataset. GraphTRSS performs better than the existing methods for all values in

\mathcal{M} for the random sampling scheme, as shown in Fig. 5(a). Our algorithm is also more robust against noise than the compared methods, as shown in Fig. 5(b). Similarly, Fig. 5(c) shows the benefits of GraphTRSS regarding different levels of smoothness α as given in Definition 1. Our algorithm shows better performance than other time-varying graph signal methods as TGSR and other static graph signal reconstruction schemes as GR. Finally, Fig. 5(d) shows the condition numbers of the Hessian associated with GraphTRSS ($\kappa(\nabla_z^2 f_S(z))$) and TGSR ($\kappa(\nabla_z^2 f_L(z))$) with several values of ϵ and β . As noted from Theorem 2, we have that: 1) there is a range of values of $\epsilon > 0$ where $\kappa(\nabla_z^2 f_S(z)) < \kappa(\nabla_z^2 f_L(z))$, 2) $\epsilon > 0$ promotes good values of the condition number $\kappa(\nabla_z^2 f_S(z))$ for GraphTRSS, and 3) big values of β harm the condition number $\kappa(\nabla_z^2 f_S(z))$ of GraphTRSS.

B. Real Datasets Summary

Table I shows the summary of all error metrics for random, entire snapshots sampling, and forecasting in the real datasets. The results are computed as the mean of all results over the

⁴<https://github.com/jhonygiraldo/GraphTRSS>

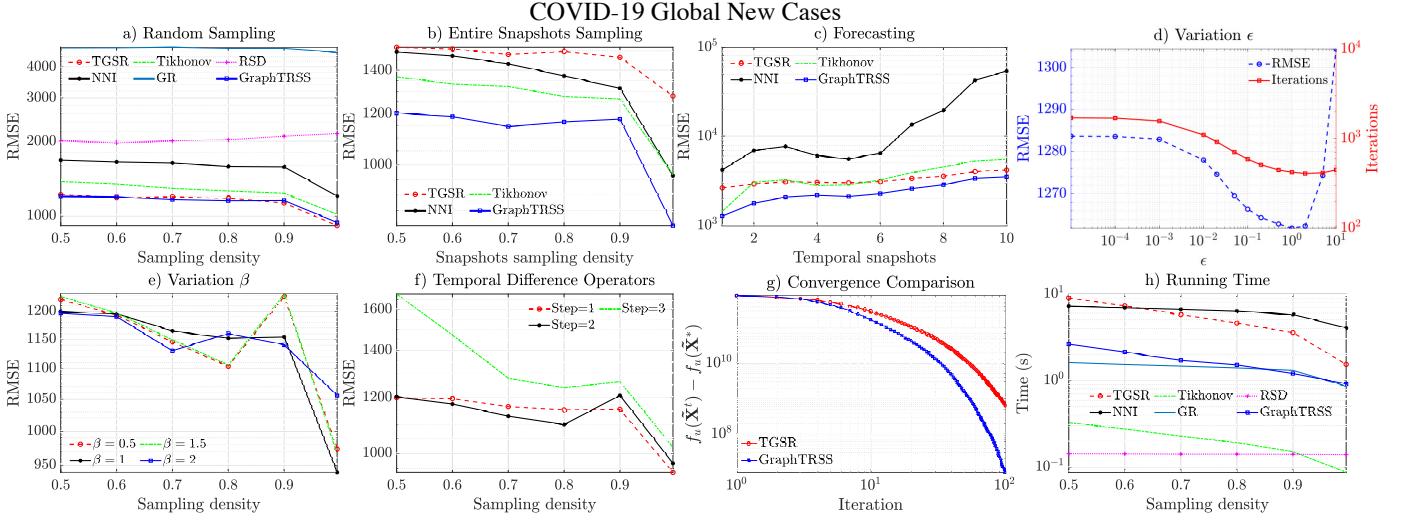


Fig. 6: Comparison of GraphTRSS with several methods in the literature on COVID-19 global dataset for several experiments in terms of: a) random sampling, b) entire snapshots sampling, c) forecasting, d) variation of parameter ϵ , e) variation of parameter β , f) several temporal difference operators, g) convergence comparison, and h) running time.

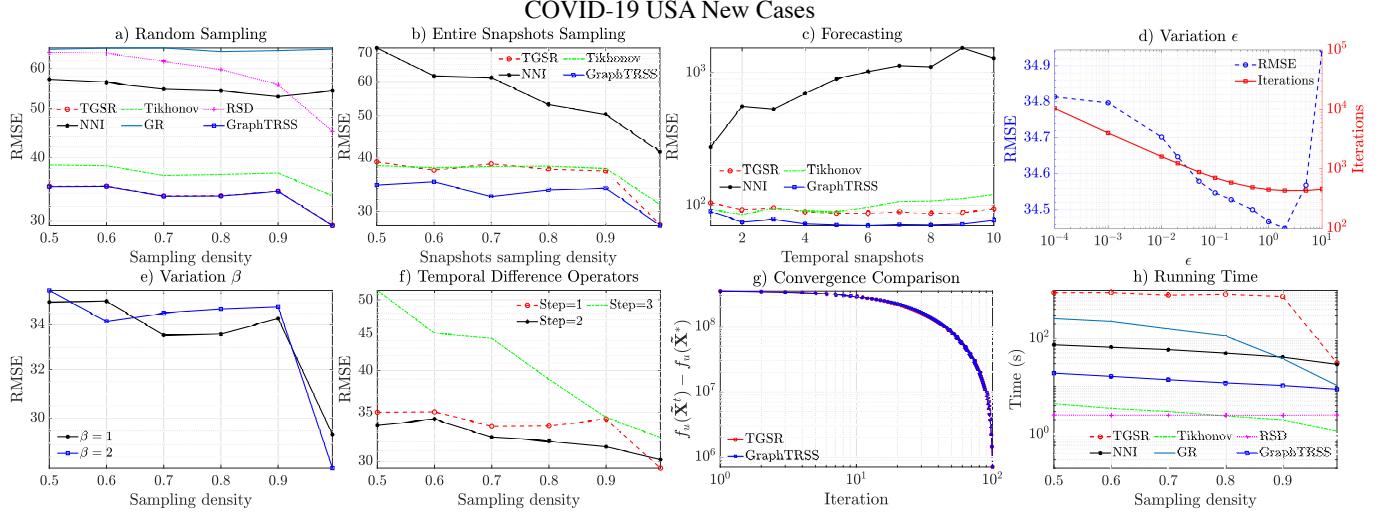


Fig. 7: Comparison of GraphTRSS with several methods in the literature on COVID-19 USA dataset for several experiments in terms of: a) random sampling, b) entire snapshots sampling, c) forecasting, d) variation of parameter ϵ , e) variation of parameter β , f) several temporal difference operators, g) convergence comparison, and h) running time.

set \mathcal{M} . GraphTRSS shows the best performance in almost all cases against the other methods.

C. COVID-19 Datasets

Fig. 6 and 7 show the results of the experiments in the COVID-19 datasets. GraphTRSS is better than the other methods for the reconstruction experiments when using different sampling strategies. We can notice that the results in the COVID-19 USA dataset have lower RMSE than the global dataset, *i.e.*, we can get better estimations if we have more fine-grained data as in the case of the COVID-19 USA dataset. From the experiment with entire temporal snapshots, we can notice that TGSR method [26] performs poorly compared to the other methods for what we observe in random sampling. This behavior is expected because entire snapshots sampling

does not satisfy the second condition of Theorem 1 for TGSR by Qiu *et al.* [26].

All tested methods, including ours, are designed to reconstruct time-varying graph signals, and therefore these methods do not have prior assumptions about the behavior in the time domain of the underlying processes. Correspondingly, in forecasting, we get weaker results than the performance shown in Fig. 6 and 7 for random and entire snapshots sampling. However, our algorithm readily shows better results than the other methods in both COVID-19 datasets, even forecasting multiple time steps ahead. Thus, the Sobolev function introduced in Definition 3 could potentially improve other graph-based forecasting methods of the literature [54], using proper time-domain prior assumptions or other machine learning strategies, which we leave for future work.

From the studies to analyze the parameters of GraphTRSS,

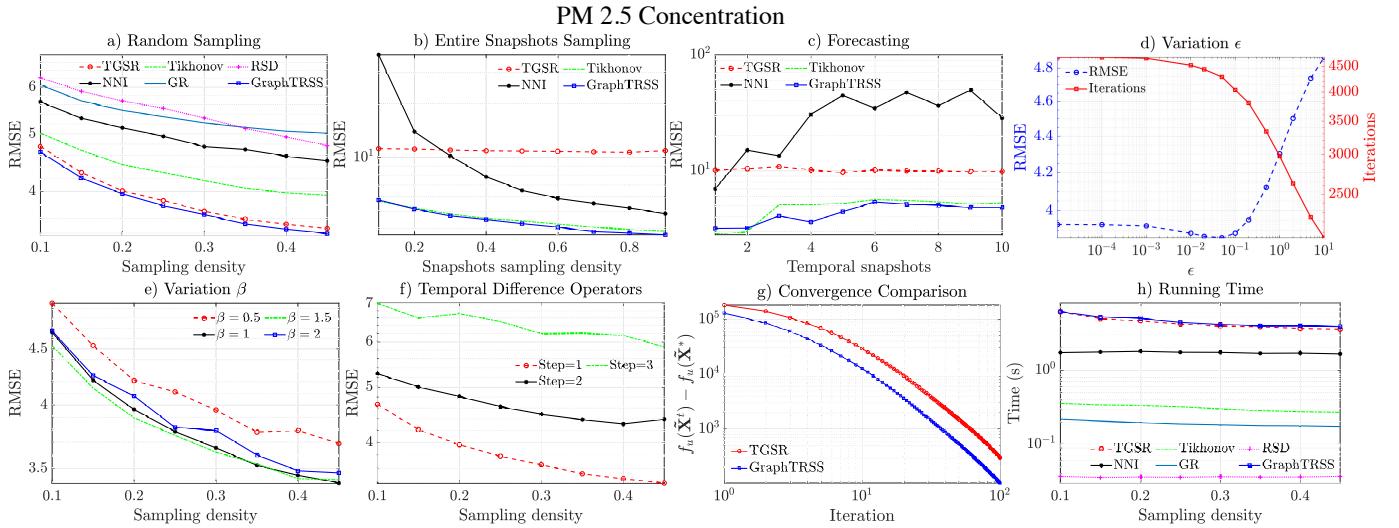


Fig. 8: Comparison of GraphTRSS with several methods in the literature in the PM 2.5 dataset for several experiments in terms: a) random sampling, b) entire snapshots sampling, c) forecasting, d) variation of parameter ϵ , e) variation of parameter β , f) several temporal difference operators, g) convergence comparison, and h) running time.

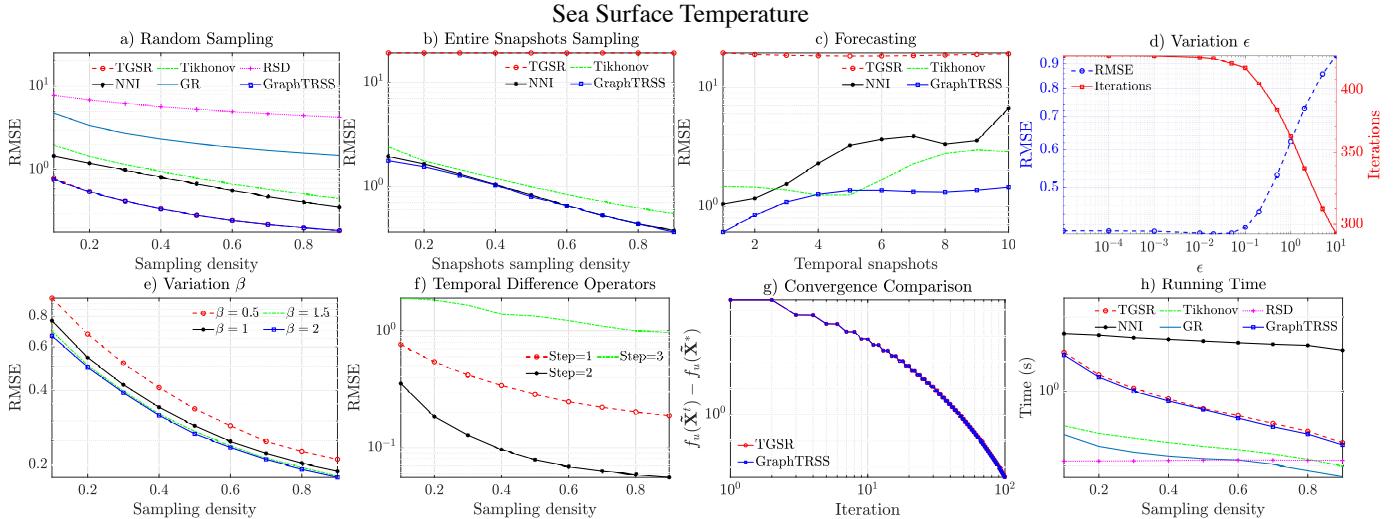


Fig. 9: Comparison of GraphTRSS with several methods in the literature in the sea surface temperature dataset for several experiments in terms of: a) random sampling, b) entire snapshots sampling, c) forecasting, d) variation of parameter ϵ , e) variation of parameter β , f) several temporal difference operators, g) convergence comparison, and h) running time.

one can notice that values of $\epsilon > 0$ improve the condition number; this is reflected in the number of iterations to satisfy the stopping condition as shown in Fig. 6(d) and 7(d) for COVID-19. However, one should be careful since big values of ϵ can heavily modify the structure of the graph leading to a degradation of the performance, as shown in Fig. 6(d) and 7(d) for the RMSE. For the variations of β and the temporal difference operators, we notice that values different from $\beta = 1$ and one time-step for the temporal difference operator might bring benefits for COVID-19 datasets.

Fig. 6(g) shows that GraphTRSS converges faster than TGSR method for COVID-19 global, where $\tilde{\mathbf{X}}^*$ is the solution of the optimization problem, and $\tilde{\mathbf{X}}^t$ is the solution at iteration t . Fig. 6(h) and 7(h) show the running time for each method. These experiments were computed with the best parameters of

each method from the random sampling experiment. Arguably, GraphTRSS shows the best compromise between accuracy and running time among several methods for COVID-19 datasets, as shown in Fig. 6 and 7.

D. Environmental Datasets

Fig. 8 and 9 show the results for the experiments in the environmental datasets. The analysis of the comparison between our algorithm and the other methods is mostly similar to the analysis for Fig. 6 and 7, *i.e.*, GraphTRSS shows the best compromise between accuracy and running time. However, we should notice two interesting results in Fig. 8 and 9. Firstly, we notice that the parameters $\beta = 1.5$ or $\beta = 2$ show better results than $\beta = 1$ in the environmental datasets. Secondly, the two-time steps temporal operator \mathbf{D}_{h2} shows a big performance

TABLE II: The average number of iterations to satisfy the stopping condition for GraphTRSS and TGSR [26].

Method	COVID-19 Global	COVID-19 USA	PM 2.5 Conc.	Sea Surface Temp.
TGSR [26]	1735.9	48659	4690.1	418.8
GraphTRSS	510.0	427	4266.6	416.0

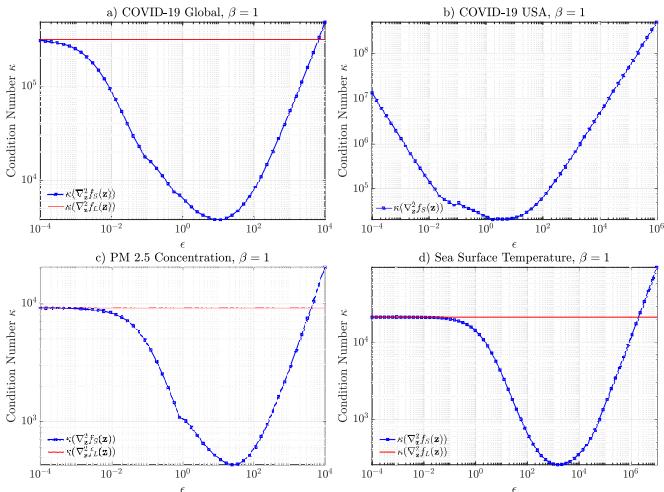


Fig. 10: Condition number of the Hessian associated with the optimization problems for GraphTRSS ($\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$) and TGSR [26] ($\kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}))$) for all datasets.

improvement compared to the one-step temporal operator for the sea surface temperature dataset. Perhaps, we could propose other temporal operators and more elaborated optimization functions based on several β parameters, which we leave for future work.

E. Additional Analysis

Table II shows the number of iterations that TGSR method and GraphTRSS require to satisfy the stopping condition in (19). Table II shows that GraphTRSS satisfies the stopping condition around 3.4 times faster than TGSR in global COVID-19; and about 114 times faster in USA COVID-19. We limit the maximum number of iterations of TGSR to 60000 for the USA COVID-19 dataset because of time limitations. The experiments with the lowest sampling densities would take more time to satisfy the stopping condition if we increase the maximum number of iterations for TGSR method in COVID-19 USA. This behavior is another undesirable effect of bad condition numbers.

Fig. 10 shows the condition number of the Hessian functions for the COVID and environmental datasets with random sampling with density 0.5. The condition number of the Hessian of (8) is not displayed in Fig. 10 for COVID-19 USA because, in that case, $\kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \rightarrow \infty$. Furthermore, as predicted by Theorem 2: 1) there is a range of values of ϵ where $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) < \kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}))$ and then we get benefits in convergence rate, and 2) $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}))$ grows quickly for large values of ϵ . Notice that the improvement in the condition number depends on the specific structure of each graph. Therefore, using the normalized Laplacian $\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$ is reasonable since $\lambda_{max}(\mathcal{L}) \leq 2$. The interested readers are

referred to Section VI in the supplementary material for a small experiment with normalized Laplacian matrices.

The parameters ϵ and v are optimized for each dataset to get good performance. However, we can notice in Fig. 10 that an ϵ value close to 10^{-1} is a good starting point since it improves the condition number, and it does not modify heavily the Laplacian matrix. Similar analysis about the parameter v can be found in [26]. In practice, the parameters of GraphTRSS can be optimized in a small part of the dataset.

VII. CONCLUSIONS

In this work, a new algorithm called GraphTRSS is introduced for the reconstruction of time-varying graph signals. The Sobolev norm is used to define a smoothness function for time-varying graph signals, and therefore this function is used to introduce a new optimization problem for reconstruction from samples. We showed the good convergence properties of GraphTRSS by relying on the condition number of the Hessian associated with our problem, as shown in Theorem 2. Our algorithm improves convergence rate without relying on expensive matrix inversions or eigenvalue decompositions. GraphTRSS shows promising results for the estimation of new COVID-19 cases in both global and USA datasets, as well as in the reconstruction of environmental variables (PM 2.5 concentration and sea surface temperature). GraphTRSS could be useful in several scenarios, for example, 1) one could try to estimate the number of new COVID-19 cases where we have missing or unreliable information, 2) we can use our algorithm as a pre-processing step to get more reliable data for further forecasting of new cases with other well-established models of infectious diseases [55], and 3) we can also use GraphTRSS as a pre-processing step for further forecasting of new COVID-19 cases with other graph-based techniques such as [54]. GraphTRSS is also evaluated directly on forecasting, showing promising results.

This work shows several insights for possible future directions. For example, our algorithm and other GSP techniques could potentially increase the performance of various models in infectious disease (not only for COVID-19) for forecasting and imputation of data, among others. In the same way, this work could be the foundation for applying other GSP-based methods in virology. These methods can be, for example, graph neural networks [20], [56] and learning graphs from data [57].

APPENDIX A PROOF THEOREM 2

Proof. The problem associated with the Laplacian matrix in (8) can be rewritten as follows [26]:

$$f_L(\mathbf{z}) = \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{Q}[\mathbf{z} - \text{vec}(\mathbf{Y})]\|_2^2 + \frac{v}{2} \mathbf{z}^\top [(\mathbf{D}_h \mathbf{D}_h^\top) \otimes \mathbf{L}] \mathbf{z}, \quad (22)$$

where $\mathbf{z} = \text{vec}(\tilde{\mathbf{X}})$. Moreover, the gradient of $f_L(\mathbf{z})$ in (22) is such that:

$$\nabla_{\mathbf{z}} f_L(\mathbf{z}) = \mathbf{Q}[\mathbf{z} - \text{vec}(\mathbf{Y})] + v[(\mathbf{D}_h \mathbf{D}_h^\top) \otimes \mathbf{L}] \mathbf{z}, \quad (23)$$

and the Hessian matrix of $f_L(\mathbf{z})$ is given as follows:

$$\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}) = \mathbf{Q} + [v(\mathbf{D}_h \mathbf{D}_h^T) \otimes \mathbf{L}]. \quad (24)$$

Similarly, the Hessian matrix associated with the Sobolev formulation is given by:

$$\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}) = \mathbf{Q} + [v(\mathbf{D}_h \mathbf{D}_h^T) \otimes (\mathbf{L} + \epsilon \mathbf{I})^\beta]. \quad (25)$$

$\mathbf{D}_h \mathbf{D}_h^T$ is a positive semi-definite matrix, so it has a matrix of eigenvalues $\Lambda_D = \text{diag}(\lambda_{(D)1}, \lambda_{(D)2}, \dots, \lambda_{(D)N})$, with $0 \leq \lambda_{(D)1} \leq \lambda_{(D)2} \leq \dots \leq \lambda_{(D)N}$, and the corresponding matrix of eigenvectors \mathbf{U}_D . As a consequence, from (24), we have:

$$\begin{aligned} \nabla_{\mathbf{z}}^2 f_L(\mathbf{z}) &= \mathbf{Q} + [v(\mathbf{U}_D \Lambda_D \mathbf{U}_D^T) \otimes (\mathbf{U} \Lambda \mathbf{U}^T)], \\ \nabla_{\mathbf{z}}^2 f_L(\mathbf{z}) &= v \left[\frac{1}{v} \mathbf{Q} + (\mathbf{U}_D \otimes \mathbf{U})(\Lambda_D \otimes \Lambda)(\mathbf{U}_D^T \otimes \mathbf{U}^T) \right], \end{aligned} \quad (26)$$

where we used the property of Kronecker products $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ [58]. \mathbf{Q} is positive semi-definite because it is a diagonal matrix with diagonal elements either 0 or 1. As a result, from (26), we know that $\lambda_{\min}(\frac{1}{v} \mathbf{Q}) = 0$, $\lambda_{\max}(\frac{1}{v} \mathbf{Q}) = \frac{1}{v}$ if $\mathbf{J} \neq \mathbf{0}$, $\lambda_{\min}((\mathbf{D}_h \mathbf{D}_h^T) \otimes \mathbf{L}) = 0$, and $\lambda_{\max}((\mathbf{D}_h \mathbf{D}_h^T) \otimes \mathbf{L}) = \lambda_N \lambda_{(D)N}$ if $\lambda_N, \lambda_{(D)N} \geq 1$. For the Sobolev problem, we have that:

$$(\mathbf{L} + \epsilon \mathbf{I})^\beta = (\mathbf{U} \Lambda \mathbf{U}^T + \epsilon \mathbf{I})^\beta = \mathbf{U}(\Lambda + \epsilon \mathbf{I})^\beta \mathbf{U}^T, \quad (27)$$

and then:

$$\nabla_{\mathbf{z}}^2 f_S(\mathbf{z}) = v \left[\frac{1}{v} \mathbf{Q} + (\mathbf{U}_D \otimes \mathbf{U})(\Lambda_D \otimes (\Lambda + \epsilon \mathbf{I})^\beta)(\mathbf{U}_D^T \otimes \mathbf{U}^T) \right], \quad (28)$$

where $\lambda_{\min}((\mathbf{D}_h \mathbf{D}_h^T) \otimes (\mathbf{L} + \epsilon \mathbf{I})^\beta) = 0$ and $\lambda_{\max}((\mathbf{D}_h \mathbf{D}_h^T) \otimes (\mathbf{L} + \epsilon \mathbf{I})^\beta) = (\lambda_N + \epsilon)^\beta \lambda_{(D)N}$ if $(\lambda_N + \epsilon)^\beta, \lambda_{(D)N} \geq 1$. We have a summation of two Hermitian matrices in (26) and (28) (we can trivially prove that (26) and (28) are Hermitian matrices since $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$). Therefore, we cannot get an exact value for the eigenvalues of these Hessian functions. The problem of describing the possible eigenvalues of the sum of two Hermitian matrices in terms of their spectra has attracted the interest of mathematicians for decades. The most complete description was conjectured by Horn *et al.* [59]. The interested readers are referred to [60] for an interesting discussion in the topic. However, we can get some bounds using the developments of Weyl [58]:

Theorem 3 (Weyl's Theorem (4.3.1 in [58])). *Let \mathbf{A} and \mathbf{B} be Hermitian matrices with set of eigenvalues $\{a_1, a_2, \dots, a_N\}$ and $\{b_1, b_2, \dots, b_N\}$, respectively, where the eigenvalues are in ascending order. Then, $c_i \leq a_{i+j} + b_{N-j}$, with $j = 0, 1, \dots, N-i$, and $a_{i-j+1} + b_j \leq c_i$, with $j = 1, \dots, i$, where c_i is the i th eigenvalue of $\mathbf{A} + \mathbf{B}$.*

Proof: see [58].

Using Theorem 3 and (26), we can have the following inequalities for $\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})$:

$$\lambda_N \lambda_{(D)N} \leq \lambda_{\max}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \leq \lambda_N \lambda_{(D)N} + \frac{1}{v}, \quad (29)$$

$$0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \leq \frac{1}{v}, \quad (30)$$

$$0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \leq \lambda_N \lambda_{(D)N}. \quad (31)$$

Similarly, for $\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})$ we have:

$$\begin{aligned} (\lambda_N + \epsilon)^\beta \lambda_{(D)N} &\leq \lambda_{\max}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \\ &\leq (\lambda_N + \epsilon)^\beta \lambda_{(D)N} + \frac{1}{v}, \end{aligned} \quad (32)$$

$$0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \leq \frac{1}{v}, \quad (33)$$

$$0 \leq \lambda_{\min}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \leq (\lambda_N + \epsilon)^\beta \lambda_{(D)N}, \quad (34)$$

Notice that we ignored the v value that multiplies (26) and (28) in these inequalities since $\kappa(v\mathbf{A}) = \kappa(\mathbf{A})$. We can have some asymptotic analysis based on the inequalities (29)-(34):

- 1) For $\epsilon > 0$ and $\beta > 1$ we have that the upper bound in (34) is looser than the upper bound in (31) given that $\lambda_N + \epsilon \geq 1$, which favors better condition numbers for the Sobolev problem. For example, if we assume that the eigenvalues are equal to the upper bounds in (29), (31), (32), and (34) we have that:

$$\begin{aligned} \kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) &= \left(1 + \frac{1}{v(\lambda_N + \epsilon)^\beta \lambda_{(D)N}} \right)^2 \\ &\leq \kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) = \left(1 + \frac{1}{v \lambda_N \lambda_{(D)N}} \right)^2. \end{aligned} \quad (35)$$

- 2) When $v \rightarrow \infty$, $\lambda_{\min}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \rightarrow 0$ and $\lambda_{\min}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow 0$ according to (30) and (33), and then $\kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) \rightarrow \infty$ and $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$.
- 3) When $\epsilon \rightarrow \infty$ and $\beta > 0$, $\lambda_{\max}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$ according to (32) and then $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$. When $\epsilon \rightarrow \infty$ and $\lambda_{\min}(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z})) > 0$ we have that $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) > \kappa(\nabla_{\mathbf{z}}^2 f_L(\mathbf{z}))$.
- 4) when $\beta \rightarrow \infty$ and $\lambda_N + \epsilon > 1$, $\lambda_{\max}(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$ according to (32) and then $\kappa(\nabla_{\mathbf{z}}^2 f_S(\mathbf{z})) \rightarrow \infty$.

□

REFERENCES

- [1] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [2] 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.
- [3] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.
- [4] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [5] J. H. Giraldo, S. Javed, and T. Bouwmans, "Graph moving object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, early access, Dec. 9, 2020.
- [6] J. H. Giraldo and T. Bouwmans, "GraphBGS: Background subtraction via recovery of graph signals," in *International Conference on Pattern Recognition*, 2021.
- [7] G. Lewenfus, W. A. Martins, S. Chatzinotas, and B. Ottersten, "Joint forecasting and interpolation of time-varying graph signals using deep learning," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 761–773, 2020.
- [8] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

- [9] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *IEEE journal of Selected Topics in Signal Processing*, vol. 10, no. 7, pp. 1189–1203, 2016.
- [10] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *IEEE International Conference on Image Processing*, 2014.
- [11] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.
- [12] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3d point cloud sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.
- [13] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, 2020.
- [14] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *International Conference on Knowledge Discovery and Data Mining*, 2014.
- [15] A. Anis, A. El Gamal, A. S. Avestimehr, and A. Ortega, "A sampling theory perspective of graph-based semi-supervised learning," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2322–2342, 2018.
- [16] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, 2016.
- [17] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 360–374, 2018.
- [18] X. Pu, S. L. Chau, X. Dong, and D. Sejdinovic, "Kernel-based graph learning from smooth signals: A functional viewpoint," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 192–207, 2021.
- [19] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016.
- [20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.
- [21] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.
- [22] H. Shomorony and A. S. Avestimehr, "Sampling large data on graphs," in *IEEE Global Conference on Signal and Information Processing*, 2014.
- [23] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.
- [24] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.
- [25] J. H. Giraldo and T. Bouwmans, "On the minimization of Sobolev norms of time-varying graph signals: Estimation of new Coronavirus disease 2019 cases," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2020.
- [26] 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.
- [27] A. Gavili and X. P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6303–6318, 2017.
- [28] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "Online distributed learning over graphs with multitask graph-filter models," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 63–77, 2020.
- [29] W. Xia, J. Chen, and L. Yu, "Distributed adaptive multi-task learning based on partially observed graph signals," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 522–538, 2021.
- [30] I. Pesenson, "Variational splines and Paley-Wiener spaces on combinatorial graphs," *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, 2009.
- [31] 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.
- [32] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, 2016.
- [33] 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.
- [34] D. Valsesia, G. Fracastoro, and E. Magli, "Sampling of graph signals via randomized local aggregations," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 348–359, 2018.
- [35] A. Parada-Mayorga, D. L. Lau, J. H. Giraldo, and G. R. Arce, "Blue-noise sampling on graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 3, pp. 554–569, 2019.
- [36] A. Venkitaraman, S. Chatterjee, and P. Händel, "Predicting graph signals using kernel regression where the input signal is agnostic to a graph," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 698–710, 2019.
- [37] I. Pesenson, "Sampling in Paley-Wiener spaces on combinatorial graphs," *Transactions of the American Mathematical Society*, vol. 360, no. 10, pp. 5603–5627, 2008.
- [38] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *International Conference on Computational Learning Theory*, 2004.
- [39] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *IEEE Global Conference on Signal and Information Processing*, 2013.
- [40] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 539–554, 2016.
- [41] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.
- [42] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, 2017.
- [43] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, pp. 1–19, 2019.
- [44] 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, 2017.
- [45] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 728–740, 2015.
- [46] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, 2018.
- [47] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.
- [48] P. Di Lorenzo, S. Barbarossa, and P. Banelli, "Sampling and recovery of graph signals," in *Cooperative and Graph Signal Processing*. Elsevier, 2018, pp. 261–282.
- [49] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus, "Sampling and reconstruction of signals on product graphs," in *IEEE Global Conference on Signal and Information Processing*, 2018.
- [50] J. S. Arora, "More on numerical methods for unconstrained optimum design," in *Introduction to Optimum Design*, 4th ed. Boston: Academic Press, 2017, pp. 455–509.
- [51] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, 2020.
- [52] R. Sibson, "A brief description of natural neighbor interpolation," *Interpreting Multivariate Data*, 1981.
- [53] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.
- [54] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.
- [55] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.
- [56] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
- [57] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

- [58] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2013.
- [59] A. Horn *et al.*, "Eigenvalues of sums of hermitian matrices." *Pacific Journal of Mathematics*, vol. 12, no. 1, pp. 225–241, 1962.
- [60] A. Knutson and T. Tao, "Honeycombs and sums of hermitian matrices," *Notices Amer. Math. Soc.*, vol. 48, no. 2, 2001.



Dorina Thanou is a senior researcher and lecturer at EPFL, leading the development of the Intelligent Systems for Medicine and Health research pillar, under the Center for Intelligent Systems. Prior to that, she was a Senior Data Scientist at the Swiss Data Science Centre. She got her M.Sc. and Ph.D. in Communication Systems and Electrical Engineering respectively, both from EPFL, Switzerland, and her Diploma in Electrical and Computer Engineering from the University of Patras, Greece. She is the recipient of the Best Student Paper Award at ICASSP 2015, and the co-author of the Best Paper Award at PCS 2016. She is an Ellis Scholar. Her research interests lie in the broader area of signal processing and machine learning, with a special focus on graph-based representation learning algorithms. She is particularly interested in applying her expertise to intelligent systems for healthcare.



Jhony H. Giraldo received the B.Sc. and M.Sc. degrees in Electronics Engineering from the University of Antioquia, Medellin, Colombia. He is currently working toward a Ph.D. degree in Computer Sciences with the University of La Rochelle, France. His research interests include the fundamentals and applications of computer vision, machine learning, graph signal processing, and graph neural networks. He has worked on image and video processing, supervised and semi-supervised learning, and on sampling and reconstruction of graph signals.



Arif Mahmood is currently a Professor with Information Technology University (ITU). He received his M.Sc. and Ph.D. in Computer Science from the Lahore University of Management Sciences (LUMS). He has served as a Research Assistant Professor at the School of Computer Science and Software Engineering, and later at the School of Mathematics and Statistics in the University of Western Australia from 2012-2015. Previously, he was an Assistant Professor at the Punjab University College of Information Technology from 2008-2012.

He also worked as a Postdoctoral Researcher at the College of Engineering in Qatar University from 2015-2018. His major research interests include data clustering, classification, action and object recognition using image sets. He has also worked on computation elimination algorithms for fast template matching, video compression, object removal and image mosaicking. At the School of Mathematics and Statistics, he worked on community detection in complex networks.



Thierry Bouwmans is an Associate Professor at the University of La Rochelle, France. His research interests consist mainly in the detection of moving objects in challenging environments. He has recently authored more than 30 papers in the field of background modeling and foreground detection. These papers investigated particularly the use of fuzzy concepts, discriminative subspace learning models and robust PCA. They also develop surveys on mathematical tools used in the field and particularly on Robust PCA via Principal Component Analysis. He has supervised Ph.D. students in this field. He is the creator and administrator of the Background Subtraction Web Site. He served as the lead guest editors in two editorial works: 1) the special issue in MVA on Background Modeling for Foreground Detection in Real-World Dynamic Scenes and 2) the Handbook on "Background Modeling and Foreground Detection for Video Surveillance" in CRC press. He has served as a reviewer for numerous international conferences and journals.



Belmar Garcia-Garcia received the B.Sc. degree in Communications and Electronics Engineering, the M.Sc. degree in Electronics Engineering, and the Ph.D. degree in Communications and Electronics Engineering from Instituto Politécnico Nacional de México (IPN). He is currently a Postdoctoral Researcher at the University of La Rochelle, France. He is focused on the development of algorithms for prediction and diagnosis of COVID-19. Other topics of interest are tracking, digital electronic, real time processing, and artificial intelligence.