



HAL
open science

Large Graph Signal Denoising with Application to Differential Privacy

Elie Chedemail, Basile de Loynes, Fabien Navarro, Baptiste Olivier

► **To cite this version:**

Elie Chedemail, Basile de Loynes, Fabien Navarro, Baptiste Olivier. Large Graph Signal Denoising with Application to Differential Privacy. *IEEE Transactions on Signal and Information Processing over Networks*, 2022, 8, 10.1109/TSIPN.2022.3205555 . hal-03769828

HAL Id: hal-03769828

<https://hal.science/hal-03769828v1>

Submitted on 5 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Large Graph Signal Denoising with Application to Differential Privacy

Elie Chedemail

Orange Labs / CREST, ENSAI

Cesson-Sévigné, France

elie.chedemail@orange.com

Basile de Loynes

ENSAI

Bruz, France

basile.deloynes@ensai.fr

Fabien Navarro

SAMM, Paris 1 Panthéon-Sorbonne

Paris, France

fabien.navarro@univ-paris1.fr

Baptiste Olivier

Ericsson

Stockholm, Sweden

baptiste.olivier@ericsson.com

Abstract—Over the last decade, signal processing on graphs has become a very active area of research. Specifically, the number of applications, for instance in statistical or deep learning, using frames built from graphs, such as wavelets on graphs, has increased significantly. We consider in particular the case of signal denoising on graphs via a data-driven wavelet tight frame methodology. This adaptive approach is based on a threshold calibrated using Stein’s unbiased risk estimate adapted to a tight-frame representation. We make it scalable to large graphs using Chebyshev-Jackson polynomial approximations, which allow fast computation of the wavelet coefficients, without the need to compute the Laplacian eigendecomposition. However, the overcomplete nature of the tight-frame, transforms a white noise into a correlated one. As a result, the covariance of the transformed noise appears in the divergence term of the SURE, thus requiring the computation and storage of the frame, which leads to an impractical calculation for large graphs. To estimate such covariance, we develop and analyze a Monte-Carlo strategy, based on the fast transformation of zero mean and unit variance random variables. This new data-driven denoising methodology finds a natural application in differential privacy. A comprehensive performance analysis is carried out on graphs of varying size, from real and simulated data.

Index Terms—Chebyshev polynomial approximation, Monte-Carlo methods, differential privacy, graph signal processing, Stein’s unbiased risk estimate

I. INTRODUCTION

Data acquired from large-scale interactive systems, such as computer, ecological, social, financial or biological networks, become increasingly widespread and accessible. In modern machine learning, the effective representation, processing or analysis of these large-scale structured data with graphs or networks are some of the key issues [5], [31]. The emerging field of Graph Signal Processing (GSP) highlights connections between signal processing and spectral graph theory [33], [36], while building bridges to address these challenges. Indeed, GSP has led to numerous applications in the field of machine learning: convolutional neural networks (CNN) on graphs [6], [12], [24], semi-supervised classification with graph CNN [22], [28] or community detection [39] to name just a few. We refer the reader to [14] for a recent review providing new perspectives on GSP for machine learning including, for instance, the important role it played in some of the early designs of graph neural networks (GNNs) architectures. Moreover, the recent study in [19] shows that popular GNNs designed from a spectral perspective, such as spectral graph convolutional

networks or graph attention networks, are implicitly solving graph signal denoising problems.

In the past decades, sparse approximation with respect to a frame played a fundamental role in many areas such as signal compression and restoration, data analysis, and GSP in general. Indeed, over-complete representations like wavelet frames have several advantages and offer more flexibility over orthonormal bases. One representative family of over-complete systems derived from the orthonormal *Diffusion Wavelets* of [8] is the so-called *Spectral Graph Wavelet Transform* (SGWT) of [23] constructed from a general wavelet frame. In a denoising context, SGWT has recently been adapted by [21] to form a tight frame using the Littlewood-Paley decomposition inspired by [9]. Based on SGWT, [11] proposed an automatic calibration of the threshold parameter by adapting Stein’s unbiased risk estimate (SURE) for a noisy signal defined on a graph and decomposed in a given wavelet tight frame. Even if this selection criterion produces efficient estimators of the unknown mean squared error (MSE), the main limitation is the need for a complete eigendecomposition of the Laplacian matrix, making it intractable for large-scale graphs.

We propose here to extend this methodology to large sparse graphs by avoiding this eigendecomposition, thus extending its range of application. Different strategies have been proposed in the context of GSP, one of the most popular is based on Chebyshev polynomial approximations [23]. However, even if Chebyshev expansions are a good choice in many scenarios, approximations of discontinuous or non-periodic functions suffer from the *Gibbs phenomenon*. A simple strategy commonly used in GSP [37] to reduce possible spurious oscillations without additional computational cost is the introduction of Jackson’s *damping coefficients* [13], [27] which allows for higher orders of approximation.

As SURE can be evaluated in the wavelet domain, its calculation benefits directly from these efficient numerical approximations. In order to make it suitable for large sparse graphs, the only problematic step is the computation of the weights appearing in its expression. Indeed, since the SGWT is no longer orthogonal a white Gaussian noise in the graph domain is transformed into a correlated one thus involving the covariance of the transformed noise in the resulting SURE divergence term. The latter requires the explicit computation and storage of the frame in order to be calculated. Inspired by

the estimation of the correlation between wavelets centered at different nodes proposed in [39], our contribution is to take advantage of the interpretation of the SURE weights as the covariance between wavelet transforms of random signals in order to estimate them with Monte-Carlo approximation. We then plug this weight estimator in the SURE formula to obtain an estimator of SURE that extends well to large graph signals. In addition, we provide expressions for the variance of our proposed estimators and show that drawing Monte-Carlo samples from the centered Rademacher distribution gives a smaller variance compared to the standard Gaussian distribution. Our approach is in line with other methods [35], [41] that also use Monte-Carlo strategy, but to estimate the entire divergence term involved in the calculation of SURE, in the case of uncorrelated noise.

Our proposed method can remove noise from any signal defined on a graph, this includes images [36] and 3D meshes [32] which can have a large number of vertices. Here, we focus on an interesting application in differential privacy [17] whose purpose is to protect sensitive data used by algorithms. Such privacy guarantees are usually achieved by adding white noise to the signal which inevitably reduces its statistical utility as the relevant information it contains is perturbed. This utility can be partially recovered through denoising on the condition that no information about the original signal is used. As our proposed data-driven methodology lends itself well to this usage for graph signals, we incorporate it in our numerical experiments. These give an evaluation of our Monte-Carlo estimator of SURE and its weights, along with the overall denoising methodology on both small and large graphs. In summary, the contributions of this paper are as follows:

- We propose a Monte-Carlo estimation of Stein’s unbiased risk estimate (SURE) that extends to signals defined on large-scale graphs. This method avoids the computationally expensive eigendecomposition of the graph Laplacian matrix required to compute weights that appear in the SURE expression.
- Provided expressions for the variance of our estimators show that Monte Carlo samples drawn from a Rademacher distribution is more efficient than with a Gaussian distribution. This theoretical result is illustrated through numerical experiments that compare both distributions.
- A performance analysis of the proposed graph signal denoising methodology shows its performance on real data protected with differential privacy and simulated large graph signals.

The paper is structured as follows. We introduce our notation of graph signals and briefly recall the SGWT definition of [23], its construction by [21] and polynomial approximations in Section II. Our proposed Monte-Carlo estimators of SURE and its weights along with their respective variance are presented in Section III. In Section IV we present the notion of differential privacy from [17] and two methods to achieve it in the context of graph signals. Finally, we numerically evaluate

our estimators and compare our methodology to the DFS fused lasso introduced in [25] for small and large graphs in Section V.

II. GRAPH SIGNAL DENOISING

Consider a signal $f \in \mathbb{R}^V$ defined on an undirected weighted graph G , with set of vertices V of cardinality n , and weighted adjacency matrix W with entries $(w_{ij})_{i,j \in V}$. The (unnormalized) graph Laplacian matrix $\mathcal{L} \in \mathbb{R}^{V \times V}$ associated with G is the symmetric matrix defined as $\mathcal{L} = D - W$, where D is the diagonal matrix with diagonal coefficients $D_{ii} = \sum_{j \in V} w_{ij}$. We present here our methodology with this particular Laplacian matrix but it can be easily adapted to its normalized and random walk counterparts like presented below in Section II-C.

The noise corruption model can be written as

$$\tilde{f} = f + \xi,$$

where $\xi \sim \mathcal{N}(0, \sigma^2 I_n)$. The purpose of denoising is to build an estimator \hat{f} of f that depends only on \tilde{f} .

A simple way to construct an effective non-linear estimator is obtained by thresholding the SGWT coefficients of f on a frame (see [23] for details about the SGWT). Given the Laplacian and a given frame, denoising in this framework can be summarized as follows:

- Analysis: compute the SGWT transform $\mathcal{W}\tilde{f}$;
- Thresholding: apply a given thresholding operator (*e.g.*, *soft* or *hard*) to the coefficients $\mathcal{W}\tilde{f}$;
- Synthesis: apply the inverse SGWT transform to obtain an estimate \hat{f} of the original signal.

This procedure can be viewed as an extension of the wavelet denoising methodology from Donoho and Johnstone [15] to the SGWT.

A. Spectral Graph Wavelet Transform

The SGWT decomposes a signal into a frame $\mathfrak{F} = \{r_i\}_{i \in I}$ of vectors of \mathbb{R}^V with frame bounds $A, B > 0$ satisfying for all $f \in \mathbb{R}^V$

$$A\|f\|_2^2 \leq \sum_{i \in I} |\langle f, r_i \rangle|^2 \leq B\|f\|_2^2.$$

When $A = B = 1$, the above inequality becomes Parseval’s identity and such a frame is said to be *tight*.

As \mathcal{L} is a symmetric matrix, its spectral decomposition is given by $\mathcal{L} = \sum_{\ell} \lambda_{\ell} \langle \chi_{\ell}, \cdot \rangle \chi_{\ell}$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n = 0$ are the (ordered) eigenvalues of \mathcal{L} and $(\chi_{\ell})_{1 \leq \ell \leq n}$ are the associated eigenvectors. Then for any function $\rho : \text{sp}(\mathcal{L}) \rightarrow \mathbb{R}$ defined on the spectrum of \mathcal{L} , we have the functional calculus formula $\rho(\mathcal{L}) = \sum_{\ell} \rho(\lambda_{\ell}) \langle \chi_{\ell}, \cdot \rangle \chi_{\ell}$.

We build a tight frame following [21], [29] with a finite partition of unity $(\psi_j)_{j=0, \dots, J}$ on the compact $[0, \lambda_1]$ defined as follows: let $\omega : \mathbb{R}^+ \rightarrow [0, 1]$ be some continuous function with support in $[0, 1]$, satisfying $\omega \equiv 1$ on $[0, b^{-1}]$, for some $b > 1$, and set

$$\psi_0(x) = \omega(x), \quad \psi_j(x) = \omega(b^{-j}x) - \omega(b^{-j+1}x),$$

for $j = 1, \dots, J$, where $J = \lfloor \log \lambda_1 / \log b \rfloor + 2$. In our numerical experiments, we use the following piecewise linear function ω :

$$\omega(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq b^{-1} \\ \frac{b}{1-b}x + \frac{b}{b-1} & \text{if } b^{-1} < x \leq 1, \\ 0 & \text{if } x > 1 \end{cases}$$

with parameter $b = 2$. An alternative ω function is the \mathcal{C}^∞ function $h_c(x) = g_c(x+1)g_c(1-x)$ where $g_c(x) = f(x)/(f(x) + f(c-x))$ and $f(x) = e^{-1/x}\mathbf{1}_{\{x>0\}}$. Another choice is to take a \mathcal{C}^3 piecewise polynomial plateau function like the authors of [21]. Figure 1 illustrates the ω and h_c functions with parameters $b = 2$ and $c = 1$, respectively.

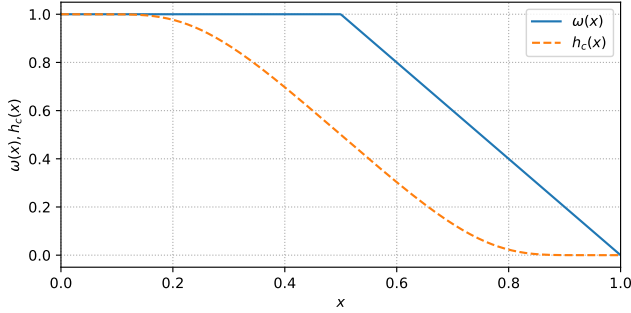


Fig. 1. ω and h_c functions on $[0, 1]$

The partition of unity $(\psi_j)_{j=0, \dots, 4}$ on $[0, \lambda_1]$ obtained with the graph presented in the first experiment from Section V is shown in Figure 2.

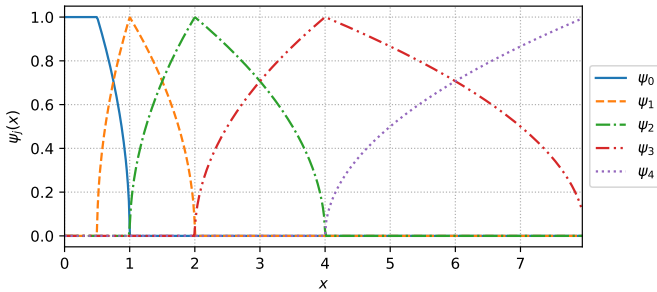


Fig. 2. Finite partition of unity on $[0, \lambda_1]$

Using Parseval's identity, we can show that the following set is a tight frame:

$$\mathfrak{F} = \left\{ \sqrt{\psi_j(\mathcal{L})} \delta_i, j = 0, \dots, J, i \in V \right\}.$$

Decomposing a signal $f \in \mathbb{R}^V$ into this frame results in its SGWT along the $(J+1)$ scales:

$$\mathcal{W}f = \left(\sqrt{\psi_0(\mathcal{L})}f^T, \dots, \sqrt{\psi_J(\mathcal{L})}f^T \right)^T \in \mathbb{R}^{n(J+1)}.$$

With the tightness property of the frame, the inverse transform is directly given by the application of the adjoint matrix to the wavelet coefficients:

$$\mathcal{W}^* \left(\eta_0^T, \eta_1^T, \dots, \eta_J^T \right)^T = \sum_{j \geq 0} \sqrt{\psi_j(\mathcal{L})} \eta_j.$$

B. SGWT Polynomial Approximation

Direct computation of the SGWT entails functional calculus on the graph Laplacian matrix \mathcal{L} and thus the computation of its eigenvectors and eigenvalues. This limits applications to reasonably sized graphs that have less than a few thousand nodes. For larger ones, the computationally expensive eigen-decomposition can be avoided through a fast transform based on Chebyshev polynomial approximation [23].

The Chebyshev polynomials of the first kind $T_k(x)$ are obtained from the recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. They form an orthogonal basis of the Hilbert space $\mathbb{L}^2([-1, 1], dy/\sqrt{1-y^2})$. Any filter ρ can be approximated with the truncated Chebyshev expansion of degree K

$$\rho_K(\mathcal{L}) = \sum_{i=0}^K \theta_i(\tilde{\rho}) T_i(\tilde{\mathcal{L}}),$$

where $\theta_i(\tilde{\rho})$ is the i -th coefficient of the Chebyshev expansion of function $\tilde{\rho}(x) = \rho(\frac{\lambda_1}{2}(x+1))$ and $T_i(\tilde{\mathcal{L}})$ is the i -degree Chebyshev polynomial computed for $\tilde{\mathcal{L}} = \frac{2}{\lambda_1}\mathcal{L} - I_n$. This transformation of \mathcal{L} extends the expansion to any Laplacian matrix by mapping $[0, \lambda_1]$ into $[-1, 1]$. According to [23], for all filter ρ defined on $\text{sp}(\mathcal{L})$ and all signal f , the approximation $\rho_K(\mathcal{L})f$ is close to $\rho(\mathcal{L})f$.

While this first approximation is more practical than the complete SGWT, it is subjected to the *Gibbs phenomenon*. A solution is to include Jackson coefficients g_i^K as damping multipliers in the Chebyshev expansion:

$$\rho_K(\mathcal{L}) = \sum_{i=0}^K g_i^K \theta_i(\tilde{\rho}) T_i(\tilde{\mathcal{L}}).$$

An expression of these damping factors can be found in [27], a shorter form proposed in [13] is given by

$$g_i^K = \frac{\sin(i+1)\alpha_K}{(K+2)\sin(\alpha_K)} + \left(1 - \frac{i+1}{K+2}\right) \cos(i\alpha_K),$$

where $\alpha_K = \pi/(K+2)$. This Chebyshev-Jackson polynomial approximation reduces Gibbs oscillations resulting in a better convergence as the degree K increases.

C. Extension to Other Laplacian Matrices

As previously mentioned, this methodology also adapts well to the normalized and random walk (or asymmetric) Laplacian matrices, respectively defined as $\mathcal{L}_{\text{norm}} = D^{-\frac{1}{2}}\mathcal{L}D^{-\frac{1}{2}}$ and $\mathcal{L}_{\text{rw}} = D^{-1}\mathcal{L}$. These have been used as an alternative to the unnormalized graph Laplacian \mathcal{L} in other related methods such as the graph Fourier transform [20].

The normalized Laplacian matrix is real symmetric like \mathcal{L} which means it is diagonalizable and therefore suited for our approach. As its spectrum $\text{sp}(\mathcal{L}_{\text{norm}}) = \{\mu_1, \dots, \mu_n\}$ is always contained in the interval $[0, 2]$, its maximum eigenvalue μ_1 is bounded by 2. This represents a special case of the construction described above and requires a few modifications. First, the formula $J = \lfloor \log \mu_1 / \log b \rfloor + 2$ that determines the number of scales in the wavelet decomposition restricts

the choice of parameter b to the interval $(1, 2]$ in order to get more than $J + 1 = 3$ scales. Then, the polynomial approximation consists of the truncated Chebyshev expansion of function $\tilde{\rho}(x) = \rho(x + 1)$ with the appropriate transformation $\tilde{\mathcal{L}}_{\text{norm}} = \mathcal{L}_{\text{norm}} - I_n$.

On the other hand, the random walk Laplacian matrix is not symmetric but is diagonalizable nonetheless as it is similar to the normalized Laplacian: $\mathcal{L}_{\text{norm}} = D^{\frac{1}{2}} \mathcal{L}_{\text{rw}} D^{-\frac{1}{2}}$. Its eigenvalues and eigenvectors are easily obtained from the eigendecomposition of $\mathcal{L}_{\text{norm}}$:

$$\begin{aligned} \mathcal{L}_{\text{norm}} u_\ell &= \mu_\ell u_\ell \\ D^{\frac{1}{2}} \mathcal{L}_{\text{rw}} D^{-\frac{1}{2}} u_\ell &= \mu_\ell u_\ell \\ \mathcal{L}_{\text{rw}} (D^{-\frac{1}{2}} u_\ell) &= \mu_\ell (D^{-\frac{1}{2}} u_\ell), \end{aligned}$$

where u_ℓ is the eigenvector of $\mathcal{L}_{\text{norm}}$ associated with μ_ℓ . We see that \mathcal{L}_{rw} has exactly the same spectrum as $\mathcal{L}_{\text{norm}}$ and its set of eigenvectors is given by $\{D^{-\frac{1}{2}} u_\ell\}_{\ell=1, \dots, n}$. Since these form an orthonormal basis for \mathbb{R}^n with the inner product $\langle x, y \rangle_D = x^T D y$, we have the spectral decomposition $\mathcal{L}_{\text{rw}} = \sum_\ell \mu_\ell \langle D^{-\frac{1}{2}} u_\ell, \cdot \rangle_D D^{-\frac{1}{2}} u_\ell$. The functional calculus formula for any function ρ defined on the spectrum of \mathcal{L}_{rw} is thus $\rho(\mathcal{L}_{\text{rw}}) = \sum_\ell \rho(\mu_\ell) \langle D^{-\frac{1}{2}} u_\ell, \cdot \rangle_D D^{-\frac{1}{2}} u_\ell$. Chebyshev polynomial approximation can then be applied in the same way as for the normalized Laplacian matrix.

III. MONTE-CARLO ESTIMATION OF WEIGHTS

From [11], SURE for a general thresholding process $h : \mathbb{R}^{n(J+1)} \rightarrow \mathbb{R}^{n(J+1)}$ is given by the following identity

$$\text{SURE}(h) = -n\sigma^2 + \|h(\tilde{F}) - \tilde{F}\|^2 + 2\sigma^2 \sum_{i,j=1}^{n(J+1)} \gamma_{ij}^2 \partial_j h_i(\tilde{F}), \quad (1)$$

where $\tilde{F} = \mathcal{W}f$ is the wavelet transform of the noisy signal f . In [11], the weights $\gamma_{ij}^2 = (\mathcal{W}\mathcal{W}^*)_{ij}$, $i, j = 1, \dots, n(J+1)$, are computed from the full reduction of the Laplacian matrix which is no longer tractable for large graphs. However, as shown in [23], the SGWT can be efficiently approximated by using Chebyshev polynomials. Besides, it is clear from the probabilistic interpretation given in [11, Th. 1] that

$$\forall i, j = 1, \dots, n(J+1), \quad \gamma_{ij}^2 = \mathbf{E}[(\mathcal{W}\varepsilon)_i (\mathcal{W}\varepsilon)_j],$$

where $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ are *i.i.d.* random variables with zero mean and variance one. Thus, taking advantage of this identity, we propose to estimate the weights with Monte-Carlo approximation as follows:

- generate $(\varepsilon_{ik})_{i=1, \dots, n, k=1, \dots, N}$ of *i.i.d.* random variables such that $\mathbf{E}[\varepsilon_{ik}] = 0$ and $\mathbf{V}(\varepsilon_{ik}) = 1$;
- compute

$$\hat{\gamma}_{ij}^2 = \frac{1}{N} \sum_{k=1}^N (\mathcal{W}\varepsilon_k)_i (\mathcal{W}\varepsilon_k)_j,$$

where $\varepsilon_k = (\varepsilon_{ik})_{i=1, \dots, n}$ are random signals.

Generally speaking, whereas Monte-Carlo are simple methods to implement and can be easily parallelized, they suffer from their slow rate of convergence. In practice, a well-chosen

distribution for the random variables ε_{ik} can result in a lower variance of the estimator $\mathbf{V}[\hat{\gamma}_{ij}^2]$ whose expression is given below. In fact, it is even more interesting to compute the variance of SURE when the estimator $\hat{\gamma}_{ij}^2$ is plugged in place of the weights γ_{ij}^2 in (1).

A. Variance of $\hat{\gamma}_{ij}^2$.

A straightforward computation gives the expectation of $\hat{\gamma}_{ij}^2$

$$\begin{aligned} \mathbf{E}[\hat{\gamma}_{ij}^2] &= \mathbf{E} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{p1} \right) \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{p1} \right) \right] \\ &= \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathbf{E}[\varepsilon_{p1} \varepsilon_{q1}] = \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} = \gamma_{ij}^2. \end{aligned}$$

The variance of $\hat{\gamma}_{ij}^2$ is given by the following result and its computation is derived in Appendix A.

Proposition 1:

$$\begin{aligned} \mathbf{V}[\hat{\gamma}_{ij}^2] &= \frac{1}{N} \left\{ \mathbf{V}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jp}^2 \right. \\ &\quad \left. + 2\mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jp} \mathcal{W}_{jq} \right\}. \end{aligned}$$

Note the usual rate of convergence \sqrt{N} from Monte-Carlo estimation. In many papers in the literature ε_{11} is chosen to be distributed as a standard Gaussian random variable so that $\mathbf{V}[\varepsilon_{11}^2] = 2$. However, if ε_{11} has a centered Rademacher distribution with probability mass function $\frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$, then ε_{11}^2 is deterministic and $\mathbf{V}[\varepsilon_{11}^2] = 0$. With such a choice, the variance of $\hat{\gamma}_{ij}^2$ is then reduced to

$$\mathbf{V}[\hat{\gamma}_{ij}^2] = \frac{2}{N} \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jp} \mathcal{W}_{jq}.$$

This trick is actually well known in the literature [26]. This computation somehow provides arguments in favor of the Rademacher distribution.

Another way to further reduce the variance of $\hat{\gamma}_{ij}^2$ is to take advantage of the SGWT localization property. Let us denote by $[x]$ the integer part of a real $x \in \mathbb{R}$. Then, for any $i \in \{1, \dots, n(J+1)\}$ and any $p \in \{1, \dots, n\}$

$$|\mathcal{W}_{ip}| = \left| \left\langle \sqrt{\psi_{[i/n]}(\mathcal{L})} \delta_{i-[i/n]}, \delta_p \right\rangle \right| \leq \|\psi_{[i/n]}(\mathcal{L})\|_2^2 \leq 1.$$

Since the SGWT is localized both in the space and the frequency domain, \mathcal{W}_{ip} vanishes as the geodesic distance between $i - [i/n]$ and p grows. Thus, the performance of the Monte-Carlo estimation could be improved by a suitable calibration of the partition of unity. As a consequence, most terms in the expression of $\mathbf{V}[\hat{\gamma}_{ij}^2]$ are small thanks to the localization properties of SGWT.

B. Variance of $\widehat{\text{SURE}}$

The SURE plug-in estimator is obtained by replacing the weights γ_{ij}^2 with their Monte-Carlo estimators $\hat{\gamma}_{ij}^2$:

$$\widehat{\text{SURE}}(h) = -n\sigma^2 + \|h(\tilde{F}) - \tilde{F}\|^2 + 2\sigma^2 \sum_{i,j=1}^{n(J+1)} \hat{\gamma}_{ij}^2 \partial_j h_i(\tilde{F}).$$

Given the observed wavelet coefficients \tilde{F} , this estimator of SURE has no bias as it is a linear function of the unbiased weight estimators $\hat{\gamma}_{ij}^2$. The following proposition presents its conditional variance whose computation is detailed in Appendix B.

Proposition 2:

$$\begin{aligned} \mathbf{V}[\widehat{\text{SURE}}(h)|\tilde{F}] &= \frac{4\sigma^4}{N} \sum_{i,j,k,\ell=1}^{n(J+1)} \partial_j h_i(\tilde{F}) \partial_k h_\ell(\tilde{F}) \left\{ \right. \\ &\quad \mathbf{V}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kp} \mathcal{W}_{\ell p} \\ &\quad + \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kp} \mathcal{W}_{\ell q} \\ &\quad \left. + \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kq} \mathcal{W}_{\ell p} \right\}. \end{aligned}$$

Here again, the Rademacher distribution reduces the variance compared to the Gaussian distribution.

C. Computational Complexity

The polynomial approximation of all random signal wavelet transforms $\mathcal{W}\varepsilon_k$ is of order $\mathcal{O}(N(mK+n(J+1)K))$, where m is the number of edges in the graph [23]. Then, computing every $(\hat{\gamma}_{ij}^2)_{i,j=1,\dots,n(J+1)}$ term requires $\mathcal{O}(n^2(J+1)^2(2N-1))$ operations. The computation of all the weights is useful when performing block thresholding on the wavelet coefficients $\mathcal{W}\tilde{f}$ [11] which shows good denoising performance but is relatively computationally expensive. Alternatively, a coordinate-wise thresholding process only needs the diagonal weights $(\hat{\gamma}_{ii}^2)_{i=1,\dots,n(J+1)}$ whose computation is reduced to $\mathcal{O}(n(J+1)N)$ operations.

After this initial weight estimation, the computational complexity for the approximated wavelet transform of the noisy signal \tilde{f} is $\mathcal{O}(mK+n(J+1)K)$. The coordinate-wise thresholding step has an average cost of $\mathcal{O}(n(J+1)\log(n(J+1)))$ according to [15]. Finally, the approximated inverse transform has the same complexity as the forward transform.

IV. DIFFERENTIAL PRIVACY AND GAUSSIAN MECHANISM

We now give a definition of differential privacy [17] which constitutes a strong standard for privacy guarantees about algorithms that use sensitive data. Let X_1, \dots, X_m be a random vector containing the private data of m individuals we wish to protect with a *privacy mechanism*. This information is collected in a dataset $X = (X_1, \dots, X_m)$ that serves

as an input to the mechanism which returns a *sanitized* output $Z = (Z_1, \dots, Z_k)$ that preserves the privacy of each individual. Let $(\mathcal{X}^m, \mathcal{A}^m)$ and $(\mathcal{Z}, \mathcal{B})$ be the measurable spaces where X and Z respectively take values. A privacy mechanism $Q(\cdot|X)$ corresponds to the conditional distribution of Z given X , that is $Q(A|x) = \mathbb{P}(Z \in A|X = x)$, where $Q(\cdot|\cdot) : \mathcal{B} \times \mathcal{X}^m \rightarrow [0, 1]$ is a Markov kernel.

Let $\varepsilon \geq 0$ be the privacy budget and $\delta \geq 0$ another privacy parameter. The privacy mechanism Q is said to satisfy (ε, δ) -differential privacy if for any two datasets $x, x' \in \mathcal{X}^m$ that differ on a single entry and for any subset of outputs $A \in \mathcal{B}$, we have

$$Q(A|x) \leq e^\varepsilon Q(A|x') + \delta.$$

As it appears from this definition, smaller privacy parameters lead to closer output distributions and hence a better privacy preserving mechanism. Intuitively, differential privacy protects individuals by ensuring the inclusion or removal of their information from the input dataset does not affect much the output distribution.

In this paper, we are interested in functions $f : \mathcal{X}^m \rightarrow \mathbb{R}^n$ that map a dataset X to a graph signal $f \in \mathbb{R}^n$. In order to achieve differential privacy, a common method is to introduce just enough uncertainty in the function response to hide the participation of any single individual. The Gaussian mechanism does so by adding white Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2 I_n)$ to the response where the standard deviation σ is calibrated to the privacy parameters and a third quantity Δ called the l_2 -sensitivity. It is defined by $\Delta = \max \|f(x) - f(x')\|_2$ which corresponds to the maximum impact a single individual's information can have on the signal f . This method yields a sanitized output $Z = f(X) + \xi$ that can be interpreted as a noisy signal $\tilde{f} = f + \xi$ with our noise corruption model given in Section II.

We present two Gaussian mechanisms that use different variance formulas to sanitize a function $f : \mathcal{X}^m \rightarrow \mathbb{R}^n$ with l_2 -sensitivity Δ . First, the classical Gaussian mechanism proposed by [17] preserves (ε, δ) -differential privacy for any $\varepsilon, \delta \in (0, 1)$ if $\sigma \geq \Delta \sqrt{2 \log(1.25/\delta)}/\varepsilon$. The authors of [2] have shown that this formula is not optimal and can be further improved to reduce the amount of noise needed to achieve the same degree of privacy. Whereas the classical Gaussian mechanism uses a Gaussian tail approximation to obtain a bound for the standard deviation, their proposed approach uses numerical evaluations of the cumulative Gaussian distribution function $\Phi(x) = \mathbb{P}(\mathcal{N}(0, 1) \leq x)$ to determine an optimal variance. Their analytic Gaussian mechanism preserves (ε, δ) -differential privacy for any $\varepsilon \geq 0$ and $\delta \in [0, 1]$ if and only if

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^\varepsilon \Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta.$$

The Gaussian mechanism offers a solution to sanitize a signal at the expense of its utility as it is perturbed by the introduced noise. Indeed, there is a trade-off between privacy and utility: very small values of ε and δ ensure a strong degree of privacy but can be detrimental to the utility of the sanitized data,

and vice versa. A valuable aspect of differential privacy is its immunity to post-processing [18, Prop. 2.1] as long as no knowledge about the original signal is used. Formally, the composition of any data-independent function with an (ϵ, δ) -differentially private mechanism is also (ϵ, δ) -differentially private. Therefore, a data-driven denoising method such as ours can improve the utility of a sanitized signal with no loss of privacy. Indeed, the SGWT, SURE and their respective approximations only require information contained in the known Laplacian matrix \mathcal{L} and observed noisy signal \tilde{f} .

V. NUMERICAL EXPERIMENTS

In this section, we present an experimental evaluation of our Monte-Carlo estimator of the SURE weights described in Section III and graph signal denoising methodology. Specifically, we are interested in the sanitization and denoising of density maps of located events over a given period. First, we consider signals built from real datasets gathering positions of taxis in the cities of New York and San Francisco. The relatively small size of their corresponding graphs enables us to diagonalize their respective Laplacian matrices and directly compute the SGWT and SURE in a reasonable amount of time, thus allowing the comparison with our approximation method. Then, we generate signals on a large graph for which the eigendecomposition of the Laplacian matrix is not tractable. Hereafter, wavelet transforms are performed with the piecewise linear ω function presented in Section II-A and each polynomial approximation is of degree $K = 100$.

A. Monte-Carlo Estimator of the Weights γ_{ii}^2

This experiment makes use of the New York City (NYC) yellow taxi trip records publicly released each year by the Taxi and Limousine Commission (TLC). These datasets contain in particular the pickup and drop-off locations and times of each trip whose distribution has been the subject of different GSP applications [2], [7], [33]. In the past, a bad pseudonymization of the taxi ID led to a privacy breach [16] for the drivers and their passengers about where they might reside and the places they frequent.

Since the yellow taxis mostly operate in the borough of Manhattan, we focus our experiment on Manhattan Island and build an associated graph with **OSMnx** [4]. This Python package automatically downloads urban networks from the OpenStreetMap database, converts them to graph objects of the **NetworkX** package and offers a variety of analysis tools. The resulting graph consists of 4513 nodes and 9743 edges representing street intersections and segments, respectively.

With this first graph, we evaluate the SURE weights Monte-Carlo estimators when their samples are either drawn from a centered Rademacher or standard Gaussian distribution. Their SGWT are computed with the Chebyshev polynomial approximation and we estimate the weights for different Monte-Carlo sample sizes N . We focus on the diagonal weights $\hat{\gamma}_{ii}^2$, $i = 1, \dots, n(J+1)$ as only these are needed in the coordinate-wise thresholding process and compare them to the weights obtained from the complete transform by averaging the MSE

along the n nodes and $(J + 1)$ scales over 50 repetitions:
$$\text{MSE}((\gamma_{ii}^2)_i, (\hat{\gamma}_{ii}^2)_i) = \frac{1}{n(J+1)} \sum_{i=1}^{n(J+1)} (\gamma_{ii}^2 - \hat{\gamma}_{ii}^2)^2.$$

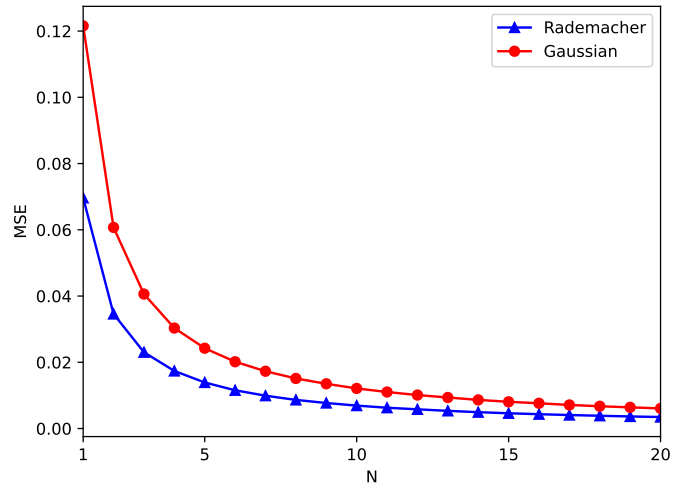


Fig. 3. Average MSE between the SURE weights γ_{ii}^2 and their Monte-Carlo estimators $\hat{\gamma}_{ii}^2$ over 50 repetitions.

Results are presented in Figure 3 where we see that drawing samples from the centered Rademacher distribution gives estimates closer to the real weights in terms of MSE for any sample size compared to the standard Gaussian distribution. This illustrates the gain in variance achieved with the former distribution as mentioned in Section III-A.

B. SURE Monte-Carlo Estimator

We now present how SURE behaves when the estimated weights $\hat{\gamma}_{ii}^2$ are plugged in. On the same graph, we build a signal f by counting the number of taxi pickups projected to the nearest intersection over a period of one hour. Here, we consider the time interval between 00:00 and 01:00 on Sept 24, 2014, as chosen by [2] to compare our results in a similar configuration. We add some white Gaussian noise with standard deviation $\sigma = 1$ to obtain a noisy signal and compute its SGWT coefficients with the Chebyshev-Jackson approximation. The denoising is done by applying the James-Stein thresholding function $\tau(x, t) = x \max\{1 - t^2|x|^{-2}, 0\}$ to the coefficients with the threshold that minimizes $\text{MSE}(f, \hat{f})$. Finally, for this optimal threshold we estimate SURE between the estimate \hat{f} and the original signal with known σ for different Monte-Carlo sample sizes and both the centered Rademacher and standard Gaussian distributions.

Figure 4 shows the average SURE estimate over 50 repetitions along with a 95 % empirical confidence interval. We visualize the SURE plug-in estimator unbiasedness as it is centered on the real SURE value. Additionally, we observe the smaller variance of the Monte-Carlo estimator when samples are drawn from the centered Rademacher distribution compared to the Gaussian distribution, as previously mentioned in Section III-B. As a result, we estimate SURE in the following experiments with 10 samples from this distribution.

TABLE I
AVERAGE SNR PERFORMANCE OVER 10 REALIZATIONS OF HIGH TO LOW PRIVACY BUDGET SANITIZATION ON THE NYC GRAPH.

ε	Classical Gaussian mechanism				Analytic Gaussian mechanism			
	0.2	0.3	0.5	1	0.2	0.3	0.5	1
σ	26.49	17.66	10.60	5.30	18.99	12.99	8.06	4.22
SNR_{in}	-12.48 ± 0.11	-8.96 ± 0.11	-4.52 ± 0.11	1.5 ± 0.11	-9.59 ± 0.11	-6.29 ± 0.11	-2.14 ± 0.11	3.47 ± 0.11
SGWT_{MSE}	0.23 ± 0.32	1.61 ± 0.26	3.55 ± 0.23	7.0 ± 0.16	1.36 ± 0.27	2.73 ± 0.23	4.79 ± 0.18	8.29 ± 0.15
$\text{SGWT}_{\text{SURE}}$	0.1 ± 0.24	1.5 ± 0.31	3.52 ± 0.2	6.94 ± 0.17	1.25 ± 0.38	2.71 ± 0.25	4.75 ± 0.2	8.24 ± 0.18
$\text{SGWT}_{\text{SURE, MC}}^{\text{CJ}}$	0.1 ± 0.28	1.51 ± 0.34	3.52 ± 0.24	6.92 ± 0.15	1.3 ± 0.34	2.71 ± 0.25	4.74 ± 0.22	8.24 ± 0.17
DFS_{MSE}	0.88 ± 0.02	1.18 ± 0.09	2.33 ± 0.18	5.58 ± 0.15	1.09 ± 0.09	1.72 ± 0.12	3.43 ± 0.17	6.96 ± 0.16
DFS_{SURE}	0.85 ± 0.03	1.11 ± 0.08	2.26 ± 0.16	5.55 ± 0.14	1.02 ± 0.07	1.65 ± 0.18	3.4 ± 0.15	6.95 ± 0.16

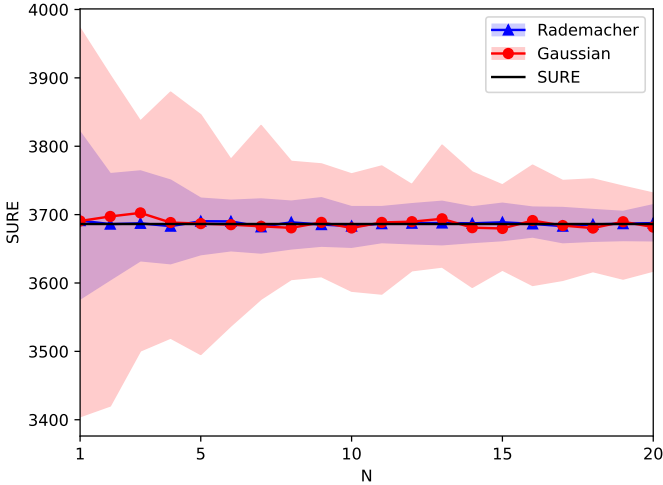


Fig. 4. Average SURE Monte-Carlo estimate and 95% CI over 50 repetitions.

C. Denoising of Differentially Private Graph Signals

We illustrate denoising performance with the SURE Monte-Carlo estimator on two relatively small graphs. This allows for the explicit eigendecomposition of their associated Laplacian matrices and computation of the SGWT and SURE with which we compare our proposed method.

1) *New York City taxis*: Considering the number of taxi pickups at each intersection from the last experiment as our signal, we now apply the differential privacy mechanisms presented in Section IV to sanitize it. Both the classical and analytic Gaussian mechanisms are used for different values of the privacy budget ε and therefore noise levels σ , while the other privacy parameter is set to $\delta = 10^{-6}$ as in [2]. Note that to satisfy the constraint associated with the maximum value taken by the ε parameter, the resulting noise levels are particularly high. We aim to protect the taxi passengers and assume they only take a taxi once within an hour. This gives an upper bound of their individual contribution to the signal and thus we have an l_2 -sensitivity of $\Delta = 1$.

We compare different denoising methods using signal-to-noise ratio $\text{SNR}(f, \hat{f}) = 20 \log_{10}(\|f\|_2 / \|f - \hat{f}\|_2)$ as a performance measure. We also compute it between the original and noisy signals to get a baseline of the amount of input noise after sanitization: $\text{SNR}_{\text{in}} = \text{SNR}(f, \hat{f})$. Three denoising methods based on the application of a level-dependent James-Stein thresholding function to wavelet coefficients are considered,

each of them uses a different criterion to select the optimal thresholds: (1) an oracle estimator that directly computes the SGWT and minimizes the real MSE; (2) a second estimator that instead minimizes SURE; and (3) our proposed estimator that approximates the SGWT with Chebyshev-Jackson polynomials and estimates SURE with Monte-Carlo. As shown by [15], for a coordinate-wise thresholding process such as James-Stein, SURE reaches its minimum for some threshold t chosen among the absolute values of the noisy wavelet coefficients $\{|\tilde{F}_i|, i = 1, \dots, n(J+1)\}$. We further reduce this set to its percentiles to find a compromise between the range and number of candidate threshold values.

These estimators are compared to the DFS fused lasso, a regularization method introduced in [25]. It first performs a standard depth-first search (DFS) traversal algorithm to reduce the initial graph to a chain graph. Then, it runs a 1-dimensional fused lasso [38], a special case of graph trend filtering [40], over this simpler graph. In doing so, this method avoids the prohibitive computational cost of standard graph trend filtering over an arbitrary graph at the expense of less statistical accuracy. Here, the comparison is made on unweighted graphs as the DFS fused lasso is limited to them, whereas the SGWT can be applied to graphs with edge weights. In the experiments, the DFS and fused lasso are respectively conducted with the **igraph** [10] and **glmgen** [1] R packages.

Table I summarizes the results of this experiment over 10 sanitization realizations. We observe that the wavelet transform oracle estimator (SGWT_{MSE}) performs better than the oracle DFS fused lasso (DFS_{MSE}) for most values of privacy budget. When considering stronger degrees of privacy with the classical Gaussian mechanism which requires the most amount of input noise, the oracle DFS fused lasso presents better results. Our approach combining Chebyshev-Jackson polynomial approximations with SURE Monte-Carlo estimation ($\text{SGWT}_{\text{SURE, MC}}^{\text{CJ}}$) gives slightly lower SNR values than its oracle counterpart but nevertheless shows better denoising performance compared to the regularization method except for the case where the noise is very high.

2) *San Francisco Taxis*: We check these initial results with a second dataset that contains the GPS coordinates of 536 taxis collected over a month in the San Francisco Bay Area [34]. Each entry consists of the taxi location and whether it currently has passengers at a given time with approximately one minute between updates. In a similar fashion as for

the previous experiment, we concentrate on the city of San Francisco and get the associated graph of the street network from **OSMnx**. It is about twice as large with 9573 nodes and 15716 edges, causing a longer but still practicable computation of the SGWT.

Pickup locations are inferred by keeping the entries whose occupancy status goes from "free" to "occupied", giving an approximation close to the minute. We build a signal by counting these pickups projected to the nearest intersection on the day of May 25, 2008. Sanitization is then applied with the analytic Gaussian mechanism and parameter values $\delta = 10^{-6}$ and $\Delta = 2$. The latter is chosen by assuming the individual passengers do a maximum of four taxi trips within a day, all starting from distinct places.

TABLE II

AVERAGE SNR PERFORMANCE OVER 10 REALIZATIONS OF HIGH TO LOW PRIVACY BUDGET SANITIZATION ON THE SAN FRANCISCO GRAPH.

ε	0.20	0.50	1
σ	37.98	16.12	8.45
SNR_{in}	-11.95 ± 0.05	-4.51 ± 0.05	1.1 ± 0.05
SGWT _{MSE}	0.80 ± 0.27	4.34 ± 0.11	8.24 ± 0.07
SGWT _{SURE}	0.77 ± 0.31	4.32 ± 0.09	8.22 ± 0.06
SGWT _{SURE, MC}	0.76 ± 0.32	4.34 ± 0.12	8.22 ± 0.08
DFS _{MSE}	0.22 ± 0.02	3.02 ± 0.1	6.77 ± 0.1
DFS _{SURE}	0.17 ± 0.07	2.99 ± 0.12	6.76 ± 0.1

Results presented in Table II are in line with those obtained above. The wavelet transform oracle estimator gives the best overall results and our method performs again better than the oracle DFS fused lasso for the considered privacy budget values.

D. Denoising of Large Graph Signals

In this experiment, we apply our method on a large graph whose scale prevents us from decomposing the Laplacian matrix due to the prohibitive computational cost. The road network of Pennsylvania from [30] is such a graph consisting of 1088092 nodes and 1541898 edges. Synthetic signals are generated on this graph following the methodology proposed in [3]: with two parameters $p \in (0, 1)$ and $k \in \mathbb{N}$, we produce a signal $f_{p,k} = W^k x_p$ where x_p is an i.i.d. realization of n Bernoulli random variables of parameter p . As this data is entirely simulated and not related to the information of real individuals, the added noise does not depend on some privacy budget ε value and is instead directly chosen. Here, a noisy signal $\tilde{f} = f_{0.001,4} + \mathcal{N}(0, \sigma^2 I_n)$ is generated for different values of σ .

Figure 5 presents the average SNR values over 5 noise realizations for our proposed estimator and the oracle DFS fused lasso. We see that the results observed on small graph signals extend well to the large-scale setting, with better performance for the Chebyshev-Jackson polynomial approximation with the SURE Monte-Carlo estimator on a range of input SNR similar to the previous experiments.

Regarding computing time, the DFS fused lasso is, however, more efficient than the approximated SGWT and estimated SURE for this application. On a standard laptop (Intel Core

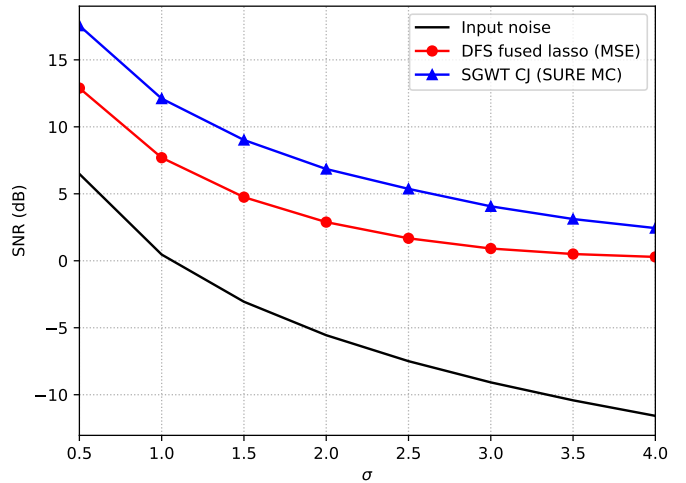


Fig. 5. Average SNR performance over 5 realizations of each noise level setting on the Pennsylvania graph

i5@1.70GHz-16Go DDR4@2400MHz), each of the realizations is denoised in 4 seconds by the regularization method while it takes less about 3 minutes for the latter after an initial estimation of the SURE weights done in 1 minute. We do not observe a significant difference between drawing Monte-Carlo samples from the Rademacher or the Gaussian distributions, both take the same amount of time. In our procedure, the most time-consuming step is the threshold optimization (2m30s), followed by the inverse wavelet transform (25s) and the forward transform (7s).

VI. CONCLUSION AND PERSPECTIVES

In this paper, we propose an extension of SURE to large-scale graphs in the context of signal denoising with thresholding of SGWT coefficients. In particular, we use Monte-Carlo and Chebyshev-Jackson polynomial approximation to build an estimator of its weights in order to avoid the computationally expensive eigendecomposition of the graph Laplacian matrix. Provided expressions for the variance of both weights and SURE estimators show that the Rademacher distribution is better suited than the Gaussian one for this method. We evaluate our data-driven approach through numerical experiments with an application in differential privacy to improve the utility of sanitized graph signals. Results show the MSE can be efficiently estimated with our extended SURE on small and large graphs. Additionally, this methodology shows better performance than the DFS fused lasso.

There is room for improvement in this approach to remove noise with better precision. For instance, the thresholding function we used can be generalized to $\tau(x, t) = x \max\{1 - t^\beta |x|^{-\beta}, 0\}$ with $\beta \geq 1$. Common choices for β include soft thresholding ($\beta = 1$) and hard thresholding ($\beta = \infty$) but an optimization algorithm for this parameter would be more beneficial to further improve performance. An additional thresholding strategy worth considering is block thresholding which partitions wavelet coefficients within each scale to identify localized features in the signal. Depending on the

regularity of the original signal, this can help to remove noise with more accuracy as a different threshold value is selected for each block. An expression of SURE provided for block thresholding processes with SGWT coefficients by [11] could be extended to large graphs with our approach.

Another direction for future research is to adapt our methodology to be run on a distributed system in order to further reduce computing time. First, sanitization with the Gaussian mechanism only consists of the addition of independent Gaussian noise to each node of the graph. Differential privacy in this case can thus be achieved in a distributed manner over subgraphs of the initial graph. Threshold selection by SURE optimization can also be computed in a distributed manner thanks to the additive nature of the SURE formula. But the Laplacian matrix and the SGWT cannot be directly decomposed over separated groups of graph nodes. This yields at least two important challenges in order to distribute 1) the computation of the weights in the SURE formula, and 2) the SGWT thresholding procedure. For specific graph structures (e.g. relatively distinct subgraphs), localization properties of the SGWT would certainly help finding accurate approximations for these distributed computations.

APPENDIX A COMPUTATION OF $\mathbf{V}[\hat{\gamma}_{ij}^2]$

The formula of the weights $\hat{\gamma}_{ij}^2$ is given by

$$\hat{\gamma}_{ij}^2 = \frac{1}{N} \sum_{k=1}^N \left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{pk} \right) \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{pk} \right).$$

The variance of $\hat{\gamma}_{ij}^2$ reads

$$\mathbf{V}[\hat{\gamma}_{ij}^2] = \frac{1}{N} \mathbf{V} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{p1} \right) \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{p1} \right) \right].$$

Then, on the one hand

$$\begin{aligned} & \mathbf{E} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{p1} \right)^2 \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{p1} \right)^2 \right] \\ &= \sum_{p,q,r,s=1}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jr} \mathcal{W}_{js} \mathbf{E}[\varepsilon_{p1} \varepsilon_{q1} \varepsilon_{r1} \varepsilon_{s1}] \\ &= \mathbf{E}[\varepsilon_{11}^4] \sum_{p=1}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jp}^2 + \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,r=1, \\ p \neq r}}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jr}^2 \\ &+ 2\mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jp} \mathcal{W}_{jq}. \end{aligned}$$

On the other hand,

$$\begin{aligned} \mathbf{E} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{p1} \right)^2 \right] &= \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathbf{E}[\varepsilon_{p1} \varepsilon_{q1}] \\ &= \mathbf{E}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip}^2. \end{aligned}$$

Finally,

$$\begin{aligned} & \mathbf{V} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{p1} \right) \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{p1} \right) \right] \\ &= \mathbf{E}[\varepsilon_{11}^4] \sum_{p=1}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jp}^2 + \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,r=1, \\ p \neq r}}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jr}^2 \\ &+ 2\mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jp} \mathcal{W}_{jq} \\ &- \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{p,q=1}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jq}^2 \\ &= \mathbf{V}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip}^2 \mathcal{W}_{jp}^2 \\ &+ 2\mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{iq} \mathcal{W}_{jp} \mathcal{W}_{jq}. \end{aligned}$$

APPENDIX B COMPUTATION OF $\mathbf{V}[\widehat{\text{SURE}}(h)|\tilde{F}]$

From Equation (1), it follows that

$$\begin{aligned} \mathbf{V}[\widehat{\text{SURE}}(h)|\tilde{F}] &= 4\sigma^4 \mathbf{E} \left[\left(\sum_{i,j=1}^{n(J+1)} (\hat{\gamma}_{ij}^2 - \gamma_{ij}^2) \partial_j h_i(\tilde{F}) \right)^2 \right] \\ &= 4\sigma^4 \sum_{i,j,k,\ell=1}^{n(J+1)} \partial_j h_i(\tilde{F}) \partial_k h_\ell(\tilde{F}) \mathbf{E}[(\hat{\gamma}_{ij}^2 - \gamma_{ij}^2)(\hat{\gamma}_{k\ell}^2 - \gamma_{k\ell}^2)] \\ &= 4\sigma^4 \sum_{i,j,k,\ell=1}^{n(J+1)} \partial_j h_i(\tilde{F}) \partial_k h_\ell(\tilde{F}) [\mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] - \gamma_{ij}^2 \gamma_{k\ell}^2]. \end{aligned}$$

Then,

$$N^2 \mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] = \sum_{a,b=1}^N \mathbf{E} \left[\left(\sum_{p=1}^n \mathcal{W}_{ip} \varepsilon_{pa} \right) \left(\sum_{p=1}^n \mathcal{W}_{jp} \varepsilon_{pa} \right) \left(\sum_{p=1}^n \mathcal{W}_{kp} \varepsilon_{pb} \right) \left(\sum_{p=1}^n \mathcal{W}_{lp} \varepsilon_{pb} \right) \right].$$

Developing each term of the sum above, it follows

$$\begin{aligned} N^2 \mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] &= \sum_{a,b=1}^N \sum_{p,q,r,s=1}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kr} \mathcal{W}_{ls} \mathbf{E}[\varepsilon_{pa} \varepsilon_{qa} \varepsilon_{rb} \varepsilon_{sb}]. \end{aligned}$$

Thus, on the one hand

$$\begin{aligned}
N^2 \mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] &= N \mathbf{E}[\varepsilon_{11}^4] \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kp} \mathcal{W}_{\ell p} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kp} \mathcal{W}_{\ell q} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kq} \mathcal{W}_{\ell p} \\
&+ N(N-1) \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\gamma_{ij}^2 \gamma_{k\ell}^2 &= \mathbf{E}[\varepsilon_{11}^2]^2 \left(\sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \right) \left(\sum_{p=1}^n \mathcal{W}_{kp} \mathcal{W}_{\ell p} \right) \\
&= \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q}.
\end{aligned}$$

Finally, the difference is given by

$$\begin{aligned}
N^2 \mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] - N^2 \gamma_{ij}^2 \gamma_{k\ell}^2 &= N \mathbf{E}[\varepsilon_{11}^4] \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kp} \mathcal{W}_{\ell p} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kp} \mathcal{W}_{\ell q} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kq} \mathcal{W}_{\ell p} \\
&+ N(N-1) \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q} \\
&- N^2 \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{p,q=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kq} \mathcal{W}_{\ell q}.
\end{aligned}$$

Hence,

$$\begin{aligned}
N^2 \mathbf{E}[\hat{\gamma}_{ij}^2 \hat{\gamma}_{k\ell}^2] - N^2 \gamma_{ij}^2 \gamma_{k\ell}^2 &= N \mathbf{V}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kp} \mathcal{W}_{\ell p} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kp} \mathcal{W}_{\ell q} \\
&+ N \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kq} \mathcal{W}_{\ell p}.
\end{aligned}$$

Summarizing,

$$\begin{aligned}
\mathbf{V}[\widehat{\text{SURE}}(h)|\tilde{F}] &= \frac{4\sigma^4}{N} \sum_{i,j,k,\ell=1}^{n(J+1)} \partial_j h_i(\tilde{F}) \partial_k h_\ell(\tilde{F}) \left\{ \right. \\
&\mathbf{V}[\varepsilon_{11}^2] \sum_{p=1}^n \mathcal{W}_{ip} \mathcal{W}_{jp} \mathcal{W}_{kp} \mathcal{W}_{\ell p} \\
&+ \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kp} \mathcal{W}_{\ell q} \\
&\left. + \mathbf{E}[\varepsilon_{11}^2]^2 \sum_{\substack{p,q=1, \\ p \neq q}}^n \mathcal{W}_{ip} \mathcal{W}_{jq} \mathcal{W}_{kq} \mathcal{W}_{\ell p} \right\}.
\end{aligned}$$

ACKNOWLEDGMENT

The authors are very grateful to the Associate Editor and the two anonymous referees for their thorough and useful comments.

REFERENCES

- [1] T. Arnold, V. Sadhanala, and R. Tibshirani. glmgen: Fast algorithms for generalized lasso problems. 2014. R package version 0.0.3.
- [2] B. Balle and Y.-X. Wang. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, pages 394–403. PMLR, 2018.
- [3] H. Behjat, U. Richter, D. Van De Ville, and L. Sornmo. Signal-Adapted Tight Frames on Graphs. *IEEE Transactions on Signal Processing*, 64(22):6017–6029, nov 2016.
- [4] G. Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, sep 2017.
- [5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [6] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.
- [7] S. Chen, A. Singh, and J. Kovaevi. Multiresolution representations for piecewise-smooth signals on graphs, 2018.
- [8] R. R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53 – 94, 2006. Special Issue: Diffusion Maps and Wavelets.
- [9] T. Coulhon, G. Kerkycharian, and P. Petrushev. Heat kernel generated frames in the setting of dirichlet spaces. *Journal of Fourier Analysis and Applications*, 18(5):995–1066, 2012.
- [10] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.
- [11] B. de Loynes, F. Navarro, and B. Olivier. Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. *J. Comput. Appl. Math.*, 389:113319, 2021.

- [12] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, 2016.
- [13] E. Di Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, 2016.
- [14] 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*, 37(6):117–127, 2020.
- [15] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.
- [16] M. Douriez, H. Doraiswamy, J. Freire, and C. T. Silva. Anonymizing nyc taxi data: Does it matter? In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 140–148, 2016.
- [17] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [18] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.
- [19] G. Fu, Y. Hou, J. Zhang, K. Ma, B. F. Kamhoua, and J. Cheng. Understanding graph neural networks from graph signal denoising perspectives. *arXiv preprint arXiv:2006.04386*, 2020.
- [20] B. Girault, A. Ortega, and S. S. Narayanan. Irregularity-aware graph fourier transforms. *IEEE Transactions on Signal Processing*, 66(21):5746–5761, 2018.
- [21] F. Göbel, G. Blanchard, and U. von Luxburg. Construction of tight frames on graphs and application to denoising. In *Handbook of big data analytics*, Springer Handb. Comput. Stat., pages 503–522. Springer, Cham, 2018.
- [22] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.
- [23] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [24] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. In *NIPS*, 2015.
- [25] O. Hernan Madrid Padilla, J. Sharpnack, J. G. Scott, and R. J. Tibshirani. The DFS Fused Lasso: Linear-Time Denoising over General Graphs. *Journal of Machine Learning Research*, 18(176):1–36, 2018.
- [26] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990.
- [27] L. O. Jay, H. Kim, Y. Saad, and J. R. Chelikowsky. Electronic structure calculations for plane-wave codes without diagonalization. *Computer Physics Communications*, 118(1):21–30, 1999.
- [28] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [29] N. Leonardi and D. Van De Ville. Tight wavelet frames on multislice graphs. *IEEE Transactions on Signal Processing*, 61(13):3357–3367, 2013.
- [30] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics*, 6(1):29–123, 2009.
- [31] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.
- [32] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka. Graph signal denoising via trilateral filter on graph spectral domain. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):137–148, 2016.
- [33] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.
- [34] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAW-DAD dataset epfl/mobility (v. 2009-02-24). Downloaded from <https://crawdad.org/epfl/mobility/20090224>, Feb. 2009.
- [35] S. Ramani, T. Blu, and M. Unser. Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, 2008.
- [36] D. Shuman, S. 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*, 3(30):83–98, 2013.
- [37] D. I. Shuman. Localized spectral graph filter frames: A unifying framework, survey of design considerations, and numerical comparison. *IEEE Signal Processing Magazine*, 37(6):43–63, 2020.
- [38] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.
- [39] N. Tremblay and P. Borgnat. Graph wavelets for multiscale community mining. *IEEE Trans. Signal Process.*, 62(20):5227–5239, 2014.
- [40] Y.-X. Wang, J. Sharpnack, A. Smola, and R. J. Tibshirani. Trend Filtering on Graphs. *Journal of Machine Learning Research*, 17:1–41, 2016.
- [41] D. S. Weller, S. Ramani, J.-F. Nielsen, and J. A. Fessler. Monte carlo sure-based parameter selection for parallel magnetic resonance imaging reconstruction. *Magnetic Resonance in Medicine*, 71(5):1760–1770, 2014.

Elie Chedemail Elie Chedemail received the B.Sc. degree in mathematics and economics from the University of Rennes 1, Rennes, France in 2016 and the M.Sc. degree in statistics from the National School for Statistics and Data Analysis (ENSAI), Bruz, France in 2019. He is currently working towards the Ph.D. degree at ENSAI and Orange Labs, Cesson-Svign, France. His research interests include machine learning, graph signal processing and differential privacy.

Basile de Loynes Basile de Loynes received a Ph.D. degree in fundamental mathematics from the University of Rennes (France) in 2012. From 2012 to 2013, he was a postdoctoral researcher with the Department of Mathematics at the University of Neuchtel (Switzerland). From 2013 to 2014, he was an assistant professor at the University of Burgundy (France) and from 2014 to 2016 he was an assistant professor at the University of Strasbourg (France). He is currently an associate professor at CREST-Ensaï. His research interests include random walks on graphs, long memory stochastic processes, functional analysis and graph signal processing.

Fabien Navarro received the B.Sc., M.Sc. and Ph.D. degrees in Applied Mathematics from the University of Caen, Caen, France, in 2008, 2010 and 2013, respectively. From 2014 to 2015, he was a Research Assistant Professor with the department of Mathematics and Statistics, Concordia University, Montreal, Canada. From 2015 to 2021, he was an Assistant Professor with the Center for Research in Economics and Statistic, Ecole Nationale de la Statistique et de l'Analyse de l'Information, Bruz, France. He is currently an Associate Professor with the University of Paris 1 Panthéon-Sorbonne, Paris, France. His research interests include nonparametric statistics, inverse problems, computational harmonic analysis, sparse representations, machine learning and statistical approaches in graph signal processing.

Baptiste Olivier Baptiste Olivier received a Ph.D. degree in fundamental mathematics from the University of Rennes (France) in 2013. From 2013 to 2015, he was a postdoctoral researcher with the Department of Mathematics at the TECHNION in Haifa (Israel). From 2016 to 2021, he was a research and data scientist at Orange in Rennes (France). He is currently a senior data scientist at Ericsson in Stockholm (Sweden). His research interests include property (T), functional analysis, machine learning, graph signal processing and differential privacy.