Node-Adaptive Regularization for Graph Signal Reconstruction

MAOSHENG YANG [ID], MARIO COUTINO, GEERT LEUS [ID] (Fellow, IEEE), AND ELVIN ISUFI [ID] (Member, IEEE)

Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft, 2628 CD Zuid-Holland, The Netherlands

CORRESPONDING AUTHOR: MAOSHENG YANG (e-mail: m.yang-2@tudelft.nl)

ABSTRACT A critical task in graph signal processing is to estimate the true signal from noisy observations over a subset of nodes, also known as the reconstruction problem. In this paper, we propose a node-adaptive regularization for graph signal reconstruction, which surmounts the conventional Tikhonov regularization, giving rise to more degrees of freedom; hence, an improved performance. We formulate the node-adaptive graph signal denoising problem, study its bias-variance trade-off, and identify conditions under which a lower mean squared error and variance can be obtained with respect to Tikhonov regularization. Compared with existing approaches, the node-adaptive regularization enjoys more general priors on the local signal variation, which can be obtained by optimally designing the regularization weights based on Prony's method or semidefinite programming. As these approaches require additional prior knowledge, we also propose a *minimax* (worst-case) strategy to address instances where this extra information is unavailable. Numerical experiments with synthetic and real data corroborate the proposed regularization strategy for graph signal denoising and interpolation, and show its improved performance compared with competing alternatives.

INDEX TERMS Bias-variance trade-off, graph regularization, graph signal denoising, graph signal processing.

I. INTRODUCTION

Graphs, as models to represent data, do not only capture information about entities (nodes) that comprise them, but also encode the interactions between these entities (edges). Graphs are promising to deal with high-dimensional data arising in many applications such as social media, sensor networks, and transportation networks, to name a few [2]–[8]. To analyze such networked data, tools from signal processing have been extended in the area of *graph signal processing* (GSP) [5], [9].

As in classical signal processing, the task of estimating the underlying signal from noisy observations, is critical in GSP as well. Over the past few years, a large amount of research has been focused on this topic [5], [9]–[20]. Within this body of work, most approaches solve a least squares problem penalized by different regularizers. The most commonly penalization method is the one based on the so-called Tikhonov regularization [21]. It recovers the graph signal by penalizing the data fitting term with the graph Laplacian quadratic form, also known as graph Laplacian regularizer (GLR) [17], [22]. This GLR based method and its variants have succeeded in many applications such as image smoothing, point cloud denoising, semi-supervised learning, data classification and graph signal denoising [5], [23]–[28].[1]

Other regularizers have also shown their applicability in different scenarios. For example, the definition of total variation of discrete signals has been extended to graphs, i.e., the graph signal total variation notion based on the adjacency matrix, graph Laplacian and their variants [24], [29]–[31]. In [32], it is used as a graph signal regularizer to perform

---

[1]Note that in [25], [27], [28] the graph edge weights (the graph Laplacian) are not fixed but (designed) updated in a specific way for tasks like image and point could denoising. This edge weights updating idea seem to be related to the idea introduced in our paper, but they are specially designed for/based on the given tasks, for instance by exploiting the properties of images and point clouds. This is conceptually different from the idea in this article.

trend filtering on graphs, effectively extending the generalized Lasso problem [33] to graphs. By generalizing total variation to the graph setting, we can consider higher-order graph signal differences across the nodes and performs graph signal reconstruction in instances where the graph signal is not necessarily globally smooth. Similarly, in [18]–[20], the notion of graph Laplacian regularization is extended by means of graph kernels such as the diffusion kernel, random walk based kernels and graph bandlimited kernels. As these regularizers tend to increase the computational complexity of the reconstruction task, several works have been focused on reducing of the complexity for finding regularized solutions by means of iterative and distributed implementations, such as adaptive graph signal estimation, primal-dual gradient method, Chebyshev polynomial approximation and so on [10]–[16].

Although state-of-the-art address the graph signal reconstruction, almost all methods adopt a single parameter to control the reconstruction performance; that is, to balance the fitting error with regularization term. For instance, in the recent work [17], the authors investigate the bias-variance trade-off for Tikhonov regularizer and propose an optimally design strategy for the regularization parameter which matches the order of the bias and variance term. Despite this effort, it is clear that a scalar regularization parameter is insufficient to impose local penalties over the graph signal. To see this, consider an instance of the Tikhonov denoising that penalizes the error fitting term with signal smoothness. A scalar regularization weight can only penalize in terms of the *global* smoothness, instead of local variability. Such a local penalty is important, for instance, for signals that are piecewise-smooth or piecewise-constant [25], [26], [32], [34], or in some networks where we want to detect anomaly nodes, etc. The global penalty term limits the denoising performance to deal with signals with general characteristics. Thus an increased number of regularization parameters for handling such situations is needed. In an attempt to increase the degrees of freedom (DoFs) of the regularization, in [15], it is proposed to minimize the signal total smoothness by regularizing separately the fitting error of each individual nodal measurement. Despite that this approach can be considered as a multi-parameter based regularization, it only focuses on a measurement-wise regularization and ignores the coupling between the graph signal and the topology.

Due to the need of a multi-parameter regularization, which considers the connectivity of the graph, in this paper, we propose a *node-adaptive (NA) regularizer* to increase the DoFs by applying node-dependent weights to fine-tune the trade-off between the fitting error and regularization term. With these enhanced DoFs, we expect to achieve a better reconstruction performance without affecting the method complexity. For the sake of exposition, we solely focus on the *NA Tikhonov-based reconstruction* to make direct comparisons with earlier works, e.g., [17], however, our findings can be generalized with the adequate changes to other regularization penalties, e.g., a graph shift operator based ridge regression penalty [19], [20].

A. CONTRIBUTIONS

To be more specific, the main questions we address in this work are: (i) *how the bias-variance trade-off behaves for the NA Tikhonov regularization?* and (ii) *how to design the NA weights optimally?* Aimed to give answers to these questions, we make the following main contributions:

1) We formulate the NA Tikhonov regularization problem under a deterministic signal model assumption, derive its solution in closed-form, and study the respective bias-variance trade-off.

2) We derive the conditions for the NA weights to allow a smaller mean-squared error (MSE) and variance compared to traditional Tikhonov regularization.

3) To design the NA weights, we propose three methods based on minimizing the MSE. The first two methods leverage Prony's method from classical signal processing and convex relaxation techniques. The third method uses a *minimax* strategy to design the weights in a worst-case setting. The latter addresses scenarios where only upper- and lower-bounds of graph signals are available.

4) We corroborate the theoretical findings of this work, using both synthetic and real-world data, and show that the proposed NA denoising performs well with respect to the competing alternatives, especially in low signal-to-noise ratio (SNR) settings.

B. OUTLINE AND NOTATION

The rest of this paper is structured as follows: Section II reviews the conventional Tikhonov regularizer. Section III formulates the NA Tikhonov regularization problem and studies its bias-variance trade-off. Section IV presents the proposed optimal designs for the NA regularizer weights. Section V contains numerical results to validate the theoretical findings. Finally, Section VI concludes the paper.

Notation: Scalars, vectors, matrices and sets are denoted by lowercase letters (x), lowercase bold letters ($\mathbf{x}$), uppercase bold letters ($\mathbf{X}$), and calligraphic letters ($\mathcal{X}$), respectively. x_i represents the i-th entry of the vector $\mathbf{x}$, and X_{ij} denotes the (i, j)th element of the matrix $\mathbf{X}$. $\mathbf{X}^\top$ and $\mathbf{X}^{-1}$ are the transpose and inverse of the matrix $\mathbf{X}$. $\mathbf{1}$ and $\mathbf{I}$ are the all-one vector and identity matrix. $\mathrm{diag}(\cdot)$ is a diagonal matrix with its arguments on the main diagonal. $\|\cdot\|_2$ is the Euclidean norm. We use $\mathbb{R}$ to denote the set of real numbers and $\mathcal{S}_+^{N \times N}$ to denote the set of $N \times N$ positive semi-definite matrices. $\mathbb{E}[\cdot]$ is the expectation operator, $\mathrm{tr}(\cdot)$ is the trace operator and $\mathrm{supp}(\mathbf{X})$ is the support of $\mathbf{X}$. If $\mathbf{x}$ is a random variable, then its covariance matrix is $\mathrm{cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top]$.

II. NODE-INVARIANT REGULARIZER

Consider an *undirected* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of N nodes and $\mathcal{E}$ the set of M edges such that if nodes i and j are connected, then $(i, j) \in \mathcal{E}$. The neighborhood set of node i is $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$. The graph can be represented by its adjacency matrix $\mathbf{A}$ with entry

$A_{ij} \geq 0$ if $(i, j) \in \mathcal{E}$ and $A_{ij} = 0$, otherwise. Alternatively, the graph can also be represented by its graph Laplacian matrix $\mathbf{L} = \text{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}$.

On the vertices of $\mathcal{G}$, we define a graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top$ whose i-th element x_i is the signal value on node i. The graph signal smoothness can be measured by the *graph Laplacian quadratic form*

$$S_2(\mathbf{x}) = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} A_{ij}(x_i - x_j)^2 = \mathbf{x}^\top \mathbf{L} \mathbf{x}. \tag{1}$$

A signal $\mathbf{x}$ is said to be *smooth* over $\mathcal{G}$, if $S_2(\mathbf{x})$ is small [5]. This measure is commonly used as a regularizer to recover smooth graph signals from noisy measurements [5], [9], [17].

Consider the measurements $\mathbf{y} = \mathbf{x}^* + \mathbf{n}$, where $\mathbf{x}^* \in \mathbb{R}^N$ is a smooth graph signal and $\mathbf{n} \in \mathbb{R}^N$ is zero-mean noise with covariance matrix $\mathbf{\Sigma}$. Assuming $\mathbf{x}^*$ is smooth over the graph, we can recover it by solving the Tikhonov regularized problem

$$\hat{\mathbf{x}}(\omega_0) = \underset{\mathbf{x} \in \mathbb{R}^N}{\arg \min} \ \|\mathbf{y} - \mathbf{x}\|_2^2 + \omega_0 \mathbf{x}^\top \mathbf{L} \mathbf{x} \tag{2}$$

where scalar $\omega_0 > 0$ is the regularization parameter. The first term in (2) forces the estimate to be close to the observed signal (fitting term), while the second term promotes global signal smoothness. The trade-off between these two quantities is governed by the scalar weight ω_0. The closed-formed solution of (2) is

$$\hat{\mathbf{x}}(\omega_0) = (\mathbf{I} + \omega_0 \mathbf{L})^{-1} \mathbf{y} := \mathbf{H}(\omega_0)\mathbf{y} \tag{3}$$

where we defined the graph filter $\mathbf{H}(\omega_0) \triangleq (\mathbf{I} + \omega_0 \mathbf{L})^{-1}$ [35].

The smooth regularizer in (2) biases the estimator $\hat{\mathbf{x}}(\omega_0)$ in (3). The bias $\mathbf{b}(\omega_0)$ is given by

$$\mathbf{b}(\omega_0) = \mathbb{E}[\hat{\mathbf{x}}(\omega_0)] - \mathbf{x}^* = (\mathbf{H}(\omega_0) - \mathbf{I})\mathbf{x}^* \tag{4}$$

which is controlled by ω_0. The variance is also controlled by ω_0 and has the form

$$\text{var}(\omega_0) = \mathbb{E}[\|\hat{\mathbf{x}}(\omega_0) - \mathbb{E}(\hat{\mathbf{x}}(\omega_0))\|^2] = \text{tr}(\mathbf{H}^2(\omega_0)\mathbf{\Sigma}). \tag{5}$$

By combining the bias and the variance, we can quantify the performance of the estimator in (3) through its MSE

$$\text{mse}(\omega_0) = \mathbb{E}[\|\hat{\mathbf{x}}(\omega_0) - \mathbf{x}^*\|_2^2] = \|\mathbf{b}(\omega_0)\|_2^2 + \text{var}(\omega_0)$$
$$= \text{tr}((\mathbf{I} - \mathbf{H}(\omega_0))^2 \mathbf{x}^* \mathbf{x}^{*\top}) + \text{tr}(\mathbf{H}^2(\omega_0)\mathbf{\Sigma}). \tag{6}$$

The MSE shows the bias-variance trade-off imposed by the smoothness regularizer. If scalar ω_0 is reduced, we achieve a lower bias but a higher variance, and vice-versa.

Ways to select the parameter ω_0 can be devised using, e.g., the discrepancy principle [36], [37], the L-curve criterion [38] and the generalized cross-validation [39]. In GSP, a natural optimal parameter selection criterion is based on the minimization of the MSE, which is investigated in [17, Thm. 3], where parameter ω_0 is optimally found by matching the order of the bias contribution and the variance.

The regularizer in (1) is a global graph signal measure, i.e., the signal smoothness over the entire graph. And the impact of this global regularizer in (2) is controlled by the scalar ω_0. We can differently think of ω_0 as a common coefficient that equally pre-weights the signal on all nodes when computing the smoothness measure, i.e., $\sqrt{\omega_0} \mathbf{x}^\top \mathbf{L} \sqrt{\omega_0} \mathbf{x}$. Due to this characteristic, we refer to problem (2) as a node-invariant (NI) regularization problem. If the signal is globally smooth over the underlying graph, the NI regularization will result in a good signal reconstruction performance. However, as it only focuses on global behaviours, ignoring local signal details, it might lead to unsatisfactory recovery performance when the graph signal is not globally smooth and/or when the noise level is different at each node.

Although the graph signal is not globally smooth in most situations, it presents a smooth behaviour in different regions, e.g., smoothness over local details. This prior generalizes the global smoothness and is amenable for learning. To enhance the role of local signal detail and improve the estimator's MSE, we propose a graph signal regularization strategy that substitutes the global penalty term $S_2(\mathbf{x})$ with a local penalty on each node. The proposed approach, named node-adaptive (NA) regularization, allows each node i to weight its signal x_i with an individual scalar ω_i. The enhanced DoFs of the NA regularizer can in turn improve the bias-variance trade-off and reduce the overall MSE. To understand the NA regularizer, we first conduct a detailed bias-variance trade-off analysis and then propose a design strategy for the node-adaptive weights $\omega_1, \ldots, \omega_N$ to optimize such a trade-off. Therefore, our goal in the upcoming sections is to formalize the NA regularization problem, analyze its statistical properties, and develop optimal weight design strategies that minimize the MSE.

III. NODE-ADAPTIVE REGULARIZER

Consider a vector of parameters $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_N]^\top \in \mathbb{R}^N$ and define the *node-adaptive Laplacian operator*

$$\mathbf{S}(\boldsymbol{\omega}) \triangleq \text{diag}(\boldsymbol{\omega})\mathbf{L}\text{diag}(\boldsymbol{\omega}) = \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L} \tag{7}$$

where $\odot$ is the element-wise Hadamard product. Note that for any $\boldsymbol{\omega}$, the parametric shift operator matrix $\mathbf{S}(\boldsymbol{\omega})$ is positive semi-definite (see Lemma 5 in Appendix A) and has the same support as the graph Laplacian $\mathbf{L}$ –properties that will result useful in the sequel. The parameterized graph Laplacian quadratic form for (7) has the form [cf.(1)]

$$S_2(\mathbf{x}, \boldsymbol{\omega}) = \mathbf{x}^\top \mathbf{S}(\boldsymbol{\omega})\mathbf{x} = (\text{diag}(\boldsymbol{\omega})\mathbf{x})^\top \mathbf{L}(\text{diag}(\boldsymbol{\omega})\mathbf{x})$$
$$= \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} A_{ij}(\omega_i x_i - \omega_j x_j)^2. \tag{8}$$

This quadratic form can be seen as, first pre-weighting each entry x_i of the signal $\mathbf{x}$ with a parameter w_i and then computing the regular quadratic measure in (1) w.r.t. Laplacian $\mathbf{L}$. We can now use (8) to recover a graph signal $\mathbf{x}^*$ from the noisy measurements $\mathbf{y}$ by solving the convex problem

$$\hat{\mathbf{x}}(\boldsymbol{\omega}) = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \ \|\mathbf{y} - \mathbf{x}\|_2^2 + S_2(\mathbf{x}, \boldsymbol{\omega}), \tag{9}$$

where the trade-off between the fitting error and the regularization term is now controlled by the N parameters in $\boldsymbol{\omega}$. We name (9) as *node-adaptive graph signal regularization*. We give the following comments on this regularization framework.

- When $\boldsymbol{\omega} = \omega_0 \mathbf{1}$, NA graph signal regularization particularizes to the NI case in (2).
- NA regularization is not restricted to the form in (9). The same generalization can be applied when variants of the graph Laplacian are used in the regularizer. For example, if we use a random walk graph Laplacian in (9), the GLR used in [26] is a special case.
- We can see that NA graph signal regularization uses a parametric graph Laplacian of form (7) where the edge weights are adapted by $\boldsymbol{\omega}\boldsymbol{\omega}^\top$ which will be optimally designed in the sense of minimizing MSE. In this way, our work acts as an alternative or a generalization to the proposed GLR in image and point could denoising [25], [28] where the graph edge weights are (designed) updated in a specific task-driven way, for instance, as a function of (feature) difference between two connected nodes.

To see the impact on local graph signal details in problem (9), consider a simple piecewise smooth graph signal with slowly varying nonzero values in a node subset $\mathcal{V}' \subseteq \mathcal{V}$ and zero on the remaining nodes in $\mathcal{V}\backslash\mathcal{V}'$. We can then weight the nodes in $\mathcal{V}'$ with a common scalar, and another different scalar for the remaining nodes, since the signal is not globally smooth but locally in two separate parts. Problem (9) will seek for a graph signal that is locally smooth on $\mathcal{V}'$ and on $\mathcal{V}\backslash\mathcal{V}'$. Instead, problem (2) will weight every node with ω_0, effectively looking for a graph signal that is smooth over all nodes and ignores the local detail. This simplified example shows how the NA regularization generalizes the NI case by considering a local smoothness penalty and relates to the piecewise-constant, -smooth notions. In the sequel, we design parameter $\boldsymbol{\omega}$ to learn the local signal smoothness priors.

The optimal analytical solution for problem (9) can be found by setting its gradient to zero, i.e.,

$$\hat{\mathbf{x}}(\boldsymbol{\omega}) = (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))^{-1}\mathbf{y} := \mathbf{H}(\boldsymbol{\omega})\mathbf{y} \qquad (10)$$

where we have defined the NA filter[2] $\mathbf{H}(\boldsymbol{\omega}) \triangleq (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))^{-1}$, which is PSD by definition. Despite the similarity with (3), the optimal solution in (10) is now governed by the vector of parameters $\boldsymbol{\omega}$. This vector changes the bias-variance trade-off as we discuss in the next section.

Remark 1: The optimal regularized solutions (3) and (10) can also be interpreted as graph filtering operations [5]. In particular, while (3) filters the measurements $\mathbf{y}$ with an autoregressive graph filter with denominator coefficients $(1; \omega_0)$ which is common for all nodes [35], the node-adaptive expression (10) filters $\mathbf{y}$ with an autoregressive edge-varying filter with edge varying coefficients $(\mathbf{I}; \boldsymbol{\omega}\boldsymbol{\omega}^\top)$ [41].

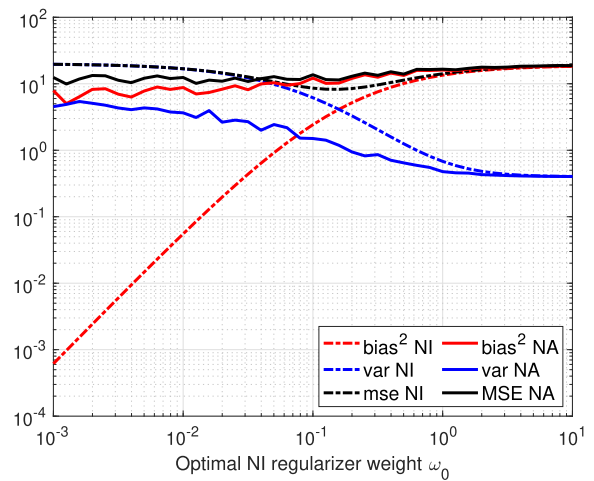[2]Here the definition of NA filter is different from the node variant graph filter in [40]

FIGURE 1. Bias-variance trade-off for recovering a graph signal with the NA and NI regularizer over an Erdos-Renyi graph, SNR = 0 dB. The detailed settings are in Section V-A. The node adaptive weights are chosen randomly to satisfy the result in Lemma 1.

A. BIAS-VARIANCE TRADE-OFF

Similar to (6), the MSE for estimator (10) has the form

$$\mathrm{mse}(\boldsymbol{\omega}) = \|\mathbf{b}(\boldsymbol{\omega})\|_2^2 + \mathrm{var}(\boldsymbol{\omega})$$
$$= \mathrm{tr}((\mathbf{I} - \mathbf{H}(\boldsymbol{\omega}))^2 \mathbf{x}^* \mathbf{x}^{*\top}) + \mathrm{tr}(\mathbf{H}(\boldsymbol{\omega})^2 \boldsymbol{\Sigma}) \qquad (11)$$

with bias

$$\mathbf{b}(\boldsymbol{\omega}) = ((\mathbf{I} + \mathrm{diag}(\boldsymbol{\omega})\mathbf{L}\mathrm{diag}(\boldsymbol{\omega}))^{-1} - \mathbf{I})\mathbf{x}^* \qquad (12)$$

and variance

$$\mathrm{var}(\boldsymbol{\omega}) = \mathrm{tr}((\mathbf{I} + \mathrm{diag}(\boldsymbol{\omega})\mathbf{L}\mathrm{diag}(\boldsymbol{\omega}))^{-2}\boldsymbol{\Sigma}). \qquad (13)$$

As it follows from (11)–(13), the bias-variance trade-off is now controlled by $\boldsymbol{\omega}$. If all entries of $\boldsymbol{\omega}$ are close to zero the bias is low and the MSE is governed by a high variance. If all entries of $\boldsymbol{\omega}$ are far from zero, the bias is large and governs the MSE, achieving a small variance. However, the enhanced DoFs of the NA regularizer compared to the NI one allow us to identify an interval for $\boldsymbol{\omega}$ that guarantees a smaller reconstruction variance while maintaining a lower MSE.

Lemma 1: Consider the node-invariant and the node-adaptive estimates $\hat{\mathbf{x}}(\omega_0)$ and $\hat{\mathbf{x}}(\boldsymbol{\omega})$ as in (3) and (10), respectively. Consider also the respective variances over all nodes $\mathrm{var}(\omega_0)$ in (5) and $\mathrm{var}(\boldsymbol{\omega})$ in (13). If all node-adaptive weights $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_N]^\top$ satisfy

$$\omega_0 \leq \omega_i^2, \quad \text{for } i = 1, 2, \ldots, N \qquad (14)$$

then $\mathrm{var}(\boldsymbol{\omega}) \leq \mathrm{var}(\omega_0)$.

Proof: See Appendix B. ∎

While a reduced variance is useful for signal recovery, it often comes at expenses of an increased bias. To see how sensitive the changes in the two quantities are, we illustrate in Fig. 1 the bias, the variance, and the MSE for the NI and the NA regularizers. We see there exists a region for the NA

weights where both the variance and the MSE of the NA regularizer are lower compared with those of the NI regularizer, while the bias is not increased significantly. The following theorem provides sufficient conditions on $\boldsymbol{\omega}$ to identify this region.

Theorem 1: Consider the measurements $\mathbf{y} = \mathbf{x}^* + \mathbf{n}$ with desired graph signal $\mathbf{x}^*$ and noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Let $\mathbf{L}$ be the graph Laplacian with maximum eigenvalue $\lambda_{\max}(\mathbf{L})$. Further, define a rank-one matrix $\mathbf{P} := \mathbf{x}^* \mathbf{x}^{*\top} \boldsymbol{\Sigma}^{-1}$ and let ρ be its only non-zero eigenvalue. Define also a rank-one matrix $\boldsymbol{\Gamma} = \mathbf{P}(\mathbf{I} + \mathbf{P})^{-1}$ and let $\gamma = \rho(1 + \rho)^{-1} \in (0, 1)$ be its only non-zero eigenvalue. Consider also the mean squared error of the node-invariant estimate $\hat{\mathbf{x}}(\omega_0)$ in (3) and node-adaptive estimate $\hat{\mathbf{x}}(\boldsymbol{\omega})$ in (10). Then, if

$$\omega_0 \leq \omega_i^2, \text{ for } i = 1, 2, \ldots, N \quad (15a)$$

$$2\gamma \leq \frac{1}{1 + \omega_0 \lambda_{\max}(\mathbf{L})} + \frac{1}{1 + \max\left\{\omega_i^2\right\} \lambda_{\max}(\mathbf{L})} \quad (15b)$$

both the variance and the mean squared error of the node-adaptive regularizer are smaller than those of the node-invariant one; i.e., $\mathrm{var}(\boldsymbol{\omega}) \leq \mathrm{var}(\omega_0)$ and $\mathrm{mse}(\boldsymbol{\omega}) \leq \mathrm{mse}(\omega_0)$.

Proof: See Appendix C. ∎

Condition (15b) is easier satisfied when the eigenvalue $\gamma \rightarrow 0$, i.e., the signal-to-noise ratio (SNR) is low, or $\rho \rightarrow 0$. This indicates that the NA is more powerful in harsher scenarios. In contrast, when $\gamma \rightarrow 1$ and, thus $\rho \rightarrow \infty$, i.e., the SNR is high, the condition for the NA regularization to outperform NI one is hard to satisfy.

Corollary 1: Under the same settings of Theorem 1, the condition

$$\max\left\{\omega_i^2\right\} \leq (\rho \lambda_{\max}(\mathbf{L}))^{-1}, \text{ for } i = 1, 2, \ldots, N \quad (16)$$

guarantees condition (15b) is satisfied.

Proof: See Appendix D. ∎

Corollary 1 provides an easier condition for the NA parameters $\boldsymbol{\omega}$ compared to Theorem 1. In addition, condition (16) provides a clearer link between $\boldsymbol{\omega}$ and the SNR. If $\rho \rightarrow 0$, hence $\gamma \rightarrow 0$, the noise is high and the upper bound for ω_i^2 increases, implying a larger weight on local smoothness is needed. If $\rho \rightarrow \infty$, hence $\gamma \rightarrow 1$, the noise vanishes and the upper bound for ω_i^2 goes to zero, implying the regularization has little effect. This performance analysis the NA regularizer compared to the NI one is supported by our numerical findings in Section V.

B. IMPLEMENTATION

The matrix inversion in optimal estimate $\hat{\mathbf{x}}(\boldsymbol{\omega})$ [cf. (10)] makes the node-adaptive regularizer challenging to be implemented on large graphs or in a distributed manner. Fortunately, in both cases the inverse can be approximated with a cost linear w.r.t. the number of graph edges, i.e., $\mathcal{O}(M)$, by leveraging the graph filtering equivalence of (10); see, e.g., [35], [41], [42]. The key to such linear cost lies in the sparsity of the parametric shift operator $\mathbf{S}(\boldsymbol{\omega})$, which coincides with the sparsity of the

---

Algorithm 1: Conjugate Gradient Method for Solving (18).

1:    **Input:** $\hat{\mathbf{x}}_{(0)}$, node-adaptive regularizer weights $\boldsymbol{\omega}$, accuracy ϵ, number of iterations T
2:    **Initialization:**
3:    $\mathbf{S}(\boldsymbol{\omega}) = \mathrm{diag}(\boldsymbol{\omega}) \mathbf{L} \mathrm{diag}(\boldsymbol{\omega})$
4:    $\mathbf{b}_{(0)} = \mathbf{r}_{(0)} = \mathbf{y} - (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))\hat{\mathbf{x}}_{(0)}$
5:    $d_{(0)} = d_{new} = \mathbf{r}_{(0)}^{\top} \mathbf{r}_{(0)}$
6:    **while** $\tau < T$ and $d_{new} > \epsilon^2 d_{(0)}$
7:      $c_{(\tau)} = \frac{d_{new}}{\mathbf{b}_{(\tau)}^{\top}(\mathbf{I}+\mathbf{S}(\boldsymbol{\omega}))\mathbf{b}_{(\tau)}}$
8:      $\hat{\mathbf{x}}_{(\tau+1)} = \hat{\mathbf{x}}_{(\tau)} + c_{(\tau)} \mathbf{b}_{\tau}$
9:      $\mathbf{r}_{(\tau+1)} = \mathbf{r}_{(\tau)} - c_{(\tau)}(\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))\mathbf{b}_{(\tau)}$
10:     $d_{old} = d_{new}, d_{new} = \mathbf{r}_{(\tau+1)}^{\top} \mathbf{r}_{(\tau+1)}$
11:     $\mathbf{b}_{(\tau+1)} = \mathbf{r}_{(\tau+1)} + \frac{d_{new}}{d_{old}} \mathbf{b}_{(\tau)}$
12:     $\tau = \tau + 1$
13:    **Output:** $\hat{\mathbf{x}}(\boldsymbol{\omega}) = \hat{\mathbf{x}}_{(\tau+1)}$

---

graph [cf. (7)]. Because of this sparsity, the graph signal shifting operation $\mathbf{x}^{(1)} = \mathbf{S}(\boldsymbol{\omega})\mathbf{x} = (\boldsymbol{\omega}\boldsymbol{\omega}^{\top} \odot \mathbf{L})\mathbf{x}$ has a cost of order linear in the number of edges M. Moreover, this operation is local over the graph and the i-th signal value is given by

$$x_i^{(1)} = \omega_i \sum_{j \in \mathcal{N}_i} A_{ij}(\omega_i x_i - \omega_j x_j). \quad (17)$$

By exploring (17), we detail next how the NA filter can be implemented using the conjugate gradient method [43] and distributed graph filters [35], [44].

Centralized. To implement (10) efficiently, we first rephrase it as a linear system

$$(\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))\hat{\mathbf{x}}(\boldsymbol{\omega}) = \mathbf{y} \quad (18)$$

and then employ conjugate gradient [43] to obtain $\hat{\mathbf{x}}(\boldsymbol{\omega})$. For completeness, Algorithm 1 summarizes the required steps. Exploiting the local operation (17) in Steps 7 and 9 of Algorithm 1, which are the main sources of computing exhausts, and running the conjugate gradient method for T iterations, we have a cost of order $\mathcal{O}(TM)$.

Distributed. To implement (10) distributively with graph filters, we start with a random initialization $\hat{\mathbf{x}}_0$ for estimate $\hat{\mathbf{x}}(\boldsymbol{\omega})$. At iteration τ, the distributed estimate follows the recursion

$$\hat{\mathbf{x}}_\tau(\boldsymbol{\omega}) = -\mathbf{S}(\boldsymbol{\omega})\hat{\mathbf{x}}_{\tau-1}(\boldsymbol{\omega}) + \mathbf{y}. \quad (19)$$

From (17), we can see the term $\mathbf{S}(\boldsymbol{\omega})\hat{\mathbf{x}}_{\tau-1}(\boldsymbol{\omega})$ implies nodes communicate with their neighbors and exchange information about the previous estimate $\hat{\mathbf{x}}_{\tau-1}(\boldsymbol{\omega})$, which has a cost of order $\mathcal{O}(M)$.

When $\boldsymbol{\omega}$ satisfies the spectral norm inequality $\|\mathbf{S}(\boldsymbol{\omega})\| < 1$, recursion (19) leads to the steady-state ($\tau \rightarrow \infty$) estimate

$$\hat{\mathbf{x}}(\boldsymbol{\omega}) \triangleq \lim_{\tau \to \infty} \hat{\mathbf{x}}_\tau(\boldsymbol{\omega}) = \sum_{\tau=0}^{\infty} (-\mathbf{S}(\boldsymbol{\omega}))^\tau \mathbf{y} = (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))^{-1}\mathbf{y},$$

$$(20)$$

which matches the optimal solution (10). Arresting, therefore, (19) in T iterations leads again to a distributed communication and computational cost of order $\mathcal{O}(TM)$.

The sparsity of parametric shift operator $\mathbf{S}(\boldsymbol{\omega})$ leads to a reduced implementation complexity for both centralized and distributed ways, which is critical in practice. In the following, we investigate how to optimally design the NA weights so to obtain the optimal performance.

IV. WEIGHT DESIGN

In this section, we focus on designing the NA weights $\boldsymbol{\omega}$ in a minimum MSE (MMSE) sense. The estimation error between the optimal estimate $\hat{\mathbf{x}}(\boldsymbol{\omega})$ [cf. (10)] and the true value $\mathbf{x}^*$ is

$$\mathbf{e} \triangleq \hat{\mathbf{x}}(\boldsymbol{\omega}) - \mathbf{x}^*. \tag{21}$$

We can then formulate the optimal design of $\boldsymbol{\omega}$ as solving the optimization problem

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^N} \mathbb{E}\left\{ \|\hat{\mathbf{x}}(\boldsymbol{\omega}) - \mathbf{x}^*\|_2^2 \right\}, \tag{22}$$

where $\hat{\mathbf{x}}$ is defined in (10). The inverse relation in $\hat{\mathbf{x}}(\boldsymbol{\omega})$ renders problem (22) challenging to solve in its original form. To overcome this challenge, we first propose two relaxation methods to solve (22) for $\boldsymbol{\omega}$ leveraging strong (approximate) prior knowledge of the graph signal. Then, we use the *minmax* strategy to adapt these two methods to scenarios where only signal bounds are available.

A. PRONY'S METHOD

Given the estimate $\hat{\mathbf{x}}(\boldsymbol{\omega}) = (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))^{-1}\mathbf{y}$ and the error $\mathbf{e}$ in (21), a typical approach to design parameters in inverse relationships is to consider Prony's modified error [45]

$$\mathbf{e}' \triangleq \mathbf{y} - (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))\mathbf{x}^*, \tag{23}$$

which is obtained by multiplying both sides of (21) by $(\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))$. Albeit not equivalent to the true error $\mathbf{e}$, minimizing the modified error $\mathbf{e}'$ is easier due to the linear relationship in $\boldsymbol{\omega}$ and the resulting performance is often satisfactory [45]. The NA weights that minimize the error $\mathbf{e}'$ in (23) can be obtained by solving the following problem

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^N} \mathbb{E}\left\{ \|\mathbf{y} - (\mathbf{I} + \mathbf{S}(\boldsymbol{\omega}))\mathbf{x}^*\|_2^2 \right\}$$

$$\text{s.t.} \quad \omega_0^* \le \omega_i^2, \text{ for } i = 1, \dots, N \tag{24}$$

where the constraint imposes all entries of $\boldsymbol{\omega}$ to satisfy Lemma 1. It would be more natural to also add the condition on a smaller MSE [cf. (15b) and (16)] as a constraint. However, this condition depends on the SNR, i.e., when the SNR is low, it is easier for them to be satisfied; it leads to a smaller set of feasible solutions. Thus, the complication of this condition does not encourage us to add it in the optimization problem. Instead, we observe that Lemma 1 already improves the optimal weight design in practice.

The quadratic relation in the optimization variable $\boldsymbol{\omega}$ in $\mathbf{S}(\boldsymbol{\omega})$ [cf. (7)] makes problem (24) non-convex. To obtain $\boldsymbol{\omega}$, we follow a two step approach. First we define a positive

semi-definite rank-one matrix $\boldsymbol{\Omega} \triangleq \boldsymbol{\omega}\boldsymbol{\omega}^\top$ and solve (24) w.r.t the new variable $\boldsymbol{\Omega}$. Then, we find a vector estimate $\hat{\boldsymbol{\omega}}$ by performing a rank-one approximation of the obtained matrix [46].

Rewriting (24) w.r.t. the new variable $\boldsymbol{\Omega}$, we obtain

$$\min_{\boldsymbol{\Omega} \in \mathcal{S}_+^{N \times N}} \text{tr}\left\{ (\boldsymbol{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*\top} \right\}$$

$$\text{s.t.} \quad \omega_0^* \le \Omega_{ii}, \text{ for } i = 1, \dots, N \tag{25}$$

where the derivation of the cost function is reported in Appendix E. Note that in (25), we followed [46] and dropped the non-convex constraint $\text{rank}(\boldsymbol{\Omega}) = 1$. Then, problem (25) becomes a convex semi-definite program (SDP), solvable with off-the-shelf tools [47], [48], and for instance with interior-point method [49], [50], it has a cost[3] of $\mathcal{O}(N^3)$ per iteration. We also observe that the returned solution from (25) is consistently a rank-one matrix. Given then $\boldsymbol{\Omega}^*$ from (25), the node-adaptive weight vector $\boldsymbol{\omega}^*$ is equal to the eigenvector of $\boldsymbol{\Omega}^*$ with the largest eigenvalue, multiplied with the square root of the eigenvalue. However, more sophisticated rank-one approximations are also possible [46].

B. SEMI-DEFINITE RELAXATION

Despite its simplicity, Prony's method does not directly relate to the true error $\mathbf{e}$ in (21). Working with the modified error might be viable when the signal-to-noise ratio (SNR) is high but it might lead to a degraded performance when the SNR is low. To overcome the latter, we propose here an optimization problem relying on semi-definite relaxation when minimizing the true error in (22). This approach follows again a two-step procedure: first, we formulate (22) w.r.t. the matrix variable $\mathbf{H}(\boldsymbol{\Omega}) = (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^{-1}$ where $\boldsymbol{\Omega} := \boldsymbol{\omega}\boldsymbol{\omega}^\top$, and then we obtain $\boldsymbol{\Omega}$ from $\mathbf{H}(\boldsymbol{\Omega})$ by means of inversion. Finally, $\boldsymbol{\omega}^*$ is extracted from $\boldsymbol{\Omega}^*$ by rank-one approximation [46].

To start, let us recall the node-adaptive estimate $\hat{\mathbf{x}}(\boldsymbol{\omega}) = (\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L})^{-1}\mathbf{y}$ and rewrite (22) w.r.t. $\mathbf{H}(\boldsymbol{\Omega})$ as

$$\min_{\boldsymbol{\Omega}, \mathbf{H}(\boldsymbol{\Omega}) \in \mathcal{S}_+^{N \times N}} \mathbb{E}\left\{ \|\mathbf{H}(\boldsymbol{\Omega})\mathbf{y} - \mathbf{x}^*\|_2^2 \right\}$$

$$\text{s.t.} \quad \mathbf{H}(\boldsymbol{\Omega}) = (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^{-1}$$

$$\text{rank}(\boldsymbol{\Omega}) = 1,$$

$$\omega_0^* \le \Omega_{ii}, \text{ for } i = 1, \dots, N. \tag{26}$$

where the last constraint is again Lemma 1. The cost function in (26) can be expanded further as (cf. Appendix. F)

$$\mathbb{E}\left\{ \|\mathbf{H}(\boldsymbol{\Omega})\mathbf{y} - \mathbf{x}^*\|_2^2 \right\} = \text{tr}\{(\mathbf{H}^2(\boldsymbol{\Omega})) - 2\mathbf{H}(\boldsymbol{\Omega}) + \mathbf{I})\mathbf{x}^*\mathbf{x}^{*\top} + \mathbf{H}^2(\boldsymbol{\Omega})\boldsymbol{\Sigma}\} \tag{27}$$

which depends on both the signal and noise covariance. Problem (26) presents two non-convex constraints: the inverse relationship $\mathbf{H}(\boldsymbol{\Omega}) = (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^{-1}$ and the rank-one constraint $\text{rank}(\boldsymbol{\Omega}) = 1$. We address the former, by leveraging

---

[3]We also refer to the recent SDP work [51] to provide a complete complexity analysis.

semi-definite relaxation and by writing the constraint in its positive semi-definite convex form [46]

$$\mathbf{H}(\mathbf{\Omega}) - (\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L})^{-1} \succeq \mathbf{0}. \qquad (28)$$

Since $\mathbf{L}$, $\mathbf{\Omega}$, and $\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L}$ are positive semi-definite matrices, we can use the Schur complement (see Appendix A) to reformulate (28) into a convex linear matrix inequality

$$\begin{pmatrix} \mathbf{I} + \mathbf{\Omega} \odot \mathbf{L} & \mathbf{I} \\ \mathbf{I} & \mathbf{H}(\mathbf{\Omega}) \end{pmatrix} \succeq \mathbf{0}.$$

Regarding the non-convexity of the rank-one constraint, we can relax it as in (25). With these relaxation techniques, we rephrase problem (26) into the convex form

$$\min_{\mathbf{\Omega}, \mathbf{H}(\mathbf{\Omega}) \in \mathcal{S}_+^{N \times N}} \mathbb{E}\left\{\|\mathbf{H}(\mathbf{\Omega})\mathbf{y} - \mathbf{x}^*\|_2^2\right\}$$

$$\text{s.t.} \quad \begin{pmatrix} \mathbf{I} + \mathbf{\Omega} \odot \mathbf{L} & \mathbf{I} \\ \mathbf{I} & \mathbf{H}(\mathbf{\Omega}) \end{pmatrix} \succeq \mathbf{0},$$

$$\omega_0^* \leq \Omega_{ii}, \text{ for } i = 1, \ldots, N. \qquad (29)$$

This is also an SDP problem but with more constraints compared to (25). After obtaining $\mathbf{H}^*(\mathbf{\Omega})$ by solving (29), we can find $\mathbf{\Omega}$ from the inverse relation below

$$\mathbf{H}^*(\mathbf{\Omega})(\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L}) = (\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L})\mathbf{H}^*(\mathbf{\Omega}) = \mathbf{I}$$

by solving the following least squares problem

$$\min_{\mathbf{\Omega} \in \mathcal{S}_+^{N \times N}} \|\mathbf{H}^*(\mathbf{\Omega})(\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L}) - \mathbf{I}\|_2^2$$

$$+ \|(\mathbf{I} + \mathbf{\Omega} \odot \mathbf{L})\mathbf{H}^*(\mathbf{\Omega}) - \mathbf{I}\|_2^2. \qquad (30)$$

Given then matrix $\mathbf{\Omega}^*$ from (30), we extract $\boldsymbol{\omega}^*$ by a rank-one approximation [46]. With the obtained NA weights $\boldsymbol{\omega}^*$, we subsequently build the filter $\mathbf{H}(\boldsymbol{\omega}^*) = (\mathbf{I} + \boldsymbol{\omega}^*\boldsymbol{\omega}^{*\top} \odot \mathbf{L})^{-1}$ and obtain the estimate $\hat{\mathbf{x}}(\boldsymbol{\omega}^*)$ as in (10).

The main advantage of the semidefinite relaxation (SDR) approach (29)–((30)) is that it focuses directly on the true error (21) rather than the modified error (23). However, due to the complexity issue, semidefinite relaxation methods are applicable to medium-sized graphs with up to a thousand nodes.

C. MIN-MAX ADAPTATION

As it follows from (24) and (26), both Prony's method and the SDR method require knowledge of signal $\mathbf{x}^*$ to design weights $\boldsymbol{\omega}$. This is possible in a data-driven fashion under the condition that the test and training data have a similar distribution. In this section, we depart from this assumption and propose a design method that is independent of $\mathbf{x}^*$ but only requires side information such as signal evolution bounds. The latter is simpler to acquire from a small set of data or by physical considerations.

Consider signal $\mathbf{x}^*$ has an evolution bounded in the interval $[\mathbf{x}_l, \mathbf{x}_u]$. We can then design the parameter vector $\boldsymbol{\omega}$ for estimator $\hat{\mathbf{x}}(\boldsymbol{\omega})$ [cf. (10)] as the one that minimizes the MSE of the worst-case scenario, i.e.,

$$\min_{\boldsymbol{\omega}} \max_{\mathbf{x}^*} \mathbb{E}\left\{\|\hat{\mathbf{x}}(\boldsymbol{\omega}) - \mathbf{x}^*\|_2^2\right\}$$

$$\text{s.t.} \quad \mathbf{x}_l \leq \mathbf{x}^* \leq \mathbf{x}_u,$$

$$\omega_0^* \leq \omega_i^2, \text{ for } i = 1, \ldots, N. \qquad (31)$$

Problems of the form (31) are known as *min-max* problems. The inner maximization seeks for the signal $\mathbf{x}^*$ that leads to the worst MSE performance, while the outer minimization finds the parameter $\boldsymbol{\omega}$ that minimizes the worst MSE among all possible choices. Different from problems (24) and (26), signal $\mathbf{x}^*$ is now an optimization variable in (31) and only $\mathbf{x}_l$ and $\mathbf{x}_u$ are needed. There are efficient methods to solve *min-max* problems of the form (31) such as iterative first-order methods [52] or gradient descent-ascent [53]. In the sequel, we detail how (31) specializes to Prony's method and SDR method.

Following the same rationale as in (25), we can write the *min-max* Prony's method as

$$\min_{\mathbf{\Omega} \in \mathcal{S}_+^{N \times N}} \max_{\mathbf{x}^*} \text{tr}\{(\mathbf{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*\top}\}$$

$$\text{s.t.} \quad \mathbf{x}_l \leq \mathbf{x}^* \leq \mathbf{x}_u,$$

$$\omega_0^* \leq \Omega_{ii}, \text{ for } i = 1, \ldots, N. \qquad (32)$$

Since $\mathbf{\Omega} \odot \mathbf{L}$ is positive semidefinite, the cost function is quadratic (convex) over the inner optimization variable $\mathbf{x}^*$. Further, it can be shown that the maximizer of the inner problem is at the boundaries, i.e., either $\mathbf{x}^* = \mathbf{x}_l$ or $\mathbf{x}^* = \mathbf{x}_u$. To solve the outer minimization problem w.r.t. $\mathbf{\Omega}$, we proceed similarly as in (25). For the SDR problem (29) we can also solve the min-max version since the cost function for the latter is also convex in $\mathbf{x}^*$. Here again, we optimize the true error [cf. (21)] at the price of a higher computational complexity.

V. NUMERICAL RESULTS

In this section, we compare the performance of the proposed design schemes with state-of-the-art alternatives and illustrate the different trade-offs inherent to NA regularization on synthetic and real-world data from the Molene[4] and the NOAA[5] data sets. We measure the recovery accuracy between the estimate $\hat{\mathbf{x}}$ and the true signal $\mathbf{x}^*$ through the normalized mean squared error, NMSE $= \|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2/\|\mathbf{x}^*\|_2^2$. In these simulations, we used the GSP [54] and CVX [47] toolboxes.

A. SYNTHETIC DATA

In the first set of experiments, we consider synthetic Erdos-Renyì graphs of $N = 50$ nodes and link formation probability 0.5. We generate a synthetic graph signal $\mathbf{x}^*$, whose graph Fourier transform is one in the first 20 coefficients and zero

---

[4]Raw data available at [Online]. Available: https://donneespubliques. meteofrance.fr/donnees libres/Hackathon/RADOMEH.tar.gz

[5]Raw data available at [Online]. Available: https://www.ncdc.noaa. gov/data-access/land-based-station-data/land-based-datasets/climate-normals/1981-2010-normals-data
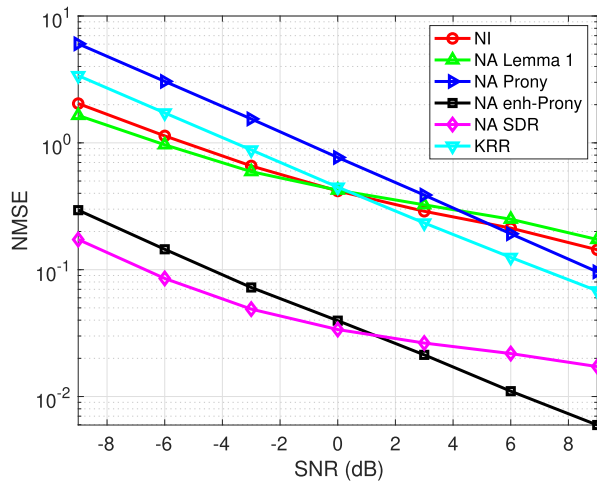
FIGURE 2. NMSE of different methods as a function of the SNR over an Erdos-Renyì graph. The true graph signal is bandlimited to the first 20 graph frequencies.

FIGURE 3. Interpolation performance of the different methods as a function of the number of observed nodes with SNR = 0 dB over an Erdos-Renyì graph. The true graph signal is bandlimited in the first 20 graph frequencies.

elsewhere. This signal is called a bandlimited graph signal and varies smoothly over the graph [5]; hence, it fits the assumption for Tikhonov regularization (2). We corrupted the signal with a zero-mean Gaussian noise with variance σ_n^2, so to obtain a signal-to-noise ratio SNR $= \|\mathbf{x}^*\|_2^2/(N\sigma_n^2)$. We average the performance over 50 graphs and 100 noise realizations leading to a total of 5000 Monte-Carlo runs. This scenario is also the one used in Fig. 1, where we showed the NA regularizer can achieve both a lower variance and MSE.

We first evaluate Prony's method when the true signal is known and the SDR method when both the true signal and noise variance are known. Our rationale is to avoid biases induced by a training set or by focusing solely on the worst-case scenario. We address the latter in the subsequent section with real data. The specific approaches we consider are:

1) The benchmark node-invariant regularizer with optimal weight $\omega_0^* = \mathcal{O}(\sqrt{\frac{\theta}{\lambda_2 \lambda_N}})$, where $\theta = \sqrt{\frac{1}{SNR}}$, and λ_2, λ_N are the smallest and the largest non-zero eigenvalues of the graph Laplacian $\mathbf{L}$, respectively [17].

2) The naive node-adaptive regularization where the NA weights $\boldsymbol{\omega}$ are chosen randomly to satisfy Lemma 1; i.e., $\omega_i = \sqrt{\omega_0^* + \omega_0^* \cdot c_i}$ for $i = 1, \ldots, N$, where c_i is uniformly distributed in $[0,1]$.

3) Prony's node-adaptive design, where we do not enforce the constraint of Lemma 1 in problem (25).

4) Prony's method [cf. (25)].

5) The SDR node-adaptive design [cf. (29), (30)].

6) The diffusion kernel ridge regression (KRR) with parameters $\sigma_{KRR}^2 = 1$ and $\mu_{KRR} = 10^{-4}$, chosen to achieve the best performance [19].

Fig. 2 shows the NMSE recovery performance of the considered methods with respect to the SNR. First, we observe the proposed Prony's method and the SDR reduce the NMSE by one order of magnitude compared with the optimal NI approach and the KRR. As we anticipated in Fig. 1, even the
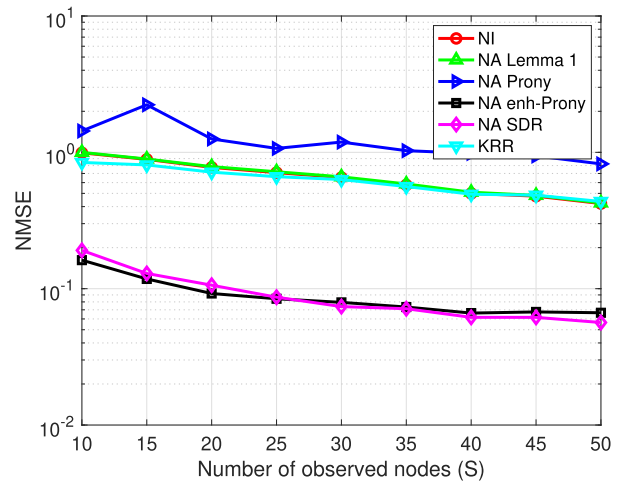
naive random NA regularizer achieves a comparable performance with these competitive alternatives, ultimately, highlighting the potential of the NA regularizer for graph signal recovery. We also remark the importance of the theoretical result in Lemma 1 in the Prony's problem (25), which reduces the MSE by one order of magnitude. We attribute the latter to the fact that Prony's approaches focus on the modified error rather than the true one. This is also evidenced by the comparison with the SDR technique, where Prony's method has a worse NMSE for lower SNRs; i.e., where considering the true error is more effective to deal with the large noise.

Next, we evaluate the NMSE performance of six different methods for interpolating missing values. We consider noisy observations from the random subset $\mathcal{M} \in \{10, 15, 20, \ldots, 50\}$ with an SNR of 0 dB to show the robustness of NA regularization. From Fig. 3, we observe again the superior performance of Prony's method and the SDR method. On the contrary, the naive weight setting and Prony's unconstrained method offer a similar performance as the benchmark NI regularization. The latter highlights the improvement brought by Lemma 1.

In the sequel, we will conduct experiments over real-world measurements. We omit the results of the NA design based on the ground truth signal since they behave the same as for the synthetic data. Instead, we consider the earlier mentioned data-driven and *min-max* scenarios.

B. MOLENE DATA SET

The Molene data set comprises $T = 744$ hourly temperature measurements collected in January 2014 from $N = 32$ weather stations in the region of Brest, France. We treat each weather station as a node of a graph and build a geometric distance graph in which each node is connected to its five nearest neighbours. The weight of edge (i, j) is $A_{ij} =$

IEEE
Signal
Processing
Society
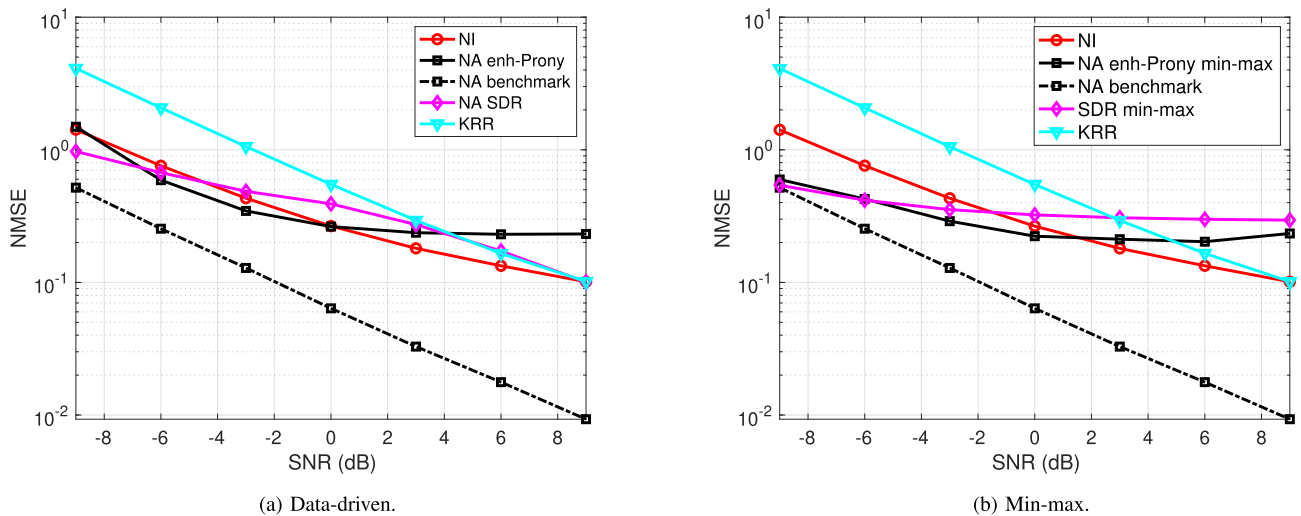
IEEE Open Journal of
Signal Processing

FIGURE 4. NMSE denoising performance of different methods w.r.t SNRs in the Molene data set. (a) Data-driven scenario. Half of the data are used to estimate the node-adaptive parameters. (b) Min-max scenario based on signal evolution bounds.

$\exp\{-5\,d^2(i, j)\}$, where $d(i, j)$ is the Euclidean distance between stations i and j. After removing the mean temperature across space and time, we can view every temporal snapshot as a graph signal. Among the six approaches listed in the former section, we omit the naive Lemma 1 based approach as well as Prony's unconstrained method to avoid overcrowded plots since their performance trend is similar to that observed with synthetic data. For the KRR, we tune $\sigma_{\mathrm{KRR}}^2 = 5$ to reach its best performance. The performance measure is still the NMSE averaged over all the 744 graph signals. The collected measurements are assumed to be the true signal and we artificially add noise as before. We consider 50 noise realizations per signal.

To show how different methods behave in the ideal and in a more practical setting, we considered two scenarios. First, we considered a data-driven scenario, where half of the temporal data are used to learn the NA weights. Specifically, we use these data to compute an average of $\mathbf{x}^*\mathbf{x}^{*\top}$ to be used in Prony's design [cf. (25)] and the SDR problem [cf.(29)]. The remaining half graph signals are used for testing. As the objective function of (29) depends on the noise covariance, for a fixed SNR, the noise level for each signal is different. Thus, we need to design NA weights based on (29) for each training signal. To avoid doing so, we use one instance from the whole recordings to compute the noise covariance matrix needed in the SDR method, which degrades its performance. Second, the min-max method is tested where only the signal lower and upper bounds are known. The bounds are obtained as the lowest and highest temperature records in the dataset. For both scenarios, we consider the NMSE denoising performance as a function of the SNR.

From Fig. 4, we observe that for more practical scenarios, the performance of the NA approaches degrades by approximately one order of magnitude, where the benchmark is done based on the oracle design (25) with true signal available.

However, they still outperform the KRR counterpart. Specifically, from the results in Fig. 4(a), we observe that Prony's method degrades substantially and achieves an NMSE similar to the NI regularizer. Though in the large SNR regime, it gets worse than the NI regularizer, this is because when noise is negligible, the training strategy will result in a suboptimal approximation of $\mathbf{x}^*\mathbf{x}^{*\top}$, and thus, suboptimal NA weights in problem (25).

Fig. 4(b) shows the NA algorithms still perform better than competing alternatives in the low SNR regime although the only information is the signal variation range. We attribute the saturation in the high SNR regime to the lack of information needed for designing the NA weights; i.e., the NA regularizer will impose a stronger bias on the solution that is not needed to denoise the signal in the high SNR regime.

We then evaluate the interpolation performance of different methods. We collect noisy observations at nodes $\mathcal{M} \in \{2, 4, 6, \ldots, 28, 30\}$ and SNR = 0 dB. The results are shown in Fig. 5. For the data-driven case in Fig. 5(a), Prony's method degrades significantly with a performance worse than the NI regularizer. This could be due to over-fitting the training data. On the other hand, the SDR is not obtaining satisfactory results unless the observations are collected from all of the nodes. This is because the obtained estimate of $\mathbf{x}^*\mathbf{x}^{*\top}$ does not apply on the SDR method for the interpolation case. In the *min-max* scenario in Fig. 5(b), we only make use of the signal bounds and obtain a consistently better performance by Prony's method, while the SDR method behaves similar to the NI regularizer.

The performance of the NA regularizer on the Molene data set is in general not surprisingly good. This is because the temperature data is collected geographically in a small region, which inherently leads to a set of smooth signal measurements over the graph. Next, we will test the performance of the NA regularizer over a different data set which is collected from a
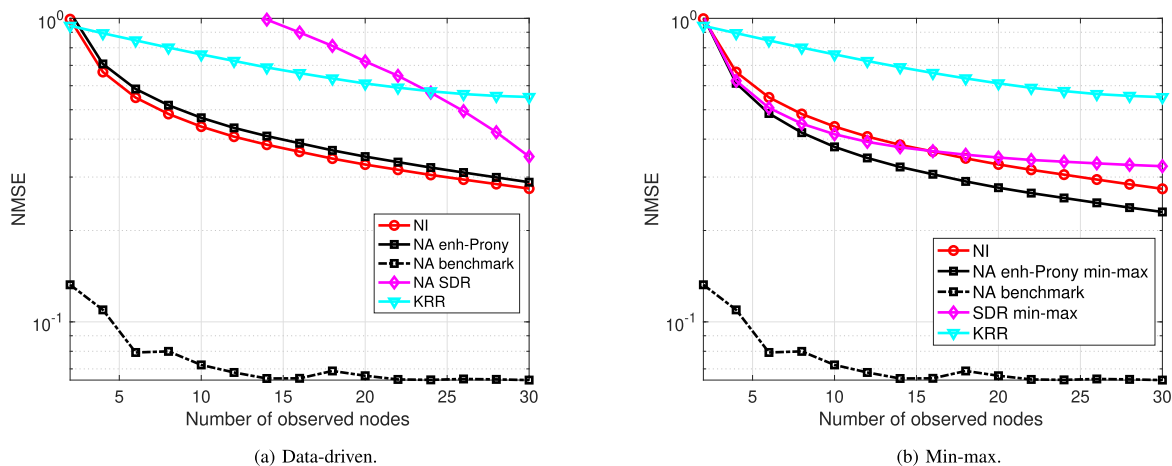
FIGURE 5. NMSE interpolation performance of different methods w.r.t the number of sampled nodes on Molene data set. The signal-to-noise ratio is SNR = 0 dB. (a) Data-driven scenario. Half of the data are used to estimate the signal covariance matrix. (b) Min-max scenario based on signal evolution bounds.
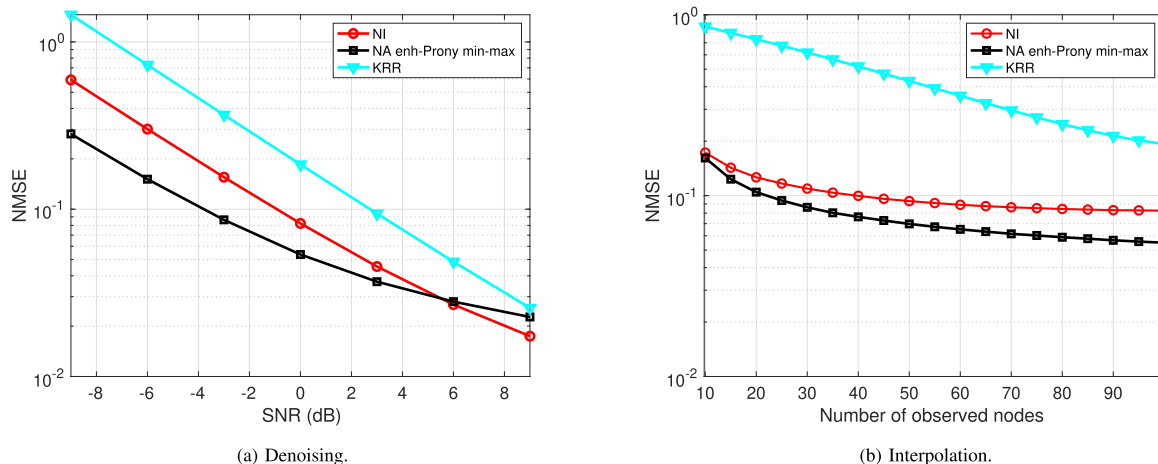

FIGURE 6. (a) Denoising performance of the enhanced Prony's method w.r.t. SNRs with different methods in *min-max* scenario. (b) Interpolation performance w.r.t. SNRs the number of observed nodes, SNR = 0 dB.

very large geographical region. The global smoothness then is not intrinsically guaranteed, therefore a superior performance of the NA regularizer is expected.

C. NOAA DATA SET

The NOAA data set comprises $T = 8759$ hourly temperature measurements collected across the continental U.S. from $N = 109$ weather stations in 2010. It is collected from a much larger geographical region compared to the Molene weather data. Following [55], we treat each station as a node of a seven nearest neighbor graph based on geographical distances. We measure again the NMSE performance averaged over all the 8759 signals and 50 noise realizations per signal. The parameters of all methods are the same as in the former section.

Fig. 6 shows the results for the min-max method applied to denoising and interpolation. From Fig. 6(a), we observe the improved performance of the proposed NA regularizer over other alternatives. In specific, we see a 3 dB SNR improvement for a fixed NMSE in Prony's method, which is more accentuated at low SNRs. This shows the benefits of the NA regularizer in situations with a high noise level compared with the NI regularizer.

In the interpolation setting, the observed nodes are $\mathcal{M} \in \{10, 15, 20, \ldots, 90, 95\}$. From Fig. 6(b), we observe that the NA regularizer achieves consistently a smaller NMSEs compared with the NI regularizer and KRR method. This improved performance gets more noticeable when the number of observed nodes is larger since the node-adaptive regularizer can better exploit the local signal behavior to find the missing values. In turn, this indicates that when the true signal is not available but only the signal variation bounds are known, i.e., the maximal and minimal signal observations, the NA regularizer can still perform well based on the *min-max* strategy.

In practical situations, with proper prior information on the graph signals, we recommend to design the NA regularizer using the *min-max* strategy, since it does not require the true signal or training data, but only the signal measurement bounds. From the above results, we see that in harsher situations with a high level of noise present, based on the *min-max* strategy, the NA regularizer will behave better than the NI and KRR counterparts in graph signal reconstruction. But when the SNR is high or the noise level is low, we can see that the NI approach is reasonably good. In addition, when the graph signal is globally smooth, such as the Molene weather data, the NA regularizer would behave similar to the NI one. However, if the signal does not have global smoothness, e.g., the NOAA data, the reconstruction ability of the NA regularizer is shown to be much stronger than the NI one.

VI. CONCLUSION

This paper proposed a node-adaptive regularizer for graph signal reconstruction by enhancing the degrees of freedom of the regularization and generalizing the typical global signal smoothness prior. We considered the popular Tikhonov regularizer (also known as graph Laplacian regularizer), and proposed a node-adaptive extension of it. Instead of adopting a shared weight for all nodes, we assign each node a weight to deal with local signal smoothness and possible node heterogeneity. This strategy allows the signal priors to be adapted locally by tuning the regularization parameters, instead of assuming a global smoothness prior. In addition, we evaluated the related bias-variance trade-off, and showed its potential in achieving a lower variance and MSE. Furthermore, we formalized optimal design of the node weights based on Prony's method and semi-definite relaxation. We also proposed a *min-max* formulation to deal with situations when no training data is available but only signal side information such as evolution bounds. Finally, numerical results on both synthetic and different type real data were performed to corroborate our findings.

Meanwhile, in our parallel work [1], we proposed a first-order iterative method to optimally design the NA weights based on a *min-max* game where an energy bound is assumed on the signal measurements. In the future, we expect to further extend the node-adaptive regularization to other regularizers, and aim to reduce the complexity of optimal weight design.

APPENDIX A
IMPORTANT LEMMAS AND THEOREMS

Lemma 2: Schur complement lemma: Given any symmetric matrix, $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{pmatrix}$, the following conditions are equivalent:

- $\mathbf{M} \succeq \mathbf{0}$ ($\mathbf{M}$ is positive semi-definite);
- $\mathbf{A} \succeq \mathbf{0}$, $(\mathbf{I} - \mathbf{A}\mathbf{A}^\dagger)\mathbf{B} = \mathbf{0}$, $\mathbf{C} - \mathbf{B}^\top\mathbf{A}^\dagger\mathbf{B} \succeq \mathbf{0}$
- $\mathbf{C} \succeq \mathbf{0}$, $(\mathbf{I} - \mathbf{C}\mathbf{C}^\dagger)\mathbf{B} = \mathbf{0}$, $\mathbf{A} - \mathbf{B}^\top\mathbf{C}^\dagger\mathbf{B} \succeq \mathbf{0}$

Lemma 3: Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be $n \times n$ symmetric matrices, then

$$\mathrm{tr}\left((\mathbf{A}^2 - \mathbf{B}^2)\mathbf{C}\right)$$

$$= \mathrm{tr}\left((\mathbf{A} - \mathbf{B})(\mathbf{A} + \mathbf{B})\mathbf{C}\right) - \mathrm{tr}\left((\mathbf{AB} - \mathbf{BA})\mathbf{C}\right)$$

$$= \mathrm{tr}\left((\mathbf{A} - \mathbf{B})(\mathbf{A} + \mathbf{B})\mathbf{C}\right) \tag{33}$$

because $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}((\mathbf{ABC})^\top) = \mathrm{tr}(\mathbf{CBA}) = \mathrm{tr}(\mathbf{ACB})$.[6]

Lemma 4: Let $\mathbf{A}$ be an $n \times n$ positive semi-definite matrix, and $\mathbf{B}$ an $n \times n$ negative semi-definite matrix, then $\mathrm{tr}(\mathbf{AB}) \leq 0$.

Proof: Consider an eigenvalue-eigenvector equation of matrix $\mathbf{AB}$ as follows

$$\mathbf{ABx} = \lambda \mathbf{x}, \tag{34}$$

where eigenvalue λ is a scalar, and eigenvector $\mathbf{x}$ is a vector of length n. If we left multiply (34) by $\mathbf{x}^\top \mathbf{B}$ on both sides, then

$$\mathbf{x}^\top \mathbf{BABx} = \lambda \mathbf{x}^\top \mathbf{Bx}. \tag{35}$$

The eigenvalue of $\mathbf{AB}$ can be represented as

$$\lambda = \frac{\mathbf{x}^\top \mathbf{BABx}}{\mathbf{x}^\top \mathbf{Bx}}. \tag{36}$$

From [56, Thm. 7.2.7], matrix $\mathbf{BAB}$ is positive semi-definite independent of matrix $\mathbf{B}$. Thus, the numerator $\mathbf{x}^\top \mathbf{BABx}$ is nonnegative. Since matrix $\mathbf{B}$ is negative semi-definite, the denominator $\mathbf{x}^\top \mathbf{Bx}$ is nonpositive. This results into a nonpositive eigenvalue λ. Thus, the trace of matrix $\mathbf{AB}$ is nonpositive. ∎

Lemma 5: Let $n \times n$ matrices $\mathbf{A}, \mathbf{B}$ be positive semi-definite, then $\mathbf{A} \odot \mathbf{B}$ is positive semi-definite [56, Thm. 7.5.3] and

$$\lambda_{\max}(\mathbf{A} \odot \mathbf{B}) \leq \lambda_{\max}(\mathbf{A})\max\{b_{ii}\} \tag{37}$$

where b_{ii} is the i-th diagonal element of $\mathbf{B}$ [56, 7.5.P24]. Since the inequality is given in the exercise part of [56], so we give the proof below.

Proof: Let λ_n be the maximal eigenvalue of $\mathbf{A}$, then $\lambda_n\mathbf{I} - \mathbf{A} \succeq \mathbf{0}$. Thus, we have $(\lambda_n\mathbf{I} - \mathbf{A}) \odot \mathbf{B}$ is positive semi-definite. Let $\mathbf{x} \in \mathbb{R}^n$ be a nonzero vector, then

$$\mathbf{x}^\top \left((\mathbf{A} - \lambda_n\mathbf{I}) \odot \mathbf{B}\right) \mathbf{x}$$
$$= \mathbf{x}^\top (\mathbf{A} \odot \mathbf{B}) \mathbf{x} - \lambda_n\mathbf{x}^\top (\mathbf{I} \odot \mathbf{B}) \mathbf{x} \leq 0, \tag{38}$$

which leads to

$$\mathbf{x}^\top (\mathbf{A} \odot \mathbf{B}) \mathbf{x} \leq \lambda_n\mathbf{x}^\top (\mathbf{I} \odot \mathbf{B}) \mathbf{x}$$

$$= \lambda_n \sum_{i=1}^{n} b_{ii}|x_i|^2 \leq \lambda_n\max\{b_{ii}\} \|\mathbf{x}\|^2. \tag{39}$$

If $\mathbf{x}$ is the eigenvector, the claim holds competing the proof. ∎

Theorem 2: (Weyl's Inequality) [56, Thm. 4.3.1] Let $\mathbf{A}, \mathbf{B}$ be $n \times n$ Hermitian matrices and let the respective eigenvalues of $\mathbf{A}, \mathbf{B}$ and $\mathbf{A} + \mathbf{B}$ be $\{\lambda_i(\mathbf{A})\}_{i=1}^n$, $\{\lambda_i(\mathbf{B})\}_{i=1}^n$ and $\{\lambda_i(\mathbf{A} + \mathbf{B})\}_{i=1}^n$, each of which is algebraically ordered as $\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} \leq \lambda_n = \lambda_{\max}$. Then

$$\lambda_i(\mathbf{A} + \mathbf{B}) \leq \lambda_{i+j}(\mathbf{A}) + \lambda_{n-j}(\mathbf{B}), \ j = 0, 1, \ldots, n - i \tag{40}$$

for each $i = 1, \ldots, n$, with equality for some pair (i, j) if and only if there is a nonzero vector $\mathbf{x}$ such that $\mathbf{Ax} = \lambda_{i+j}(\mathbf{A})\mathbf{x}$,

---

[6]Due to the cyclic property of trace and $\mathrm{tr}(\mathbf{A}) = \mathrm{tr}(\mathbf{A}^\top)$

$\mathbf{B}\mathbf{x} = \lambda_{n-j}(\mathbf{B})\mathbf{x}$ and $(\mathbf{A} + \mathbf{B})\mathbf{x} = \lambda_i(\mathbf{A} + \mathbf{B})\mathbf{x}$. Also,

$$\lambda_{i-j+1}(\mathbf{A}) + \lambda_j(\mathbf{B}) \leq \lambda_i(\mathbf{A} + \mathbf{B}), \; j = 1, \ldots, i \quad (41)$$

for each $i = 1, \ldots, n$, with equality for some pair (i, j) if and only if there is a nonzero vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \lambda_{i-j+1}(\mathbf{A})\mathbf{x}$, $\mathbf{B}\mathbf{x} = \lambda_j(\mathbf{B})\mathbf{x}$ and $(\mathbf{A} + \mathbf{B})\mathbf{x} = \lambda_i(\mathbf{A} + \mathbf{B})\mathbf{x}$. If $\mathbf{A}$ and $\mathbf{B}$ have no common eigenvector, then every inequality above is a strict one.

APPENDIX B
PROOF OF LEMMA 1
To show $\text{var}(\boldsymbol{\omega}) \leq \text{var}(\omega_0)$, given $\omega_i^2 \geq \omega_0$, for $i = 1, 2, \ldots, n$, it suffices to show

$$\text{var}(\boldsymbol{\omega}) - \text{var}(\omega_0) = \text{tr}(\mathbf{H}^2(\boldsymbol{\omega})\boldsymbol{\Sigma}) - \text{tr}(\mathbf{H}^2(\omega_0)\boldsymbol{\Sigma})$$
$$= \text{tr}([(\mathbf{H}(\boldsymbol{\omega}) - \mathbf{H}(\omega_0))(\mathbf{H}(\boldsymbol{\omega}) + \mathbf{H}(\omega_0))]\boldsymbol{\Sigma}) \leq 0 \quad (42)$$

where the second equality comes from Lemma 3. Since matrices $\boldsymbol{\Sigma}$, $\mathbf{H}(\boldsymbol{\omega})$ and $\mathbf{H}(\omega_0)$ are positive semi-definite by definition, the sum of two positive semi-definite matrices is also positive semi-definite. Thus, from Lemma 4, it suffices to show $\mathbf{H}(\boldsymbol{\omega}) - \mathbf{H}(\omega_0) \preceq \mathbf{0}$ with the condition $\omega_i^2 > \omega_0$ for $i = 1, \ldots, N$.

With this condition, we have $\boldsymbol{\omega}\boldsymbol{\omega}^\top \succeq \omega_0 \mathbf{1}\mathbf{1}^\top$. Since $\text{diag}(\boldsymbol{\omega})\mathbf{L}\text{diag}(\boldsymbol{\omega}) = \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L}$ and $\omega_0 \mathbf{L} = \omega_0 \mathbf{1}\mathbf{1}^\top \odot \mathbf{L}$, we further have

$$(\mathbf{I} + \omega_0 \mathbf{1}\mathbf{1}^\top \odot \mathbf{L}) - (\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L})$$
$$= (\omega_0 \mathbf{1}\mathbf{1}^\top - \boldsymbol{\omega}\boldsymbol{\omega}^\top) \odot \mathbf{L} \preceq \mathbf{0} \quad (43)$$

where the equality holds because the Hadamard product is distributive over addition. Then, we left multiply both sides by $(\mathbf{I} + \omega_0 \mathbf{1}\mathbf{1}^\top \odot \mathbf{L})^{-1}$ and right multiply both sides by $(\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L})^{-1}$. This does not change the sign because they are both PSD. Hence, we have

$$(\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L})^{-1} - (\mathbf{I} + \omega_0 \mathbf{1}\mathbf{1}^\top \odot \mathbf{L})^{-1} \preceq \mathbf{0}$$
$$\mathbf{H}(\boldsymbol{\omega}) - \mathbf{H}(\omega_0) \preceq \mathbf{0} \quad (44)$$

which completes the proof.

APPENDIX C
PROOF OF THEOREM 1
Proving $\text{mse}(\boldsymbol{\omega}) \leq \text{mse}(\omega_0)$ with the given conditions, is equivalent to showing $\Delta = \text{mse}(\boldsymbol{\omega}) - \text{mse}(\omega_0) \leq 0$. We expand the latter as

$$\Delta = \text{tr}((\mathbf{I} - \mathbf{H}(\boldsymbol{\omega}))^2 \mathbf{x}^* \mathbf{x}^{*T}) + \text{tr}(\mathbf{H}^2(\boldsymbol{\omega})\boldsymbol{\Sigma})$$
$$- \text{tr}((\mathbf{I} - \mathbf{H}(\omega_0))^2 \mathbf{x}^* \mathbf{x}^{*T}) + \text{tr}(\mathbf{H}^2(\omega_0)\boldsymbol{\Sigma}). \quad (45)$$

Since $\mathbf{x}^* \mathbf{x}^{*T} = \mathbf{P}\boldsymbol{\Sigma}$ holds by definition, by working out the above equation, we have

$$\Delta = \text{tr}(\boldsymbol{\Sigma}(\mathbf{H}(\omega_0) - \mathbf{H}(\boldsymbol{\omega}))$$
$$\cdot [2\mathbf{P} - (\mathbf{H}(\boldsymbol{\omega}) + \mathbf{H}(\omega_0))(\mathbf{I} + \mathbf{P})]). \quad (46)$$

Due to the covariance matrix $\boldsymbol{\Sigma} \succeq \mathbf{0}$ and from the first condition (15a) we have also that the filter difference is positive semi-definite $\mathbf{H}(\omega_0) - \mathbf{H}(\boldsymbol{\omega}) \succeq \mathbf{0}$. Thus, from Lemma 4, it suffices to show that

$$2\mathbf{P} - (\mathbf{H}(\boldsymbol{\omega}) + \mathbf{H}(\omega_0))(\mathbf{I} + \mathbf{P}) \preceq \mathbf{0}. \quad (47)$$

Further, since $\mathbf{I} + \mathbf{P} \succ \mathbf{0}$ and it is invertible, we can focus on proving the smallest eigenvalue is greater than or equal to zero, i.e.,

$$\lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega}) + \mathbf{H}(\omega_0) - 2\mathbf{P}(\mathbf{I} + \mathbf{P})^{-1}\} \geq 0. \quad (48)$$

Let us then define the matrix $\boldsymbol{\Gamma} \triangleq \mathbf{P}(\mathbf{I} + \mathbf{P})^{-1}$ and scalar $\gamma \in (0, 1)$ as its only nonzero eigenvalue to simplify notations. From Theorem 2, we have

$$\lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega}) + \mathbf{H}(\omega_0) - 2\boldsymbol{\Gamma}\} \geq \lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega})\}$$
$$+ \lambda_{\min}\{\mathbf{H}(\omega_0)\} + \lambda_{\min}\{-2\boldsymbol{\Gamma}\}. \quad (49)$$

So, a sufficient condition for $\text{mse}(\boldsymbol{\omega}) \leq \text{mse}(\omega_0)$ is

$$\lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega})\} + \lambda_{\min}\{\mathbf{H}(\omega_0)\} - \lambda_{\max}\{2\boldsymbol{\Gamma}\} \geq 0 \quad (50)$$

where $\lambda_{\max}\{2\boldsymbol{\Gamma}\} = 2\gamma \in (0, 2)$, and $\lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega})\}$, $\lambda_{\min}\{\mathbf{H}(\omega_0)\}$ can be found as follows. From the eigendecomposition

$$\mathbf{H}(\omega_0) \triangleq (\mathbf{I} + \omega_0 \mathbf{L})^{-1} = \sum_{i=1}^N \frac{1}{1 + \omega_0 \lambda_i} \mathbf{u}_i \mathbf{u}_i^\top \quad (51)$$

with λ_i being the i-th eigenvalue of the Laplacian $\mathbf{L}$, and $\mathbf{u}_i$ the corresponding eigenvector, we have

$$\lambda_{\min}\{\mathbf{H}(\omega_0)\} = \lambda_{\min}\{(\mathbf{I} + \omega_0 \mathbf{L})^{-1}\} = \frac{1}{1 + \omega_0 \lambda_{\max}(\mathbf{L})}. \quad (52)$$

From Lemma 5, we have

$$\lambda_{\max}\{\boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L}\} \leq \lambda_{\max}(\mathbf{L})\max\{\omega_i^2\} \quad (53)$$

which results in

$$\lambda_{\max}\{\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L}\} \leq 1 + \lambda_{\max}(\mathbf{L})\max\{\omega_i^2\}. \quad (54)$$

The lower bounds of $\lambda_{\min}\{\mathbf{H}(\boldsymbol{\omega})\}$ then follows

$$\lambda_{\min}\{(\mathbf{I} + \boldsymbol{\omega}\boldsymbol{\omega}^\top \odot \mathbf{L})^{-1}\} \geq \frac{1}{1 + \lambda_{\max}(\mathbf{L})\max\{\omega_i^2\}}. \quad (55)$$

Finally, the sufficient conditions for $\text{mse}(\boldsymbol{\omega}) \leq \text{mse}(\omega_0)$ are

$$\omega_i \geq \sqrt{\omega_0} > 0, \; \text{for } i = 1, 2, \ldots, n \quad (56a)$$

$$\frac{1}{1 + \lambda_{\max}(\mathbf{L})\max\{\omega_i^2\}} + \frac{1}{1 + \omega_0 \lambda_{\max}(\mathbf{L})} \geq 2\gamma \quad (56b)$$

which completes the proof.

APPENDIX D
PROOF OF COROLLARY 1
If we let the following two hold

$$\frac{1}{1 + \lambda_{\max}(\mathbf{L})\max\{\omega_i^2\}} \geq \gamma \quad (57a)$$

$$\frac{1}{1 + \omega_0 \lambda_{\max}(\mathbf{L})} \geq \gamma \qquad (57b)$$

then we have the condition (15a) sufficiently satisfied. Since $\gamma = \rho/(1+\rho) = 1/(1+\frac{1}{\rho})$, we substitute this relation into above two conditions and have

$$\max\left\{\omega_i^2\right\}\lambda_{\max}(\mathbf{L}) \leq \frac{1}{\rho} \qquad (58a)$$

$$\omega_0 \lambda_{\max}(\mathbf{L}) \leq \frac{1}{\rho} \qquad (58b)$$

$$\omega_i \geq \sqrt{\omega_0} > 0, \ \text{for } i = 1, 2, \ldots, n. \qquad (58c)$$

Condition (58b) can be omitted since $\max\{\omega_i^2\} \geq \omega_0$, which completes the proof.

APPENDIX E
SIMPLIFYING THE COST FUNCTION FOR (24)

First, we expand the inner term as

$$\mathbb{E}[\|\mathbf{y} - (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})\mathbf{x}^*\|_2^2]$$

$$= \mathbb{E}[\text{tr}\{[\mathbf{y} - (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})\mathbf{x}^*][\mathbf{y} - (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})\mathbf{x}^*]^\top\}]$$

$$= \mathbb{E}[\text{tr}\{\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\mathbf{x}^{*T}(\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L}) + (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*T}\}]. \qquad (59)$$

Since the trace operation can be switched with the expectation operation, the cost function is equal to

$$\text{tr}\{\mathbb{E}(\mathbf{y}\mathbf{y}^\top) - 2\mathbb{E}(\mathbf{y}\mathbf{x}^{*T})(\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L}) + (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*T}\}. \qquad (60)$$

Further, since $\mathbb{E}(\mathbf{y}\mathbf{y}^\top) = \boldsymbol{\Sigma} + \mathbf{x}^* \mathbf{x}^{*T}$ and $\mathbb{E}(\mathbf{y}\mathbf{x}^{*T}) = \mathbf{x}^* \mathbf{x}^{*T}$, the above is equivalent to

$$\text{tr}\{\boldsymbol{\Sigma} + \mathbf{x}^* \mathbf{x}^{*T} - 2\mathbf{x}^* \mathbf{x}^{*T}(\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L}) + (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*T}\}$$

$$= \text{tr}\{\boldsymbol{\Sigma} + [\mathbf{I} - (\mathbf{I} + \boldsymbol{\Omega} \odot \mathbf{L})]^2 \mathbf{x}^* \mathbf{x}^{*T}\}$$

$$= \text{tr}\{\boldsymbol{\Sigma} + (\boldsymbol{\Omega} \odot \mathbf{L})^2 \mathbf{x}^* \mathbf{x}^{*T}\}. \qquad (61)$$

As our optimization variable is $\boldsymbol{\Omega}$, we can drop the unrelated covariance matrix $\boldsymbol{\Sigma}$, which gives the cost function in problem (25).

APPENDIX F
DETAILING THE COST FUNCTION IN (26)

The cost function in (26) is

$$\mathbb{E}\left\{\|\mathbf{H}(\boldsymbol{\Omega})\mathbf{y} - \mathbf{x}^*\|_2^2\right\} \qquad (62)$$

which can be further expressed as

$$\mathbb{E}\text{tr}(\mathbf{H}^2(\boldsymbol{\Omega})\mathbf{y}\mathbf{y}^\top - 2\mathbf{H}(\boldsymbol{\Omega})\mathbf{y}\mathbf{x}^{*\top} + \mathbf{x}^* \mathbf{x}^{*\top}). \qquad (63)$$

Then, exchanging the expectation and trace operator, we obtain the cost function

$$\text{tr}\{(\mathbf{H}^2(\boldsymbol{\Omega})) - 2\mathbf{H}(\boldsymbol{\Omega}) + \mathbf{I})\mathbf{x}^* \mathbf{x}^{*\top} + \mathbf{H}^2(\boldsymbol{\Omega})\boldsymbol{\Sigma}\}. \qquad (64)$$

From this expression, we indeed see it includes both the signal and the noise.

REFERENCES

[1] M. Yang, C. Mario, E. Isufi, and G. Leus, "Node varying regularization for graph signals," in *Proc. Euro. Signal Process. Conf.*, 2020, pp. 845–849.

[2] W. Huang, A. G. Marques, and A. R. Ribeiro, "Rating prediction via graph signal processing," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5066–5081, Oct. 2018.

[3] M. A. Russell, *Mining the Social Web: Analyzing Data From Facebook, Twitter, LinkedIn, and Other Social Media Sites*, Gravenstein Highway North, Sebastopol, CA, USA: O'Reilly Media, Inc., 2011.

[4] A. Mehri, A. H. Darooneh, and A. Shariati, "The complex networks approach for authorship attribution of books," *Physica A: Stat. Mechanics Appl.*, vol. 391, no. 7, pp. 2429–2437, 2012.

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

[6] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. 13th Int. Conf. Very Large Data Bases*, 2004, pp. 588–599.

[7] K.-M. Lee, B. Min, and K.-I. Goh, "Towards real-world complexity: An introduction to multiplex networks," *Eur. Phys. J. B*, vol. 88, no. 2, pp. 1–20, 2015.

[8] M. C. Scroggie, M. E. Kacaba, D. A. Rochon, and D. M. Diamond, "System and method for providing shopping aids and incentives to customers through a computer network," U.S. Patent 6014634, Jan. 11, 2000.

[9] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[10] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[11] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 491–494.

[12] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4. pp. 728–740, Jun. 2015.

[13] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584–3598, Jul. 2018.

[14] E. Isufi, P. Di Lorenzo, P. Banelli, and G. Leus, "Distributed Wiener-based reconstruction of graph signals," in *Proc. IEEE Stat. Signal Process. Workshop*, 2018, pp. 21–25.

[15] P. Berger, G. Hannak, and G. Matz, "Graph signal recovery via primal-dual algorithms for total variation minimization," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 842–855, Sep. 2017.

[16] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 736–751, Dec. 2018.

[17] P. Chen and S. Liu, "Bias-variance tradeoff of graph Laplacian regularizer," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1118–1122, Aug 2017.

[18] A. Venkitaraman, S. Chatterjee, and P. Händel, "Predicting graph signals using kernel regression where the input signal is agnostic to a graph," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 4, pp. 698–710, Dec. 2019.

[19] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.

[20] A. Venkitaraman, S. Chatterjee, and P. Händel, "Multi-kernel regression for graph signal processing," in *2018 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4644–4648.

[21] C. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Equations.* Marshfield, MA, USA: Boston Pitman Publication, 1984.

[22] F. R. Chung and F. C. Graham, *Spectral Graph Theory.* Providence, Rhode Island, USA: American Mathematical Soc., 1997.

[23] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[24] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 3042–3054, Jun. 2014.

[25] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[26] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Random walk graph Laplacian-based smoothness prior for soft decoding of jpeg images," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 509–524, Feb. 2017.

[27] J. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1770–1785, Apr. 2017.

[28] C. Dinesh, G. Cheung, and I. V. Bajić, "Point cloud denoising via feature graph laplacian regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 4143–4158, Jan. 2020.

[29] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Roy. Stat. Society: Ser. B. (Stat. Methodol.)*, vol. 67, no. 1, pp. 91–108, 2005.

[30] S. Mallat, *A Wavelet Tour of Signal Processing*. Amsterdam, The Netherlands: Elsevier, 1999.

[31] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot, "Dual constrained tv-based regularization on graphs," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1246–1273, 2013.

[32] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani, "Trend filtering on graphs," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 3651–3691, 2016.

[33] R. J. Tibshirani *et al.*, "The solution path of the generalized lasso," *Ann. Statist.*, vol. 39, no. 3, pp. 1335–1371, 2011.

[34] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Representations of piecewise smooth signals on graphs," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2016, pp. 6370–6374.

[35] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[36] O. Scherzer, "The use of Morozov's discrepancy principle for Tikhonov regularization for solving nonlinear ill-posed problems," *Computing*, vol. 51, no. 1, pp. 45–60, 1993.

[37] S. W. Anzengruber and R. Ramlau, "Morozov's discrepancy principle for Tikhonov-type functionals with nonlinear operators," *Inverse Problems*, vol. 26, Dec. 2009, Art. no. 025001.

[38] P. C. Hansen and D. P. O'Leary, "The use of the l-curve in the regularization of discrete ill-posed problems," *SIAM J. Sci. Comput.*, vol. 14, no. 6, pp. 1487–1503, 1993.

[39] G. H. Golub and U. von Matt, "Generalized cross-validation for large-scale problems," *J. Comput. Graphical Statist.*, vol. 6, no. 1, pp. 1–34, 1997.

[40] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.

[41] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320–2333, May 2019.

[42] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 47–60, Mar. 2019.

[43] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Tech. Rep., Pittsburgh, PA, USA, Aug. 4, 1994.

[44] E. Isufi, "Graph-time signal processing: Filtering and sampling strategies," Ph.D. dissertation, Delft Univ. Technol., 2019.

[45] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. 1st ed., New York, NY, USA: Wiley, 1996.

[46] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[47] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," Version 2.1, Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[48] D. P. Bertsekas, *Convex Optimization Theory*. Belmont, MA, USA: Athena Scientific, 2009.

[49] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[50] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, 1996.

[51] H. Jiang, T. Kathuria, Y. T. Lee, S. Padmanabhan, and Z. Song, "A faster interior point method for semidefinite programming," in *Proc. 61st Annu. Symp. Found. Comput. Sci.*, Durham, NC, USA, 2020, pp. 910–918.

[52] M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn, "Solving a class of non-convex min-max games using iterative first order methods," in *Proc. Adv. Neural Inf. Process. Syst.*, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/25048eb6a33209cb5a815bff0cf6887c-Paper.pdf

[53] T. Lin, C. Jin, and M. I. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6083–6093.

[54] N. Perraudin *et al.*, "GSPBOX: A toolbox for signal processing on graphs," Aug. 2014. [Online]. Available: https://epfl-lts2.github.io/gspbox-html/

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

[56] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge, U.K.: Cambridge Univ. Press, 2013.