A major line of work in graph signal processing [2] during the past 10 years has been to design new transform methods that account for the underlying graph structure to identify and exploit structure in data residing on a connected, weighted, undirected graph. The most common approach is to construct a dictionary of atoms (building block signals) and represent the graph signal of interest as a linear combination of these atoms. Such representations enable visual analysis of data, statistical analysis of data, and data compression, and they can also be leveraged as regularizers in machine learning and ill-posed inverse problems, such as inpainting, denoising, and classification.

In general, the desirable properties when designing dictionaries for graph signals include the following:

1) The atoms have an interpretable form that accounts for the underlying graph structure so that the inner products between a graph signal and each atom are informative.

2) The dictionary comprises an orthonormal basis or tight frame for the signal space, so that the contribution of each atom can be computed via an inner product with the graph signal and the energy of the graph signal is equal to a constant multiple of the energy of the transform coefficients.

3) It is numerically efficient to apply the dictionary analysis and synthesis operators (forward and inverse transforms).

4) Signals of certain mathematical classes can be represented exactly or approximately as sparse linear combinations of a subset of the dictionary atoms.

By our count, approximately 100 conference and journal articles written during the past decade have introduced new dictionaries for graph signals. These include designs for analytic dictionaries that are adapted to the graph structure but not any specific training data, as well as techniques for learning dictionaries from training data. Broad classes of dictionaries include graph Fourier transforms (GFTs); windowed GFTs; vertex domain designs, including spatial wavelets, hierarchical trees, lifting transforms, and top-down approaches; diffusion-based designs; spectral domain designs; pyramid transforms;
and generalized filter banks (see [1] for references). Despite, or perhaps because of, the number of new graph signal dictionary designs, it remains difficult to identify which dictionary might be best suited for a specific task and to understand subtle qualitative tradeoffs when specifying the parameters of a given dictionary construction.

In this survey, we restrict our attention to localized spectral graph filter frames, whose atoms are created by localizing patterns (spectral filters) to different regions of the graph. The seminal example of a dictionary of such atoms is spectral graph wavelets [3]. However, localized spectral graph filter frames are broad in their scope, including more recently proposed methods, such as single-level filter banks for graph signals (e.g., [4]–[10]; see [1] for more examples); variational or interpolating splines (e.g., [11]); frames adapted to training data [12],[13]; frame constructions for general graph signals [14]–[17]; frame constructions tailored to specific applications, such as functional magnetic resonance imaging (fMRI) data analysis [18] and community mining [19]; “natural” wavelets [20]; and even some vertex domain constructions. Our motivations for examining these dictionaries include:

1) The design framework is flexible: it can yield highly redundant dictionaries to sparsely represent graph signals and new bases to efficiently extract structure from data on graphs, and it can also incorporate representative training signals when they are available.

2) The atoms have a physically interpretable structure, and their closed-form definition opens the door to formal mathematical analysis.

3) Fast numerical approximations exist to efficiently apply these dictionary transforms and their inverses to data residing on large, sparse graphs, which are increasingly common in signal processing and machine learning applications.

Due to the multiscale and localized structure of their atoms, these dictionaries are particularly relevant for applications where interesting phenomena are expressed in discontinuities and quick changes in signal values in smaller regions of a graph, analogous to edges in images.

The MATLAB code for all figures and numerical experiments in this article is available at http://www.macleaster.edu/~dshuman1/publications.html. The extended cut [1] contains additional references and numerical experiments.

**Dictionaries of localized spectral patterns**

Keeping with the notation of [2], we consider data residing on a connected, weighted, undirected graph \( G = (V,E,W) \) characterized by a finite set of vertices \( V \) with \(|V|=N\), a set of edges \( E \), and a weighted adjacency matrix \( W \). A signal or function \( f:V \rightarrow \mathbb{R} \) defined on the vertices of the graph may be represented as a vector \( f \in \mathbb{R}^N \), where the \( i \)th element of the vector \( f \) represents the graph signal value at vertex \( i \) in \( V \).

The dictionaries we consider feature atoms of the form

\[
\mathbf{v}_{ij} := \tilde{g}_i(\mathbf{L}) \mathbf{e}_j = \tilde{g}_i(\mathbf{L}) \mathbf{u}_j^\top \mathbf{u}_i,
\]

In (1), \( \mathbf{e}_j \) is a graph signal with a value of one at vertex \( i \) and zero elsewhere, \( \mathbf{L} = \mathbf{D} - \mathbf{W} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top \) is the (combinatorial) graph Laplacian, the columns of \( \mathbf{U} \) are the orthogonal eigenvectors of \( \mathbf{L} \), the " * " symbol denotes a conjugate transpose, and \( \mathbf{\Lambda} \) is a diagonal matrix whose \( j \)th diagonal element \( \lambda_j \) is the eigenvalue of \( \mathbf{L} \) associated with the eigenvector \( \mathbf{u}_i \), which is the \( j \)th column of \( \mathbf{U} \). While we use the combinatorial (nonnormalized) graph Laplacian \( \mathbf{L} \) throughout, the ideas we discuss apply to dictionaries comprised of atoms of the form (1), with the graph Fourier basis \( \mathbf{U} \) chosen as the eigenvectors of other symmetric generalized graph Laplacian operators, such as the normalized graph Laplacian \( \mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \). Each spectral graph filter or kernel \( \hat{g}_i(\cdot) \) is a function from \( \sigma(\mathbf{L}) = \{\lambda_0, \lambda_1, \ldots, \lambda_{N-1}\} \), the set of Laplacian eigenvalues, to the real numbers. Thus, \( \hat{g}_i(\mathbf{L}) \) is a diagonal matrix, with the \( j \)th diagonal entry equal to \( \hat{g}_i(\lambda_j) \). In practice, these filter functions are often defined on the continuous range \([0, \hat{\lambda}]\), where \( 0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1} = \hat{\lambda} \).

At a more intuitive level, we can think of each spectral graph filter \( \hat{g}_i(\cdot) \) as defining a spectral pattern that is localized to different regions of the graph, with vertex \( i \) being the center of the localized pattern \( \mathbf{v}_{ij} = \tilde{g}_i(\mathbf{L}) \mathbf{e}_j \). As an extreme example, if the spectral pattern is \( \hat{g}(\lambda) = 1 \) for all \( \lambda_i \), then the localized pattern centered at vertex \( i \) is \( \tilde{g}_i(\mathbf{L}) \mathbf{e}_j \).

Figure 1 and “Example 1: Variational/Interpolating Splines” display more localized spectral patterns on the Stanford bunny graph.

We refer to a collection of atoms of the form (1) as a localized spectral graph filter dictionary (LSGFD), denoted by

\[
\mathcal{D} = \{ \tilde{g}_i(\mathbf{L}) \mathbf{e}_j \}_{j=1,2,...,J; i \in \mathcal{V}},
\]

In (2), \( \mathcal{V}_j \subseteq \mathcal{V} \) is the set of center vertices to which the \( j \)th spectral pattern \( \tilde{g}_j(\cdot) \) is localized, and each atom \( \mathbf{v}_{ij} := \tilde{g}_i(\mathbf{L}) \mathbf{e}_j \) is a graph signal in \( \mathbb{R}^N \). Therefore, to fully specify an LSGFD \( \mathcal{D} \), we need to answer the following questions, which are the focus of the next two sections, respectively:

1) How many spectral patterns should we use, and what should those patterns be? That is, we must specify the number of filters, \( J \), and the form of the filters, \( \{ \tilde{g}_1(\cdot), \tilde{g}_2(\cdot), \ldots, \tilde{g}_J(\cdot) \} \).

2) For each spectral pattern \( \tilde{g}_j(\cdot) \), how many center vertices should the pattern be localized to, and which vertices should those be? That is, we must specify the sets \( \mathcal{V}_j \) for each \( j \).

In specifying the spectral patterns and sets of center vertices for LSGFDs, it is also important to keep in mind what information is available and the graph size. In all cases in this survey, we assume the underlying graph structure \( G \) is known, although learning graph structures is a vibrant area of ongoing
research [21]. In terms of data available in the design of the dictionary, there are three possibilities:

1) No data are available (which is the default unless otherwise specified).

2) The design of the dictionary atoms may also take into account a set of one or more specific graph signals that is to be analyzed by the dictionary (we refer to the resulting dictionaries as signal adapted).

3) A set of training data is available to learn parameters of the LSGFD, but the dictionary is then used to analyze a different set of (presumably similar) graph signals.

For small-to-medium-size graphs (say, on the order of 10,000 or fewer vertices), the full Laplacian eigendecomposition \( \mathbf{L} = \mathbf{U} \Lambda \mathbf{U}^\top \) can be computed, and therefore the exact Laplacian eigenvectors and eigenvalues can be used in the dictionary design. For larger graphs, however, it may not be tractable to perform this decomposition, and we thus put an emphasis, in the next two sections, on methods that do not require these quantities. Without the Laplacian eigenvectors, we almost always need an estimate of the maximum eigenvalue \( \lambda_{\text{max}} \) via, e.g., a few steps of the Lanczos algorithm [22] or a closed-form upper bound. For example, \( \bar{\lambda} \) can be taken to be the maximum sum of the degrees of any two vertices connected by an edge, \( \lambda_{\text{max}} \leq \max_{(m, n) \in \mathcal{E}} \{ d(m) + d(n) \} \), where \( d(n) \) is the degree of vertex \( n \) [23, Cor. 3.2].

![Figure 1. Localized spectral graph filter frame atoms. (a)–(c) Three different filters/patterns localized to the same center vertex. (c)–(e) The same filter/pattern localized to three different center vertices.](image)
In Figure S1, we show examples of interpolating and then interpolates \(\hat{g}(\cdot)\) localized to a subset \(\mathcal{V}_i\) of the vertices. They are used as a basis to interpolate an entire graph signal from its sample values at the vertices in \(\mathcal{V}_i\). In Figure S1, we show examples of interpolating kernels and an atom generated from each on the Stanford bunny graph.

In addition to obtaining a fast estimate for the spectral range \([0, \lambda_{\text{max}}]\), it is often also beneficial to estimate the distribution of the Laplacian eigenvalues across the spectral range. Specifically, the cumulative spectral density function of \(L\), defined as

\[
P_A(z) := \frac{1}{N} \sum_{\ell=0}^{N-1} \mathbb{1}_{[\lambda_{\ell} \leq z]},
\]

can be efficiently estimated by different methods. We use a variant of the kernel polynomial method detailed in Algorithm 2 that leverages a stochastic trace estimator to approximate the number of eigenvalues below linearly spaced points between zero and \(\lambda_{\text{max}}\) and then interpolates these values via monotonic piecewise cubic interpolation to generate an estimate of the cumulative spectral density function (3). The computational cost is proportional to the number of edges in the graph. As an order-of-magnitude example, for a sparse graph with more than 469,000 vertices, estimates for the maximum eigenvalue and the density function can be computed on a laptop in approximately 1 and 16 s, respectively.

In Figure 2, we show examples of exact and approximate cumulative spectral density functions on six different graphs. In summary, while the full Laplacian ED is necessary to exactly compute the atoms in (1) and their inner products with a graph signal, the spectral range and density function can be estimated inexpensively and leveraged in the design of the filters, the selection of the center vertices, and the approximate computation of the inner products between the graph signal and each dictionary atom. We discuss these details further in the next two sections.

### Example 1: Variational/Interpolating Splines

Variational, or interpolating, splines on graphs, which were pioneered in [11], are atoms of the form (1) with a single low-pass filter \(g(\cdot)\) localized to a subset \(\mathcal{V}_i\) of the vertices. They are used as a basis to interpolate an entire graph signal from its sample values at the vertices in \(\mathcal{V}_i\). In Figure S1, we show examples of interpolating kernels and an atom generated from each on the Stanford bunny graph.

We represent the synthesis operator with the matrix \(\Phi \in \mathbb{R}^{N \times M}\) where the columns of \(\Phi\) are the \(M = \sum_j |\mathcal{V}_j|\) dictionary atoms in \(D\). We refer to its adjoint \(\Phi^*\) as the analysis operator; this matrix maps a graph signal to the analysis coefficients \(\{\langle f, \phi_{j,i} \rangle\}\). If the dictionary \(D\) satisfies the frame condition for all vectors \(f\) in some subspace \(S\) of \(\mathbb{R}^N\) (or all of \(\mathbb{R}^N\)),

\[
\|f\|_2^2 \leq \|\Phi^* f\|_2^2 = \sum_j \sum_{i \in \mathcal{V}_j} |\langle f, \phi_{j,i} \rangle|^2 \leq B \|f\|_2^2,
\]

then any graph signal in the subspace can be exactly recovered from its analysis coefficients \(\Phi^* f\). Moreover, if \(A = B\) in (4), the dictionary is known as a tight frame, and \(f = (1/A) \sum_{j=1}^{J} \sum_{i \in \mathcal{V}_j} \langle f, \phi_{j,i} \rangle \phi_{j,i} = (1/A) \Phi \Phi^* f\). A tight frame with frame bounds \(A = B = 1\) is called a Parseval frame and has the added benefit that \(\|\Phi \mathbf{1}\|_2 = \|f\|_2\); i.e., the energy of the analysis coefficients is the same as the energy of the graph signal [24].

Finally, we mention the connection between the aforementioned analysis coefficients and the graph spectral filter banks. As shown in Figure 3, in a \(J\)-channel graph filter bank, \(J\) different filters are applied to the signal, and the values of \(g_j(L) f\) (the filtered signal in the \(j\)th channel) at a specified set of downsampled vertices \(\mathcal{V}_j\) are stored. The dictionary analysis coefficients \(\{\langle f, \phi_{j,i} \rangle\}_{i \in \mathcal{V}_j}\) correspond exactly to the downsampled values in the \(j\)th channel of the filter bank.

### Design of spectral filters

Three are three broad classes of spectral filter designs: 1) those adapted only to the spectral range \([0, \lambda_{\text{max}}]\) e.g., [3]–[5], [7],
FIGURE 2. Estimated and actual cumulative spectral density functions (3) for six graph Laplacians: (a) a random Erdős–Rényi graph with $N = 500$ vertices and an edge probability of 0.2, (b) the Minnesota traffic network ($N = 2,642$), (c) the Andrianov net25 matrix ($N = 9,520$), (d) the Stanford bunny graph ($N = 2,503$), (e) an eight-neighbor local graph for eastern Massachusetts ($N = 877$), and (f) the cerebellum region of the brain ($N = 4,465$). See [1] for graph source references. CDF: cumulative distribution function.

FIGURE 3. The equivalence between single-level graph spectral filter banks and localized spectral graph filter dictionaries. (a) In the four-channel filter bank, the graph signal is filtered by each of the $J = 4$ spectral filters and then downsampled on the corresponding vertex sets $\{V_j\}$ to yield the filter bank coefficients $\{a_j\}$. (b) and (c) Each coefficient $a_i = \delta_i (\mathbb{L} f)(i) = \delta_i g_i (\mathbb{L}) f = f (\mathbb{G} \mathbb{L}) \delta_i$. In the final column of (a) corresponds exactly to the inner product between the graph signal and the dictionary atom $\phi_{i}\rho_{j}$ as $\langle f, \phi_{i}\rangle = \langle f, T g_i \rangle = \langle f, g_i \rangle \delta_i = f (\mathbb{G} \mathbb{L}) \delta_i$. 

0.2 0.4 0.6 0.8 1
0 20 40 60
$\lambda$
(a)

0.2 0.4 0.6 0.8 1
0 5 6
$\lambda$
(b)

0.2 0.4 0.6 0.8 1
0 10
$\lambda$
(c)

0.2 0.4 0.6 0.8 1
0 10 20 30
$\lambda$
(d)

Estimated Spectral CDF
Actual Spectral CDF

0.2 0.4 0.6 0.8 1
0 50 100 150
$\lambda$
(e)

0.2 0.4 0.6 0.8 1
0 100
$\lambda$
(f)
where the infimum in (6) is taken across all polynomial kernels of degree \( K_m \), as defined in (5). If \( \hat{g}(\cdot) \) is real analytic on \([0, \lambda_{\text{max}}]\), the upper bound in (6) converges geometrically to zero as \( d_g(i, n) \) increases.

In short, and less precisely mathematically, the smoother the filter \( \hat{g}(\cdot) \) is in the spectral domain, the more concentrated is the energy of the atom \( \varphi_{ij} = T_i \hat{g}_j \) around the center vertex \( i \); compare, e.g., the first two atoms in Figure 1.

**Eigenvector groupings**

Recall from the introduction that, for the inner products between a graph signal and each atom to be informative, the atoms should have interpretable structural features that account for the underlying graph. The localization in the vertex domain described previously is one such structural feature. The shape of the filter \( \hat{g}(\cdot) \) in the graph spectral domain leads to another: smoothness in terms of how much the atom’s values vary between neighboring vertices, particularly those connected by a high edge weight. The unit norm Laplacian eigenvectors satisfy

\[
\lambda_i = u_i^T L u_i = \sum_{(m,n) \in E} W_{m,n} (u_i(m) - u_i(n))^2,
\]

which is the total variation with respect to the graph. Therefore, the eigenvectors associated with the lower eigenvalues vary less from vertex to neighboring vertex. Moreover, the eigenvectors are roughly ordered in terms of the number of zero crossings, which are defined as edges where the values of the eigenvector at the two connected vertices have opposite signs [2, Fig. 3].

Based off the analogy between this smoothness of Laplacian eigenvectors and the frequency of complex exponentials in 1D signal processing, the most common spectral design approach in the graph signal processing literature is to choose filters concentrated on one part of the graph spectrum, grouping together eigenvectors with similar levels of total variation with respect to the graph. In particular, when localized to different center vertices via (1), filters whose support is concentrated on the eigenvectors associated with small eigenvalues lead to scaling functions or windows around the center vertex [see Figure 1(a) and “Example 1: Variational/Interpolating Splines”]. The inner products between such atoms and a graph signal provide information about the trend and the local average of the signal in the neighborhood of the center vertex. On the other hand, all Laplacian eigenvectors associated with eigenvalues greater than zero sum to zero because they are orthogonal to \( u_0 \), which is constant across all vertices. Thus, any filter with \( \hat{g}(0) = 0 \) yields atoms \( \varphi_{ij} \) that have a mean of zero and feature some oscillation [see Figure 1(b) and (c)].

Noting that the Laplacian eigenvalues correspond to the total variation of the eigenvectors but not necessarily the directions of their oscillations on the graph (see Figure 4), some more recent works ([20] and [26] and the references therein) investigate other ways to group the eigenvectors. For example, [20] suggests viewing the eigenvectors as probability distributions on the graph, quantifying the distances between eigenvectors using optimal transport theory, clustering the eigenvectors...
Example 2: Spectral Filter Designs That Use Only the Spectral Range

We show 10 different sets of six filter patterns for the cerebellum graph [18], whose spectral range is \([0, \lambda_{\text{max}}] = [0, 32.4]\). In Figures S2 and S3, the vertical axis represents the value of the filter, and the shaded gray/black circles indicate the values of \(G(\lambda) = \sum_{m=0}^{\infty} |g_m(\lambda)|^2\) at each of the Laplacian eigenvalues, with darker areas denoting regions of higher spectral density. For each of the filters \(g_j(\lambda)\) in the set of uniform translates (Figure S2), the corresponding filter in the log-warped set (Figure S3) is given by \(g_{\lambda}(\lambda) = g_j(\lambda) \omega(\lambda)\), where the warping function is \(\omega(\lambda) = \log(1 + \nu\lambda)\) for a parameter \(\nu > 0\) (\(\nu = 10\) here).

We highlight the following design considerations:

1) All of these designs cover the entire spectrum; i.e., \(G(\lambda) > 0\) for all \(\lambda \in [0, \lambda_{\text{max}}]\). Thus, each design yields a frame when every filter is localized to be centered at every vertex.

2) The last two sets of wavelet filters (Figure S3(e) and (f)) are the only two amongst those shown that do not satisfy the Parseval frame condition, \(G(\lambda) = 1\) for all \(\lambda \in \sigma(L)\), from Theorem 4.

3) Because the ideal filters in Figures S2(a) and S3(a) do not overlap, the atoms generated from one filter are orthogonal to the atoms generated from any other filter. On the other hand, atoms generated by localizing strongly overlapping filters (e.g., the yellow and red filters of the fast tight wavelet frame) to the same center vertex are likely to be highly correlated.

4) As detailed in Theorem 2, filters that are smooth (well approximated by low-order polynomials) yield dictionary atoms that are more localized around the center vertex; i.e., most of their energy is close to the center. For example, for Meyer-type wavelets, log-warped translates, and spectral graph wavelets, the filters that cover the upper end of the spectrum yield more localized atoms than the scaling and wavelet filters at the low end of the spectrum.

Reference

[S1] N. Perraudin, J. Paratte, D. I Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, “GSPBOX: A toolbox for signal processing on graphs,” 2014, arXiv:1408.5781.
based on their distances, and constructing a spectral filter for each cluster, with the support of the filter set to match the eigenvectors of that cluster.

Orthogonality and near orthogonality

To reduce the correlation between atoms (and, in turn, improve the discriminatory power of taking inner products between each atom and a target signal, as discussed in the “Theoretical Considerations and Metrics” section), it may be desirable for all atoms that are generated from a single filter to be orthogonal or nearly orthogonal to all atoms that are generated from all other filters; i.e., \( \langle T g_j, T g_j \rangle \approx 0 \) for all \( j \neq j' \). This can be ensured via the filter design, with a sufficient condition for the orthogonality of atoms generated from different patterns being that \( g_j(\lambda) \approx 0 \) for all \( j \neq j' \) and all \( \lambda \in \sigma(L) \). For the specific case when \( J = 2 \) and \( \mathcal{G} \) is a bipartite graph

Example 3: Spectral Filter Designs Adapted to the Spectral Density of the Graph or the Ensemble Energy Density of the Training Signals

The ideal filters and uniform translates from “Example 2: Spectral Filter Designs That Use Only the Spectral Range” are designed to cover equal portions of the spectral range \([0, \lambda_{\text{max}}]\), but they are the same for any two graphs with the same value of \( \lambda_{\text{max}} \). One option to further adapt the filters to the graph structure is to warp them so that each filter contains roughly the same number of Laplacian eigenvalues [15]. This is accomplished by defining spectrum-adapted filters \( \tilde{g}_j(\lambda) = \tilde{g}_j(\lambda_{\text{max}} P_{\lambda}(\lambda)) \), where \( \{	ilde{g}_j(\cdot)\} \) are the uniform translates from “Example 2: Spectral Filter Designs That Use Only the Spectral Range” and \( P_{\lambda}(\cdot) \) is an estimate of the cumulative spectral density defined in (3). For the cerebellum graph, the Laplacian eigenvalues are concentrated in the middle of the spectral range (Figure S4), and therefore the spectrum-adapted Meyer-type filters shown in Figure S5(b) are narrower in this region of high spectral density. The same idea can be used to generate spectrum-adapted wavelets \( \tilde{h}_j(\lambda) = \tilde{g}_j((\lambda_{\text{max}} P_{\lambda}(\lambda)) \omega(\lambda_{\text{max}})) \omega(\lambda) \) by using a warping function \( \omega(\lambda) = \log(1 + \nu \lambda_{\text{max}} P_{\lambda}(\lambda)) \).

When training data are available, another option is to adapt the filters to be narrower in the regions of the spectrum where the energy of the training signals is concentrated [13]. This can again be achieved via warping, using an estimate of the ensemble energy cumulative spectral density function \( P_{\lambda}(\cdot) \) defined in (8) in place of the spectral density estimate \( P_{\lambda}(\cdot) \) in the warping function. The plot of the density functions (Figure S4) shows that, despite the Laplacian eigenvalues being concentrated in the middle portion of the cerebellum graph spectrum, the energies of 292 functional magnetic resonance imaging training signals are heavily concentrated in the lower end of the spectrum. Therefore, the signal-adapted design that aims to have roughly the same signal energy in each filter band features narrower filters at lower eigenvalues.

FIGURE S4. The density functions. CDF: cumulative distribution function.

FIGURE S5. (a) Meyer-type uniform translates. (b) Spectrum-adapted Meyer-type filters [15]. (c) Signal-adapted Meyer-type filters [13].
with the normalized Laplacian eigenvectors as the graph Fourier basis, it is possible to go a step further and generate $N$ atoms that are orthogonal not only to atoms generated from the other filter but also to other atoms generated from the same filter. (These conditions can be adapted for a regular bipartite graph with the nonnormalized Laplacian eigenvectors as the graph Fourier basis.)

Theorem 3 [4]
Let $\mathcal{G}$ be a bipartite graph with a bipartition $\{V_1, V_2\}$, and consider an LSGFD $\mathcal{D}$ of the form $(2)$ with $J = 2$ [i.e., $\hat{g}_j(\cdot)$ is localized to the center vertices in $V_j$ for $i = 1, 2$], using the normalized Laplacian graph Fourier basis. Then, necessary and sufficient conditions on the filters to ensure that the $N$ atoms of $\mathcal{D}$ form an orthogonal basis for $\mathbb{R}^N$ are that $g_j(\lambda) \hat{g}_j(2 - \lambda) = g_j(\lambda) \hat{g}_j(2 - \lambda)$ and $|\hat{g}_j(\lambda)|^2 + |\hat{g}_j(\lambda)|^2 = c^2$ for all $\lambda$ and any constant $c$.

Coverage of the spectrum
A necessary condition for $\mathcal{D}$ to be a frame for all graph signals in $\mathbb{R}^N$ is that, for all $\lambda \in \sigma(\mathcal{L})$, $g_j(\lambda) \neq 0$ for some $j \in \{1, 2, \ldots, J\}$. If this is not true for some $\lambda$, then $\langle u_i, \phi_{ij} \rangle = 0$ for all $i$ and $j$ and thus $\|\Phi u_i\|_2 = 0$, contradicting the frame condition (4). Therefore, choosing a set of filters that covers the full spectral range $[0, \lambda_{\max}]$ (or at least the portion of it whose Laplacian eigenvectors span the subspace of the signals of interest) is a good place to start. In the case that $\mathcal{V}_j = \mathcal{V}$ for all $j$ (i.e., every spectral pattern is localized to every vertex), which is often referred to as complete sampling or an undecimated filter bank, it is possible to design the spectral filters such that $\mathcal{D}$ is a Parseval frame.

Theorem 4 [3, Th. 5.6], [14], [15, Lemma 1]
Let $\mathcal{V}_j = \mathcal{V}$ for all $j$ and $\mathcal{D} := \{\phi_{ij}\}_{i=1, \ldots, N, j=1, \ldots, J}$ be a dictionary of atoms with $\phi_{ij} := T_{ij} \hat{g}_j$. Define $G(\lambda) := \sum_{j=1}^J \hat{g}_j(\lambda)^2$. If $G(\lambda) > 0$ for all $\lambda \in \sigma(\mathcal{L})$, then the frame condition (4) is satisfied for all $\mathbf{f} \in \mathbb{R}^N$, with the frame bounds $A = \min_{\lambda \in \sigma(\mathcal{L})} G(\lambda)$ and $B = \max_{\lambda \in \sigma(\mathcal{L})} G(\lambda)$. In particular, if $G(\lambda)$ is constant on $\sigma(\mathcal{L})$, $\mathcal{D}$ is a tight frame with $A = B$. Moreover, if $G(\lambda) = 1$, $\forall \lambda \in \sigma(\mathcal{L})$, then $\mathcal{D}$ is a Parseval frame; i.e., $\|\Phi \mathbf{f}\|_2^2 = \sum_{j=1}^J \sum_{i=1}^N |\langle \mathbf{f}, \phi_{ij} \rangle|^2 = \|\mathbf{f}\|_2^2$, $\forall \mathbf{f} \in \mathbb{R}^N$.

“Example 2: Spectral Filter Designs That Use Only the Spectral Range” includes filters that satisfy the sufficient condition of Theorem 4 for a Parseval frame.

It is important to distinguish between coverage of the spectral range and coverage of the spectrum. One subtlety about Theorem 4 is that, while the filters are often designed across the interval $[0, \lambda_{\max}]$ or $[0, \tilde{\lambda}]$, the condition for a tight frame is that $G(\lambda)$ needs to be constant only on the set of actual Laplacian eigenvalues $\sigma(\mathcal{L})$, as these are the values of the filter that contribute to the definition of the atom in (1). Related to this point, if a filter is defined on the interval $[0, \lambda_{\max}]$ but $\hat{g}_j(\lambda) = 0$ for all $\lambda \in \sigma(\mathcal{L})$, then any atoms $T_{ij} \hat{g}_j$ derived from this filter are equal to the zero vector and therefore do not provide any useful information about the graph signal. To avoid such noninformative atoms, the spectrum-adapted designs of [15] leverage the estimated spectral density function (3) to warp a set of nonadapted filters so that the support of each filter approximately contains a desired number of Laplacian eigenvalues, as shown in “Example 3: Spectral Filter Designs Adapted to the Spectral Density of the Graph or the Ensemble Energy Density of the Training Signals.”

Computational efficiency and approximation
As mentioned in the “Dictionaries of Localized Spectral Patterns” section, exactly computing the graph Laplacian eigenvectors is feasible only for small-to-medium graphs, implying that, for large graphs, the computation of the analysis coefficients $\langle \mathbf{f}, \phi_{ij} \rangle = \delta_{ij} \hat{g}_j(\mathcal{L}) \mathbf{f}$ must be efficiently approximated. Methods for approximating a matrix function times a vector [i.e., $\hat{g}_j(\mathcal{L}) \mathbf{f}$] include Krylov subspace methods, such as the Lanczos method; contour integral methods; the conjugate gradient; algebraic multigrid methods; rational approximations (also referred to as infinite impulse response filters in the graph signal processing community); spline approximations; and polynomial approximations (see [27] and [28] for surveys of these methods in centralized and distributed settings).

We focus our attention and numerical experiments on degree $K$ polynomial approximations $\hat{g}_{j,K}(\lambda)$ of the form (5) to each filter $\hat{g}_j(\cdot)$ (also referred to as finite impulse response filters in the graph signal processing community). The approximation $\mathbf{x}_j^{(K)} = \hat{g}_{j,K}(\mathcal{L}) \mathbf{f}$ to $\hat{g}_j(\mathcal{L}) \mathbf{f}$ can be computed recursively through either a three-term recurrence for specific types of polynomials (e.g., Chebyshev) or the nested multiplication iteration

$$\mathbf{x}_j^{(0)} = a_{j,K} \mathbf{f} + \mathcal{L} \mathbf{x}_j^{(l-1)}, \quad l = 1, 2, \ldots, K,$$

with $\mathbf{x}_j^{(0)} = a_{j,K} \mathbf{f}$. The computational complexity of computing $\mathbf{x}_j^{(K)} = \hat{g}_{j,K}(\mathcal{L}) \mathbf{f}$ through (7) or a three-term recurrence is $O(K^2 \mathcal{E})$, which, for a large, sparse graph, is approximately linear in the number of vertices, as opposed to the $O(N^3)$ required to naively compute the full ED of $\mathcal{L}$.

Additional advantages of the polynomial approximations include 1) the atoms are strictly localized, as described in Theorem 1; 2) in addition to the analysis operator, the synthesis operator $\Phi$ can be efficiently applied; and 3) both the analysis and the synthesis computations can be performed in a distributed setting where each vertex knows only its own signal value and can communicate solely with its neighboring vertices [28]. Polynomial approximation methods commonly used in the graph signal processing literature include Chebyshev [3], [29] and Jackson–Chebyshev [30]. “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation” shows filters resulting from these polynomial approximation methods. For the specific case of approximating an ideal low-pass filter with a small-degree polynomial, [31] introduces energy compaction filters that maximize the power of the polynomial filter that is concentrated on the specified band.
Approximating each spectral filter \( \hat{g}_i(\cdot) \) by a degree \( K \) polynomial \( \hat{p}_i(\cdot) \) reduces the complexity of applying the dictionary analysis operator from \( O(N^2) \) to \( O(K|E|) \), which, for large, sparse graphs, is \( O(N) \). The drawback of this scalable approximation is that the tight Parseval frame condition \( G(\lambda) = 1 \) for all \( \lambda \in \sigma(\mathcal{L}) \) of Theorem 4 is not typically satisfied by the polynomial approximations. Let \( \Phi^* \) be the approximate analysis operator with the polynomial filters and \( \alpha = \Phi^* \Gamma \) be the resulting analysis coefficients. There are three common options for fast, approximate inverse transforms. The first is to solve \( \Phi^* \Gamma \mu = \Phi \alpha \) via the conjugate gradient method [3], and the second is the frame inversion algorithm [32, Ch. 3] that sets \( \Gamma \mu = (2(A + B)) \Phi \alpha \) and iterates \( \Gamma \mu = \Gamma \mu - (2(A + B)) \Phi \Phi^* \Gamma \mu \).

Both of these iterative methods have complexity \( O(2TK|E|) \), where the number of iterations \( T \) is typically small (five to 10) and the speed of convergence depends on how close the ratio of the frame bounds \( B/A \) is to one [recall that, when \( \mathcal{V} = \mathcal{V}' \) for all \( j \), the lower frame bound

![Figure S6](https://example-image-url.com)

**Figure S6.** Chebyshev polynomial approximation of three different filter banks on the net25 graph: (a) spectrum-adapted uniform ideal filters [10], (b) spectrum-adapted and shifted uniform ideal filters [10], and (c) spectrum-adapted uniform translates of an Itersine kernel. (d)–(f) The degree 40 Chebyshev polynomial approximations of the filters in (a)–(c), respectively.

However, there are also tradeoffs to using polynomial filters. First, polynomial approximations to filter designs that meet the orthogonality and tight frame criteria may no longer satisfy these conditions. In fact, [8] shows that it is not possible to find \( J \) polynomial filters with the property that \( G(\lambda) \) is constant for all \( \lambda \) in the interval \([0, \lambda_{\text{max}}] \). Although it may be possible to satisfy this condition for all \( \lambda \) in \( \sigma(\mathcal{L}) \), it is not usually tractable to do so.

When using ideal filters, one option to mitigate the approximation error at the Laplacian eigenvalues (recall that the filter values at these eigenvalues are the only filter values that actually affect the form of the dictionary atoms) is to attempt to place the endpoints of the subband filters in areas of the spectrum with a low density (or even better, in spectral gaps), as the error is typically highest near the endpoints [see Figure S6(e) in “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation”] [10].

A second option for mitigating the approximation error is to choose polynomials that control the error in specific parts of the spectrum, such as transformed linear phase multirate filters [8], which reduce the error near the eigenvalue zero (with no dc leakage) and spectrum-adapted polynomial approximation [32], which can reduce the error in high-density areas of the spectrum. A third option is to directly choose the initial set of filters to be polynomials or, at least, smoother functions that are more accurately approximated by polynomials (e.g., [7]).

**The number of filters**

For sets of filters that cover the whole spectrum, typical choices of \( J \) in the literature are in the four-to-eight range; however, we are not aware of a theoretical analysis concerning the choice of \( J \). In general, increasing \( J \) may lead to sparser representations by expanding the number of atoms and the ability to distinguish between signals by capturing the behavior of the signal.
Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation

For this faster synthesis, $A$ is small [S2, Ch. 3]. For this faster synthesis of the polynomial filters, near perfect reconstruction is still possible at the same $O(N)$ complexity, but the inverse transform may require 10–20 times the number of computations as the fast analysis operator.

A third, faster $[O(K|E|)]$ but less accurate option is to just take $f_{m} = \left(2/(A + B)\right)\Phi \alpha$ (i.e., stop the frame inversion algorithm after the initial guess). The intuition is that $(2/(A + B))\Phi \Phi^* \approx$ identity. If $r = B/A - 1$ is small [S2, Ch. 3]. For this faster synthesis operator, the reconstruction error can be upper-bounded by $\|\hat{f} - f_{m}\| \leq \left(2/(A + B)\right)\|\Phi \alpha\| \leq n(2 + r)\|\Phi\|_2$. 

$\Phi$ is close to the identity matrix $I$, if $r = B/A - 1$ is small [S2, Ch. 3]. So, regardless of the choice of fast inverse transform, it is desirable for $\hat{G}(\lambda)$ to be close to one for each $\lambda$, yielding a frame bound ratio $B/A$ that is close to one and a small value of $r$.

In Figure S6, we show degree 40 Chebyshev polynomial approximations to three different sets of five filters on the net25 graph, which features many repeated eigenvalues and therefore has areas of the spectrum that have a high density. The polynomial approximants to the ideal filter bank in Figure S6(a) yield a $\hat{G}(\lambda)$ (the black points in Figure S6(d)) that fluctuates across $\lambda \in \mathcal{L}$, and the ratio of the frame bounds $B/A$ is equal to 2.78. By shifting the filter endpoints slightly to be in regions with a lower spectral density [Figure S6(b)], the frame bound ratio drops to 1.85 [Figure S6(e)]. The smoother spectrum-adapted translates [Figure S6(c)], are more amenable to polynomial approximation; the polynomial filters for this design have a frame bound ratio of 1.16 [Figure S6(f)].

While the Chebyshev polynomial approximations [3], [29] are good general choices, as they are nearly optimal in terms of minimizing the maximum approximation error across the spectrum, they may not be the most desirable in certain situations, such as approximating a series of ideal bandpass filters. This is because the oscillations of the Chebyshev polynomials may lead to more energy at the spectral values farther from the bandpass region. The Jackson–Chebyshev polynomial approximations [30] damp these Gibbs oscillations, resulting in less energy farther from the bandpass region (Figure S7).

Figure S7. Comparison of Chebyshev and Jackson–Chebyshev polynomial approximations of an ideal bandpass filter.

Reference

[S2] I. Daubechies, Ten Lectures on Wavelets, vol. 61. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

Available data

Graph spectral filters can also be adapted to an ensemble of $T$ training signals, $\{y_j\}_{j=1,2,...,T}$, that are exemplary of the data to be analyzed by the dictionary $\mathcal{D}$ when such training signals are available. We briefly review two parametric dictionary learning approaches, both of which set $\mathcal{V}_j = \mathcal{V}$ for all $j$ and adapt the filters $\{\hat{g}_j(.)\}$ to the training data. The first approach, presented in [13] and shown in “Example 3: Spectral Filter Designs Adapted to the Spectral Density of the Graph or the Ensemble Energy Density of the Training Signals,” is to design the filters so that 1) each one captures a roughly equal amount of the energy of the ensemble of training signals and 2) the filters satisfy the tight frame condition $G(\lambda) = 1$ for all $\lambda \in [0, \lambda_{\text{max}}]$. Similar to the spectrum-adapted filter design of [15], this signal-adapted filter design first constructs a set of prototype filters that uniformly covers the spectrum, and then warps the filters.

Whereas the warping function in [15] estimates the cumulative distribution of the Laplacian eigenvalues (3), the warping function in [13] estimates the ensemble energy cumulative spectral density.
which can also be done in a computationally efficient manner [33]. An example where a signal-adapted spectral design is particularly beneficial is in fMRI, where the energy of the data tends to be highly concentrated at the low end of the spectrum of the cerebellum graph, even though there are more eigenvalues at the upper end of the spectrum [13].

A second approach to incorporate the training data, presented in [12], is to force the J spectral filters to be polynomials and, through optimization, find polynomial coefficients that 1) lead to sparse representations of the training data and 2) yield filters that cover the spectrum so that the frame is close to being tight (i.e., the ratio of frame bounds $B/A$ is close to one). Specifically, [12] suggests alternating between the following:

1) a sparse approximation step that fixes the dictionary (i.e., fixes the polynomial filters) and uses an orthogonal matching pursuit (OMP) to find the coefficient matrix $X$ that minimizes $\|Y - \Phi X\|^2_2$ subject to $\|x_i\|_0 \leq K_0$ for all $t$, where the columns of $Y$ are the training signals, $\Phi$ is the current dictionary with normalized atoms, and $K_0$ is a fixed sparsity level

$$P_\mathcal{Y}(z) := \sum_{t = 1}^{\mathcal{K}} \frac{1}{T} \sum_{t = 1}^{N} \left( \frac{y_t \cdot u_t}{\| y_t \|_2} \right)^2,$$

(8)

2) a dictionary update step that fixes that coefficient matrix $X$ and updates the polynomial filter coefficients by minimizing $\|Y - \Phi X\|^2_2 + \mu \sum_{j=1}^{J} \| a_j \|^2_2$, where $a_j$ is a vector of the $K + 1$ polynomial coefficients in (5) for the $j$th filter, subject to constraints ensuring that the learned polynomial filters are nonnegative and cover the whole spectrum $[c - \epsilon \leq \Sigma_{j=1}^{J} \hat{g}(\lambda) \leq c + \epsilon$ for some constants $c$ and $\epsilon$.

Selection of the center vertices

When selecting the center vertices $\mathcal{V}$ for the $j$th filter, four broad options are most commonly used:

1) Take $\mathcal{V}_j = \mathcal{V}$ for every filter (i.e., localize every pattern to every vertex as done in [3]).

2) Select the center vertex sets such that $\sum_{j=1}^{J} |\mathcal{V}_j| = N$ (i.e., the total number of atoms is equal to $N$, which is the number of vertices in the graph), which is typically referred to as critical sampling (e.g., [10]).

3) Do not localize every filter to every vertex, but do not restrict the total number of atoms, resulting in a frame with more than $N$ atoms but fewer than $JN$ atoms.

4) Localize a single filter $\hat{g}_1(\cdot)$ to a strict subset $\mathcal{V}_1 \subset \mathcal{V}$ of vertices to generate an interpolation basis for a subspace of graph signals, as discussed in “Example 1: Variational/Interpolating Splines.”

For option 1, there is no choice to be made regarding the selection of the center vertices; for options 2–4, the specific center vertices for each filter $\hat{g}_j(\cdot)$ must be chosen, and this

![Figure 4](image-url)

FIGURE 4. Six Laplacian eigenvectors of the eastern Massachusetts eight-neighbor graph. Associated eigenvalues: (a) $\lambda = 0.39$. (b) $\lambda = 0.45$. (c) $\lambda = 0.46$. (d) $\lambda = 1.07$. (e) $\lambda = 1.10$. (f) $\lambda = 1.14$. While the first three eigenvectors are more similar in terms of the total variation (captured by the eigenvalue), the directions of the oscillations and regions where the eigenvectors’ energies are concentrated are not necessarily in linear order. For example, both $u_{0.39}$ and $u_{1.07}$ have more of their energy concentrated on the vertices corresponding to Cape Cod.
selection process may also involve deciding how many center vertices to use for each filter. We now outline the main considerations when choosing between these four options and/or selecting the specific center vertices.

Frame bounds and reconstruction error

Recall that when \( \mathcal{V}_j = \mathcal{V} \) for every \( j \), Theorem 4 outlines the computation of the frame bounds and provides a sufficient condition on the filters to yield a (tight) Parseval frame. When \( \mathcal{D} \) is a Parseval frame,

\[
    f = \sum_{j=1}^{J} \sum_{i=1}^{N} \langle f, \varphi_{ij} \rangle \varphi_{ij} = \sum_{j=1}^{J} \hat{g}_j(\mathcal{L}) \alpha_j,
\]

so the graph signal \( f \) can be perfectly reconstructed from the vectors of analysis coefficients, \( \alpha_j = \{\langle f, \varphi_{ij} \rangle\}_{i \in \mathcal{V}_j} \).

When \( \mathcal{V}_j = \mathcal{V} \) for every \( j \) and \( G(\lambda) > 0 \) for all \( \lambda \in \sigma(\mathcal{L}) \), the atoms form a frame but not necessarily a tight one. This is also the case when each filter is not centered to each vertex, as long as the dictionary atoms span the space of signals under consideration (typically \( \mathbb{R}^N \)). “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation” details three options for inverse transforms when the dictionary atoms form a frame but not a tight frame. The least-squares solution via the conjugate gradient and the iterative frame inversion algorithm are accurate but converge slower when the frame is further from being tight (\( BI/A \gg 1 \)). We are not aware of any investigations into how to select the center vertices from general weighted graphs in a manner that explicitly controls the ratio of the frame bounds.

Band-by-band reconstruction and connections to sampling and interpolation theory

When the filters are not localized to every vertex but 1) they do not overlap too much and 2) they evenly cover the whole spectrum \( \sum_{j=1}^{J} \hat{g}_j(\mathcal{L}) = 1 \) for all \( \lambda \), an alternative approach to synthesis is to try to reconstruct the signal from each band separately and add these reconstructions up, similar to (9), except that we replace \( \hat{g}_j(\mathcal{L}) \) with a different synthesis operator for each band. The main idea is that \( f \approx \sum_{j=1}^{J} \hat{g}_j(\mathcal{L}) f \), where each filtered signal \( \hat{g}_j(\mathcal{L}) f \) belongs to the subspace \( \mathcal{U}_j \) spanned by the eigenvectors \( \{u_i\} \) \( i \in \sigma(\mathcal{L}) \neq 0 \). Thus, \( \dim(\mathcal{U}_j) \) provides an estimate for the number of center vertices, \( |\mathcal{V}_j| \), required to recover \( \hat{g}_j(\mathcal{L}) f \) from the analysis coefficients \( \alpha_j \).

The problem of first selecting the center vertices \( \mathcal{V}_j \) and then specifying a method to recover \( \hat{g}_j(\mathcal{L}) f \) from \( \alpha_j \) falls into the broader category of sampling and interpolating graph signals. Generally speaking, algorithm development in this area proceeds as follows:

1) Define a signal model and, if appropriate, a measurement noise model.

2) Specify a reconstruction method that maps a given set of (possibly noisy) sample values of the signal to the entire signal in a way that optimizes a specified error criterion, accounting for the signal model (and the noise model) and the graph structure.

3) Given the signal model and the reconstruction method, select vertices (often constrained to a fixed number of them) on which to sample the graph signal values.

See [34] and [35] for more detailed literature reviews and theoretical results covering the sampling and interpolation of graph signals. The majority of sampling selection methods 1) focus on smooth and low-pass graph signals and 2) require the Laplacian eigenvectors associated with the eigenvalues \( \lambda_i \) for which \( \hat{g}_j(\lambda) \neq 0 \). We focus our review in this section on sampling strategies where at least one of these conditions is not met, as these strategies are particularly relevant for LSGFD design (also see “Example 5: Scalable Sampling Strategies for Localized Spectral Graph Filter Dictionary Design”).

For the special case of ideal filters that partition the spectrum, as shown in “Example 2: Spectral Filter Designs That Use Only the Spectral Range,” “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation,” and Figure 5, the critically sampled sets of center vertices can be constructed to ensure that the signal can be recovered perfectly from the \( N \) analysis coefficients.

Theorem 5 [10, Prop. 2 and Cor. 1]

Let \( \hat{g}_1(), \hat{g}_2(), \ldots, \hat{g}_J() \) be a set of spectral filters with the properties that, for all \( \lambda \in \sigma(\mathcal{L}) \), \( G(\lambda) = 1 \) and \( \hat{g}_j(\lambda) \hat{g}_j(\lambda) = 0 \) for all \( j \neq j' \) (i.e., the filters form a partition of the spectrum \([0,\lambda_{\max}]\) such that each eigenvalue is in the support of exactly one filter). Then, there exists a partition \( \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_J\} \) of \( \mathcal{V} \) with \( |\mathcal{V}_j| = \sum_{\lambda_{j-1} < \lambda \leq \lambda_j} 1 \) such that the resulting dictionary \( \mathcal{D} \) of the form (2) is a basis. Each atom in the basis is orthogonal to all atoms generated from a different filter.

The authors of [10] provide a constructive algorithm for finding the center vertex sets \( \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_J\} \) in Theorem 5, the choice of which is not unique. This requires a full ED and therefore is only applicable to small and medium graphs. Each set \( \mathcal{V}_j \) is a uniqueness set for the subspace \( \mathcal{U}_j \) spanned by the eigenvectors \( \{u_i\} \) \( i \in \sigma(\mathcal{L}) \neq 0 \). That is, any graph signal in \( \mathcal{U}_j \) can be uniquely recovered from its values at the vertices in \( \mathcal{V}_j \). In Figure 5, we display a set of five spectrum-adapted ideal octave-band filters on the bunny graph, and the corresponding partition of the vertices into the uniqueness sets \( \mathcal{V}_1 \) to \( \mathcal{V}_5 \).

Which sampling and interpolation techniques are applicable to smoother approximations to ideal filter banks [e.g., Figure S6(e) in “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation”] for large, sparse graphs, where the computation of the Laplacian eigenvectors is tractable? The fastest methods are random, not necessarily uniform, sampling techniques. For example, [10] and
We compare the sampling and reconstruction of a low-pass and a bandpass filtered signal using 1) greedy eigendecomposition-free (ED-free) sampling [34], 2) uniform random sampling, 3) nonuniform random sampling [10], [36], and 4) signal-adapted nonuniform random sampling [10]. For each polynomial filter type (Figures S8 and S9), we show the random sampling distributions, examples of center vertices selected by these methods, and the normalized mean-square reconstruction error (NMSE) \( \| \zeta - \hat{\zeta} \|_2^2 / \| \zeta \|_2^2 \) between the filtered signal and the reconstruction \( \zeta^* \), computed via (11), averaged across 50 trials of choosing the random center locations (samples). The NMSEs are in Figures S10 and S11.

The greedy ED-free method explicitly prioritizes choosing samples that are not too close to previously chosen ones. The nonuniform random sampling weights are derived from a computationally efficient approximation to the ideal distribution, for which the probability of sampling vertex \( i \) is proportional to \( \| \hat{U}_i \|_2^2 \). This approach has a close connection to leverage score sampling in the statistics and numerical linear algebra literature. The signal-adapted nonuniform random sampling distribution is computed by multiplying the initial nonuniform weight associated with vertex \( i \) by \( \log (1 + \| \hat{L}_i \|_2^2 ) \) and then renormalizing. The intuition is that it is beneficial to take additional

---

(continued)
propose to sample according to a nonuniform distribution with the weight at vertex \( i \) proportional to an approximation of \( |\mathbf{U} R_j|_2^2 \), where \( \mathbf{U}_R_j \) is the submatrix of \( \mathbf{U} \) containing the columns corresponding to the support of the ideal filter \( \hat{g}_j \).

The \( j \)th filtered signal is then reconstructed via the optimization problem

\[
\min_{\mathbf{z}_j} \left\{ \frac{1}{2} \mathbf{z}_j^\top \phi_j(\mathcal{L}) \mathbf{z}_j + \kappa \| \mathbf{M}_j \mathbf{z}_j - \mathbf{a}_j \|_2^2 \right\},
\]

(10)

where \( \mathbf{M}_j \) is a \( |\mathcal{V}_j| \times 1 \) diagonal matrix with the \( j \)th channel sampling weights of \( \mathcal{V}_j \) along the diagonal, \( \mathbf{a}_j \) is a downsampling matrix that maps a full graph signal to a vector of its values on \( \mathcal{V}_j \), and \( \kappa > 0 \) is a parameter to trade off the two optimization objectives: the regularization term \( \frac{1}{2} \mathbf{z}_j^\top \phi_j(\mathcal{L}) \mathbf{z}_j \) in (10) penalizes reconstructions with support outside the desired spectral band, and the data fidelity term \( \kappa \| \mathbf{M}_j \mathbf{z}_j - \mathbf{a}_j \|_2^2 \) penalizes reconstructions that do not match the analysis coefficients (filtered signal values). From the first-order optimality conditions, the solution to (10) is the solution to the linear system of equations

\[
(\kappa \mathbf{M}_j^\top \mathbf{M}_j + \phi_j(\mathcal{L})^\top \mathbf{M}_j^\top)(\mathbf{M}_j \mathbf{z}_j - \mathbf{a}_j) = \kappa \mathbf{M}_j^\top \mathbf{a}_j,
\]

(11)

which can be solved with the preconditioned conjugate gradient method (see [10] for more on the choice of the preconditioner).

For this particular bandpass signal, the regions with the highest energy are around the midsection and tail of the bunny. Indeed, the signal-adapted method leads to more samples in these areas, and, in turn, better reconstruction performance [10]. The average NMSE of the signal-adapted sampling method at the dashed vertical line in Figures S10 and S11 represents a 79%, 78%, and 83% reduction of the errors of the ED-free, nonuniform random sampling and uniform random sampling methods, respectively.

Key insights highlighted by this example include:

1) For the low-pass filter (Figure S10), the results are consistent with the common insight from the graph sampling literature that the scalable greedy methods are slower than random sampling but can yield a better reconstruction performance by forcing the samples to be more spread out.

2) For bandpass and high-pass filters, however, forcing and incentivizing the samples to be more spread out does not necessarily (or even usually) improve the reconstruction performance (Figure S11).

3) When performing random sampling to choose the center vertices, using a nonuniform sampling distribution is more important for bandpass and high-pass filtered signals. For low-pass filtered signals, the nonuniform random sampling distribution is much closer to uniform, as the energy distributions of the eigenvectors at the low end of the spectrum tend to be more evenly spread across the graph than those associated with higher Laplacian eigenvalues.

4) In the classical sampling and interpolation problem, the complete signal is not usually available when selecting the sample locations. However, in the context of subsampled localized spectral graph filter dictionaries, it is often feasible and highly beneficial to adapt the nonuniform random selection of the center vertices to the energy distribution of each filtered signal.

![FIGURE S10. Reconstruction error for the low-pass signal.](image1)

![FIGURE S11. Reconstruction error for the bandpass signal.](image2)
The other more scalable options are efficient greedy methods that do not rely on the Laplacian ED, such as [34] and [37]. For each $j$, the ED-free method of [34] attempts to select center vertices $i$ such that $1)$ $\|T_j g_i\|_1 = \|\phi_{ij}\|_1$ is large and $2)$ the atoms $\phi_{ij}$ do not overlap too much in the vertex domain. The binary search Gershgorin disk alignment method of [37] aims to select vertices in a way that reduces the condition number of the matrix $\Omega_j^{-1} M_j + \phi_j(\mathcal{L})$ in (11) [with $\Omega_j = I_N$ and a regularization term of $\phi_j(\mathcal{L}) = \mathcal{L}$], improving the reconstruction stability. These methods may lead to better reconstruction for low-pass signals, but they are slower than random sampling and are either not applicable to or may lead to a worse reconstruction for bandpass signals (see the “Computational Complexity” section and “Example 5: Scalable Sampling Strategies for Localized Spectral Graph Filter Dictionary Design”).

Allocation of center vertices across the filters

When the sum of the cardinalities $\{1 \, | \, V_j| \}$ is constrained (e.g., to $N$), how should we allocate the center vertices (samples) across the filters? One option is to estimate the spectral cumulative density and match the number of samples to the projected number of eigenvalues in each band. Another option is to adjust the distribution of samples to the signal $f$ by multiplying the initial allocation of samples to $V_j$ by a factor that increases with the amount of energy in the filtered signal $\hat{g}_j(\mathcal{L}) f$ [10].

**Figure 5.** The selection of center vertices for a critically sampled LSGFD with five spectrum-adapted ideal octave-band filters on the bunny graph. Exactly one of the five filters in (a) is localized to each vertex $i$, according to the mapping in (b), yielding a total of $N$ atoms [10].

**Computational complexity**

We briefly analyze the computational complexity implications of the choice of method for selecting the center vertices on large, sparse graphs, which is the most typical and important case for complexity considerations. Throughout this analysis, we assume the filters are degree $K$ polynomials. First, for selecting the center vertices, the costs of the random sampling methods (either signal adapted or not) are negligible if the spectral density of the Laplacian has been estimated [or $O(K |E|)$ if it has not], at least an order of magnitude faster than the $O(K N |E|) = O(N^3)$ complexity of the greedy ED-free method. Second, regardless of the selection of the center vertices, applying the analysis operator (forward transform) has the same complexity $O(K |E|)$ as the case where we localize each filter to every vertex; i.e., there is not a significant computational savings in the analysis step from subsampling the center vertices. Third, as detailed in “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation,” the complexity of the inverse transform is $O(2TK |E|)$ if either the least-squares solution via the conjugate gradient or the iterative frame inversion algorithm is used and $O(K |E|)$ if the inverse is approximated by a constant multiple of the analysis operator.

The band-by-band reconstruction method of (11) has a similar complexity of $O(TJK |E|)$, where $K$ is the degree of the polynomial penalty function $\phi$. These inverse transform complexities do not depend on the method used to select the center vertices or the number of center vertices for each filter. In summary, for large, sparse graphs, the complexities of the design, forward transform, and inverse transform are all no more than linear in the number of vertices, as long as each filter is localized to every vertex or the center vertices are selected through nonuniform random sampling.

**Memory**

The total number of analysis coefficients to store for a graph signal of length $N$ is $\Sigma_{j=1}^J |V_j|$. Thus, complete sampling yields $NJ$ coefficients to store, while critical sampling reduces that number to $N$. Except for special filters, such as the heat kernel, the ED-free method requires storing the entire matrices $\{\hat{g}_j(\mathcal{L})\}$. So, when memory is an important concern and a subsampled dictionary is desired, random sampling should be used to select the center vertex sets.

**Theoretical considerations and metrics**

Whether the dictionary of atoms is critically sampled (a basis for the vector space of signals) or redundant (a frame for the vector space of signals), different mathematical characteristics can be beneficial for various applications. The desirable characteristics of dictionaries may include, for example, the following: 1) The atoms are not too correlated, so as to enhance the discriminatory power of taking inner products between each atom and a target signal.
2) The atoms are jointly localized in the vertex domain and the spectral domain.

3) Classes of signals on the graph (e.g., globally smooth and locally smooth signals) can be represented as sparse combinations of the dictionary atoms.

We briefly describe each of these considerations, in order.

One common metric for capturing the correlation between dictionary atoms is the cumulative coherence [38] of the dictionary \( \mathcal{D} \), which, for a given sparsity level \( k \), is defined as

\[
\mu_1(k) := \max_{\psi \in \mathcal{D}} \max_{\|\theta\|_0 \leq k} \sum_{\psi \in \mathcal{D}} |\langle \psi, \theta \rangle| / (\|\psi\|_2 \|\theta\|_0).
\]

Uncertainty principles for signals on graphs (see [1] for references) characterize the degree to which graph signals can be jointly localized in both the vertex (spatial) domain and the spectral (frequency) domain. Particularly interesting for guiding the selection of the center vertices of LGSFDs are the uncertainty principles developed in [39] and [40]. Let \( \mathbf{y} \in \mathbb{R}^N \) be a spatial filter or window function, i.e., a set of weights assigned to the vertices, with \( 0 \leq y_i \leq 1 \) for all \( i \in \mathcal{V} \) and \( \max_{i \in \mathcal{V}} y_i = 1 \). Typical examples of such spatial filters include 1) setting the weights equal to one for all vertices in a certain region of the graph and zero elsewhere and 2) choosing each \( y_i \) to be a measure of the distance from vertex \( i \) to a fixed center vertex \( n \). Then, for any graph signal \( \mathbf{f} \), the quantity \( \mathbf{m}_i(\mathbf{f}) := \mathbf{f}^T \mathbf{y}^T / \| \mathbf{f} \|_2^2 \), where \( \mathbf{Y} \) is a diagonal matrix with diagonal elements equal to the weights \( \mathbf{y} \), captures the portion of the energy of \( \mathbf{f} \) that is located in the specified region of the graph (i.e., those vertices with high weights \( y_i \)). Similarly, for each filter satisfying \( 0 \leq g_j(\lambda_i) \leq 1 \) for all \( \ell \) and \( \max_{i \in \mathcal{V}} g_j(\lambda_i) = 1 \), the quantity \( \mathbf{c}_{ij}(\mathbf{f}) := \mathbf{f}^T g_j(\mathcal{L}) \mathbf{f} / \| \mathbf{f} \|_2^2 \) captures the portion of the energy of \( \mathbf{f} \) that is located in the region of the spectrum specified by the filter. The uncertainty principles in [39] and [40] characterize and provide algorithms to approximate the sets \( \mathcal{W}_{Y_{ij}} := \{ (\mathbf{m}_i(\mathbf{f}), \mathbf{c}_{ij}(\mathbf{f})) : \| \mathbf{f} \|_2 = 1 \} \subseteq [0, 1]^2 \). For example, for a given filter pattern, such a principle can inform how localized a dictionary atom of the form \( 1 \) can be in the vertex domain (typically around the center vertex \( i \)). Due to the irregularity of general graphs and the possibility of highly localized Laplacian eigenvectors, uncertainty does not always exist, in which case \( \mathcal{W}_{Y_{ij}} \) may be equal to \([0, 1]^2\). Nearly a decade after being listed as an open issue in [2], relatively little progress has been made in developing a mathematical theory of approximation linking structural properties of graph signals and their underlying graphs to the sparsity of the analysis coefficients \( \{ (\mathbf{f}, \varphi_{ij}) \} \) that is analogous to the theory for wavelet transform coefficients in Euclidean domains. In [41], vertex domain dictionary designs are proposed that sparsely represent defined classes of piecewise constant and piecewise smooth graph signals. For the special case of signals on circulant graphs, [9] defines a family of complex exponential polynomial graph signals and designs a class of filters that annihilates graph signals from this class; i.e., \( \langle \mathbf{f}, T_{g_j} \rangle = 0 \) for all \( i \). The authors of [42] define notions of global and local regularity for graph signals and begin to connect the regularities of the signals and the degree of the polynomial filters to the decay of the magnitudes of spectral graph wavelet analysis coefficients.

### Application examples and comparisons via numerical experiments

In this section, we first describe LSGF transform methods for two signal processing tasks—denoising and nonlinear approximation (compression)—and then perform a set of targeted numerical experiments, in an attempt to answer high-level design questions and help focus the community’s research going forward.

#### Denoising

We consider the denoising problem of recovering a graph signal \( \mathbf{f} \) from a noisy observation, \( \mathbf{y} = \mathbf{f} + \xi \), where \( \xi \in \mathbb{R}^N \) is a white Gaussian noise vector whose entries are independent and identically distributed (i.i.d.) normal random variables with a mean of zero and a known variance of \( \sigma^2 \). We use the common wavelet denoising method of performing soft thresholding on the LSGF transform coefficients and then resynthesizing the signal with the inverse transform. Specifically, we take \( \mathbf{f}_{\text{denoised}} = \Phi^{-1}(\tilde{\mathbf{a}}) \), where each soft thresholded coefficient in the vector \( \tilde{\mathbf{a}} \) is set to

\[
\tilde{a}_{ij} = \text{sgn}(\langle \mathbf{y}, \varphi_{ij} \rangle) \cdot \max \{0, |\langle \mathbf{y}, \varphi_{ij} \rangle| - Y_{ij}\}.
\]

(12)

If the dictionary used to transform the noisy signal is a Parseval frame, then \( \Phi^{\ast -1}(\tilde{\mathbf{a}}) = \Phi \tilde{\mathbf{a}} \); otherwise (e.g., not every filter is localized to every vertex or the filters do not satisfy the tight frame condition of Theorem 4, due to polynomial approximation), \( \Phi^{\ast -1} \) can be taken to be any of the three approximate inverse transforms discussed in “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation.” We set the soft thresholds \( Y_{ij} \) in (12) to zero for the scaling functions [atoms generated from filters satisfying \( g_j(0) > 0 \) since these coefficients are not expected to be sparse. For other thresholds, as in [16], to account for the fact that the dictionary atoms have different norms, we use atom-adapted thresholds of the form \( Y_{ij} = Y_j \sigma \| \varphi_{ij} \|_2 \), where \( \{ Y_j \} \) are the \( J \) scalar parameters. The optimal value of each \( Y_j \) is estimated with Stein’s unbiased risk estimator. As detailed in [43], for soft thresholding on a dictionary with a single lowpass filter \( j = 1 \), this amounts to solving

\[
\arg\min_{Y_j} \sum_{i=1}^N \left\{ \min \{ (|\langle \mathbf{y}, \varphi_{ij} \rangle|)^2, Y_j^2 \sigma^2 \| \varphi_{ij} \|_2^2 \} 
\right. 
+ 2 \sigma^2 \| \varphi_{ij} \|_2^2 \left\{ 1_{|\langle \mathbf{y}, \varphi_{ij} \rangle| \leq \sigma} - 1_{|\langle \mathbf{y}, \varphi_{ij} \rangle| > \sigma} \right\} \right\}
\]

(13)

for each \( j = 2, 3, \ldots, J \).
Importantly, the objective in (13) does not depend on the unknown signal \( f \). The thresholds \( \{ T_{ij}\} \) and the objective in (13) depend on the atom norms \( \| \phi_{ij} \|_2 \). For small-to-medium graphs, these can be exactly computed; for polynomial filter designs on large, sparse graphs, they can be efficiently estimated as \( \| \phi_{ij} \|_2 \approx 1/\sigma \) s.d. \((\delta^T \bar{p}_{j,k}(L) \eta_j)_{j=1}^f\), where \( \bar{p}_{j,k} \) is a polynomial approximation to \( g_j \) and \( \{ \eta_j \}_{j=1}^f \) is a sequence of i.i.d. random vectors, each with i.i.d. entries normally distributed with a mean of zero and a variance of \( \sigma^2 \).

In Table 1, we examine the denoising performance for different graph signals, noise levels (\( \sigma_f \) is the standard deviation of the signal values), and filter designs. For each filter design type, we report the best signal-to-noise ratio improvement across the range of three to 12 filters, each localized to every vertex in the graph to create the dictionary. The signal-to-noise ratio improvement is defined as \( \Delta_{\text{SNR}} = 10 \log_{10}(\| f \|_2^2 / \| f_{\text{denoised}} \|_2^2) + 10 \log_{10}(\| f \|_2^2 / \| f_{\text{denoised}} \|_2^2) \), where \( f \) is the original signal vector and \( f_{\text{denoised}} \) is the denoised signal vector. The four test signals are the sparse, piecewise smooth, and piecewise constant signals on the Minnesota road network, and the average of 292 fMRIs on the cerebellum.

### Table 1. Denoising performance for different graph signals, noise levels, and filter designs, shown in the signal-to-noise ratio improvement, \( \Delta_{\text{SNR}} \).

| \( \sigma/\sigma_f \) | Sensor Network | Bunny | Minnesota | Cerebellum |
|----------------------|----------------|-------|-----------|------------|
|                      |                |       |           |            |
| \( \sigma/\sigma_f \) | 1/8            | 1/4   | 1/2       | 1          |
| Spectral graph waves  | 1.56           | 2.29  | 4.04      | 4.99       |
| Spectral graph waves  | 5.92           | 7.69  | 10.46     | 5.81       |
| Spectral graph waves  | 6.69           | 8.5   | 10.93     | 0.53       |
| Spectral graph waves  | 1.74           | 3.76  | 6.85      |            |
| Spectral graph waves  | 1.77           | 2.3   | 3.94      | 6.26       |
| Spectral graph waves  | 4.87           | 5.88  | 7.61      | 10.32      |
| Spectral graph waves  | 6.85           | 7.14  | 12.06     | 0.41       |
| Spectral graph waves  | 1.39           | 2.07  | 3.85      | 6.26       |
| Spectral graph waves  | 4.51           | 5.76  | 7.6       | 10.37      |
| Spectral graph waves  | 6.29           | 7.08  | 9.1       | 12.06      |
| Spectral graph waves  | 0.41           | 1.39  | 3.59      | 6.31       |
| Spectral graph waves  | 1.33           | 2.14  | 3.93      | 6.37       |
| Spectral graph waves  | 4.61           | 5.69  | 7.36      | 10.41      |
| Spectral graph waves  | 6.69           | 7.94  | 9.57      | 0.69       |
| Spectral graph waves  | 2.03           | 4.10  | 7.09      |            |
| Spectral graph waves  | 1.42           | 2.15  | 3.98      | 6.38       |
| Spectral graph waves  | 4.4            | 5.76  | 7.48      | 10.44      |
| Spectral graph waves  | 6.76           | 8.66  | 11.94     | 0.55       |
| Spectral graph waves  | 1.7            | 3.9   | 6.99      |            |
| Spectral graph waves  | 1.48           | 2.11  | 3.97      | 6.27       |
| Spectral graph waves  | 4.35           | 5.46  | 7.14      | 10.32      |
| Spectral graph waves  | 6.23           | 8.48  | 11.22     | 0.68       |
| Spectral graph waves  | 2.05           | 4.05  | 7.03      |            |
| Spectral graph waves  | 0.39           | 1.38  | 3.53      | 6.85       |

### Nonlinear approximation

One approach to the compression of smooth and piecewise smooth graph signals is to represent them as sparse linear combinations of LSGFIs. To find such sparse representations for a graph signal \( f \), the sparse coding optimization \( \min_{\alpha} \| f - \Phi \alpha \|_2^2 \) subject to \( \| \alpha \|_0 \leq T_0 \), where \( T_0 \) is a predefined sparsity level can be approximately solved, e.g., by normalizing the dictionary atoms and then applying the greedy OMP algorithm [38]. Figure 6 displays an example of the results from this approach. When the graph is very large and OMP becomes computationally impractical, a common approximation method is to hard-threshold the analysis coefficients (normalized by an estimate of the corresponding atom norm) and then resynthesize the signal from the \( T_0 \) largest-magnitude coefficients via one of the inverse transform methods described in “Example 4: Fast Transforms and Inverse Transforms via Polynomial Approximation.” In Figure 7, we apply this hard-thresholding method to compress an average temperature signal containing \( N = 469,404 \) values, reconstructing it from \( N/2 \) of the analysis coefficients attained by using two different LSGFIs (one redundant and one critically sampled).
Design considerations revisited

What are the design takeaways from these examples? First, there is no clear “best” filter design method or number of filters across applications, which is not surprising but worth stating. Second, with exact computations, there is not a significant drawback from using a nontight frame; e.g., the spectral graph wavelets have excellent performance in the denoising results in Table 1 and the compression example in Figure 6. Third, using redundant dictionaries generally enables sparser representations of the data and a better compression performance than bases, particularly when using exact computations and OMP, as in Figure 6. However, it appears to be the most beneficial to include fewer center vertices in applications where either 1) memory is a critical issue or 2) the graph signals under consideration reside on very large, sparse graphs, necessitating approximations or less complex reconstruction methods at all stages of the signal processing and machine learning pipelines, as in Figure 7.

Summary and future directions

In summary, localized spectral graph filter frames feature structured atoms with analytically tractable properties, such as localization around the center vertex and spectral patterns that carry a notion of smoothness with respect to the graph. At the same time, efficient numerical approximations exist to the forward and inverse transforms, rendering these dictionaries useful in myriad applications on large, sparse graphs. For small and medium graphs, it is typical to perform an exact ED, localize each filter to every vertex, and choose the filters to satisfy the condition of Theorem 4, guaranteeing that the frame is tight. For large, sparse graphs, we reviewed fast techniques to bound or approximate the maximum graph

FIGURE 6. The nonlinear approximation of a piecewise smooth signal on a random sensor network with 500 vertices. The signal is shown in (a) the vertex domain and (b) the spectral domain. (c) The normalized mean-square reconstruction error (NMSE) incurred by applying the OMP algorithm to the analysis coefficients from nine different dictionary transforms. The first four are bases, with errors shown in dashed lines, and the next five are frames generated from six filters, with all but the spectral graph wavelet frame being Parseval frames. DiffWav: diffusion wavelets; MCSFB: M-channel critically sampled filter bank; SGWT: spectral graph wavelet transform; Unif: uniform; Tr: translates; Hann: Hann kernel; Wav: wavelets.
Laplacian eigenvalue and the cumulative spectral density function and, subsequently, how to leverage these approximations in the design and application of the spectral filters, the selection of center vertices via signal-adapted nonuniform random sampling, and fast reconstruction methods.

Open issues and future directions in this line of research include:

1) The continued investigation of algorithms that use LSGFDs to efficiently extract information in the context of specific application domains and signal processing and machine learning tasks is important; as just one example, [44] begins to investigate how to leverage the structured sparsity of LSGFD analysis coefficients to recover piecewise smooth graph signals in the semisupervised learning problem.

2) Many unanswered or partially answered questions remain regarding connections between theoretical metrics, dictionary design, and applications: What mathematical classes of graph signals are sparsely represented by LSGFDs with specific spectral patterns? Are there easily computable metrics on the dictionary that are good predictors of performance in application tasks, as demonstrated either empirically or via mathematical analysis? In what applications is it most beneficial to adapt the filters to the spectrum or the energy density of training signals?

3) Faster greedy and hybrid greedy/random graph sampling methods that are applicable to both smooth and nonsmooth graph signals as well as accompanying fast, scalable interpolation methods should be developed.

4) The dictionary design principles reviewed here may be beneficial in the settings of data on directed graphs, time series data on graphs, and deep learning/convolutional neural networks on graphs.

Acknowledgments

The author would like to thank the anonymous reviewers and Hamid Behjat for constructive feedback on earlier versions of this article, and Pierre Vandergheynst for introducing him to this area of research.

Author

David I. Shuman (dshuman1@macalester.edu) received his B.A. degree in economics and M.S. degree in engineering-economic systems and operations research from Stanford University,
California, in 2001 and his M.S. degree in electrical engineering systems, M.S. degree in applied mathematics, and Ph.D. degree in electrical engineering systems from the University of Michigan, Ann Arbor, in 2006, 2009, and 2010, respectively. He is currently an associate professor in the Department of Mathematics, Statistics, and Computer Science, Macalester College, St. Paul, Minnesota. From 2010 to 2013, he was a post-doctoral researcher at the Institute of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. His research interests include signal processing on graphs, computational harmonic analysis, and stochastic scheduling and resource allocation problems. He has served as an associate editor of IEEE Transactions on Signal and Information Processing Over Networks (2019–present) and IEEE Signal Processing Letters (2017–2019). He received the 2016 IEEE Signal Processing Magazine Best Paper Award.

References
[1] D. I. Shuman, “Localized spectral graph filter frames: A unifying framework, survey of design considerations, and numerical comparison (extended cut),” 2020, arXiv:2006.11220.
[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83–98, May 2013. doi: 10.1109/MSP.2012.2235192.
[3] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” Appl. Comput. Harmon. Anal., vol. 30, no. 2, pp. 129–150, Mar. 2011. doi: 10.1016/j.acha.2010.04.005.
[4] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” IEEE Trans. Signal Process., vol. 60, no. 6, pp. 2786–2799, June 2012. doi: 10.1109/TSP.2012.2188718.
[5] D. B. Tay and Z. Lin, “Design of near orthogonal graph filter banks,” IEEE Signal Process. Lett., vol. 22, no. 6, pp. 701–704, June 2015. doi: 10.1109/LSP.2014.2368128.
[6] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacević, “Discrete signal processing on graphs: Sampling theory,” IEEE Trans. Signal Process., vol. 63, no. 24, pp. 6510–6523, Dec. 2015. doi: 10.1109/TSP.2015.2469645.
[7] A. Sakiyama, K. Watanabe, and Y. Tanaka, “Spectral graph wavelets and filter banks with low approximation error,” IEEE Trans. Signal Inform. Process. Netw., vol. 2, no. 3, pp. 230–245, 2016. doi: 10.1109/TISPEN.2016.2581303.
[8] D. B. Tay, Y. Tanaka, and A. Sakiyama, “Almost tight spectral graph wavelets with polynomial filters,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 6, pp. 812–824, 2017. doi: 10.1109/JSTSP.2017.2726972.
[9] M. S. Kotzigannidis and P. L. Dragotti, “Splines and wavelets on circular graphs,” Appl. Comput. Harmon. Anal., vol. 47, no. 2, pp. 481–515, Sept. 2019. doi: 10.1016/j.acha.2017.10.002.
[10] S. Li, Y. Jin, and D. I. Shuman, “Scalable M-channel critically sampled filter banks for graph signals,” IEEE Trans. Signal Process., vol. 65, no. 15, pp. 3954–3969, Aug. 2019. doi: 10.1109/TSP.2019.2923342.
[11] I. Pesenson, “Variational splines and Paley-Wiener spaces on combinatorial graphs,” Constr. Approx., vol. 29, pp. 1–21, Feb. 2009. doi: 10.1007/s00365-007-0009-4.
[12] D. Thanou, D. I. Shuman, and P. Frossard, “Learning parametric dictionaries for signals on graphs,” IEEE Trans. Signal Process., vol. 62, no. 15, pp. 3849–3862, Aug. 2014. doi: 10.1109/TSP.2013.2233241.
[13] H. Behjat, U. Richter, D. Van De Ville, and L. Storino, “Signal-adapted tight frames on graphs,” IEEE Trans. Signal Process., vol. 64, no. 22, pp. 6017–6029, Nov. 2016. doi: 10.1109/TSP.2016.2591513.
[14] N. Leonardi and D. Van De Ville, “Tight wavelet frames on multiscale graphs,” IEEE Trans. Signal Process., vol. 61, no. 13, pp. 3357–3367, July 2013. doi: 10.1109/TSP.2013.2392825.
[15] D. I. Shuman, C. Wiesmeyer, N. Holighaus, and P. Vandergheynst, “Spectrum-adapted tight graph wavelet and vertex-frequency frames,” IEEE Trans. Signal Process., vol. 63, no. 16, pp. 4223–4235, Aug. 2015. doi: 10.1109/TSP.2015.2424203.
[16] F. Göbel, G. Blanchard, and U. von Luxburg, “Construction of tight frames on graphs and application to denoising,” in Handbook of Big Data Analytics, W. Hardle, H. Li, and X. Shen, Eds. New York: Springer-Verlag, 2018, pp. 503–522.
[17] B. Dong, “Sparse representation on graphs by tight wavelet frames and applications,” Appl. Comput. Harmon. Anal., vol. 42, no. 3, pp. 452–479, 2017. doi: 10.1016/j.acha.2015.09.005.