



HAL
open science

What's in a frequency: new tools for graph Fourier Transform visualization

Benjamin Girault, Antonio Ortega

► To cite this version:

Benjamin Girault, Antonio Ortega. What's in a frequency: new tools for graph Fourier Transform visualization. 2019 Information Theory and Applications Workshop, Feb 2019, San Diego, California, United States. hal-02074862

HAL Id: hal-02074862

<https://hal.science/hal-02074862>

Submitted on 21 Mar 2019

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

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

WHAT'S IN A FREQUENCY: NEW TOOLS FOR GRAPH FOURIER TRANSFORM VISUALIZATION

Benjamin Girault and Antonio Ortega

Signal and Image Processing Institute
Department of Electrical and Computer Engineering,
University of Southern California, Los Angeles, USA

ABSTRACT

Recent progress in graph signal processing (GSP) has addressed a number of problems, including sampling and filtering. Proposed methods have focused on generic graphs and defined signals with certain characteristics, *e.g.*, bandlimited signals, based on the graph Fourier transform (GFT). However, the effect of GFT properties (*e.g.*, vertex localization) on the behavior of such methods is not as well understood. In this paper, we propose novel GFT visualization tools and provide some examples to illustrate certain GFT properties and their impact on sampling or wavelet transforms.

Index Terms— Graph signal processing, graph Fourier transform, visualization, filtering, sampling.

1. INTRODUCTION

Significant recent progress has been made in developing graph signal processing (GSP) methods for a number of basic problems such as graph signal filtering or sampling [1, 2]. Most of these approaches start by selecting a graph operator, typically the graph Laplacian or the adjacency matrix, and using the eigenvectors of this operator as elementary graph signal frequencies. In addition to the Laplacian and adjacency matrices, which were already studied in the context of spectral graph theory [3], some new graph operators have been proposed recently. Examples include operators designed to address graph signal stationarity [4] or graph irregularity [5], or leading to orthogonal basis for directed graphs [6].

Letting the variable λ denote the graph frequency. A given graph with n nodes will have n , not necessarily distinct, discrete frequencies $(\lambda_0, \lambda_1, \dots, \lambda_{n-1})$, each corresponding to an eigenvalue of its corresponding graph Laplacian, assuming a GFT based on the graph Laplacian. Then, \mathbf{U}^T , the transpose matrix of orthogonal eigenvectors¹, is usually called the graph Fourier transform (GFT) [1, 2].

The key observation in our work is that once an operator has been defined, *e.g.*, the combinatorial graph Laplacian for undirected graphs, much of the recent literature has focused on developing methods that make use of that operator and are *generic*, in the sense of being suitable for any graph within a certain class (*e.g.*, any undirected graph). Thus, basic GSP techniques are often designed so that exact knowledge of the distribution of eigenvalues and exact values of the eigenvectors are not required. For example, a popular design of wavelet transforms for graph signals [7] is based on defining a bandpass filter $g(\lambda)$. Exact knowledge of the eigenstructure of the graph is not required as long as $g(\lambda)$ can be approximated by a polynomial. As another example, most methods proposed to date for graph signal sampling (*e.g.*, [8, 9]) set up the problem by

selecting a sampling rate (a fraction of graph nodes to be observed) and then optimizing node selection. These decisions are made under the assumption that graph signals will be smooth in some sense, *e.g.*, bandlimited. The choice of sampling rate a priori does not require knowledge of the eigenstructure of the graph. While this leads to a mathematically meaningful definition of which signals can be reconstructed (*e.g.*, bandlimited signals) other characteristics of these signals (such as their localization) are often unclear.

Clearly, it is desirable to design algorithms, such as those discussed above, that can be applied to generic graphs. However, there is an inherent risk in not having sufficient knowledge about the frequency characteristics of the graph. In particular, working with signal models that can be interpreted and related to actual observed behavior becomes essential when considering specific application domains. In this paper we argue that current methods fall short in this regard and that not enough attention has been devoted to developing a more specific understanding of graph signal frequencies. For example, a given $g(\lambda)$ may have a pass band containing a relatively small number of discrete frequencies. Similarly, a given sampling rate may partition the range of frequencies in such a way that some subsets of nodes may be “over-represented” due to the localization properties of graph frequencies. These and similar issues arise due to the irregular nature of graph topologies, which in turn leads to irregular distribution of discrete frequencies on the real line, and localization (and even compact support) of some of the elementary frequencies. This is in contrast with elementary frequencies in conventional signal processing, which are evenly spaced and are not localized in time (or space).

Our main goals in this paper are twofold. First, we introduce a series of novel graph frequency visualization tools, which are also being made available online². For relatively small size graphs these tools make it possible, for the first time, to observe specific characteristics of all graph frequencies, including their spread and vertex domain localization properties. While our proposed tools require computing the GFT, which may not be practical in some cases, and visualization may be difficult for large graphs, these ideas can be the basis to develop suitable techniques for larger graphs. Second, by presenting several examples of GFTs plotted using our tools, we show some key differences between GFTs that our visualization techniques can highlight, including localization of GFT eigenvectors, sampling, and localization of SGWT atoms, as a few examples.

Our work was initially inspired by techniques used to represent eigenstates in dynamical systems (*e.g.*, [10]). While the importance of eigenvector (graph frequency) structure in graph signal processing is undeniable, surprisingly little effort has been devoted to visualization. One exception is [11], which uses visualization examples to

¹Unless otherwise stated we will only consider undirected graphs here.

²<https://github.com/STAC-USC/GrasP/>

make the case that ordering frequencies on the eigenvalues of the Laplacian may not be adequate. Our approach also suggests that a better understanding of the implications of such an ordering is required. Unlike [11] our focus is primarily visualization, with the goal of providing insights about existing methods.

The rest of this paper is organized as follows. In Section 2 we introduce basic notation and motivate the need for visualization tools. In Section 3, we present our proposed visualization tools and provide simple examples comparing results for different graph operators. In Section 4, we show experimental results illustrating the irregular behavior of frequencies in an example graph. These results suggest that a more detailed understanding of graph frequencies is important, before GSP methods can be applied in specific applications. Finally we conclude this paper in Section 5.

2. MOTIVATION

2.1. Basic definitions

We consider weighted undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where \mathcal{V} is the vertex set and \mathcal{E} is the edge set of the graph. The entry $w_{i,j} \geq 0$ in the weight matrix \mathbf{W} represents the weight of edge $(i, j) \in \mathcal{E}$, and $w_{i,j} = 0$ if $(i, j) \notin \mathcal{E}$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where \mathbf{D} is the diagonal degree matrix with $d_{i,i} = \sum_{j=1}^n w_{i,j}$. Based on the definition of Laplacian matrix, for a given graph signal $\mathbf{x} \in \mathbb{R}^n$, the *Laplacian quadratic form*

$$\mathbf{x}^\top \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} (x_i - x_j)^2 \quad (1)$$

measures the variation of \mathbf{x} on the graph.

The graph Fourier transform (GFT) is an important tool in graph signal processing [1, 12, 2]. One possible definition of the GFT would be to choose \mathbf{U} the matrix of eigenvectors of the graph Laplacian: $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$. Based on this definition, the GFT basis functions $\mathbf{u}_0, \dots, \mathbf{u}_{n-1}$ are mutually orthogonal unit norm vectors that correspond to the smallest to the largest variations on the graph:

$$\mathbf{u}_0 = \underset{\|\mathbf{f}\|=1}{\operatorname{argmin}} \mathbf{f}^\top \mathbf{L} \mathbf{f}, \quad \mathbf{u}_k = \underset{\mathbf{f} \perp \mathbf{u}_0, \dots, \mathbf{u}_{k-1}, \|\mathbf{f}\|=1}{\operatorname{argmin}} \mathbf{f}^\top \mathbf{L} \mathbf{f}.$$

Each of these basis vectors corresponds to a graph frequency, *i.e.*, the corresponding eigenvalue $\lambda_0, \lambda_1, \dots, \lambda_{n-1}$, where $\lambda_0 = 0$ corresponds to the constant eigenvector, $\mathbf{u}_0 = n^{-1/2} \mathbf{1}$, similar to the DC frequency for conventional signal processing.

2.2. Filtering

A significant amount of research effort has been devoted to developing transformations for graph signals. As a representative example, consider spectral graph wavelet transforms (SGWTs) [7], one of the earliest and most widely used approaches. This approach is based on designing a bandpass prototype $g(\lambda)$ (with $g(0) = 0$) and a scaling function $h(\lambda)$ (with $h(0) \neq 0$). Multiple scaled versions of $g(\lambda)$, $g_t(\lambda) = g(t\lambda)$ combined with $h(\lambda)$ can be shown to form a frame, whose multi-resolution properties mimic those of wavelet transforms for conventional signals. SGWTs can be implemented in the frequency domain, by computing $\tilde{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$, the GFT of a signal \mathbf{x} , and then scaling each of the frequencies by the corresponding filter weights $f(\lambda_i)$ before computing the inverse GFT. In order to avoid computing the GFT, which could be costly for large graphs, it is possible to compute the SGWT in the vertex domain by using filters $g(\lambda)$ that can be approximated by polynomials of λ . If the

filters are polynomial the transformation can be applied directly in the vertex domain.

Thus, assuming a polynomial design, one can construct a complete SGWT without having any information about the eigenvalues and eigenvectors of the graph. While not having to compute the GFT is compelling in terms of complexity, we will show that this could lead to undesirable results. For example, assume that for a specific graph and graph signal we observe that a certain band, corresponding to a scale t , $g_t(\lambda)$ has low energy. This could be due to the signal not having energy in those specific frequencies, or it may be due to that particular passband not containing many discrete frequencies to begin with.

2.3. Sampling

As a second example, consider graph signal sampling [8, 9]. Some approaches require computing the GFT (*e.g.*, [8]), while for others (*e.g.*, [9]) this is not needed. In either case, we face the question of deciding what the sampling rate should be, *i.e.*, if the graph has n nodes we need to choose $k < n$, the number of nodes to be sampled. This question is more straightforward for conventional signal sampling. k allows us to quantify the frequency content of signals of interest. Since frequencies are equally spaced and all basis vectors are global (*e.g.*, the DFT), it is easy to interpret the effect of increasing the number of samples. In particular, since frequencies are equally spaced, increasing the number of samples by one always increases the maximum signal frequency by the same amount (*i.e.*, $1/N$ -th of the maximum frequency, if the signal has length N).

This is not as straightforward for graph signals. Given that frequencies are no longer evenly spaced, increasing the size of the sampling set by 1 will increase the maximum signal frequency of a signal that can be reconstructed, but this increase will not be the same for each additional sampled node. As an example, consider two cases where k_1 and k_2 samples, respectively, have already been selected for sampling and we would like to choose an additional one. Then, the respective increases in the size of the signal frequency band would respectively be $\lambda_{k_1+1} - \lambda_{k_1}$ and $\lambda_{k_2+1} - \lambda_{k_2}$, which are not guaranteed to be equal. In fact, in some cases, if an eigenvalue has multiplicity greater than one, say $\lambda_{k_1+1} = \lambda_{k_1}$ there will be no increase in the size of the signal band as the size of the sampling set is increased. Moreover, since elementary graph frequencies can be localized in the vertex domain, the choice of sampling rate can also lead to localized sampling sets: for example, if the graph nodes can be clustered, many nodes in one cluster could be added to the sampling set, while only few will be added from another cluster.

3. PROPOSED VISUALIZATION TOOLS

In this section, we present our proposed visualization technique, and discuss the design choices to be made. To give a concrete example, we use the unweighted graph of Figure 1 and represent the GFT of that graph based on its combinatorial Laplacian. Focusing on the problem of displaying the GFT basis vectors, *i.e.*, the columns of \mathbf{U} , we provide an overview of the visualization challenges and how we tackle them. Note that each of these vectors is a graph signal by itself, *i.e.*, it associates a scalar value to each of the nodes.

3.1. Visualization challenges and design choices

Embedding Note that there is no “natural” way to plot a graph signal. Some graphs can be plotted in 2D space, *i.e.*, each node is given

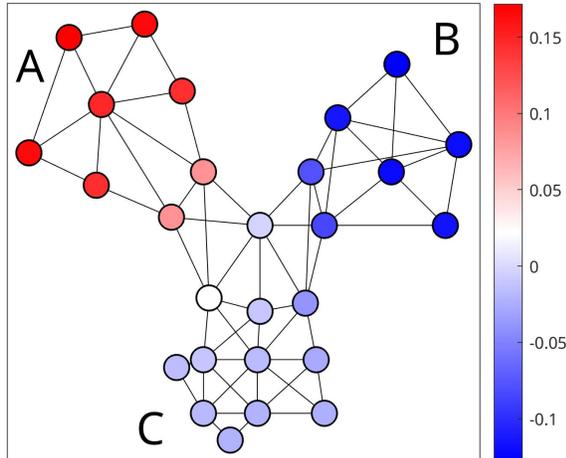


Fig. 1. Unweighted toy graph example, with 1D embedding.

horizontal and vertical coordinates, and the graph signal can be represented as a node attribute (see Figure 1). However, nodes are not necessarily points in a metric space, and if they are, the dimension of the space could be high. In order to achieve a compact representation, here we choose to map all graph nodes onto the real line via an *embedding*. Clearly, information will be lost in the embedding, but since our goal is to compare the GFT basis vectors, using the same embedding for all of them will provide an easy and consistent way to visualize them. Note that, although the embedding may not be that robust in terms of ordering individual nodes, it will generally preserve clusters. It is the global structure associated to clusters that matters most to understand graph frequencies and the GFT.

Frequency spacing Graph frequencies (the λ_k 's) are not equally spaced. We decided to display each of the GFT basis vectors as a signal on the real line, and to stack them all vertically. These are the horizontal signals plotted on Figure 2. When doing so one option is to make the vertical coordinate of each component \mathbf{u}_k a function of the corresponding λ_k . This proved not to be practical for graphs with very irregular frequency spacing, or when there are numerous eigenvalues with multiplicity greater than one, since basis vectors corresponding to similar (or identical) frequencies would be hard to distinguish (see for example Figure 6(b)). Instead, we chose to use the index of the component (k for a given \mathbf{u}_k) to space them (*i.e.*, they are equispaced). Then, in order to convey the irregular spacing of eigenvalues we plot them along the vertical right axis in their original spacing and plot a line linking each component to the corresponding graph frequency. Notice that in the case of a more general graph signal linear transform, one has to associate a graph frequency to each of the transform components. We will give an example of such an association in the case of the SGWT in subsection 4.3.

Magnitude We had to decide on how to convey the magnitude of the value associated to each node for a given GFT basis vector. That is, denoting \mathbf{u}_k the k -th eigenvector, we are interested in plotting its i^{th} entry $[\mathbf{u}_k]_i$ in the position of the i^{th} node in our embedding. To do this, we first considered plotting this information as a signal with each sample connected to the next using a black line. Unfortunately, these black lines were too distracting, since many of the magnitudes

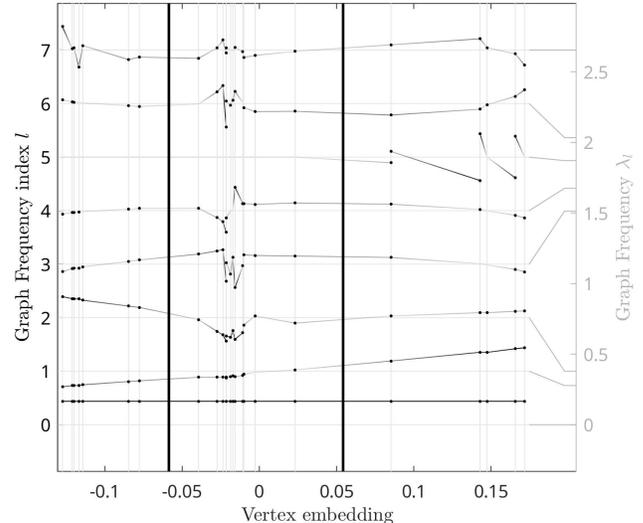


Fig. 2. Partial combinatorial Laplacian GFT of the unweighted toy graph in Figure 1. Only the first 8 GFT basis vectors are shown here: $\{\mathbf{u}_0, \dots, \mathbf{u}_7\}$, with 'amplitude_scale' set to 1 (no overlap between GFT basis vectors).

are actually very close to zero (see for example \mathbf{u}_5 on Figure 2). To better see the support without the distracting influence of nodes with zero magnitude, we use a shade of gray that corresponds to the magnitude: highest magnitude is black, zero is a light gray, and lines between two consecutive nodes in the embedding have a shade of gray that is interpolated between the magnitudes (see Figure 2 for an illustration). Moreover, since all zero or close to zero entries are difficult to identify, we implemented two additional features to better highlight the support of the signals. The most noticeable one in Figure 2 are the dots. These are added each time the magnitude is greater than some threshold parameter (see subsection 3.3), hence showing the support of the signals. The second feature is the vertical light gray bars: these show the embedding. In addition, these vertical bars can be used to spot the entries that are not in the support of a GFT basis vector: a lack of dot around the intersection the vertical bar and horizontal line of a GFT basis vector corresponds to approximately zero magnitude.

Additional information for visualization Along with the basic information described in the previous paragraph, we decided to add visual elements to make explicit the presence of structure in the graph. In particular, we included an option to plot cluster boundaries, which are shown as thick black vertical bars in Figure 2. There are additional visualization features that will be discussed in the following sections. Two of these are of particular interest and are shown in Figure 3: spectral bands represented by colored regions of the plot, and highlighted entries using red circles (useful in the context of sampling). We next illustrate these choices with an example, followed by details about our implementation in Section 3.3.

3.2. Illustration on a Toy Graph

The function `grasp_show_transform` of the GraSP toolbox [13] implements our proposed visualization technique³. In Figure 3

³Available since version 1.2.0.

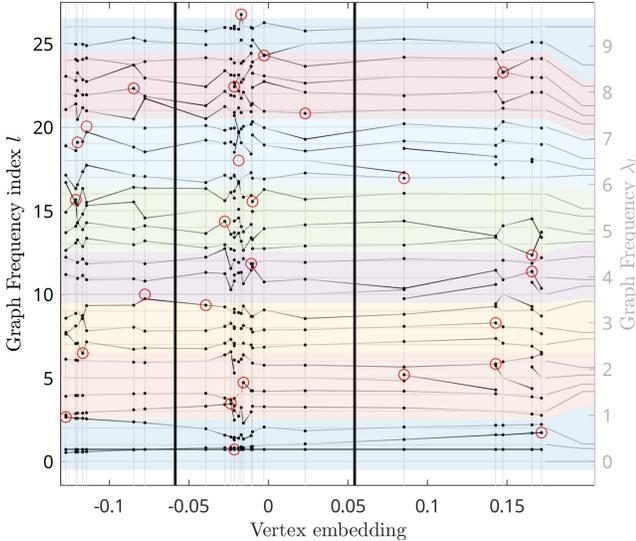


Fig. 3. Combinatorial Laplacian GFT visualization using the graph of Figure 1. The visualization includes regularly spaced spectral bands from the lowest graph frequency (0) and the maximum. The red circles correspond to a sampling using [9] (cf. subsection 4.2 for details).

we show the visualization obtained for the graph Fourier transform based on the combinatorial Laplacian of the unweighted graph shown in Figure 1, and using the code in Figure 4. Additionally, we show in Figure 2 a subset of the GFT basis vectors to better illustrate key properties of the embedding and representation of the magnitudes.

Several features are of importance. First and foremost, we can see the stacked GFT basis vectors, appearing as irregularly sampled time series thanks to the 1D embedding of nodes. Note the light gray horizontal lines that depict for each of the basis vectors the value 0, and the light gray vertical lines that identify each vertex embedding. These pseudo-time series continue then to the right of the figure until they are mapped on the right axis to the associated graph frequency. We notice then the importance of this mapping due to the irregular spacing of graph frequencies.

In Figure 3, the frequencies are then grouped into equal size bands using colored backgrounds, leading to colored groups of GFT basis vectors. This shows an effect of the irregular nature of graph frequencies: even though the bands are of equal size, they do not correspond the same number of GFT basis vector.

Next, the approximate support of the GFT basis vector is depicted using black dots: each time an entry has a large enough magnitude a dot is added. Note that some of the GFT basis vector have a very small support (e.g. \mathbf{u}_5). This is one property that makes GFT basis vectors very different from Fourier modes in conventional signal processing. Making it possible to visualize this behavior is one of the main motivations for our work.

Finally, the clusters are separated by thick black vertical lines, and some of the entries of the GFT matrix are highlighted using red circles in Figure 3. These highlighted entries can be used to represent a specific sampling algorithm. Assuming that a sampling algorithm selects one node to sample at each iteration, we represent the first chosen node by highlighting the corresponding entry in the first GFT basis vector, the second selected sample is shown in the second GFT basis vector, and so on. Each additional node sampled allows for an increase in bandwidth of the graph signal we can reconstruct.

3.3. Parameters of the GraSP implementation

The GraSP toolbox implements our proposed visualization technique in the function `grasp_show_transform`. This section provides a description of visualization parameters and serves as a reference of the implementation in version 1.2.0 of GraSP.

Embedding By default, `grasp_show_transform` uses the second eigenvector of the random walk Laplacian, a typical choice in the literature [14], to generate a 1D embedding of the vertices. To override that choice, the parameter `'embedding'` can be used, and can be set to either i) 0 (default) for an embedding equal to the second eigenvector of the random walk Laplacian, ii) 1 for a regular (equispaced) embedding based on the second eigenvector of the random walk Laplacian, and iii) some n dimensional vector for an arbitrary embedding.

Clusters The parameter `'clusters'` can be set as a vector giving the cluster id of each vertex. In that case, the resulting figure will show thick vertical bars (see Figure 3) between clusters. Alternatively, a single integer value k will trigger the computation of k clusters using an the embedding in k D based on the random walk Laplacian eigenvectors.

Magnitude Normalization If the magnitude of the GFT basis vectors is not constrained, the visualization can quickly become hard to read with too much overlap between stacked GFT basis vectors (magnitudes too large), or details too small to read (magnitudes too small). To avoid this problem, we implemented three normalization schemes for the GFT basis vectors to plot. The parameter `'amplitude_normalization'`⁴ takes three possible values: i) `'l2'` to have basis vectors of unit norm, $\ell_2(\mathbf{u}_k) = 1, \forall k$, ii) `'max_abs'` (default) to have basis vectors of maximum unit amplitude, $\ell_\infty(\mathbf{u}_k) = 1, \forall k$, and iii) `'overall_max_abs'` to have the maximum amplitude over all modes equal to 1, $\max_k \ell_\infty(\mathbf{u}_k) = 1$, (same normalization factor for all modes). This constrains the magnitude to a predefined range, and the various options above control the relative amplitude normalization between GFT basis vectors. Once the amplitudes are normalized, the value of the parameter `'amplitude_scale'` controls the overlap between the GFT basis vectors: a value 1 means no overlap, and 1.5 a 50% overlap (anything smaller 1 corresponds then to a gap between GFT basis vectors).

Support Visualization Often the support of a GFT basis vector spans the entire vertex set. However, in some cases the magnitude at a specific node may be so small that considering the node to be in the support may be undesirable. To account for this, we propose the parameter `'epsilon_support'` as a threshold on the magnitude to decide whether a given vertex is in the approximate support of the mode. This is based on the values after normalization (see `'amplitude_normalization'` above), and by default the threshold is 0.05 (e.g. 5% of the the maximum magnitude of the mode for `'max_abs'`). Moreover, the parameter `'support_scatter_size'` controls the size of the dots shown for the vertices in the support of the modes (default value: 36).

⁴Amplitude refers here to the maximum magnitude over vertices of a mode.

```

grasp_start_opt_3rd_party('usc_graphs');
load('toy_graph.mat');
toy_graph = grasp_eigendecomposition(toy_graph);
sampling_sets = gsp_sampling(toy_graph);
5 clusters = [ones(12, 1) ; 2 * ones(8, 1) ; 3 * ones(7, 1)];
grasp_show_transform(gcf, toy_graph, ...
    'clusters', clusters, ...
    'embedding', 0, ...
    'bands', max(toy_graph.eigvals) / 8 * [(0:7)' (1:8)'], ...
10 'highlight_entries', sampling_sets);

```

Fig. 4. Matlab code to generate Figure 3 using GraSP version 1.2.0. `embedding` is set to 0 to automatically compute the embedding from the second eigenvector of the random walk Laplacian. `gsp_sampling` is a generic function that returns a matrix \mathbf{S} such that $[\mathbf{S}]_{li}$ is one whenever i is the l^{th} sampled vertex (and 0 otherwise). See subsection 4.2 for details.

Bands A bandpass filter is a filter that attenuates the graph signal spectral components outside of its band. Given such a filter, in order to visualize its effect on the GFT basis vectors, we added the possibility to highlight a band, *i.e.*, a given interval of frequencies. The parameter `'bands'` can be set to a matrix with 2 columns, one for the beginning of the band, and one with the end of the band, for as many bands as rows in the matrix. Each band is then colored similarly to what we observe on Figure 3 (shown here with 8 bands of equal size and spanning the entire spectrum). Additionally, colors can be set using the parameter `'bands_colors'` and providing a matrix with 3 columns (RGB channels) and as many rows as there are bands to plot.

Vertical Alignment of GFT Basis Vectors As described earlier, we decided to represent each of the GFT basis vectors equispaced along the vertical direction. However, we also implemented the scheme where each mode is aligned vertically with its graph frequency, for reference. This is set using the parameter `'graph_signal_y_scheme'` set to `'freq'` (this parameter default value is `'regular'` for equispaced modes). This can serve as a reference and may also be useful in instances where information about specific basis vectors is not as important and thus overlap could be acceptable.

Alternative Transform `grasp_show_transform` displays by default the GFT of the graph given as argument. To visualize another linear transform given by a matrix \mathbf{H} of size $M \times N$, one can set the parameter `'transform_matrix'` to \mathbf{H} . The columns of \mathbf{H}^* are then used instead of the GFT basis vectors. We will call these columns the *atoms* of the linear transform⁵. Note that \mathbf{H} is effectively an *analysis* transform. For example, in the case of the SGWT discussed in subsection 4.3, the transform is defined as the orthogonal projection on the set of localized and scaled wavelet atoms. This defines a transform matrix that given an input graph signal \mathbf{x} , outputs the set of wavelet coefficients $\mathbf{H}\mathbf{x}$.

Custom Frequencies When providing a custom graph transform matrix using `'transform_matrix'`, the atoms of the transform are not naturally associated with a set of graph frequencies, since the atoms of the transform may not overlap with the GFT basis vectors. For example, the SGWT uses bandpass filters, and we may be interested in mapping each SGWT atom to some graph frequency that

⁵Note that in case of using an inner product for graph signals $\mathbf{Q} \neq \mathbf{I}$ (*i.e.*, not the dot product), then the atoms are given by the columns of the matrix $\mathbf{Q}^{-1}\mathbf{H}^*$. See [5] for details.

depends on the band of the filter. For this purpose, the parameter `'graph_frequencies'` can be set to a vector of frequencies, with as many entries as there are distinct frequencies. In subsection 4.3, we use the frequency of maximum response of the bandpass filter to map each atom to a graph frequency.

Sampling and Highlighted Entries In the context of sampling on graphs, one can be interested in understanding the relation between the spectrum of a graph and the nodes sampled (see subsection 4.2 for details). To achieve this we created an option to highlight some of the entries of the transform matrix. The parameter `'highlight_entries'` can be set to a matrix of 0/1 values of the same size as the transform matrix, where the (l, i) entry is one when the corresponding entry in the transform matrix needs to be highlighted (red circles on Figure 3). In the case of the GFT, this would highlight the entry $[\mathbf{u}_l]_i$.

Verbosity Similar to several other functions in the GraSP toolbox, verbosity of the function `grasp_show_transform` can be set using the parameter `'verbose'`. By default it is set to `true` as it prints important information.

Future Features GraSP being an actively developed toolbox, its features can change overtime. Documentation is however provided for each function, and we encourage the interested reader to look at the source code for documentation of the various parameters of any function in the toolbox, including `grasp_show_transform`.

4. EXPERIMENTAL RESULTS

In this section, we use again our toy graph, but this time we are giving weights to the edges. More precisely, this graph consists of nodes in 2D and we choose the edge weights using a Gaussian kernel:

$$\mathbf{W}_{ij} = \exp\left(-\frac{d(i, j)^2}{2\sigma^2}\right),$$

where we chose $\sigma = 1.5$ and $d(i, j)$ is the Euclidean distance between i and j . This setting allows us to highlight important properties we wish to study.

Before delving into the visualization, notice how the nodes are grouped into three natural clusters in Figure 6(a): the cluster A with an embedding with large values is mapped to right hand side of Figure 5, the cluster C with an embedding with the smallest values is mapped to the left hand side of Figure 5, and finally the cluster B is mapped to middle section of Figure 5.

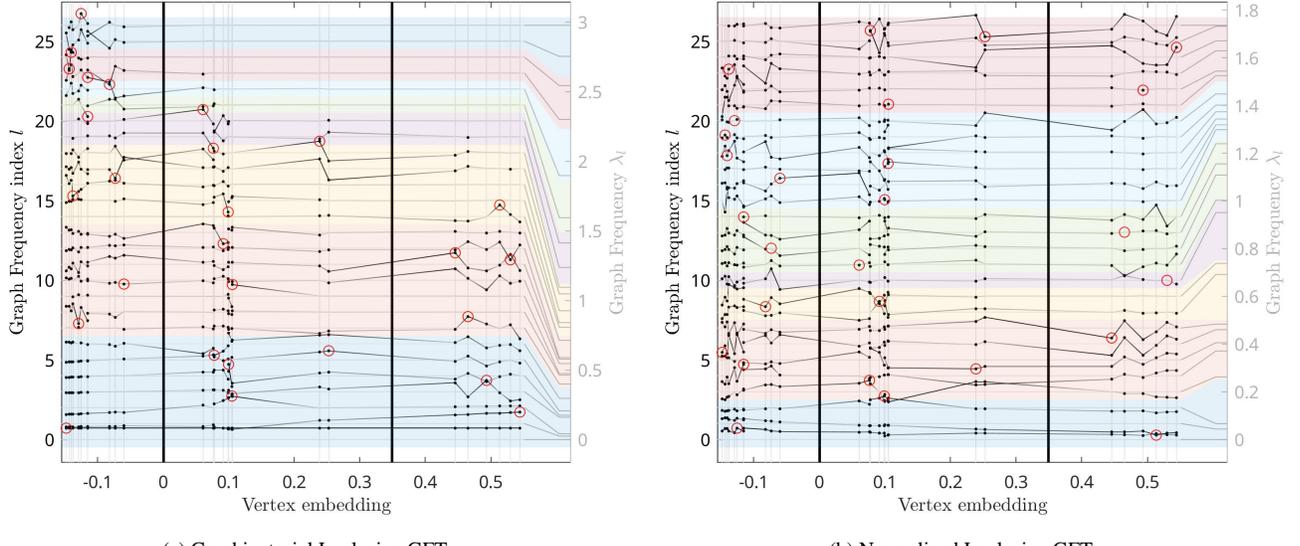


Fig. 5. GFT visualization for (a) the combinatorial Laplacian approach and (b) the normalized Laplacian approach.

4.1. GFT

For this first experiment, we want to highlight a key difference between the classical Fourier transform and the GFT: the GFT basis vectors can be exactly (or approximately) compactly supported. Indeed, as the literature shows [15, 16, 17, 18, 19], uncertainty principles for GFT basis vectors allow them to be localized in the vertex domain.

The GFT based on the combinatorial Laplacian for the weighted toy graph shows such a property. In Figure 5(a), cluster A is not in the support of GFT basis vectors associated to the highest frequencies in the spectrum. In other words, with such a GFT, an ideal high pass filter that filters out any frequency below 1.5 (purple band and below) is actually removing all energy that the signal has on any of the cluster A nodes. Therefore, such a high pass filter is mostly useful to describe high variations on cluster C (a couple of nodes in cluster B have some energy in the support).

This example also shows that a signal that has energy that is completely localized in cluster A cannot exhibit frequencies greater than a certain threshold. This does not mean that the signal is bandlimited in the conventional sense, it just means that signals with energy only in that cluster can only achieve a certain maximum frequency. Here, Figure 5(a) shows that the spectral band of such a signal is actually smaller than that of a signal localized in cluster C. In general, this case shows that if we have $\hat{\mathbf{x}}(k) = \langle \mathbf{x}, \mathbf{u}_k \rangle \simeq 0$, this could correspond to two distinct cases: either \mathbf{x} is orthogonal to \mathbf{u}_k , or \mathbf{x} is localized on some vertices that are not in the support of \mathbf{u}_k . Strictly speaking, of course, both are instances of orthogonality, but in the latter case we get information about localization: no signal compactly supported in that region would have any energy along that basis vector.

On the other hand, for this example, the normalized Laplacian approach in Figure 5(b) does not show such behavior, and most of the GFT basis vectors have support across most of the nodes. Note that this simple experiment cannot be used to generalize. A more thorough study needs to be performed to draw any conclusion on the topic of combinatorial versus normalized Laplacian with respect to GFT basis vector support.

4.2. Sampling

Our visualization technique can also be leveraged to understand sampling schemes. Typical algorithms for sampling set selection choose k out of the n nodes in order to optimize interpolation to the n nodes from data observed on the chosen k . In this formulation, k can be viewed as the largest frequency index for a bandlimited signal that could be reconstructed from k samples. Assuming a greedy procedure such as [9] is used, we can use the parameter ‘highlight_entries’ that highlights $[\mathbf{u}_k]_i$ if the $(k + 1)^{\text{th}}$ sampled node is i , for all $k \in \{0, \dots, n - 1\}$.

We implemented this using the sampling method in [20, 9] and show the results in Figure 5. These results show that the difference in sampling behavior depending on which Laplacian is selected. In particular, when the combinatorial Laplacian is used and the graphs have clusters with different densities, we can see that the samples are not uniformly distributed across the clusters. Indeed we can see that the nodes on the left of the GFT plot (corresponding to the denser cluster C in the graph), continue to be sampled even after the other two clusters have been completely sampled. Since edge weights are a function of distance between node, and we expect better interpolation if nodes are close, this is a sensible result: data from nodes in the dense cluster is easier to interpolate, and thus nodes in that cluster are given lower priority for sampling.

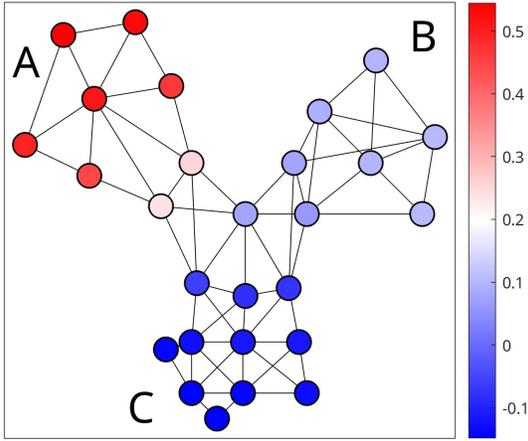
4.3. SGWT

In this experiment, we are interested in studying the support of the SGWT atoms across scales and nodes in the graph. More precisely SGWT atoms are formally defined as the impulse response of band-pass filters [7]:

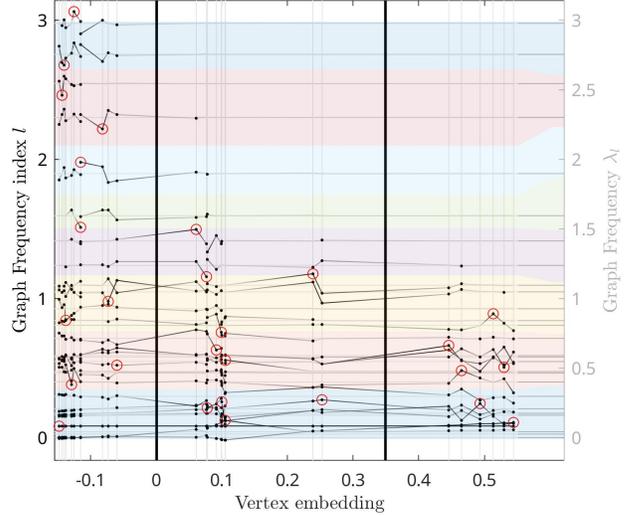
$$\mathbf{g}_{m,i} = \mathbf{G}_m \delta_i,$$

where $\mathbf{G}_m = g_m(\mathbf{L}) = g(t_m \mathbf{L})$ is the matrix of a bandpass filter whose band is defined by the function g and the scaling factor t_m . We are interested in the support of the atoms $\mathbf{g}_{m,i}$, and in particular whether we can see differences in support at coarse and fine scales.

In the classical setting, a wavelet atom is merely a scaled and translated version of a mother wavelet [21]. Two observations are



(a) Embedding



(b) Combinatorial Laplacian

Fig. 6. (a) Embedding used in Figure 5 and in (b) which is the same as Figure 5(a) with GFT basis vectors aligned vertically with their graph frequency instead of equispaced GFT basis vectors. Notice how cluster A is mapped to the left section of (b) and Figure 5, cluster B to the right section, and cluster C to the middle section.

important here: i) the scale of the wavelet completely defines the size of the support of said wavelet, and ii) the translation operator does not change the magnitude of the Fourier coefficient of the scaled wavelet. In other words, the scale completely defines the support in both the direct and spectral domains. However, as we saw in subsection 4.1 for the GFT, some spectral bands may have a localized support. As such, the impulse responses of the bandpass graph filter \mathbf{G}_m have varying support, especially for the combinatorial Laplacian approach: depending on which node the atom is centered on, the support of the atom may have varying size on the graph.

This is validated with Figure 7(a), where the sets of atoms $\{\mathbf{g}_{m,i}\}_{m,i}$ for each scale are stacked, from coarsest scale on the bottom to finest scale on top. Within each scale, the ordering of the nodes is based on the 1D embedding: if node i is assigned to a real value e_i in the 1D embedding, then the corresponding atom will be plotted above all nodes j assigned embedding locations e_j such that $e_j < e_i$. Finally, the values are normalized using 'overall_max_abs' to maintain the relative amplitude difference between atoms.

Two observations are of importance in Figure 7(a). First, as expected, most atoms of cluster B are vanishing. This finest scale corresponds to a high pass filter where the support is limited to cluster C. Therefore, the filter impulse response on those nodes of cluster A is close to zero. This property is not shared by the nodes of the cluster C (top left corner of the figure), where many of the atoms have several nodes in their support. Given these observation, it is clear that the study of the SGWT decomposition of a graph signal needs to take into account the support of the atoms themselves: a zero in the decomposition may correspond to either the signal being orthogonal to the atom, or the atom itself being zero in the vertices where the signal is localized.

The second observation is more significant. Indeed, even in the coarsest wavelet scale (second row of atoms from the bottom in Figure 7(a)), some atoms have zero support. This is the case for most atoms centered in cluster C. This property needs some thorough in-

vestigation, however, we can already identify some explanation using the GFT. In Figure 8(a), we plotted, for each scale, the output of the corresponding bandpass filter g_m applied to the GFT (each row corresponds to $\mathbf{G}_m \mathbf{u}_k$ with \mathbf{u}_k a GFT basis vector). For cluster C, it is quite clear that most of the nodes of the cluster have almost zero magnitude on this filtered GFT basis. In other words, since we have:

$$\mathbf{g}_{m,i} = \sum_k \langle \delta_i, g_m(\lambda_k) \mathbf{u}_k \rangle,$$

and $g_m(\lambda_k) \mathbf{u}_k$ is vanishing on nodes i in cluster C, then the $\mathbf{g}_{m,i}$ are also vanishing. Therefore, the property of non consistent support for a given scale can happen for coarser and finer scales alike.

To end this section, we observe that the SGWT based on the normalized Laplacian is slightly better with more consistent support size, but it still shows some issues (see Figure 7(b)). It should be noted that spectrum adapted tight graph wavelets [22] tackle part of the issue, where some bands defined by the SGWT may actually correspond to a very small set of graph frequencies⁶, but it does not solve the issue of localized GFT basis vectors, which is the root cause of some of the vanishing SGWT atom.

5. CONCLUSION

In this paper we have described a new set of visualization tools for graph signal processing, and their implementation in the `GrASP` Matlab toolbox. While it is well known that the GFT differs significantly from transforms for signals in regular domain, our goal is to encourage further research into this behavior, and into tools for graph signal representation that can be used in practice, by making it easier to visualize graph signal frequency information.

⁶This issue is not arising clearly here since the graph frequency distribution is not skewed as much as with other graphs (see Figure 5(a)).

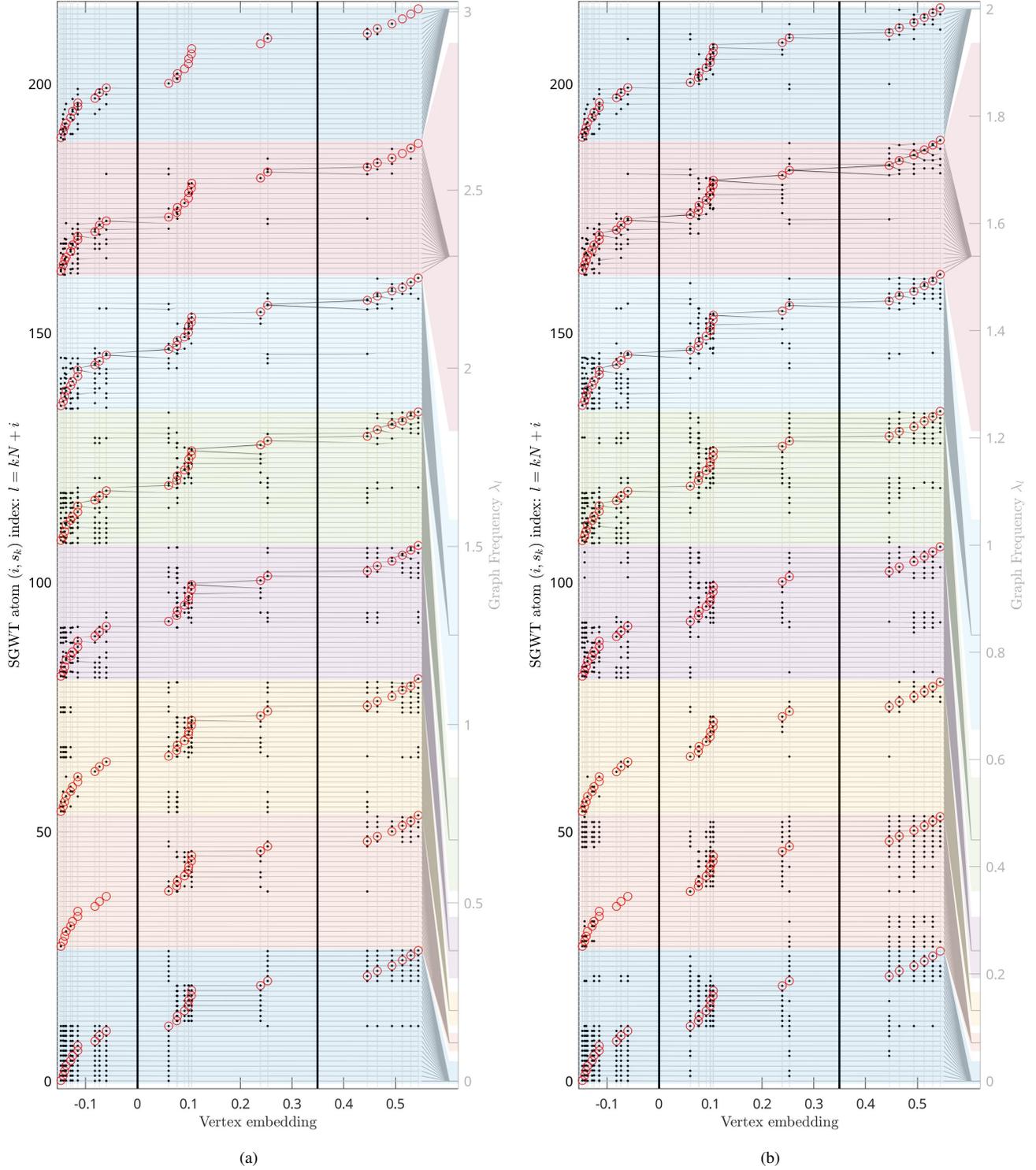


Fig. 7. SGWT atoms using 7 wavelet scales (and 1 scaling function) for (a) the combinatorial Laplacian approach and (b) the normalized Laplacian approach. Atoms are sorted from bottom to top by scale first (coarsest to finest) and then by node embedding. Highlighted entries correspond to the i^{th} node of the atom $g_{i,k}$. Each atom is mapped to the frequency of maximum spectral response of its associated bandpass filter $g_k(\lambda)$. The associated band is obtained by keeping all frequencies such that $g_k(\lambda) \geq 1.2$.

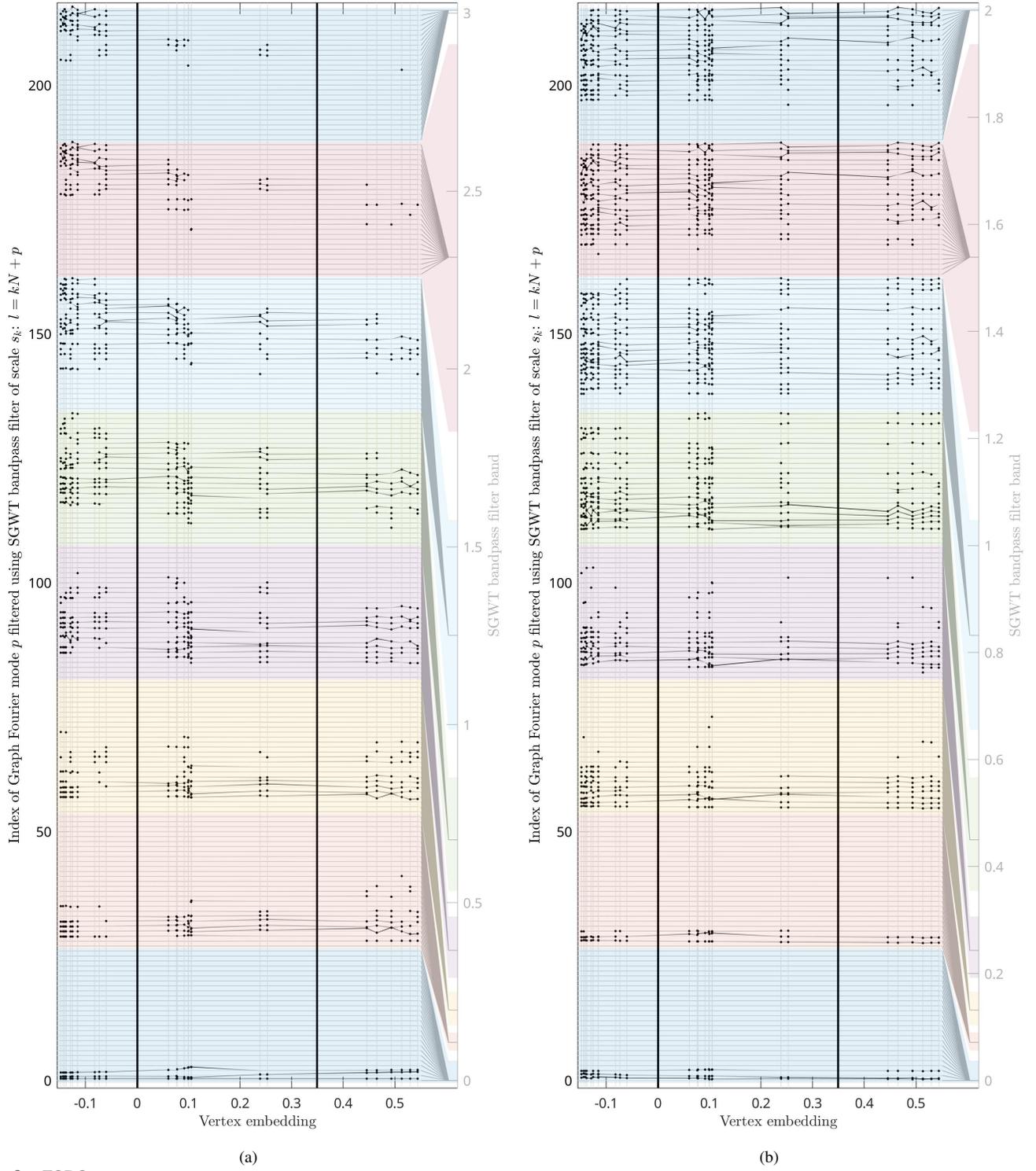


Fig. 8. TODO.

6. REFERENCES

- [1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
- [2] A. Ortega, P. Frossard, J. Kovačević, JMF Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [3] Fan RK Chung, "Spectral graph theory (CBMS regional conference series in mathematics, no. 92)," 1996.
- [4] Benjamin Girault, "Stationary graph signals using an isometric graph translation," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1516–1520.
- [5] B. Girault, A. Ortega, and S. S. Narayanan, "Irregularity-aware graph Fourier transforms," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5746–5761, Nov 2018.
- [6] Rasoul Shafipour, Ali Khodabakhsh, Gonzalo Mateos, and Evdokia Nikolova, "A directed graph Fourier transform with spread frequency components," *arXiv preprint arXiv:1804.03000*, 2018.
- [7] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [8] Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE transactions on signal processing*, vol. 63, no. 24, pp. 6510–6523, 2015.
- [9] Aamir Anis, Akshay Gadde, and Antonio Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.
- [10] Lloyd N Trefethen, Ásgeir Birkisson, and Tobin A Driscoll, *Exploring ODEs*, vol. 157, SIAM, 2017.
- [11] Naoki Saito, "How can we naturally order and organize graph laplacian eigenvectors?," *arXiv preprint arXiv:1801.06782*, 2018.
- [12] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [13] Benjamin Girault, Shrikanth S. Narayanan, Antonio Ortega, Paulo Gonçalves, and Eric Fleury, "GraSP: A Matlab toolbox for graph signal processing.," in *ICASSP. 2017*, p. 6574–6575, IEEE.
- [14] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [15] Ameya Agaskar and Yue M. Lu, "A Spectral Graph Uncertainty Principle.," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.
- [16] Bastien Padeloup, Réda Alami, Vincent Gripon, and Michael Rabbat, "Toward An Uncertainty Principle For Weighted Graphs," in *Signal Processing Conference (EUSIPCO), 2015 Proceedings of the 21st European*, Sept. 2015, pp. 1511–1515.
- [17] B. Padeloup, V. Gripon, G. Mercier, and D. Pastor, "Towards a characterization of the uncertainty curve for graphs," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4558–4562.
- [18] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on Graphs: Uncertainty Principle and Sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.
- [19] Nathanael Perraudin, Benjamin Ricaud, David I Shuman, and Pierre Vandergheynst, "Global and local uncertainty principles for signals on graphs," *APSIPA Transactions on Signal and Information Processing*, vol. 7, pp. e3, 2018.
- [20] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2014, KDD '14, pp. 492–501, ACM.
- [21] Stéphane Mallat, *A wavelet tour of signal processing (2. ed.)*, Academic Press, 1999.
- [22] David I. Shuman, Christoph Wiesmeyr, Nicki Holighaus, and Pierre Vandergheynst, "Spectrum-Adapted Tight Graph Wavelet and Vertex-Frequency Frames," *IEEE Trans. Signal Processing*, vol. 63, no. 16, pp. 4223–4235, 2015.