Two Channel Filter Banks on Arbitrary Graphs With Positive Semi Definite Variation Operators

Eduardo Pavez, Member, IEEE, Benjamin Girault, Member, IEEE, Antonio Ortega, Fellow, IEEE, and Philip A. Chou, Fellow, IEEE

Abstract—We propose novel two-channel filter banks for signals on graphs. Our designs can be applied to arbitrary graphs, given a positive semi definite variation operator, while using arbitrary vertex partitions for downsampling. The proposed generalized filter banks (GFBs) also satisfy several desirable properties including perfect reconstruction and critical sampling, while having efficient implementations. Our results generalize previous approaches that were only valid for the normalized Laplacian of bipartite graphs. Our approach is based on novel graph Fourier transforms (GFTs) given by the generalized eigenvectors of the variation operator. These GFTs are orthogonal in an alternative inner product space which depends on the downsampling and variation operators. Our key theoretical contribution is showing that the spectral folding property of the normalized Laplacian of bipartite graphs, at the core of bipartite filter bank theory, can be generalized for the proposed GFT if the inner product matrix is chosen properly. In addition, we study vertex domain and spectral domain properties of GFBs and illustrate their probabilistic interpretation using Gaussian graphical models. While GFBs can be defined given any choice of a vertex partition for downsampling, we propose an algorithm to optimize these partitions with a criterion that favors balanced partitions with large graph cuts, which are shown to lead to efficient and stable GFB implementations. Our numerical experiments show that partition-optimized GFBs can be implemented efficiently on 3D point clouds with hundreds of thousands of points (nodes), while also improving the color signal representation quality over competing state-of-the-art approaches.

Index Terms—Two-channel filter banks, graph Fourier transform, graph signal, multiresolution representation.

I. INTRODUCTION

GRAPHS are powerful tools to model unstructured data. On a graph, nodes and edges represent objects of interest and their similarity relations, respectively. A function on the

Manuscript received 4 March 2022; revised 7 September 2022 and 21 February 2023; accepted 27 February 2023. Date of publication 21 March 2023; date of current version 4 April 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. David Shuman. This work was supported in part by Google Faculty Research Award and in part by NSF under Grants CCF-1410009 and CCF-2009032. (Corresponding author: Eduardo Pavez.)

Eduardo Pavez and Antonio Ortega are with the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: pavezcar@usc.edu; aortega@usc.edu).

Benjamin Girault is with the ENSAI, CNRS, CREST-UMR, Université de Rennes, 9194 Rennes, France (e-mail: benjamin.girault@ensai.fr).

Philip A. Chou is with Google Research, Seattle, WA 98013 USA (e-mail: pachou@ieee.org).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2023.3257983, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3257983

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We introduce a new vertex partition, so that GFBs can be applied to bipartite graphs and the disconnected graph. While the GFB theory is designed for bipartite graphs, we introduced a numerical stability criterion so that solutions can be found for non-bipartite graphs, as well as with operators other than the normalized Laplacian. Instead, Generalized Filter Bank (GFB) solutions are obtained for arbitrary graphs, for any positive semi-definite graph variation operator and using arbitrary vertex partitions for downsampling, while still satisfying properties (i)–(iv). We show that for sparse graphs, GFBs have scalable and eigendecomposition-free implementations, relying only on sparse matrix-vector products and sparse linear system solvers.

The key innovation enabling GFBs is the adoption of a new inner product, which allows us to depart from the traditional Hilbert space (induced by the dot product) underlying the majority of GSP methods. We build upon [29] where graphs are represented by a positive semi-definite variation operator $M \succeq 0$, which measures signal smoothness, and an inner product $(x,y)_Q = y^T Q x$, with $Q \succ 0$. The (M, Q) Graph Fourier Transform (M, Q)-GFT is defined as the generalized eigenvectors of M, which form a Q-orthonormal basis.

In our solution, Q is chosen as a function of M and the downsampling operator (determined by a vertex partition). More precisely, for a given a variation operator M our GFB theory can find a valid Q for any vertex partition, so that GFBs can be constructed using spectral graph filters of the (M, Q)-GFT. Moreover, we show that it is important to optimize these partitions so that they are balanced (Section V) and result in Q matrices that are sparse and close to diagonal (Section VI). Our main contributions are summarized next.

**Theory of GFBs (Section IV):** We introduce a new spectral folding property for the (M, Q)-GFT, analogous to that satisfied by the eigenvectors and eigenvalues of the normalized Laplacian of bipartite graphs [39]. For a given variation operator M and a vertex partition for downsampling, we show that there exists a unique inner product matrix Q such that the (M, Q)-GFT obeys the spectral folding property. Based on this result, we propose perfect reconstruction and Q-orthogonal filter banks using spectral graph filters of the (M, Q)-GFT. Our conditions in the graph frequency domain are exactly those developed in [10, 11] for the normalized Laplacian of bipartite graphs and therefore we can reuse any of the previously proposed filter designs, including those in [10, 11] and improved solutions such as [12, 13]. When M is the normalized Laplacian and the graph is bipartite, we recover the BFB framework.

In our preliminary version of this work [40] we introduced the spectral folding property and two-channel filter banks on arbitrary graphs (Sections IV-D and IV-F), while an application of GFBs to image compression has been recently published [28]. In this paper, we further develop the GFB theory, providing all proofs not given in [40], discussing the results in more depth and introducing the following novel contributions.

**Properties, interpretations and examples (Section V):** Since (M, Q)-GFTs are relatively new, especially when Q is not diagonal, Section V is devoted to studying their properties. We show that some spectral properties of the normalized and random walk Laplacians can be extended to the proposed (M, Q)-GFTs. We also provide examples of (M, Q)-GFTs with and without the spectral folding property (Fig. 6). Representing an arbitrary graph as the sum of a bipartite and a disconnected graph (Fig. 2) leads to a vertex domain interpretation of the proposed spectral graph filters. Finally, we use Gaussian graphical models to give a probabilistic interpretation of GFBs and the spectral folding property.

**Vertex partitioning (Section VI):** While the GFB theory is valid for arbitrary vertex partitions, not all vertex partitions are desirable for downsampling. As an example, in Section V we prove that unbalanced vertex partitions lead to GFBs with poor frequency selectivity, and thus balanced partitions should be used. In Section VI we propose a numerical stability criterion for vertex set partition along with a computationally efficient algorithm based on an approximate solution to a maximum cut (max-cut) problem. Essentially, our algorithm finds vertex partitions leading to a graph decomposition (Fig. 2) where the bipartite part has the largest cut, while the disconnected graph is sparse and has small weights, which is also helpful to reduce the complexity of GFB implementations. Note that bipartite approximation algorithms for BFBs have been formulated as max-cut problems [23]. However, these approaches result in the removal of non-bipartite edges, while in our case edge removal is unnecessary.

---

1The use of non traditional Hilbert spaces has proven effective in various studies involving irregularly structured data, such as graph sparsification [33], machine learning [34], compression of 3D point clouds [35, 36], graph signal sampling [37], and perceptual coding [38].
In this paper, the inner product defined by $Q$ plays a fundamental role: It not only leads to a definition of the $(M, Q)$-GFT and to conditions for $Q$-orthogonal filter banks, but it is also needed to understand biorthogonal designs (i.e., good biorthogonal filters are such that they are approximately $Q$-orthogonal). Since the choice $Q = I$ is the dominant one in both signal processing and GSP, it is important to understand the implications of not using the standard ($Q = I$) inner product. To address this point, we note that relaxing the $I$-orthogonality requirement is often an option if it is necessary to achieve some desirable properties. As an example JPEG2000 uses biorthogonal filters that are nearly $I$-orthogonal because they have finite support and are symmetric, which is desirable for imaging applications and would not be possible with orthogonal filters. Similarly, biorthogonal filter banks on bipartite graphs are preferred over $I$-orthogonal ones because they have polynomial implementations [11]. Thus, in practice a non $I$-orthogonal solution can be useful as long as it is sufficient close to being $I$-orthogonal.

In this work, we use biorthogonal filters (obtained from [11]) that lead to polynomial filters and can be designed to be nearly $Q$-orthogonal (see Section IV). Additionally, in Section VI we show that the downsampling operator can be optimized so that the $Q$ matrices have favorable properties (sparsity, close to diagonal and with small condition number in operator norm) so that the $Q$ inner product approximates the conventional dot product. How closely the best $Q$ matrix for a given graph approximates the dot product depends on the graph itself (degree irregularity, sparsity, etc), but we show examples for specific graphs (sensor network, spatial graphs, and 3D point cloud graphs) showing that good approximations can be achieved. Further analysis of this question for more general scenarios is left for future work.

The rest of the paper is organized as follows. In Section II we review related work. In Section III we introduce the fundamentals of GSP on general Hilbert spaces, while Section IV is dedicated to bipartite and generalized filter banks. Section V focuses on properties of $(M, Q)$-GFTs and GFs. Section VI studies optimal vertex partitioning for downsampling. We end this paper with numerical results, and conclusions in Sections VII and VIII respectively. Additional proofs can be found in the Supplementary Materials. Code examples can be found in https://github.com/STAC-USC/graph_filterbank_folding.

II. RELATED WORK

We discuss related filter banks and multiresolution representations (MRR) on graphs from three perspectives: graph topology, graph matrix, and downsampling sets.

Graph topology: Several existing filter bank theories are applicable only to graphs with certain types of topology such as: bipartite [10], [11], $M$-block cyclic [41], [42], circumulant [48], and acyclic [49]. While some of these graph structures appear naturally in some applications [50], [51], many cases graphs of interest do not belong to any of these categories. To apply BFBs to arbitrary graphs one can decompose any graph as a sum of bipartite graphs and apply the filter bank in a separable manner [52]. Other approaches include bipartite approximation [23], bipartite graph learning methods [24], [25], [26], [27], [53], graph oversampling [43], and vertex partition optimization [54]. Earlier filter bank designs for arbitrary graphs are difficult to invert (e.g., least squares reconstruction is needed) [55], [56], [57]. More recent MRRs fail to be simultaneously perfect reconstruction and orthogonal [44], while others, require full eigen-decomposition [45], [58]. Existing approaches that are valid for arbitrary graphs have several disadvantages. On the one hand, approximation-based methods may reduce signal representation quality while also requiring additional computational resources (to select the best bipartite approximation) [23], [24], [25], [52]. On the other hand, methods that allow the original graph to be used can do so at the expense of other desirable features, such as low complexity [45], [58], perfect reconstruction or orthogonality (in $I$ or other Q inner product) [44], [54] (see Table I for a comparison of some of these approaches). Therefore, a theoretical formulation leading to critically sampled filter banks for arbitrary graphs, without the aforementioned disadvantages, can be an attractive alternative.

Graph matrix: The choice of graph matrix, such as the graph Laplacian or the adjacency matrix, whose non-zero pattern encodes the graph structure, is an important design decision [59], [60]. Existing filter bank frameworks are built for specific types of graph matrices because of their special algebraic or spectral properties [10], [41], but these choices may not be suitable for a particular application. As an example, BFBs use the normalized Laplacian [10], while the random walk Laplacian has been shown to achieve better coding performance [28]. More recent filter bank approaches [46], [47] can be applied to the larger class of positive semi-definite variation operators. However, they lack perfect reconstruction [46] or require computing a full eigen-decomposition of the graph operator [47], which significantly limits their application to large graphs. The proposed GFBs are valid for any positive semi definite graph matrix (i.e., a variation operator), including commonly used graph Laplacians, without compromising on other properties.

Downsampling: For discrete-time signals, a downsampling by 2 operator keeps “every other sample” and discards the rest. In graphs, there is no obvious notion of “every other vertex” unless the graph is bipartite [52]. For graph filter banks, vertex partitions of the node set can be chosen under various criteria [9], [46], [54]. Recently, [47], [61] used spectral domain sampling, and while this approach leads to an attractive theory, spectral sampling requires computing all eigenvectors and eigenvalues of the graph matrix, which can have significant computation complexity. [46] extended sampling theory of graph signals to filter banks but practical implementations of this framework cannot achieve perfect reconstruction. In this work, we show that any partition of the vertex set is a valid downsampling operator. We also propose strategies for optimally choosing these partitions in Section VI.

III. GSP IN GENERAL HILBERT SPACES

A. Notation

Scalars, vectors and matrices are written in lower case regular, lower case bold and upper case bold respectively (e.g., $a$, $b$, $A$).
C). Positive definite and semi-definite matrices are denoted by $A \succ 0$, and $A \succeq 0$ respectively. We will denote by $C_{\mathcal{A}}$, the sub-matrix of $C$ whose rows and columns are indexed by the sets $\mathcal{A}$ and $\mathcal{B}$, respectively. The spectral norm or largest singular value of a matrix $A$ is denoted by $\|A\|$.

### B. Graph Signal Processing

Consider a graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, \ldots, n\}$, and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A graph signal is a function $x : \mathcal{V} \to \mathbb{R}$, which can be represented by a vector $x = [x_1, \ldots, x_n]^\top$, and $x_i$ is the signal value at vertex $i \in \mathcal{V}$.

1) **Graph Signal Variation**: The graph is equipped with a symmetric positive semi-definite variation matrix $M = (m_{ij})$, with sparsity pattern determined by the edge set, that is, $m_{ij} = m_{ji} = 0$ when $(i, j) \notin \mathcal{E}$ and $i \neq j$. Throughout the paper we will assume that the graph is connected and thus $M$ is an irreducible matrix [62], [63]. The graph signal variation is

$$\Delta(x) = x^\top M x.$$  

(1)

Intuitively, given two signals of equal energy, the one with larger variation is the one with higher energy in the higher frequencies of the spectrum. Popular choices of variation operators are listed in [29], [67].

The adjacency matrix is a non-negative symmetric matrix $W = (w_{ij})$, where $w_{ij} = 0$, whenever $(i, j) \notin \mathcal{E}$. The degree of node $i$ is $d_i = \sum_{j \in \mathcal{V}} w_{ij}$, and the degree matrix is $D = \text{diag}(d_1, \ldots, d_n)$. The combinatorial graph Laplacian (CLG) is $L = D - W$, while the normalized graph Laplacian (NGL) is $\mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$. A generalized graph Laplacian (GLG) is any positive semi-definite matrix with non-positive off diagonal entries, which includes the normalized and combinatorial graph Laplacian matrices [63], [64], [65], [66]. Other variation operators are listed in [29], [67].

2) **The $\langle M, Q \rangle$-GFT** [29]: Given a positive definite matrix $Q$, we define the $Q$ inner product between $x$ and $y$, as

$$(x, y)_Q = y^\top Qx,$$  

(2)

with induced $Q$-norm $\|x\|_Q = \sqrt{(x, x)_Q}$. We say $\{u_k\}_{k=1}^n$ is a $Q$-orthonormal set if it satisfies

$$(u_i, u_j)_Q = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.$$  

(3)

In matrix form this corresponds to $U^\top Q U = I$, where $U = [u_1, \ldots, u_n]$, which also implies that $U^{-1} = U^\top Q$. Following [29], the $\langle M, Q \rangle$-Graph Fourier Transform ((M, Q)-GFT) is defined as the $Q$-orthonormal set that minimizes the graph signal variation, that is

$$u_1 = \arg \min_{u \neq 0, \|u\|_Q = 1} u^\top M u,$$  

(4)

and for any $2 \leq k \leq n$

$$u_k = \arg \min_{u \neq 0, \|u\|_Q = 1} u^\top M u \text{s.t. } (u_i, u_j)_Q = 0, \forall i < k.$$  

(5)

Note that this definition of the $\langle M, Q \rangle$-GFT basis is consistent with the traditional definition of GFT (i.e., the $\langle M, I \rangle$-GFT). In particular, when $M = L$, the variation of $x$ is $x^\top L x = \sum_{(i,j)\in E} w_{ij} (x_i - x_j)^2$, and as in that case, the generalized eigenvectors $u_k$ have increasing variation (quantified with the same operator $M$) for larger $k$. The only difference is that now the basis vectors are $Q$-orthonormal instead of $I$-orthonormal. The $\langle M, Q \rangle$-GFT is also the solution to the generalized eigendecomposition problem [29], [62]:

$$M u = \lambda Q u.$$  

(6)

The matrix of unit $Q$-norm generalized eigenvectors is denoted by $\hat{U} = [u_1, \ldots, u_n]$, with generalized eigenvalues (or graph frequencies) forming a set $\sigma(M, Q) = \{\lambda_1, \ldots, \lambda_n\}$, where $\lambda_k = u_k^\top M u_k$ and $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$. The generalized eigenvalue matrix is $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. The fundamental matrix is $Z = Q^{-1} M$, which is diagonalized by $\hat{U}$, so

$$Z = Q^{-1} M = U A U^{-1} = U A U^\top Q.$$  

(7)

Then we can factorize the variation operator $M$ as follows:

$$M = Q U A U^\top Q.$$  

(8)

A graph signal $x$ has a $(M, Q)$-GFT representation given by:

$$x = \sum_{i=1}^n (x, u_i)_Q u_i = \hat{U} \hat{x}.$$  

(9)
where $\hat{x} = U^T Q x$ is the $(M, Q)$-GFT of $x$ [29]. Thus, (9) is the inverse $(M, Q)$-GFT, where $x$ is reconstructed as a linear combination of the frequency components in $\hat{x}$. Note that the inverse transform is given by $x = UX$, since $UU^T Q = I$.

3) Spectral Graph Filters: Given a function (i.e., a filter kernel) $h : (M, Q) \rightarrow \mathbb{R}$, then a spectral graph filter (SGF) is any matrix $H$ of the form

$$H = Uh(A)U^T Q = h(Z),$$

where $h(A) = \text{diag}(h(\lambda_1), h(\lambda_2), \ldots, h(\lambda_n))$. When $h(\lambda) = \lambda$ we have that $H = Z = Q^{-1} M$ and we recover the fundamental matrix. A spectral graph filter is implemented by first applying the $(M, Q)$-GFT to a signal $x$, then multiplying each transformed coefficient $\hat{x}_i = (x, u_i)_Q$ by the filter coefficient $h(\lambda_i)$. The resulting signal is transformed back to the vertex domain with the inverse $(M, Q)$-GFT. Filtering $x$ using $h$ produces the graph signal $Hx$, since

$$Hx = Uh(A)\hat{x} = \sum_{i=1}^{n} h(\lambda_i)\hat{x}_i u_i.$$

When $h$ is a polynomial $h(\lambda) = a_0 + a_1 \lambda + \cdots + a_d \lambda^d$ for $d \in \mathbb{N}$, the resulting SGF is a polynomial of $Z$, thus $H = h(Z) = a_0 I + a_1 Z + \cdots + a_d Z^d$, can be implemented efficiently without eigendecomposition using matrix product vectors. Thus, if $Z$ is sparse and $d$ is relatively small, graph filtering with polynomial SGFs can be applied efficiently on large graphs. Complexity of polynomial graph filters and filter banks is reviewed in more detail in Section VI.

C. Vertex and Frequency Domain Hilbert Spaces

Both a graph signal, $x$, and its $(M, Q)$-GFT, $\hat{x}$, can be represented by vectors in $\mathbb{R}^n$. However, in our formulation they belong to different Hilbert spaces, namely, $(\mathbb{R}^n, (\cdot, \cdot)_Q)$ for $x$ and $(\mathbb{R}^n, (\cdot, \cdot)_I)$ for $\hat{x}$, where $(\cdot, \cdot)_1$ is the traditional dot product. Thus, we can view the $(M, Q)$-GFT, $U^T Q$, as a mapping from $(\mathbb{R}^n, (\cdot, \cdot)_Q)$ to $(\mathbb{R}^n, (\cdot, \cdot)_I)$, while the inverse $(M, Q)$-GFT $U$ is a mapping from $(\mathbb{R}^n, (\cdot, \cdot)_I)$ to $(\mathbb{R}^n, (\cdot, \cdot)_Q)$. Making explicit the Hilbert spaces involved in this mapping allows us to state the following property.

**Theorem 1 (Parseval [29]):** Let $x, y \in \mathbb{R}^n$, and let $\hat{x}$ and $\hat{y}$ be their respective frequency domain representations, then

$$(x, y)_Q = (\hat{x}, \hat{y})_1.$$  

(12)

From (10) it is clear that SGFs are functions from $(\mathbb{R}^n, (\cdot, \cdot)_Q)$ to the same Hilbert space. For the rest of the paper we will use the $Q$ inner product for graph signals, and the $I$ inner product for their frequency domain representations.

IV. TWO-CHANNEL FILTER BANKS ON GRAPHS

In this section we introduce the two-channel filter bank theory for arbitrary graphs. In Section IV-A we formulate the problem. Section IV-B reviews the solution for bipartite graphs from [10]. We illustrate how to go from bipartite to arbitrary graphs through an example in Section IV-C. Our main theoretical result, the generalized spectral folding property is presented in Section IV-D. Generalized filter banks are constructed in Sections IV-F and IV-G.

A. Problem Formulation

A two-channel filter bank (Fig. 1) is composed of downsampling and upsampling operators on the sets $A$ and $B$, and two sets of $|V| \times |V|$ matrices, corresponding to the analysis filters ($H_0, H_1$) and the synthesis filters ($G_0, G_1$). Throughout the paper, we will consider critically sampled filter banks.

**Definition 1:** A two-channel filter bank is critically sampled if the downsampling sets $A$ and $B$ form a partition of $V$.

Without loss of generality we assume that $A = \{1, \ldots, |A|\}$, and $B = V \setminus A$. Downsampling $x$ on $A$ keeps the entries $\{x_i : i \in A\}$ and discards the rest, resulting in

$$x_A = S_A x,$$

(13)

where $S_A = [I_A, 0]$ is a $|A| \times |V|$ selection matrix. The upsampling operator $S_A^T$ maps the signal back to $V$ by filling the entries on $B$ with zeroes. Downsampling followed by upsampling corresponds to

$$S_A^T S_A x = S_A^T x_A = \begin{bmatrix} x_A^T & 0 \end{bmatrix}^T.$$

(14)

The analysis operator $T_a$ (i.e., filtering followed by downsampling) from Fig. 1 can be written as

$$T_a = S_A^T S_A H_0 + S_A^T S_B H_1 = \begin{bmatrix} S_A H_0 \\ S_B H_1 \end{bmatrix}.$$

(15)

The outputs of the low pass and high pass channels, approximation, $a$, and detail, $d$, coefficients, respectively, are

$$T_a x = \begin{bmatrix} a^T \\ d^T \end{bmatrix}^T = \begin{bmatrix} (S_A H_0) x \\ (S_B H_1) x \end{bmatrix}^T.$$

(16)

The synthesis operator can be expressed as

$$T_s = G_0 S_A^T S_A + G_1 S_B^T S_B = \begin{bmatrix} G_0 S_A^T \\ G_1 S_B \end{bmatrix}.$$

(17)

Both the analysis and synthesis operators map signals from $(\mathbb{R}^n, (\cdot, \cdot)_Q)$ to $(\mathbb{R}^n, (\cdot, \cdot)_Q)$ (see also Section III-C), thus $x, T_a x,$ and $T_s T_a x$ are vertex domain signals. We are interested in perfect reconstruction two-channel filter banks.

**Definition 2:** A two-channel filter bank is perfect reconstruction (PR) if $T_s T_a x = x$, for all $x$.

Since we only consider critically sampled filter banks, the PR condition is equivalent to $T_s = T_a^{-1}$, thus a necessary and sufficient condition for perfect reconstruction is given by

$$T_s T_a = G_0 S_A S_A^T S_A H_0 + G_1 S_B S_B H_1 = I.$$

(18)

We also introduce Q-orthogonal filter banks.

**Definition 3:** A two-channel filter bank is $Q$-orthogonal, if for every pair of graph signals $x, y$,

$$(x, y)_Q = (T_s x, T_a y)_Q.$$

(19)

In matrix form, (19) is equivalent to $T_s^T Q T_a = Q$.

The traditional notion of orthogonal filter banks corresponds to the case $Q = I$. Critically sampled $Q$-orthogonal filter banks
are always perfect reconstruction with synthesis operator
\[ T_s = Q^{-1}T_u Q. \] (20)

It is easy to verify that (20) also obeys \( T_s^T Q T_s = Q \).

Remark 1: Because the vertex domain and frequency domain Hilbert spaces are different (see Section III-C), \( Q \)-orthogonality for filter banks is different than \( Q \)-orthogonality of the \((M, Q)\)-GFT. This is because the \((M, Q)\)-GFT maps graph signals from their vertex domain representation to their frequency domain representation, that is, \( U^T Q : (\mathbb{R}^n, \langle \cdot, \cdot \rangle_Q) \rightarrow (\mathbb{R}^n, \langle \cdot, \cdot \rangle_1) \), which results in the relationship of Theorem 1 (see [29]). In contrast, the analysis and synthesis operators of a PR critically sampled filter bank stay in the same Hilbert space \((\mathbb{R}^n, \langle \cdot, \cdot \rangle_1)\).

We seek two-channel filter banks that can provide good signal representations, while having efficient implementations on large (arbitrary) graphs. In practice, this can be achieved by designing SGFs \( H_i, G_i \) that: (i) have good frequency selectivity and (ii) can be written as polynomials of \( Z \). We review bipartite graph solutions [10], [11] before introducing our proposed GFBs for arbitrary graphs.

### B. Bipartite Filter Banks [10], [11]

The necessary and sufficient filter bank design conditions of [10], [11] apply to bipartite graphs.

**Definition 4 (Bipartite graph [39]):** A graph \( G = (\mathcal{V}, \mathcal{E}) \) is bipartite on a partition of the node set \( \mathcal{A}, \mathcal{B} \), if for all \((i, j) \in \mathcal{E}, i \in \mathcal{A} \) and \( j \in \mathcal{B}, \) or \( i \in \mathcal{B} \) and \( j \in \mathcal{A} \).

Some examples of bipartite graphs include Fig. 2 (middle graph), Figs. 5(a) and 7(b). In bipartite filter banks (BFB) the sets \( \mathcal{A} \) and \( \mathcal{B} \) are used for downsampling. In a bipartite graph only edges between \( \mathcal{A} \) and \( \mathcal{B} \) exist, so we have:

\[
L = \begin{bmatrix}
D_A & -W_{AB}^T \\
-W_{BA} & D_B
\end{bmatrix}, \quad \mathcal{L} = \begin{bmatrix}
I_A & -W_{AB}^T \\
-W_{BA} & I_B
\end{bmatrix}.
\] (21)

BFBs are implemented using spectral graph filters of the \((\mathcal{L}, I)\)-GFT, that is \( Z = \mathcal{L} \), and

\[ H_i = h_i(\mathcal{L}), \quad G_i = g_i(\mathcal{L}), \quad i \in \{0, 1\}. \] (22)

The diagonal matrix \( J = S_A^T S_A - S_B^T S_B \), with entries

\[ J_{i,i} = \begin{cases} 1 & \text{if } i \in \mathcal{A} \\ -1 & \text{if } i \in \mathcal{B} \end{cases} \] (23)

is used to establish the following property of the normalized Laplacian of bipartite graphs.

**Proposition 1 (Spectral folding [39]):** Let \( \mathcal{L} = \mathcal{L} u = \lambda u \) if and only if \( Ju \) is also an eigenvector of eigenvalue \( 2 - \lambda \).

Thus, eigenvalues come in pairs \((\lambda, 2 - \lambda)\) mirrored around the middle frequency \( \lambda = 1 \). For a given \( \lambda \), the eigenvector \( u \) and its folded version \( Ju \) have the same values for entries in \( \mathcal{A} \), while signs are changed for the entries in \( \mathcal{B} \). Proposition 1 is the key property used by [10], [11] to design perfect reconstruction and I-orthogonal BFBs.

**Theorem 2 (Perfect Reconstruction [10]):** A two-channel filter bank on a bipartite graph with spectral graph filters given by (22) is PR if and only if, \( \forall \lambda \in \sigma(\mathcal{L}, I) \)

\[ g_0(\lambda) h_0(\lambda) + g_1(\lambda) h_1(\lambda) = 2, \] (24)

\[ h_1(\lambda) g_1(2 - \lambda) - h_0(\lambda) g_0(2 - \lambda) = 0. \] (25)

The proof follows from using \( S_A^T S_A = \frac{1}{2}(I + J) \) and \( S_B^T S_B = \frac{1}{2}(I - J) \) in the PR condition (18), in combination with Proposition 1 (see [10]). Proposition 1 can also be used to design I-orthogonal filter banks.

**Theorem 3 (Orthogonality [10]):** Under the same conditions of Theorem 2, a filter bank is I-orthogonal if and only if, \( \forall \lambda \in \sigma(\mathcal{L}, I) \)

\[ h_0^2(\lambda) + h_1^2(\lambda) = 2, \] (26)

\[ h_1(\lambda) h_1(2 - \lambda) - h_0(\lambda) h_0(2 - \lambda) = 0. \] (27)

The main advantage of I-orthogonal over non-I-orthogonal PR filter banks is the energy preservation property (Definition 3), essential for compression and de-noising applications. Because no polynomial solutions to (26) and (27) exist [11], I-orthogonal filter banks cannot be implemented with polynomial graph filters. Instead, polynomial biorthogonal filter banks with near I-orthogonality can be designed [11].

**Proposition 2:** [11], [13] Biorthogonal filters defined by

\[ h_0(\lambda) = g_1(2 - \lambda), \quad h_1(\lambda) = g_0(2 - \lambda), \] (28)

\[ h_0(\lambda) h_1(2 - \lambda) + h_0(2 - \lambda) h_1(\lambda) = 2 \] (29)

are perfect reconstruction.

In what follows we will show that the PR, I-orthogonality and biorthogonal conditions are not limited to bipartite graphs and the normalized Laplacian. The key insight is a generalization of Proposition 1.

### C. From Bipartite to Arbitrary Graphs: Lazy Filter Bank

Before stating our main results, we show how our extension from bipartite to arbitrary graphs would work for the “lazy” filter bank of Fig. 3, where \( C \) is given by

\[ C = \begin{bmatrix} I_A & C_{AB} \\ C_{BA} & I_B \end{bmatrix}. \] (30)

\( C_{AB} \) and \( C_{BA} \) are arbitrary rectangular matrices of dimensions \(|\mathcal{A}| \times |\mathcal{B}| \) and \(|\mathcal{B}| \times |\mathcal{A}| \), respectively. It is easy to verify that the lazy filter bank is perfect reconstruction when \( C \) is given by (30). Since \( d = S_B C x \) and \( a = S_A x \), we have that

\[ \hat{x} = S_B^T d + (2I - C) S_A^T a = (S_B^T S_B C + (2I - C) S_A^T S_A) x \]

\[ = \begin{bmatrix} 0 & 0 \\ C_{BA} & I_B \end{bmatrix} \begin{bmatrix} I_A & 0 \\ -C_{BA} & I_B \end{bmatrix} x = x. \] (31)
If we take $\mathbf{C} = \mathcal{L}$ for a bipartite graph, the lazy filter bank is a special case of a perfect reconstruction (FBF) (Fig. 1) with filters $\mathbf{H}_0 = \mathbf{I}$, $\mathbf{H}_1 = \mathcal{L}$, $\mathbf{G}_0 = 2\mathcal{I} - \mathcal{L}$, and $\mathbf{H}_1 = \mathbf{I}$, corresponding to SGFs of the $(\mathcal{L}, \mathcal{I})$-GFT with biorthogonal filter kernels $h_0(\lambda) = g_1(\lambda) = 1$, $h_1(\lambda) = \lambda$, and $g_0(\lambda) = h_1(2 - \lambda)$. In the following subsections we will show that it is also possible to choose $\mathbf{C}$ as a function of the variation operator $\mathbf{M}$ of a non-bipartite graph, which allows us to generalize the theorems from Section IV-B to arbitrary graphs.

D. Spectral Folding on Arbitrary Graphs

We generalize Proposition 1 to other graphs and variation operators by using the $\mathbf{Q}$ inner product. Using this result, we propose a new $(\mathbf{M}, \mathbf{Q})$-GFT and show that it can be used to obtain Generalized Filter Banks on arbitrary graphs. First we define the spectral folding property for a $(\mathbf{M}, \mathbf{Q})$-GFT.

Definition 5 (Spectral folding): Given a graph $\mathcal{G}$ with variation operator $\mathbf{M} \succeq 0$, inner product $\mathbf{Q} \succ 0$ and a partition $A$, $B$. The $(\mathbf{M}, \mathbf{Q})$-GFT has the spectral folding property, if for all generalized eigenpairs $(\mathbf{u}, \lambda)$ then $(\mathbf{J_u} (2 - \lambda))$ is also a generalized eigenpair, that is:

$$\mathbf{M_u} = \lambda \mathbf{Q_u} \Leftrightarrow \mathbf{M J_u} = (2 - \lambda) \mathbf{Q J_u},$$

where $\mathbf{J} = S_A^T S_A - S_B^T S_B$, as in (23).

The following theorem completely characterizes the spectral folding property. The proof can be found in Section IV-E.

Theorem 4 (Spectral folding): Given a graph $\mathcal{G}$ with variation operator $\mathbf{M} \succeq 0$, inner product $\mathbf{Q} \succ 0$ and a partition $A$, $B$. The $(\mathbf{M}, \mathbf{Q})$-GFT has the spectral folding property if and only if $\mathbf{Q}$ is chosen as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{M_{AA}} & 0 \\ 0 & \mathbf{M_{BB}} \end{bmatrix}. \quad (33)$$

The condition that $\mathbf{M_{AA}}$ and $\mathbf{M_{BB}}$ are non singular is satisfied for any vertex partition if $\mathbf{M} \succ 0$. This condition also holds if $\mathbf{M}$ is the combinatorial or normalized Laplacian of a connected graph. Theorem 4 generalizes Proposition 1, demonstrating that the spectral folding property is not unique to the $(\mathcal{L}, \mathcal{I})$-GFT of bipartite graphs, and in fact, it is always satisfied if the inner product matrix $\mathbf{Q}$ is chosen as in (33).

Using (33), the fundamental matrix $\mathbf{Z} = \mathbf{U} \mathbf{A} \mathbf{U}^T$ is

$$\mathbf{Z} = \mathbf{Q}^{-1} \mathbf{M} = \begin{bmatrix} \mathbf{I_A} & \mathbf{M_{AA}}^{-1} \mathbf{M_{AB}} \\ \mathbf{M_{BB}}^{-1} \mathbf{M_{BA}} & \mathbf{I_B} \end{bmatrix}, \quad (34)$$

so that $\mathbf{Z}$ has the same structure as $\mathbf{C}$ in (30) and thus $\mathbf{Z}$ can be used to construct “lazy” filter banks (see Fig. 3). In Section IV-F we apply Theorem 4 with the fundamental matrix $\mathbf{Z}$ in (34) to design GFBs on arbitrary graphs. To gain some intuition about $(\mathbf{M}, \mathbf{Q})$-GFTs with spectral folding, we study some of their properties from vertex, spectral and probabilistic perspectives (Section V) and propose criteria to choose vertex partitions with favorable properties (Section VI).

E. Proof of Theorem 4 (Spectral Folding)

First we prove that (33) implies the spectral folding property (Definition 5). Since $\mathbf{M}$ and $\mathbf{Q}$ are positive semidefinite, and $\mathbf{Q}$ is non singular, there is a full set of generalized eigenvectors [62]. Let $\mathbf{u} = [\mathbf{u}_A^T, \mathbf{u}_B^T]^T$ and $\mathbf{M_u} = \lambda \mathbf{Q_u}$, then

$$\mathbf{M}_{AA} \mathbf{u}_A + \mathbf{M}_{AB} \mathbf{u}_B = \lambda \mathbf{Q_A} \mathbf{u}_A, \quad (35)$$

$$\mathbf{M}_{BB} \mathbf{u}_A + \mathbf{M}_{BB} \mathbf{u}_B = \lambda \mathbf{Q_B} \mathbf{u}_B. \quad (36)$$

Set $\mathbf{v} = \mathbf{J_u} = [\mathbf{u}_A^T, -\mathbf{u}_B^T]^T$, and using (35) and (36) we get

$$\mathbf{M_v} = \begin{bmatrix} \mathbf{M}_{AA} \mathbf{u}_A - \mathbf{M}_{AB} \mathbf{u}_B \\ \mathbf{M}_{BB} \mathbf{u}_A - \mathbf{M}_{BB} \mathbf{u}_B \end{bmatrix} = \begin{bmatrix} 2\mathbf{M}_{AA} \mathbf{u}_A - \lambda \mathbf{Q_A} \mathbf{u}_A \\ \lambda \mathbf{Q_B} \mathbf{u}_B - 2\mathbf{M}_{BB} \mathbf{u}_B \end{bmatrix} = (2 - \lambda) \mathbf{Q_v}. \quad (37)$$

The second equality is implied by $\mathbf{M}_{AA} = \mathbf{Q_A}$, and $\mathbf{M}_{BB} = \mathbf{Q_B}$. To prove the other direction of the equivalence in Definition 5 we set $\gamma = 2 - \lambda$ and repeat the same steps.

Now, we prove that spectral folding implies (33). Let

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q_A} & \mathbf{Q_B} \\ \mathbf{Q_A^T} & \mathbf{Q_B} \end{bmatrix}. \quad (39)$$

We will first show that $\mathbf{Q}$ is block diagonal. Using the spectral folding assumption and given two generalized eigenvectors $\mathbf{u}$ and $\mathbf{v}$ with unit $\mathbf{Q}$-norm, we have that $\mathbf{J_u}$ and $\mathbf{J_v}$ are also generalized eigenvectors of unit $\mathbf{Q}$-norm. In addition, $\mathbf{J_u}$ is $\mathbf{Q}$-orthogonal to $\mathbf{J_v}$, and $\mathbf{u}$ is $\mathbf{Q}$-orthogonal to $\mathbf{v}$. Therefore, given the matrix of generalized eigenvectors $\mathbf{U}$,

$$\mathbf{U}^T (\mathbf{Q} + \mathbf{JQ}) \mathbf{U} = 2 \mathbf{I}. \quad (40)$$

We replace the right hand side of (40) with $2\mathbf{U}^T \mathbf{Q} \mathbf{U} = 2 \mathbf{I}$. After simplification we obtain $\mathbf{JQ} = \mathbf{Q}$, meaning that $\mathbf{Q_A} = 0$, and $\mathbf{Q_B} = 0$.

Next, we prove that $\mathbf{Q_A} = \mathbf{M}_{AA}$ and $\mathbf{Q_B} = \mathbf{M}_{BB}$. Using the spectral folding property, and given $\mathbf{u}$, a generalized eigenvector with eigenvalue $\lambda$, we obtain:

$$\mathbf{M}(\mathbf{I} + \mathbf{J}) \mathbf{u} = (\lambda \mathbf{Q} + (2 - \lambda) \mathbf{Q}) \mathbf{u} \quad (41)$$

$$\mathbf{M}(\mathbf{I} - \mathbf{J}) \mathbf{u} = (\lambda \mathbf{Q} - (2 - \lambda) \mathbf{Q}) \mathbf{u}. \quad (42)$$

(41) and (42) imply $2\mathbf{M}_{AA} \mathbf{u}_A = 2\mathbf{Q_A} \mathbf{u}_A$ and $2\mathbf{M}_{BB} \mathbf{u}_B = 2\mathbf{Q_B} \mathbf{u}_B$, respectively. Gathering all these equations for all eigenvectors into matrix form, we obtain

$$\begin{bmatrix} \mathbf{M}_{AA} & 0 \\ 0 & \mathbf{M}_{BB} \end{bmatrix} \mathbf{U} = \begin{bmatrix} \mathbf{Q_A} & 0 \\ 0 & \mathbf{Q_B} \end{bmatrix} \mathbf{U}. \quad (43)$$

Since $\mathbf{U}$ is invertible, we obtain that $\mathbf{M}_{AA} = \mathbf{Q_A}$ and $\mathbf{M}_{BB} = \mathbf{Q_B}$.

F. Generalized Filter Banks on Arbitrary Graphs

We use the spectral folding property (Definition 5) and Theorem 4 to construct perfect reconstruction (Theorem 5) and $\mathbf{Q}$-orthogonal (Theorem 6) filter banks. Our proofs (Appendices IX-B and IX-C) follow closely the proofs of Theorem 2 and Theorem 3 for the bipartite case [10], with the main difference being the use of spectral graph filters of the $(\mathbf{M}, \mathbf{Q})$-GFT.
state the conditions for perfect reconstruction and Q-orthogonal filter banks (see Definition 3) next.

**Theorem 5:** Consider a positive semi-definite variation operator $M$ and a vertex partition $A, B, Q$ is chosen according to Theorem 4 so that the $(M, Q)$-GFT has the spectral folding property. A two-channel filter bank with SGFs

$$H_i = h_i(Z) = U h_i(A)U^T Q \quad \text{and}$$

$$G_i = g_i(Z) = U g_i(A)U^T Q$$

is perfect reconstruction, if and only if for all $\lambda \in \sigma(M, Q)$

$$g_0(\lambda) h_0(\lambda) + g_1(\lambda) h_1(\lambda) = 2,$$

$$h_1(\lambda) g_1(2 - \lambda) - h_0(\lambda) g_0(2 - \lambda) = 0.$$  

**Theorem 6:** Under the conditions of Theorem 5, a two-channel filter bank is Q-orthogonal if and only if,

$$h_0^2(\lambda) + h_1^2(\lambda) = 2,$$

$$h_1(\lambda) h_1(2 - \lambda) - h_0(\lambda) h_0(2 - \lambda) = 0,$$

for all $\lambda \in \sigma(M, Q)$.

Note that the PR and Q-orthogonality conditions on $h_i$ and $g_i$ are the same for BFBs and GFBs. Thus filter designs for BFBs can be used to construct GFBs. From a computational perspective, spectral graph filters that have polynomial filter implementations are more desirable. As in the BFB case, there are no exactly polynomial solutions to the Q-orthogonality conditions (47) and (48) [11]. However, the biorthogonal filters from (28) also satisfy the PR conditions of Theorem 5 and have polynomial solutions. Since the biorthogonal designs from [11] are designed to approximately satisfy conditions (47) and (48), they are also approximately Q-orthogonal.

**Remark 2:** While I-orthogonal filter banks are preferable in many scenarios (e.g., for coding applications), nearly I-orthogonal solutions are often used in practice if they provide other useful properties. As an example, I-orthogonal filter banks cannot be constructed with finite impulse response linear phase filters (other than the Haar filters) [5], thus nearly I-orthogonal biorthogonal filters have been used because of the advantages of symmetry for image coding applications [30]. For GFBs, we will show in Section VI that by optimizing the downsampling sets based on a numerical stability criteria, the matrix Q can be close to a diagonal matrix.

**Remark 3:** The zeroDC filter bank [11] was proposed so that DC (constant) signals are mapped to the lowest graph frequency ($\lambda = 0$). This is achieved by multiplying the input signal by $D^{1/2}$ before applying the analysis filter bank (based on the $(L, I) - GFT$), and multiplying by $D^{-1/2}$ at the output of the synthesis filter bank. This ensures that a constant input signal has zero response in the high pass channel. [11] showed that biorthogonal zeroDC filter banks can be implemented with polynomials of the random walk Laplacian of a bipartite graph. The zeroDC filter banks can be derived as a special case of our framework with the $(L, D)$-GFT, by noticing that for bipartite graphs with Laplacian $L$, Theorem 4 leads to choosing $Q = D$.

**G. Tree Structured Generalized Filter Banks**

Tree structured GFBs are formed by concatenating two-channel GFBs. For all resolution levels, the graphs, variation operators and sampling sets are given and fixed, but otherwise arbitrary. See Fig. 4 for an example with $L = 3$ levels.

We assume the input signal is at resolution $L$, thus $a_L = x$. The outputs of the low and high pass channels at resolution $\ell < L$ are called approximation and detail coefficients, and are denoted by $a_\ell$ and $d_\ell$, respectively. The sampling sets obey, $V = A_L$, and for $\ell < L$, $V_\ell = A_\ell \cup A_{\ell+1}$ and $B_\ell = A_{\ell+1} \cup A_\ell$. The graph at resolution $\ell$ is denoted by $G_\ell = (V_\ell, E_\ell)$, and has variation operator $M_\ell$ with corresponding inner product matrix $Q_\ell$, chosen so that the $(M_\ell, Q_\ell)$-GFT has the spectral folding property. We will consider a family of PR filter banks, with analysis and synthesis operators at resolution $\ell$ denoted by $T_{a,\ell}$ and $T_{s,\ell} = T_{a,\ell}^{-1}$ respectively. The analysis equation at resolution $\ell$ is given by

$$T_{a,\ell} a_{\ell+1} = \begin{bmatrix} a_{\ell}^T & d_{\ell}^T \end{bmatrix}^T,$$

where $a_\ell = S_{A_\ell} H_{0,\ell} a_{\ell+1}$, and $d_\ell = S_{\partial B_\ell} H_{1,\ell} a_{\ell+1}$. The synthesis operator implements

$$a_{\ell+1} = T_{s,\ell} \begin{bmatrix} a_{\ell}^T & d_{\ell}^T \end{bmatrix}^T = G_{0,\ell} S_{A_\ell} a_\ell + G_{1,\ell} S_{\partial B_\ell} d_\ell.$$  

After applying (50) recursively we can represent $x$ as a linear combination of coefficients at various resolutions $c = [a_0, d_0, \ldots, d_{L-1}]^T$, and define the synthesis operator of the tree structured filter bank $T_s$ via the equation

$$x = T_s c.$$  

Because $T_s$ is the composition of the synthesis operators $T_{s,\ell}$ at multiple resolutions, there is a corresponding analysis operator $T_a = T_{s,\ell}^{-1}$, that can be implemented as a composition of analysis operators $T_{a,\ell}$.

**V. PROPERTIES, INTERPRETATION AND EXAMPLES**

The term **graph Fourier transform** (GFT) associated to the eigenvectors of a graph operator is often justified by the fact that the discrete Fourier transform (DFT) diagonalizes the adjacency matrix of circulant graphs. More generally, the frequency interpretation of the $(M, I)$-GFT is justified from its variational definition, i.e., (4) and (5). In this section we motivate the use of the proposed $(M, Q)$-GFTs for graph signal representation, provide insights about its frequency interpretation, and through examples help further understand the role of $Q$. 

---

**Fig. 4.** Tree structured analysis filter bank.
A. Spectral Properties of the \((M, Q)\)-GFT

1) Eigenvalue Bounds: For any graph with normalized Laplacian \(L = D^{-1/2}LD^{-1/2}\), its eigenvalues belong to the [0,2] interval, moreover \(\sigma(L, D) = \sigma(L, I) \subset [0,2] [39]\). Also, for the \((L, D)\)-GFT and \((L, I)\)-GFT, \(\lambda_n = 2\) if and only if the graph is bipartite [39]. For the \((M, Q)\)-GFT there is a direct result.

Proposition 3: If the \((M, Q)\)-GFT has the spectral folding property then \(\sigma(M, Q) \subset [0,2]\), and \(\lambda_n = 2\) if and only if \(\lambda_1 = 0\).

Proof: For all \(x, x^TMx \geq 0\) and \(x^TMJx \geq 0\), therefore
\[
0 \leq x^TMx \leq x^TMx + x^TMJx = 2x^TQx,
\]
where we use the identity \(M + JMJ = 2Q\). Now for any generalized eigenvector \(u\) with \(Q\)-norm and eigenvalue \(\lambda\), we have \(u^TMu = \lambda u^TQu = \lambda\), and using (52) we obtain
\[
0 \leq u^TMu \leq \lambda \leq 2.
\]

The statement, \(\lambda_n = 2\) if and only if \(\lambda_1 = 0\), follows directly from the spectral folding property.

2) Middle Frequency \(\lambda = 1\): In the bipartite case, the subspace associated to the eigenvectors corresponding to \(\lambda = 1\) has the least frequency discrimination because energies at this frequency contribute equally to the low pass and high pass channels. This can be seen from (27) evaluated at \(\lambda = 1\), which shows that \(h_0(1)^2 = h_1(1)^2\). Since the subspace associated to \(\lambda = 1\) has dimension at least \(|A| - |B|\), bipartite graphs with more balanced partitions are usually preferred [24]. There is a similar result for the \((M, Q)\)-GFT.

Proposition 4: If the \((M, Q)\)-GFT has the spectral folding property with vertex partition \(A, B\), then the multiplicity of \(\lambda = 1\) is at least \(|A| - |B|\).

Proof: We will show that
\[
|\{i : \lambda_i = 1\}| \geq |A| - |B|.
\]
(54)

If \(\lambda = 1\), and \(u\) is the corresponding generalized eigenvector, then \(Zu = u\), hence the dimension of the null space of \(I - Z\) is equal to the multiplicity of \(\lambda = 1\). Note that \(I - Z\) has the sparsity pattern of a bipartite adjacency matrix, thus \((I - Z)^2\) is block diagonal, and
\[
|\{i : \lambda_i = 1\}| = \dim \ker(I - Z) = n - \text{rank}(Z_{AB}) - \text{rank}(Z_{BA}) = n - 2\text{rank}(M_{AB}) \geq n - 2\min(|A|, |B|) = |A| - |B|.
\]

The rank of \(I - Z\) is the sum of the ranks of the blocks (this can be proven using the fact that \((I - Z)^2\) is block diagonal). Then the rank of each individual term is equal to the rank of \(M_{AB}\), because \(Q\) is non-singular. The rank of a matrix is upper bounded by the minimum between the number of rows and columns. Finally we use the fact that \(n = |A| + |B|\).

3) Multiplicity of \(\lambda_1\): When \(M\) is a generalized Laplacian the multiplicity of the smallest eigenvalue is equal to the number of connected components of the graph [63]. The smallest generalized eigenvalue also has this property.

Proposition 5: If \(M \geq 0\) is a generalized Laplacian, that is, \(M_{ij} \leq 0\) for all \(i \neq j\), the \((M, Q)\)-GFT has the spectral folding property, and \(n \geq 2\), then the multiplicity of \(\lambda_1\) is equal to the number of connected components of the graph.

Proof: Let \(\lambda_1\) be the smallest generalized eigenvalue. The operation \(M = M - \lambda_1Q\) preserves generalized eigenvectors, and shifts all generalized eigenvalues by \(-\lambda_1\). Then the multiplicity of \(\lambda_1\) is equal to dimension of the null space of \(M\), thus the desired result is proven if we can show that \(M\) is a positive semi-definite generalized Laplacian. To show that \(M \geq 0\), note (6) implies that \(M = \sum_{i=1}^n (\lambda_i - \lambda_1)Qu_iu_i^TQ\), thus \(x^TMx \geq 0\) for any \(x\). To see that \(M\) is also a generalized Laplacian we use the identity
\[
M = (1 - \lambda_1)Q + M - Q = (1 - \lambda_1)Q + M - Q.
\]
(60)

Because \(n \geq 2\) and the spectral folding property, \(\lambda_1 \leq 1\), and \((1 - \lambda_1)Q\) is a generalized Laplacian. Since \(M - Q\) has zero diagonal and non positive off-diagonal entries, \(M\) must be a generalized Laplacian.

4) \((M, Q)\)-GFT Examples: \(G_1\) in Fig. 5(a) is a bipartite path graph, while \(G_2\) in Fig. 5(b) is a non bipartite graph, formed by adding a few edges to \(G_1\). All edge weights in \(G_1\) and \(G_2\) are equal to 1. For each graph we use their combinatorial Laplacian \(L\) as variation operator. The vertex partition is marked by colored vertices, where blue and red correspond to the sets \(A\) and \(B\), respectively. The Laplacian and degree matrices of \(G_i\) are denoted by \(L_i\) and \(D_i\), respectively. In Fig. 6(a) and (b) we plot the \((M, Q)\)-GFTs for graphs \(G_1\) and \(G_2\) that have the spectral folding property. Since \(G_1\) is bipartite, we plot \((L_1, D_1)\)-GFT, while for \(G_2\) we plot the \((L_2, Q_2)\)-GFT where \(Q_2\) is chosen according to Theorem 4. As predicted by Theorem 3, the generalized eigenvalues lie in [0,2] and \(\lambda_n = 2\) \((n = 11)\) for both graphs. The generalized eigenvalue/eigenvector pairs \((\lambda, u)\) and \((2 - \lambda, Ju)\) can be easily observed, as predicted by Theorem 4. Note that also for both graphs the generalized eigenvalue \(\lambda_1\) is simple, thus agreeing with the lower bound \(|\lambda| = |B| = 1\) from Proposition 4. In Fig. 6(c) we plot the \((L_2, D_2)\)-GFT of \(G_2\). The graph frequencies belong the interval [0,2], but they are not symmetric around 1. While both \((L_i, D_i)\)-GFTs are the eigenvectors of the random walk Laplacian, their basis functions are very different, specially at the higher frequencies. Some of the basis functions of \((L_2, D_2)\)-GFTs present some undesirable features such as repeated eigenvalues \((\lambda_6 = \lambda_7)\) and localized
basis functions \(u_6, u_7, u_8\). In this example the \((M, Q)\)-GFTs that have the spectral folding produce better bases for signal representation.

### B. Vertex Domain Interpretation

Traditionally, implementations of SGFs using polynomials of the variation operator \(M\) (i.e., the fundamental matrix of the \((M, I)\)-GFT) have been preferred for their efficiency (sparse matrix vector products) and interpretability (localized vertex domain operations). Since \(Q\) is not diagonal unless the graph is bipartite, \(Z = Q^{-1}M\) can be much denser than \(M\), and thus the product \(Zx\) may no longer be implemented with localized vertex domain operations.

In this section we show that \(Q^{-1}\), and thus the fundamental matrix \(Z\), can be approximated by polynomials of sparse matrices. In fact, for certain choices of vertex partitions, \(Z\) may be approximately sparse and thus the operation \(Zx\) is localized in the vertex domain. We show that this product can be described in terms of vertex domain operations involving bipartite and disconnected graphs. The complexity of SGFs as a function of the vertex partition is studied in Section VI.

For simplicity we only consider the combinatorial Laplacian, so that \(M = L\). Given a vertex partition \(A, B\), any graph can be decomposed as the sum of a bipartite graph and a disconnected graph with 2 or more connected components (the adjacency matrix of a disconnected graph is block diagonal if vertices in the same connected component are labeled consecutively). An example is depicted in Fig. 2. These graphs obey the following identities

\[
L = L^b + L^{bd}, \quad W = W^b + W^{bd}, \quad D = D^b + D^{bd},
\]

where the super indices \(bi\) and \(bd\) refer to bipartite and block diagonal, respectively (see Fig. 2). In addition, we have that

\[
L^{bd} = D^{bd} - W^{bd}, \quad L^{bi} = D^{bi} - W^{bi}.
\]

We have the following factorization for \(Z\).

**Proposition 6:** If \(M = L\) and \((L, Q)\)-GFT has the spectral folding property, then the fundamental matrix is equal to

\[
Z = Q^{-1}L = I - P^{bd}P^{bi},
\]

where \(P^{bi} = (D^{bi})^{-1}W^{bi}\), and \(P^{bd} = Q^{-1}D^{bd}\), are both right stochastic non negative matrices, with bipartite and block diagonal structure, respectively.

The proof can be found in Appendix IX-A. The matrix \(P^{bd}P^{bi}\) is a two step smoothing operator, since

\[
y_i = (P^{bi}x)_i = \begin{cases} \frac{1}{\deg(v_i)} \sum_{j \in B} w_{ij}x_j & \text{if } i \in A, \\ \frac{1}{\deg(v_i)} \sum_{j \in A} w_{ij}x_j & \text{if } i \in B, \end{cases}
\]

where a smoothed signal is obtained through linear combinations of neighbors on the complement set. Since \(P^{bd}\) is block diagonal and non negative, the second filtering step is also a low pass filter that only uses connections within \(A\) or \(B\).

As an example we consider the 8-connected grid graph in Fig. 7(a), commonly used in image processing [50]. The vertex partition (red or blue) can be used to decompose the graph into (i) a bipartite 4-connected grid (Fig. 7(b)), containing all vertical and horizontal connections, and (ii) a block diagonal graph containing the diagonal connections (see Fig. 7(c)). The adjacency matrix of the 8-connected grid is depicted in Fig. 7(g), where all the edge weights have been set to 1, and the vertices are labeled so that \(A = \{1, 2, \ldots, |A|\}\). \(P^{bd}\) and \(P^{bi}\) are approximated in Fig. 7(d) and Fig. 7(e), respectively, and their product is depicted in Fig. 7(f). While \(P^{bi}\) and \(W\) are sparse, the matrices \(P^{bd}\) and \(P^{bd}P^{bi}\) are dense. However, Fig. 7(d) and Fig. 7(f) show that most of the entries of \(P^{bd}\) and \(P^{bd}P^{bi}\) are close to zero. The approximate sparsity of \(Z\) and \(P^{bd}P^{bi}\) can be explained by the following result.

**Proposition 7:** Let \(L\) be the combinatorial Laplacian of a connected graph. If \(Q\) is chosen according to Theorem 4, then

\[
Q^{-1} = D^{-\frac{1}{2}} \left( \sum_{k=0}^{\infty} (D^{-\frac{1}{2}} W^{bd} D^{-\frac{1}{2}})^k \right) D^{-\frac{1}{2}}.
\]

**Proof:** Because the graph is connected, \(Q > 0\). Since \(Q = D - W^{bd}\), we have that \(I - D^{-1/2}W^{bd}D^{-1/2} > 0\), which implies that \(\|D^{-1/2}W^{bd}D^{-1/2}\| < 1\). Then \((I - D^{-1/2}W^{bd}D^{-1/2})^{-1} = \sum_{k=0}^{\infty} (D^{-1/2}W^{bd}D^{-1/2})^k\), and \(Q^{-1} = D^{-1/2}(I - D^{-1/2}W^{bd}D^{-1/2})^{-1}D^{-1/2}\).

If we only keep the first \(m\) terms in (65), the norm of the remaining terms decays exponentially as \(O((|D^{-1/2}W^{bd}D^{-1/2}|)^{m+1})\). If the vertex partition is designed so that \(W^{bd}\) is sparse and has small weights, this error decays faster. While in this work we do not approximate \(Q^{-1}\), we propose a vertex partitioning algorithm (Section VI) that minimizes the \(\ell_1\) norm of \(D^{-1/2}W^{bd}D^{-1/2}\) that leads to a sparse \(Q\), and an approximately sparse fundamental matrix \(Z\).
Minimization of $E[\|x_A + C_{AB}x_B\|^2]$ corresponds to optimal linear prediction of $x_A$ from $x_B$, which fortunately, has a closed form solution for Gaussian distributions.

**Proposition 8**: Given a vertex partition $\mathcal{A}, \mathcal{B}$, if $M = \Sigma^{-1}$, then $Z = Q^{-1}M$ from (34) is the minimizer of (66), and
\[
Zx = x - \frac{E[x_A|x_B]}{E[x_B|x_A]} = x - \left(\Sigma_{AB}\Sigma_{BB}^{-1}x_B\right).
\]

The proof of this result is a direct consequence of
\[
\Sigma_{AB}\Sigma_{BB}^{-1} = -M_{A,B}M_{AB}, \quad \Sigma_{B,A}\Sigma_{AA}^{-1} = -M_{B,A}M_{BA},
\]
which follows from $MQ = I$. From the $(M, Q)$-GFT perspective, $Z$ is a high pass filter (see Section V-B). Proposition 8 illustrates that the filtering operation $Zx$ is the prediction error of an optimal linear predictor for the Gaussian distribution. Now we take the lazy filter bank from Fig. 3 and form a tree structured filter bank as in Fig. 4. Filters at resolution $\ell$ are given by $H_{0,\ell} = I$, $H_{1,\ell} = Z_\ell$, $G_{0,\ell} = 2I - Z_\ell$, and $G_{1,\ell} = I$. We will show that Proposition 8 implies that this tree structured filter bank produces sub-bands with uncorrelated coefficients. Assume that the input signal $x = a_\ell$ is a zero mean Gaussian with covariance $\Sigma = M^{-1}$. Since the low pass channel corresponds to down-sampling $a_{\ell+1}$ on the set $\mathcal{A}_\ell$, we have that for $\ell < L$
\[
a_\ell = x_{\ell A},
\]
is a $|\mathcal{A}_\ell|$-dimensional zero mean Gaussian vector with covariance matrix $\Sigma_{\ell A, \ell A}$. Therefore, at resolution $\ell$ we will use the variation operator $M_\ell = \Sigma_{\ell A, \ell A}$, which can be computed using Schur complements [68]
\[
M_{\ell-1} = (M_\ell)_{\ell A, \ell A} - (M_\ell)_{\ell A \ell A}B_\ell((M_\ell)_B)\Sigma_{B, A, \ell A}^{-1}(M_\ell)_B B_\ell.
\]
(71)
The detail coefficients are
\[
d_\ell = x_{B_\ell} - \Sigma_{B, \ell A}^{-1}\Sigma_{A, \ell A}^{-1}x_{\ell A}.
\]
(72)
The coefficient vector obtained after iterating the filter bank $L$ times is given by $c = [a_0^T, d_0^T, \ldots, d_{L-1}^T]^T = T_a x$. These coefficients are uncorrelated, more precisely:

**Proposition 9**: The inverse covariance matrix of $c$ is equal to
\[
(E[cc^\top])^{-1} = \begin{bmatrix}
M_0 & 0 & \cdots & 0 \\
0 & Q_{1,0} & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & Q_{1,L-1}
\end{bmatrix},
\]
(73)
where $Q_{1,\ell}$ is the submatrix given by
\[
Q_{1,\ell} = Q_{\ell+1}(E_{\ell, B_\ell}).
\]
(74)
The proof follows from Proposition 8 (see Appendix IX-D).

### VI. Vertex Partitioning

In this section we assume $M$ is given, and we wish to optimize $\mathcal{A}$ and $\mathcal{B} = V \setminus \mathcal{A}$, as a function of $M$. We start by formulating this problem as an optimization of the condition number of $Q$. We provide some theoretical justifications for this choice of
objective based on complexity and stability of GBFs. We end the section with our proposed solution.

A. Problem Formulation

J defined in (23) completely characterizes the vertex partition since \( J_{ii} = 1 \) if \( i \in \mathcal{A} \), and \( J_{ii} = -1 \) if \( i \in \mathcal{B} \). Consequently, we will use a vector \( f \) containing the diagonal terms of \( J \) as an optimization variable. We propose to minimize \( \kappa(Q) = \|Q\|\|Q^{-1}\| \), the condition number of \( Q \), subject to spectral folding and balanced partition constraints, namely,

\[
\begin{align*}
\min_{f, f^2 = 1} & \quad \kappa(Q) \\
\text{s.t.} & \quad Q = \frac{1}{2}(M + \text{diag}(f)M\text{diag}(f)) \\
& \left| \sum_{i \in V} f_i \right| \leq 1.
\end{align*}
\]

The constraint \( f^2 = 1 \) ensures that the vector \( f \) has entries equal to 1 or -1. The downsampling sets are recovered as \( \mathcal{A} = \{ i : f_i > 0 \} \), and \( \mathcal{B} = \mathcal{A}^c \). To ensure the \((M, Q)\)-GFT has the spectral folding property we incorporate the constraint \( Q = \frac{1}{2}(M + \text{diag}(f)M\text{diag}(f)) \). When there is an even number of nodes, the constraint \( |\sum_{i \in V} f_i| \leq 1 \) becomes \( \sum_{i \in V} f_i = 0 \) so that \( |\mathcal{A}| = |\mathcal{B}| \). When the number of nodes is odd, the constraint \( |\sum_{i \in V} f_i| \leq 1 \) becomes \( |\sum_{i \in V} f_i| = 1 \), so that \( |\mathcal{A}| - |\mathcal{B}| = 1 \). These constraints guarantee a balanced partition, which is a necessary condition for good frequency selectivity (see Section V-A for more details).

B. Justification for Condition Number Minimization

The reason for seeking vertex partitions so that the \( Q \) has a small condition number is due to numerical stability and complexity of GBFs. For large graphs, performing eigendecomposition to implement a SGF is infeasible due to high computational complexity, thus polynomial graph filters are used instead. For BFBs, a polynomial graph filter of degree \( d \) can be implemented by sparse matrix vector products (of the normalized Laplacian) with complexity \( \mathcal{O}(dl|\mathcal{E}|) \) [10, 12]. For arbitrary graphs, polynomials of \( Z \) require the computation of products of the form \( Zx \). However, although \( M \) may be sparse, the fundamental matrix \( M = Q^{-1}M \) is not. Hence, a naive implementation of the product \( Zx \) has complexity \( \mathcal{O}(n^2) \), while direct computation of \( Z = Q^{-1}M \) has complexity \( \mathcal{O}(n^3) \) due to matrix inversion. A more efficient approach is to decompose the product \( y = Zx \) into: 1) computing a sparse matrix vector product \( w = Mx \), and 2) solving a system of equations \( Qy = w \). The second step is the computation bottleneck, since it requires solving a linear system.

Large scale positive semi-definite linear system are solved by iterative methods such as the Conjugate Gradient (CG), whose complexity per iteration is that of a sparse matrix vector product with \( Q \). The condition number of \( Q \) is directly responsible for: 1) the convergence rate of CG algorithms (and thus the total number of iterations), and 2) stability to perturbations of the system \( Qy = w \). From a computational perspective, the best case scenario occurs when \( Q \) is sparse and has a small condition number. In practice this is attained when \( Q \) is diagonal, and the graph is bipartite, as will be shown in the next subsection.

C. Approximate Solution for Generalized Laplacians

Because of the non convex constraint \( f^2 = 1 \), (75) is a non convex minimization problem. In this section we obtain an upper bound for \( \kappa(Q) \), which allows us to pose an alternative, more tractable optimization problem. For the rest of this section we assume that \( M \) is a positive semi definite generalized graph Laplacian, that is, \( M = V - W \), where \( V \) is a diagonal matrix with positive entries and \( W \) is a non negative matrix with zero diagonal. Since \( Q \) is given by Theorem 4, we have that \( Q = V - W^{bd} \geq 0 \), where \( W^{bd} = (1/2)(W + \text{diag}(f)W\text{diag}(f)) \) is the block diagonal part of \( W \). We can bound \( \kappa(Q) \) with

\[
\kappa(Q) \leq \frac{\kappa(V)}{(1 + \rho(A))(1 - \rho(A))},
\]

where \( \kappa(V) \) is the condition number of \( V \), and \( \rho(A) = \|V^{-1/2}W^{bd}V^{-1/2}\| \). A derivation of this bound is given in Appendix IX-E. When the graph is bipartite, \( f \) can be chosen so that \( Q \) is diagonal resulting in \( \rho(A) = 0 \), making the bound tight. Since the right side of (76) is decreasing with \( \rho(A) \), we can minimize \( \rho(A) \) as a function of the vertex partition. As a simplification, we use the sequence of bounds for the operator norm of a matrix, \( \|A\| \leq \|A_{\mathcal{F}}\| \leq |\mathcal{A}| = \sum_{i,j} |a_{ij}| \), which results in \( \rho(A) \leq \sum_{i,j} (1 + f_i f_j) w_{ij} / \sqrt{\nu_{ij}} \). We propose solving instead:

\[
\begin{align*}
\min_{f, f^2 = 1} & \quad \sum_{i,j} (1 + f_i f_j) \tilde{w}_{ij} \quad \text{s.t.} \quad \left| \sum_{i \in V} f_i \right| \leq 1
\end{align*}
\]

where \( \tilde{w}_{ij} \) is the \( ij \) entry of \( \tilde{W} = V^{-1/2}W^{bd}V^{-1/2} \). Using the identity \( \sum_{i,j} f_i f_j \tilde{w}_{ij} = f^\top \tilde{W} f = \sum_{i,j} \tilde{w}_{ij} - f^\top \tilde{L} f \), we have that (77) is equivalent to

\[
\begin{align*}
\max_{f, f^2 = 1} & \quad f^\top \tilde{L} f \quad \text{s.t.} \quad \left| \sum_{i \in V} f_i \right| \leq 1,
\end{align*}
\]

where \( \tilde{D} = \text{diag}(\tilde{W}) \), and \( \tilde{L} = \tilde{D} - \tilde{W} \). (78) is an instance of weighted maximum cut (WMC) [71], with an additional balanced partition constraint. It is well known that WMC is NP hard, thus we consider a spectral partitioning approximation [9, 72], that computes \( u_n = \arg \max_{u_{\mathcal{A}} = 1} u^\top \tilde{L} u \), and sets \( f = \sigma(u_\mathcal{A} - \tau) \), and \( \mathcal{A} = \{ i \in V : f_i = 1 \} \). The parameter \( \tau \in \mathbb{R} \) can be tuned to ensure \( \left| \sum_{i \in V} f_i \right| \leq 1 \).

D. Minnesota Graph Example

We implement the proposed max-cut vertex partitioning algorithm and a random partitioning algorithm that assigns nodes to \( \mathcal{A} \) with probability 1/2. We consider the Minnesota road graph [69], which has \( n = 26490 \) nodes and \( s = 3302 \) edges with unit weights. Its adjacency matrix \( W \) is depicted in Fig. 8(a). We use the combinatorial Laplacian \( L \) as variation operator. In Fig. 8 we display the sparsity patterns of \( W^{bd} = D - Q = (1/2)(W + \text{diag}(f)W\text{diag}(f)) \), where \( f \) has been obtained via the proposed max-cut sampling (Fig. 8(b)) and via random partitioning (Fig. 8(c)). Random vertex partition produces a less sparse matrix \( Q \), which retains 49.6% of the edges from \( W \). In contrast, the proposed max-cut vertex partitioning, produces a matrix \( Q \) which only has 14.56% of the edges of \( W \), and is much closer to \( D \). We also compare the condition numbers
Fig. 8. Comparison of matrices $W^{bd} = \frac{1}{2} (W + \text{diag}(f) W \text{diag}(f))$ using different vertex partitioning algorithms for the Minnesota graph [69]. Vertex partitions are marked by blue/yellow circles. Figures were generated using the toolbox [70].

Fig. 9. Distribution of the normalized condition number of $Q$ when using random vertex partitioning.

of $Q$ as a function of the vertex partition. We generate 1000 realizations of random partitions, and display the distribution of $\kappa(Q)/\kappa(D)$ in Fig. 9. The proposed max-cut partition achieves $\kappa(Q)/\kappa(D) = 1.863$, and our simulation shows that the normalized condition number is always greater than 1.863 for the random partitions. Because the Minnesota graph is very sparse ($s = 3302 \approx 1.25n$), the sub-graph associated to $P^{bd}$ (and $Q$) have many more than 2 connected components, resulting in both $Z$ and $Q^{-1}$ also being exactly sparse. When using max-cut partitioning $Z$ has 1.2806 $s$ non-zero off diagonal entries, but when random partitioning is used the number of non-zero off diagonal entries in $Z$ ranges from 2.7733 $s$ to 4.4170 $s$, with average 3.3559 $s$. In the next section, we show that this algorithm can be efficiently implemented for large graphs (since it only requires computing an eigenvector), while improving the signal representation.

VII. EXPERIMENTS ON 3D POINT CLOUDS

In this section we present numerical results showing that: 1) GFB can be implemented efficiently on graphs with hundreds of thousands of nodes, 2) the basis functions of GFBs are localized in the vertex domain and 3) for point clouds, GFBs with optimized vertex partitions can provide better signal representations than BFBs.

A. Vertex Domain Localization

We start by visualizing the vertex domain behavior of the GFB basis functions, i.e., the columns of the synthesis operator $T_s$ of a tree structured filter bank (see Section IV-G and Fig. 4). We compare them to those of spectral domain sampling (SDS) filter banks [47], which are close to the proposed GFB in terms of graph properties (see Table I). We show that although $Z$ is (numerically) dense, for point clouds, $Z$ is approximately sparse and polynomial filters of $Z$ are localized in the vertex domain.

Since SDS filter banks require full eigendecomposition, for complexity reasons we use a small Bunny point cloud, which has $n = 2503$ nodes. For both filter banks we fix the number of levels to $L = 3$ and consider two designs: biorthogonal CDF-9/7 filters, and orthogonal Meyer filters. Note that these filters are exactly those developed for traditional 1D wavelets. For graph construction, we use nonnegative kernel regression (NNK) [73] initialized with k-nearest-neighbors (KNN) using $K = 20$ and inverse distance as edge weights. For GFBs, we use the spectral partitioning algorithm from Section VI. Fig. 10 displays low frequency basis functions, centered at a single node. For visualization purposes, we normalize the basis functions, so their entries have magnitude at most 1. For both filter designs, the basis functions of the GFBs are more localized than their SDS counterparts. The GFB with CDF-9/7 filters is implemented using polynomials of $Z$, thus producing the most localized basis functions.

B. Representation of 3D Point Cloud Attributes

3D point clouds consist of list of point coordinates $V = [v_i] \in \mathbb{R}^{n \times 3}$, and color attributes $A \in \mathbb{R}^{n \times 3}$. A graph is constructed where each point is assigned to a node and the edge weight between nodes $i$ and $j$ is $w_{ij} = 1/||v_i - v_j||$. A sparse edge set is obtained using the KNN and NNK [73] graph

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
construction algorithms. We consider the first frame of the loot and longdress sequences of the 8iVFBv2 dataset [32], which have \( n = 784, 142 \) and \( n = 765, 821 \) points respectively. We implement the tree structured GFB of Fig. 4 with \( L = 7 \) levels, using biorthogonal analysis filters \( h_0(\lambda) = \frac{1}{2\pi a_0} (2 - \lambda)(1 + \lambda) \), and \( h_1(\lambda) = a_0\lambda \), with gain \( a_0 = 0.735 \), and synthesis filters computed with (28).

We apply the GFB to each column of the color matrix \( A \) independently, and obtain the approximation by only keeping the low-pass coefficients \( a_0 \) and a subset of the high-pass coefficients \( d_i \) for \( 0 \leq i \leq m - 1 \), with \( m = 1, \ldots, L \), and zeroing out the rest. For each filter bank, we tested various graph constructions and chose the one giving the best performance. We tested KNN graphs with \( K \in \{5, 10, 15, 20\} \) and NNK graphs initialized with KNN with \( K = 20 \). To construct bipartite graphs, we take a KNN or NNK graph, a random vertex partition \( A, B \), and use the subgraph consisting of only the edges from \( A \) to \( B \). For GFBs, the best results (higher PSNR) are obtained with NNK graphs, while for BFB the best results are obtained with KNN graphs. Fig. 11 shows the PSNR between the color signal and its approximation as a function of the number of high pass channels (\( m \)). As expected, the BFB with the normalized Laplacian has the worst performance, which can be attributed to the non constant DC signal [11], [28]. By using the BFB with the \((L, D)\)-GFT, which results in the random walk Laplacian, we recover the zeroDC filter bank from [11], which has a much improved performance, consistent with previous studies [11], [28]. The proposed GFB with random sampling always outperforms the best BFB, although by a small margin. When using optimal sampling with GFBs, performance is again improved consistently across datasets.

C. Complexity

We compute the run time of the tree structured analysis and synthesis filter banks with \( L = 7 \) levels using the biorthogonal filters from the previous section, applied to the first 20 frames of the “longdress” sequence. These point clouds have an average of approximately 795,000 points per frame. We compare BFBs constructed with KNN graphs and random balanced partitioning (BFF KNN), GFB constructed with KNN graphs and max-cut based vertex partitioning (GFB NNK), NNK graphs with varying parameters \( K \). The complexity of our implementation is dominated by graph construction (KNN and NNK), vertex partitioning, and solving the sparse linear system of the form \( Qz = u \). The complexity of each of these components is proportional to the graph sparsity, and the parameter \( K \) in KNN. In Fig. 12 we plot the average run time over 20 trials as a function of \( K \). Note that \( K \) serves as a proxy for graph sparsity since the number of edges in a KNN and NNK graphs is \( \mathcal{O}(nK) \). The figure shows that our implementations of GFB has a run time higher than their BFB counterparts, but still scales approximately linearly with \( K \).

VIII. Conclusion

This paper proposed two-channel filter banks on arbitrary graphs with positive semi definite variation operators \( M \) and \( N \). The superiority of GFBs is consistent for other frames, and other sequences of the 8i dataset. Those results are omitted due to space.
arbitrary vertex partitions $A, B$ for downsampling. Because of the spectral folding property, previous designs were only valid for the normalized Laplacian of bipartite graphs. Our main contribution is showing that the spectral folding property is satisfied by the generalized eigenvalues and eigenvectors of arbitrary graphs, if the inner product matrix is properly chosen. Based on this, we proposed generalized filter banks (GFB) implemented with spectral graph filters of the generalized eigenvectors. We also studied other theoretical aspects of GFBs including: properties of the generalized eigenvalues, vertex domain localization and probabilistic interpretations. We showed that even though GFBs can use arbitrary vertex partitions for downsampling, partitions that have a large cut (the sum of edge weights from $A$ to $B$ is large) lead to computationally efficient and stable implementations. Our numerical results show that indeed, GFBs have localized basis functions and they can be efficiently implemented on large graphs with hundreds of thousands of nodes, while outperforming bipartite filter banks on a signal representation tasks. Some important directions for future work include:

Tree structured filter banks: We constructed tree structured GFBs by concatenating two channel GFBs. There is ample room for study of properties of these tree structured filter banks from both probabilistic perspectives (beyond the lazy filter bank case), and the design of variation and downsampling operators. In the supplementary material we obtain frame bounds that reveal some of these multi resolution dependencies.

Vertex partitioning: Better vertex partitions could be obtained using different approximations to (75) or other objective functions, e.g., by using probabilistic models as in Section V, or techniques from the graph signal sampling literature [74].

REFERENCES

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

[2] J. 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.

[3] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[4] A. Ortega, Introduction to Graph Signal Processing, Cambridge, U.K.: Cambridge Univ. Press, 2022.

[5] M. Vetterli and J. Kovacevic, Wavelets and Subband Coding. Englewood Cliffs, NJ, USA: Prentice-hall, 1995.

[6] I. Daubechies, Ten Lectures on Wavelets, vol. 61, Philadelphia, PA, USA: Siam, 1992.

[7] P. Vaidyanathan, Multirate Systems and Filter Banks. Noida, India: Pearson Education India, 2006.

[8] S. Mallat, A Wavelet Tour of Signal Processing. New York, NY, USA: Elsevier, 1999.

[9] D. I. Shuman, “Localized spectral graph filter frames: A unifying framework, survey of design considerations, and numerical comparison,” IEEE Signal Process. Mag., vol. 37, no. 6, pp. 43–63, Nov. 2020.

[10] 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, Jun. 2012.

[11] S. K. Narang and A. Ortega, “Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs,” IEEE Trans. Signal Process., vol. 61, no. 19, pp. 4673–4685, Oct. 2013.

[12] A. Sakizyama, K. Watambe, and Y. Tanaka, “Spectral graph wavelets and filter banks with low approximation error,” IEEE Trans. Signal Inf. Process. Netw., vol. 2, no. 3, pp. 230–245, Sep. 2016.

[13] D. B. Tay and J. Zhang, “Techniques for constructing biorthogonal bi-partite graph filter banks,” IEEE Trans. Signal Process., vol. 63, no. 21, pp. 5772–5783, Nov. 2015.

[14] H. Q. Nguyen, P. A. Chou, and Y. Chen, “Compression of human body sequences using graph wavelet filter banks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2014, pp. 6152–6156.

[15] A. Anis, P. A. Chou, and A. Ortega, “Compression of dynamic 3D point clouds using subdivisional meshes and graph wavelet transforms,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2016, pp. 6360–6364.

[16] I. Zeng, G. Cheung, Y.-H. Chao, I. Blanes, J. Serra-Sagristà, and A. Ortega, “Hyperspectral image coding using graph wavelets,” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 1672–1676.

[17] D. E. O. Tzamarias, K. Chow, I. Blanes, and J. Serra-Sagristà, “Compression of hyperspectral scenes through integer-to-integer spectral graph transforms,” Remote Sens., vol. 11, no. 19, 2019, Art. no. 2290.

[18] Y. Iizuka and Y. Tanaka, “Depth map denoising using collaborative graph wavelet shrinkage on connected image patches,” in Proc. IEEE Int. Conf. Image Process., 2014, pp. 828–832.

[19] M. Levorato, S. Narang, U. Mitra, and A. Ortega, “Reduced dimension policy iteration for wireless network control via multiscale analysis,” in Proc. IEEE Glob. Commun. Conf., 2012, pp. 3886–3892.

[20] V. K. Sharma, D. K. Srivastava, and P. Mathur, “Efficient image steganography using graph signal processing,” IET Image Process., vol. 12, no. 6, pp. 1065–1071, 2018.

[21] Y.-L. Qiao, Y. Zhao, and X.-Y. Men, “Target recognition in SAR images via graph wavelet transform and 2DPCA,” in Proc. 2nd Int. Conf. Image Graph. Process., 2019, pp. 3–7.

[22] D. B. Tay and A. Ortega, “Bipartite graph filter banks: Polyphase analysis and generalization,” IEEE Trans. Signal Process., vol. 65, no. 18, pp. 4833–4846, Sep. 2017.

[23] S. K. Narang and A. Ortega, “Local two-channel critically sampled filter-banks on graphs,” in Proc. IEEE Int. Conf. Image Proc., 2010, pp. 333–336.

[24] J. Zeng, G. Cheung, and A. Ortega, “Bipartite approximation for graph wavelet signal decomposition,” IEEE Trans. Signal Process., vol. 65, no. 20, pp. 5466–5480, Oct. 2017.

[25] A. Fang, J. Wan, Y. Tang, B. Ni, and Y. Zhu, “Admm-based bipartite graph approximation,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2019, pp. 5421–5425.

[26] E. Pavez, H. E. Eglímez, and A. Ortega, “Learning graphs with monotone topology properties and multiple connected components,” IEEE Trans. Signal Process., vol. 66, no. 9, pp. 2399–2413, May 2018.

[27] S. Kumar, J. Ying, J. D. M. Cardoso, and D. P. Palomar, “A unified framework for structured graph learning via spectral constraints,” J. Mach. Learn. Res., vol. 21, no. 22, pp. 1–60, 2020.

[28] D. E. O. Tzamarias, E. Pavez, B. Girault, A. Ortega, I. Blanes, and J. Serra-Sagristà, “Orthogonality and zero DC tradeoffs in biorthogonal graph filterbanks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2021, pp. 5509–5513.

[29] B. Girault, A. Ortega, and S. S. Narayanan, “Irregularity-aware graph Fourier transforms,” IEEE Trans. Signal Process., vol. 66, no. 21, pp. 5746–5761, Nov. 2018.

[30] D. S. Taubman and M. W. Marcellin, JPEG2000 - Image Compression Fundamentals, Standards and Practice. Norwell, MA, USA: Kluwer, 2002.

[31] K.-S. Lu and A. Ortega, “Fast graph Fourier transforms based on graph symmetry and bipartition,” IEEE Trans. Signal Process., vol. 67, no. 18, pp. 4855–4869, Sep. 2019.

[32] E. D’Eon, B. Harrison, T. Myers, and P. A. Chou, “Si voxelized full bodies-a voxelized point cloud dataset,” ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) Input Document WG11M40059/WG1M74006, 2007, pp. 49–52.

[33] D. A. Spielman and N. Srivastava, “Graph sparsification by effective resistance,” SIAM J. Comput., vol. 40, no. 6, pp. 1913–1926, 2011.

[34] M. Hein, J.-Y. Audibert, and U. V. Luxburg, “Graph Laplacians and their convergence on random neighborhood graphs,” J. Mach. Learn. Res., vol. 8, no. 6, pp. 1325–1368, 2007.

[35] P. A. Chou, M. Kotorete, and M. Krivoukač, “A volumetric approach to point cloud compression—Part I: Attribute compression,” IEEE Trans. Image Process., vol. 29, pp. 2203–2216, 2019.

[36] M. Krivoukač, P. A. Chou, and M. Kotorete, “A volumetric approach to point cloud compression—Part II: Geometry compression,” IEEE Trans. Image Process., vol. 29, pp. 2217–2229, 2019.

[37] B. Girault, A. Ortega, and S. S. Narayanan, “Graph vertex sampling with arbitrary graph signal hilbert spaces,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2020, pp. 5670–5674.
E. Pavez, B. Girault, A. Ortega, and P. A. Chou, “Spectral folding and two-channel filter-banks on arbitrary graphs,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2021, pp. 5070–5074.

O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—Part I: Fundamentals,” IEEE Trans. Signal Process., vol. 65, no. 2, pp. 409–422, Jan. 2017.

O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—Part II: M-channel filter banks,” IEEE Trans. Signal Process., vol. 65, no. 2, pp. 423–437, Jan. 2017.

A. Sakiyama and Y. Tanaka, “Oversampled graph Laplacian matrix for graph filter banks,” IEEE Trans. Signal Process., vol. 62, no. 24, pp. 6425–6437, Dec. 2014.

D. I. Shuman, M. J. Farajii, and P. Vandergheynst, “A multiscale pyramid transform for graph signals,” IEEE Trans. Signal Process., vol. 64, no. 8, pp. 2119–2134, Apr. 2016.

S. Chen, R. Varma, A. Sandyhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory,” IEEE Trans. Signal Process., vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

S. Li, Y. Jin, and D. I. Shuman, “Scalable m-channel critically sampled filter banks for graph signals,” IEEE Trans. Signal Process., vol. 67, no. 15, pp. 3954–3969, Aug. 2019.

A. Sakiyama, K. Watanabe, Y. Tanaka, and A. Ortega, “Two-channel critically sampled graph filter banks with spectral domain sampling,” IEEE Trans. Signal Process., vol. 67, no. 6, pp. 1447–1460, Mar. 2019.

M. S. Kotzagiannidis and P. L. Dragotti, “Splines and wavelets on circulant graphs,” Appl. Comput. Harmon. Anal., vol. 47, no. 2, pp. 481–515, 2019.

V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, “Spline-like wavelet filterbanks for multisolution analysis of graph-structured data,” IEEE Trans. Signal Inf. Process. Netw., vol. 1, no. 4, pp. 268–278, Dec. 2015.

G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, “Graph spectral image processing,” Proc. IEEE, vol. 106, no. 5, pp. 907–930, May 2018.

J.-Y. Kao, A. Ortega, and S. S. Narayanan, “Graph-based approach for motion capture data representation and analysis,” in Proc. IEEE Int. Conf. Image Process., 2014, pp. 2061–2065.

S. K. Narang, Y. H. Chao, and A. Ortega, “Graph-wavelet filterbanks for edge-aware image processing,” in Proc. IEEE Stat. Signal Process. Workshop, 2012, pp. 141–144.

K.-S. Lu, E. Pavez, and A. Ortega, “On learning Laplacians of tree structured graphs,” in Proc. IEEE Data Sci. Workshop, 2018, pp. 205–209.

A. Anis and A. Ortega, “Critical sampling for wavelet filterbanks on arbitrary graphs,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2017, pp. 3889–3893.

M. Crovella and E. Kolaczyk, “Graph wavelets for spatial traffic analysis,” in Proc. IEEE 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc., 2003, vol. 3, pp. 1848–1857.

R. R. Coifman and M. Maggioni, “Diffusion wavelets,” Appl. Comput. Harmon. Anal., vol. 21, no. 1, pp. 53–94, 2006.