Multiscale Basis Dictionaries on Simplicial Complexes

Naoki Saito, Stefan C. Schonsheck, and Eugene Shvarts

Department of Mathematics
University of California, Davis

Applied Mathematics Special Seminar
Yale University
November 1, 2022

Outline

Acknowledgment

- NSF Grants: DMS-1418779, DMS-1912747, CCF-1934568, DMS-2012266
- ONR Grants: N00014-16-1-2255, N00014-20-1-2381

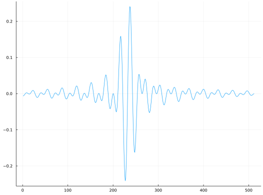
Stefan C Schonsheck (UCD)

Eugene Shvarts (UCD)

Outline

Motivation: Lifting Multiscale Basis Dictionaries to Graphs

- For conventional digital signals and images sampled on regular lattices, *Multiscale Basis Dictionaries* including *wavelet packet dictionaries* (which in turn include *wavelet bases*) and *local cosine dictionaries* have a proven track record of success, e.g.:
  - *JPEG 2000* Image Compression Standard;
  - *Modified Discrete Cosine Transform* (MDCT) in MP3;
  - Discriminant feature extraction for signal classification;
  - ...
- Want to lift/generalize these dictionaries to the graph setting for *graph signal processing* and *graph data analysis*



Shannon wavelet on $\mathbb{R}$

Graph wavelet packet vector

Roadmap So Far

- We have developed the graph versions of the *local cosine and wavelet packet dictionaries* for analysis of graph signals *sampled at nodes*.
- All these are based on the *hierarchical partitioning* of either a primary graph G or the so-called *dual graph* $G^\star$. $\Omega :=$ a domain to be hierarchically partitioned:

Classical Basis Dict.	Ω	Graph Basis Dict.	Ω
Hier. Block DCT	time axis	*HGLET*	G
LCT	time axis	*LP-HGLET*	G
Haar-Walsh WPs	time/freq. axes	*GHWT/eGHWT*	G
Cmpt-Supp. WPs	frequency axis	*LP-NGWPs*	$G^\star$
Shannon WPs	frequency axis	*NGWPs*	$G^\star$

HGLET	:=	Hierarchical Graph Laplacian Eigen Transform [Irion-Saito (2014)];
GHWT	:=	Generalized Haar-Walsh Transform [Irion-Saito (2014)];
eGHWT	:=	extended GHWT [Saito-Shao (2022)];
NGWPs	:=	Natural Graph Wavelet Packets [Cloninger-Li-Saito (2021)];
LP-HGLET/NGWPs	:=	Lapped-HGLET/NGWPs [Li (2021)]

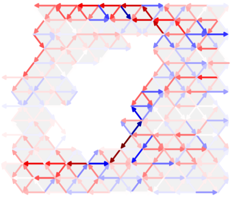
Underlying Philosophy/Basso Continuo:

Split $\Longrightarrow$ "Organize" $\Longrightarrow$ Merge

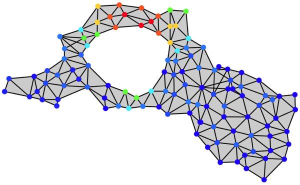
Higher-Order Graph Signals

Recently there has been great interest in analyzing and processing *higher-order signals* on graphs.

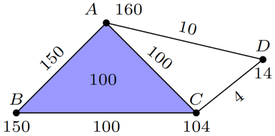
- Data are sampled over oriented *k-simplices* of a graph for some $k \in \mathbb{N}$
  - For $k = 0$, these signals take values over nodes of a graph as usual
  - For $k = 1$, these signals take values over oriented edges of a graph
  - For $k = 2$, these signals take values over oriented triangles of a graph
- Examples: regional weather data, molecular chemistry, neuronal networks, social networks, discrete exterior calculus/geometry, . . .



Buoys drifting around Madagascar



Gene expression correlations [Govek et al. (2019)]



Authorship Graph [Ebli et al. (2022)]

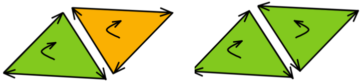
Outline

Representing Higher-Order Graphs

- A *simplicial complex* represents a combinatorial description of a topological space that can be represented and handled by a computer
- The k-simplices in a simplicial complex are typically captured by *boundary matrices* B_{k-1}, B_k expressing adjacency and relative orientation of each k-simplex σ with each $(k-1)$-simplex α or $(k+1)$-simplex β respectively.

$$[B_{k-1}]_{\alpha\sigma} = \begin{cases} 1 & \alpha,\sigma \text{ have consistent orientation} \\ -1 & \alpha,\sigma \text{ have inconsistent orientation} \\ 0 & \text{otherwise} \end{cases}$$

$$[B_k]_{\sigma\beta} = \begin{cases} 1 & \sigma,\beta \text{ have consistent orientation} \\ -1 & \sigma,\beta \text{ have inconsistent orientation} \\ 0 & \text{otherwise} \end{cases}$$



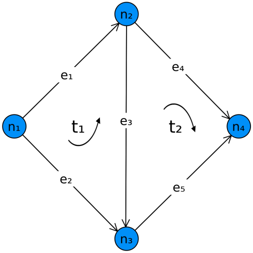
Hodge Laplacian

- The *Hodge Laplacian* [e.g., L.-H. Lim: *SIAM Review* (2020); M. T. Schaub et al.: *Signal Process.* (2021)] provides a spectral decomposition for a signal measured on k-simplices in a given simplicial complex
- Since the k-Laplacian has both "upper" and "lower" parts, we need a new notion of 'neighbors'. Two k-simplices are 'adjacent' if either:
  - they have a $(k-1)$-simplex in common as a facet
  - they are both facets of some $(k+1)$-simplex in the complex

Hodge Laplacian via Boundary Matrices

$$L_k := B_{k-1}^{\top} B_{k-1} + B_k B_k^{\top}; \quad D_k := \mathrm{diag}(L_k)$$

Example Simplicial Complex



$$B_0 = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad L_0 = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ -1 & -1 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \quad L_1 = \begin{bmatrix} 3 & 0 & 0 & -1 & 0 \\ 0 & 3 & 0 & 0 & -1 \\ 0 & 0 & 4 & 0 & 0 \\ -1 & 0 & 0 & 3 & 0 \\ 0 & -1 & 0 & 0 & 3 \end{bmatrix}$$

$$B_2 = O \quad L_2 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

Example Simplicial Complex Spectrum

$$\Lambda_0 = \operatorname{diag}\left(\begin{bmatrix}0\\2\\4\\4\end{bmatrix}\right) \qquad \Phi_0 = \begin{bmatrix}1/2 & \sqrt{2}/2 & 1/2 & 0\\ 1/2 & 0 & -1/2 & \sqrt{2}/2\\ 1/2 & 0 & -1/2 & -\sqrt{2}/2\\ 1/2 & -\sqrt{2}/2 & 1/2 & 0\end{bmatrix}$$

$$\Lambda_1 = \operatorname{diag}\left(\begin{bmatrix}2\\2\\4\\4\\4\end{bmatrix}\right) \qquad \Phi_1 = \begin{bmatrix}\sqrt{2}/2 & 0 & 0 & -\sqrt{2}/2 & 0\\ 0 & \sqrt{2}/2 & 0 & 0 & -\sqrt{2}/2\\ 0 & 0 & 1 & 0 & 0\\ \sqrt{2}/2 & 0 & 0 & \sqrt{2}/2 & 0\\ 0 & \sqrt{2}/2 & 0 & 0 & \sqrt{2}/2\end{bmatrix}$$

$$\Lambda_2 = \operatorname{diag}\left(\begin{bmatrix}2\\4\end{bmatrix}\right) \qquad \Phi_2 = \begin{bmatrix}-\sqrt{2}/2 & \sqrt{2}/2\\ \sqrt{2}/2 & \sqrt{2}/2\end{bmatrix}$$

Weighted and Normalized Hodge Laplacian

Weighted Graph Laplacian

$$L_0 = B_0 D_1 B_0^{\mathsf{T}}$$

Random-Walk Normalization

$$L_0^{\mathrm{rw}} = D_0^{-1} L_0$$

Symmetric Normalization

$$L_0^{\mathrm{sym}} = D_0^{-1/2} L_0 D_0^{-1/2}$$

Weighted Hodge Laplacian

$$L_k = (B_{k-1} D_k)^{\mathsf{T}} D_{k-1}^{-1} (B_{k-1} D_k) + B_k D_{k+1} B_k^{\mathsf{T}}$$

Random-Walk Normalization

$$L_k^{\mathrm{rw}} = D_k^{-1} L_k$$

Symmetric Normalization

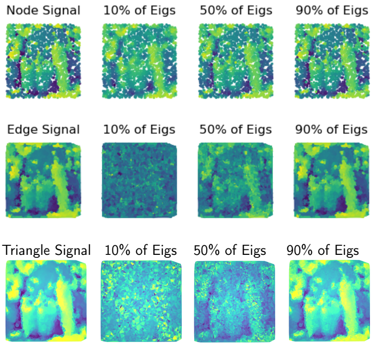
$$L_k^{\mathrm{sym}} = D_k^{-1/2} L_k D_k^{-1/2}$$

The choice of L_k^{rw}, L_k^{sym} for $k = 1$ coincides with the *Helmholtzian Eigenmap* of Chen-Meilă-Kevrekidis (2021).

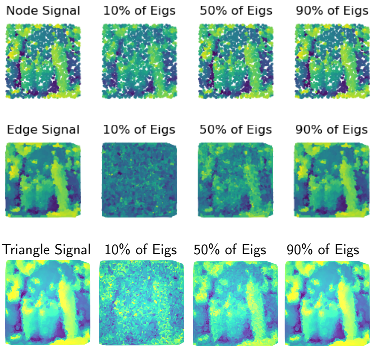
Approximation Experiments

We randomly selected $2,500$ points from the unit square to generate 'nodes', 'edges', and 'triangles', then derived the 'edge' and 'triangle' signals.

Linear approximation



Nonlinear approximation



We construct the *multiscale Fourier* basis dictionaries on k-simplices using *Hodge Laplacians*, $k = 0, 1, 2, \ldots$.

but can we do better?

Outline

Partitioning Simplicial Complexes

- The Hodge Laplacian L_k^{rw} also admits a *Fiedler vector*, whose sign provides a partition on k-simplices minimizing a relaxed version of *Normalized Cut*.
- L_k induces a *signed graph* on the k-simplices. In the combinatorial case, $[L_k]_{\sigma\tau} = 0$ when σ, τ are not adjacent or share a coface, $[L_k]_{\sigma\tau} < 0$ when σ, τ have consistent orientations, and $[L_k]_{\sigma\tau} > 0$ when σ, τ have inconsistent orientations.
- Unlike L_0, the components of $\boldsymbol{\phi}_0$ of L_k may change their signs in general; hence $\boldsymbol{\phi}_1 \odot \operatorname{sign} \boldsymbol{\phi}_0$ provides the Fiedler vector.
- Further, while the Hodge Laplacian optimizes for encoding topological information, modification such as the *signed Laplacian* is more closely connected to the appropriate Cut objective.
- As before, any other good bipartition method for simplicial complexes can be used for building our multiscale basis dictionaries.

Hierarchical Partitioning

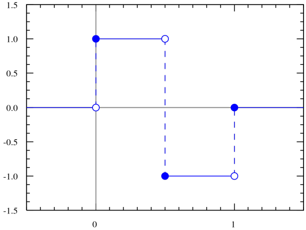
A synthetic simplicial complex with $k = 2$. Successively partitioning the subcomplexes induced by prior partitions leads to finer, nicely localized domains, illustrated by piecewise-constant functions on the triangles. Proceeding left-to-right, each complex has been partitioned to one finer level.

Outline

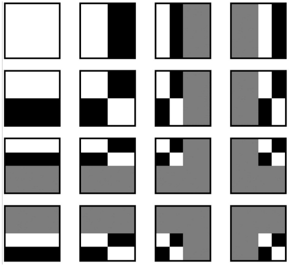
k-Haar Basis

We can use the partitioning induced by the Fiedler vector to develop a *top-down, piecewise constant, and locally concentrated* basis with good approximation properties. However, there are some challenges

- Since the division is not symmetrically dyadic, we need to compute the scaling factor for each atom separately
- The presence of both upper and lower boundary terms means that the *discrete nodal domain theorem* does not apply
- Different hierarchical bipartition schemes arise from the different weighting of the Hodge Laplacians

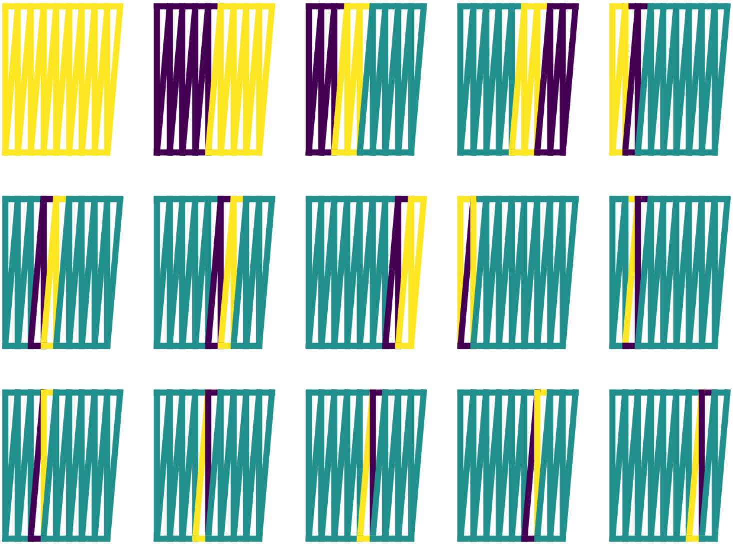


https://en.wikipedia.org/wiki/Haar_ wavelet

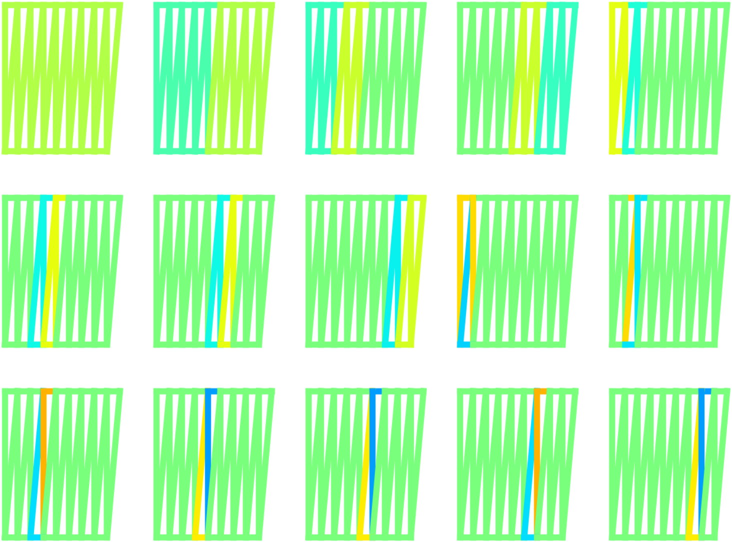


S. Chen & C. Liu, *Image & Vision Computing*, vol.33, pp.68–77, 2015

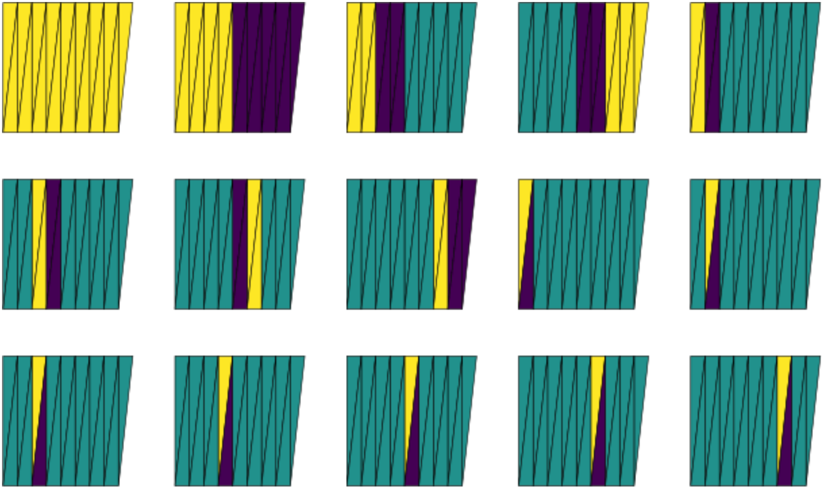
Example 1: Sign of the 1-Haar Basis on a Beam Graph



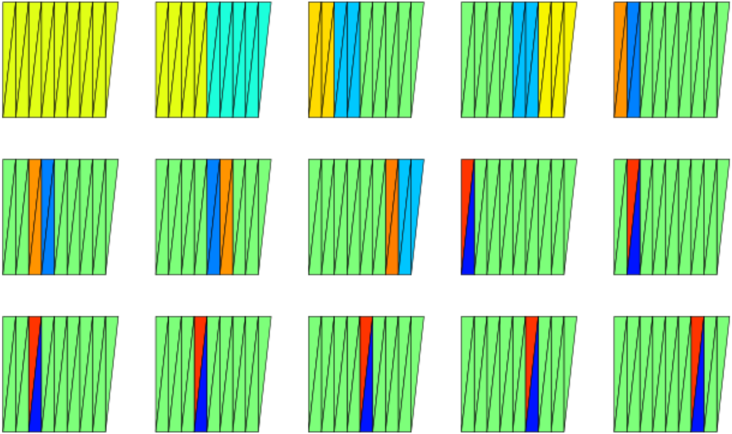
Example 1: Values of the 1-Haar Basis on a Beam Graph



Example 1: Sign of the 2-Haar Basis on a Beam Graph

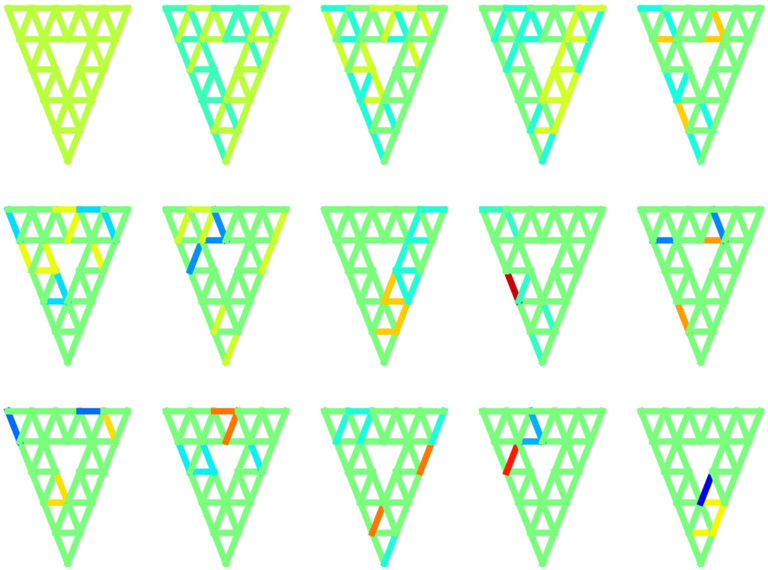


Example 1: Values of the 2-Haar Basis on a Beam Graph

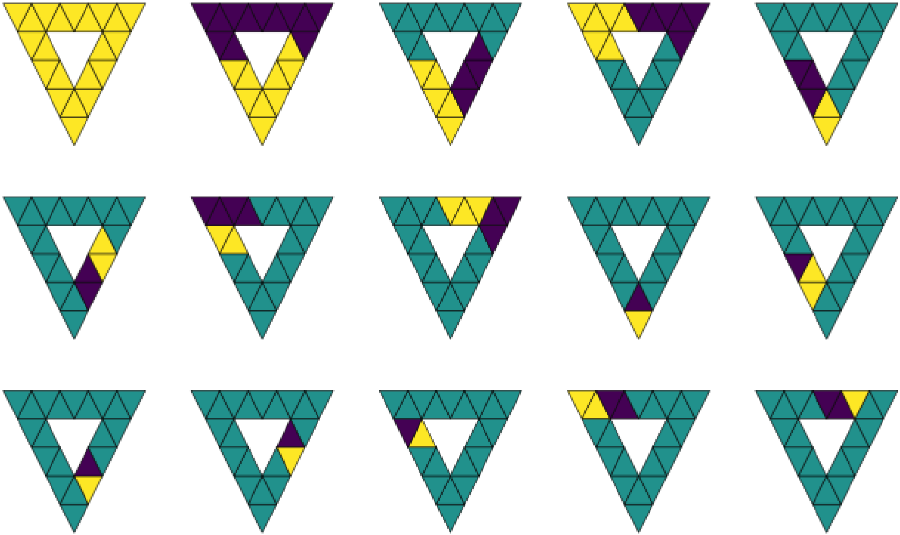


Example 2: The 1-Haar Basis on a Triangle Graph

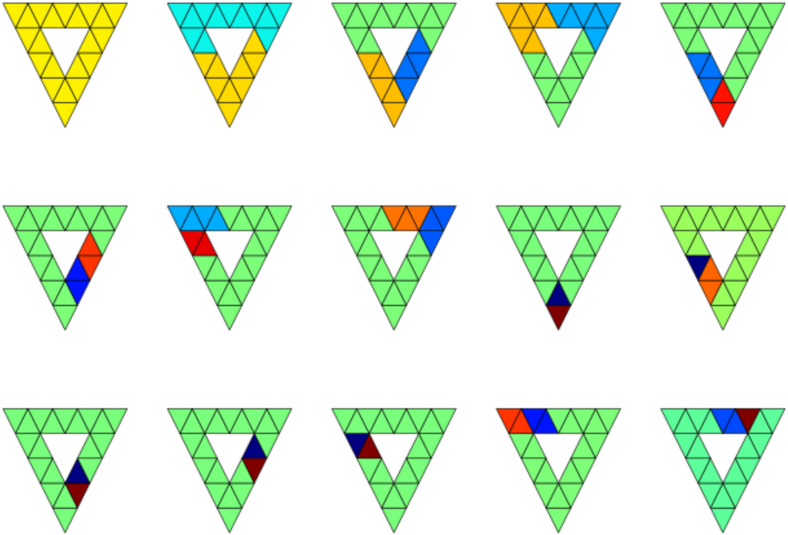
Example 2: The 1-Haar Basis on a Triangle Graph



Example 2: The 2-Haar Basis on a Triangle Graph

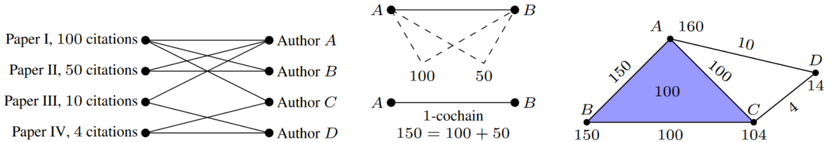


Example 2: The 2-Haar Basis on a Triangle Graph



Outline

Haar Approximation of the Citation Complex

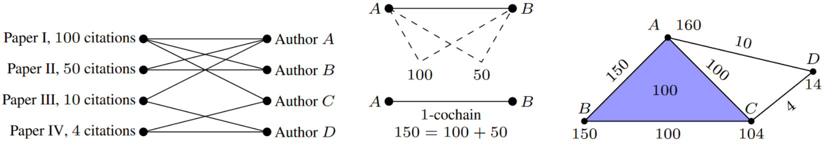


From Ebli et al. 2022

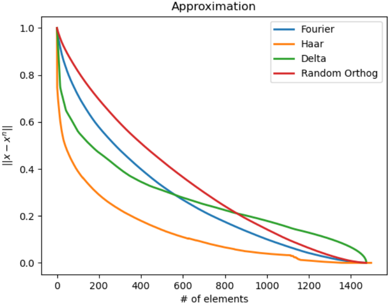
- The citation complex [Patania et al. (2017)] can be created by linking papers, authors, and co-authors from the CORA citation network. Specifically, we use the subgraph suggested by [Elbi et al. (2022)].
- Each Paper node has a citation value corresponding to the number of citations of the paper.
- Each Author node has a citation value corresponding to the total publications of the author.
- Each $(k+1)$-simplex value is equal to the sum of the k-neighbors. (see above)

Dimension	0	1	2	3	4	5	6	7	8	9	10
CC1	352	1474	3285	5019	5559	4547	2732	1175	343	61	5

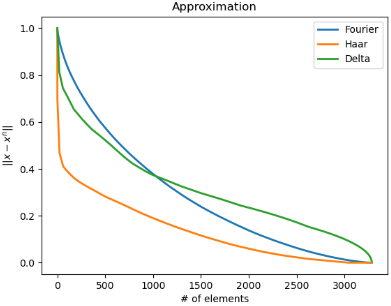
Haar Approximation of a Citation Complex: Results



From Ebli et al. 2022

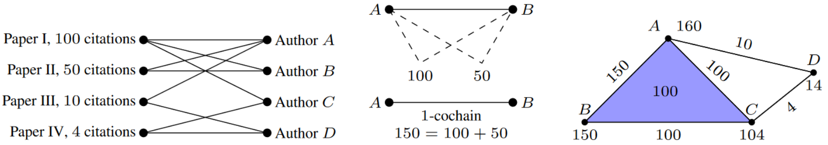


Edge Signal Approximation

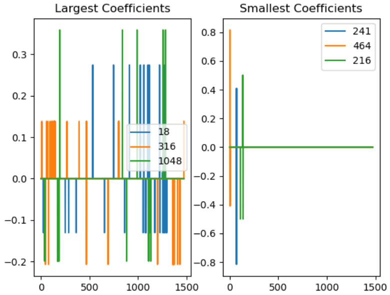


Face Signal Approximation

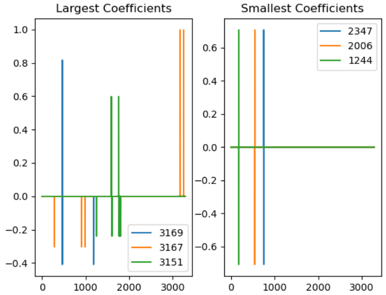
Haar Basis Vectors on the Citation Complex



From Ebli et al. 2022



Edge basis vectors



Face basis vectors

Geometric Scattering

The *Geometric Scattering Transform* (GST) [Gao et al. (2019)] provides a method for nonlinear feature extraction of node-valued data on generic graphs. Let $\{\boldsymbol{\phi}_i\}_{i=1}^N \subset \mathbb{R}^N$ be the eigenvectors of the graph Laplacian and $\boldsymbol{f}$ be a node-valued vector. Then, the GST is defined by

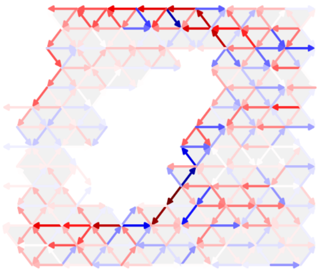
$$Sf(i,p) := \sum_{j=1}^{n} |\phi_i(x_j) f(x_j)|^p, \quad p = 1:P, \quad i = 1:N, \quad n \leq N, \text{ e.g., } n = N/2$$

- Coefficients at the deeper layers can be computed by applying the transform multiple times
- It is *efficient* (Each layer can be computed on a GPU/CPU for the same cost as a graph convolution)
- It is *invariant* to graph isomorphisms
- It is *equivariant* under function permutation–permuting then transforming is the same as transforming then permuting
- There are no trainable parameters to learn!

We can adopt this to apply to a k-simplex by substituting the usual L_0 by the k-Hodge Laplacian L_k! We call this *Hodge Scattering*.

Clustering of Buoys: Problem

Total Signal



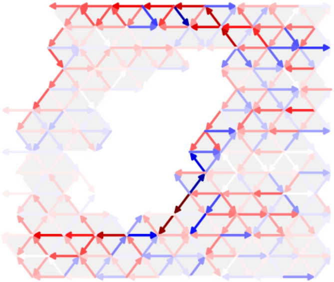
Cluster Quality:

$$\frac{1}{N_{\text{te}}}\sum_{n=1}^{N_{\text{te}}}\max_k \frac{\langle \boldsymbol{f}_n, \boldsymbol{c}_k\rangle}{\|\boldsymbol{f}_n\|_2\|\boldsymbol{c}_k\|_2}$$

- We divide a data set of buoy trajectories [Roddenberry et al. (2022)] around the island of Madagascar into training and test sets of size $N_{\text{tr}} = 251$ and $N_{\text{te}} = 83$ respectively.
- 130 Nodes, 320 Edges, 186 Triangles
- We use the delta, Fourier, and Haar basis coefficients as features
- We then use the coefficients of the training set to cluster the trajectories with the K-means algorithm $(k = 2)$ with centers $\boldsymbol{c}_k$
- Finally, we measure their *Cluster Quality* on the training set

Clustering of Buoys: Results

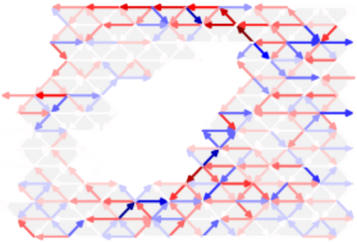
Total Signal



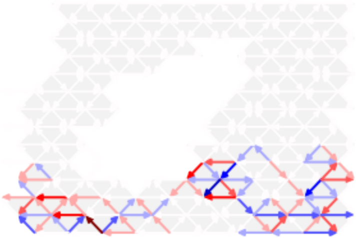
Coefficients	Cluster Quality
Delta	0.071038
Fourier	0.145108
Joint	0.133078
Separate	0.181096
Haar	**0.212611**
Hodge Scattering, P=2	**0.298434**

Here Joint and Separate are the wavelet-like overcomplete dictionaries of [Roddenberry et al. (2022)] and Haar (Orthogonal) is our basis and Haar (Scattering) is an over-complete dictionary of our own construction

Clustering of Buoys: **Fourier Clustering**



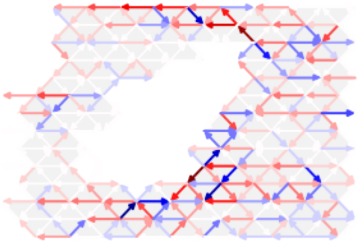
Cluster 1



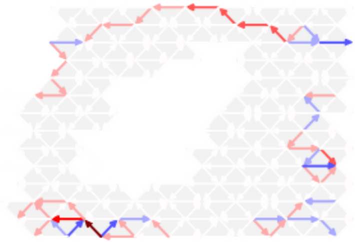
Cluster2

Sum of all of the trajectories in each cluster using Fourier coefficients

Clustering of Buoys: **Haar Clustering**



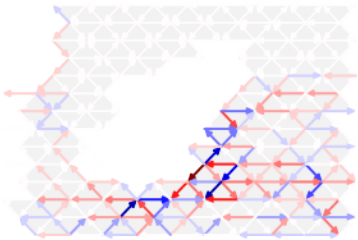
Cluster 1



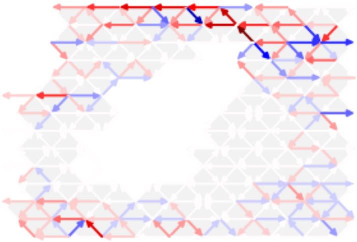
Cluster 2

Sum of all of the trajectories in each cluster using Haar coefficients

Clustering of Buoys: **Hodge-Scattering Clustering**



Cluster 1



Cluster 2

Sum of all of the trajectories in each cluster using Hodge Scattering coeff's

Clustering of Buoys: Extensive Results

We repeat this test for many different numbers of clusters (2-9) using only a sparse selection of features (3-14) selecting through Orthogonal Matching Pursuit (OMP). The scattering transform and delta basis are ill-suited for this task and are not included.

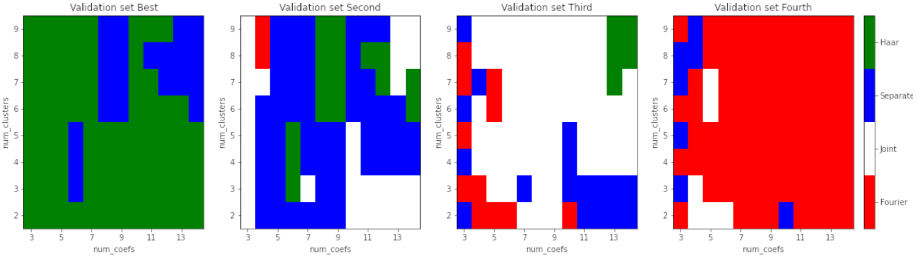


Figure: 1st, 2nd, 3rd, and 4th best bases based on cluster quality. X-axis: Number of Coefficients, Y-axis: Number of Clusters

Green: Haar, Blue: Separate, White: Joint, Red: Fourier

MNIST: Extensive Results

We repeat this test for the MNIST dataset, interpolated to edges of a random triangulation of the plane. Here, the Haar basis produces the best results for every cluster size (2-11) and number of coefficients (2-20):

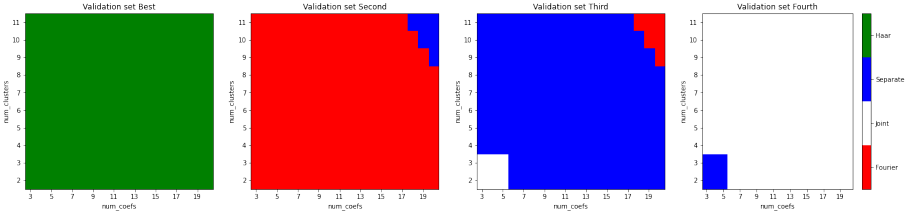


Figure: 1st, 2nd, 3rd, and 4th best bases based on cluster quality. X-axis: Number of Coefficients, Y-axis: Number of Clusters

Green: Haar, Blue: Separate, White: Joint, Red: Fourier

Outline

Motivation

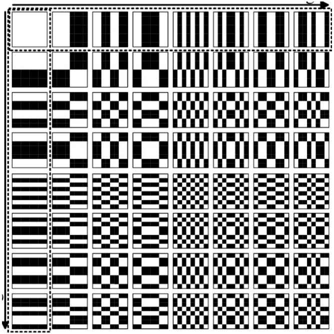
Now that we've explored particular orthonormal bases (ONBs), i.e., k-Haar bases, can we develop overcomplete dictionaries and transforms?

Some past work:

- *Generalized Haar-Walsh Transform (GHWT)* [Irion and **S.** (2014)]: Use Hierarchical bipartition in a "bottom-up" manner to generate a set of local orthonormal Walsh bases at each level of the bipartition
- *Hierarchical Graph Laplacian Eigen Transform (HGLET)* [Irion and **S.** (2014)]: Recursively bipartition a graph and use subgraph Laplacian eigenvectors to form an ONB for each level of the bipartition $\Longleftarrow$ Hierarchical Block DCT on 1D path graph
- *Hodgelets* [Roddenberry et al. (2022)]: Basis functions for edge signals which are compactly supported in the "frequency" domain –constructed via Hodge Laplacians

Generalized Haar-Walsh Transform

- HGLET can be viewed as a generalization of the block discrete cosine transform while the Generalized Haar-Walsh Transform *GHWT* can be viewed as a generalization of the *Haar* and *Walsh-Hadamard Transform*
- Rather than having *smooth* basis functions, we have *piecewise constant* functions which form a basis for each subgraph
- However, the *supports* of the HGLET and GHWT are the same for any given complex



Walsh-Hadamard Basis in 2D

C.-S. Park, *IEEE Trans. Image Process.*, vol.24, pp.155–162, 2014.

GHWT algorithm

Algorithm 1: Generating GHWT Dictionary[16–18]

Input: A binary partition tree $\{G_k^j\}$ of the graph G, $0 \le j \le j_{\max}$ and $0 \le k < K^j$. $N_k^j := |V(G_k^j)|$. K^j denotes the number of subgraphs on level j.

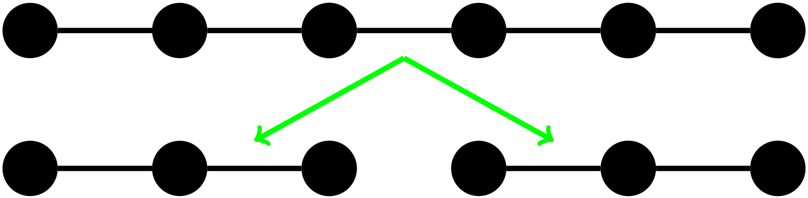
Output: An overcomplete dictionary of basis vectors $\{\psi_{k,l}^j\}$

```
for k = 0, ..., N − 1 do                                  // Basis vectors on level j_max are unit vectors
    ψ^{j_max}_{k,0} ← 1_{V(G_k^{j_max})} ∈ ℝ^N
end
for j = j_max, ..., 1 do                                  // Compose basis vectors on level j − 1 from level j
    for k = 0, ..., K^{j−1} − 1 do
        ψ^{j−1}_{k,0} ← 1_{V(G_k^{j−1})} / sqrt(N_k^{j−1})   // Compute the scaling vector
        // Basis vectors supported on V(G_k^{j−1}) are computed from those on V(G_{k'}^j) and
        V(G_{k'+1}^j).  G_{k'}^j and G_{k'+1}^j are the two subgraphs of G_k^{j−1}
        if N_k^{j−1} > 1 then
            ψ^{j−1}_{k,1} ← (N^j_{k'+1} sqrt(N^j_{k'}) ψ^j_{k',0} − N^j_{k'} sqrt(N^j_{k'+1}) ψ^j_{k'+1,0}) / sqrt(N^j_{k'} (N^j_{k'+1})^2 + N^j_{k'+1} (N^j_{k'})^2)   // Compute the Haar vector
        end
        if N_k^{j−1} > 2 then
            for l = 1, ..., 2^{j_max − j} − 1 do          // Compute the Walsh-like vectors
                if both subregions k' and k' + 1 have a basis vector with tag l then
                    ψ^{j−1}_{k,2l} ← (ψ^j_{k',l} + ψ^j_{k'+1,l}) / √2;
                    ψ^{j−1}_{k,2l+1} ← (ψ^j_{k',l} − ψ^j_{k'+1,l}) / √2;
                else if (without loss of generality) only subregion k' has a basis vector with tag l then
                    ψ^{j−1}_{k,2l} ← ψ^j_{k',l}
                else if Neither subregion has a basis vector with tag l then
                    Do nothing
            end
        end
    end
end
```

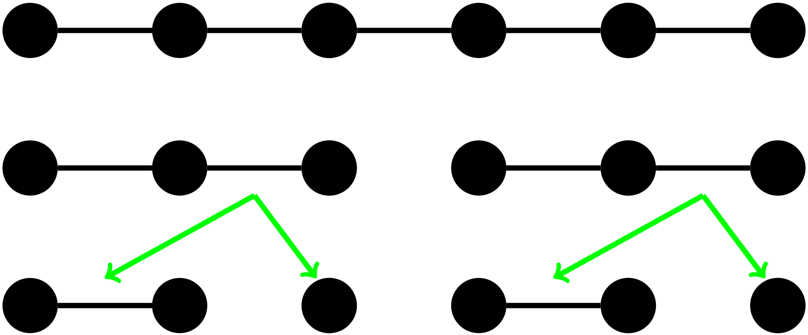
$$\psi_{k,1}^{j-1} \leftarrow \frac{N_{k'+1}^j \sqrt{N_{k'}^j}\,\psi_{k',0}^j - N_{k'}^j \sqrt{N_{k'+1}^j}\,\psi_{k'+1,0}^j}{\sqrt{N_{k'}^j (N_{k'+1}^j)^2 + N_{k'+1}^j (N_{k'}^j)^2}}$$

0-GHWT on P_6

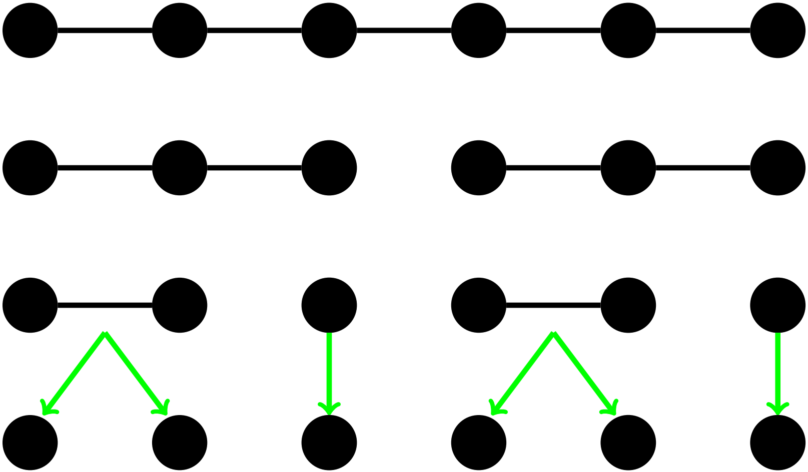
0-GHWT on P_6



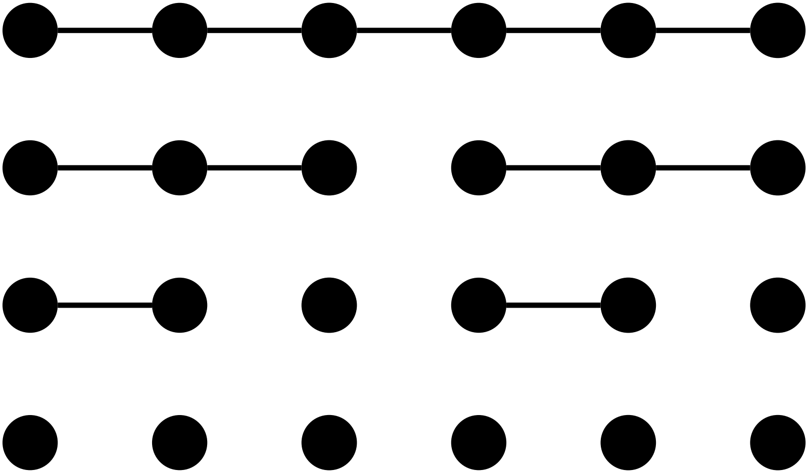
0-GHWT on P_6



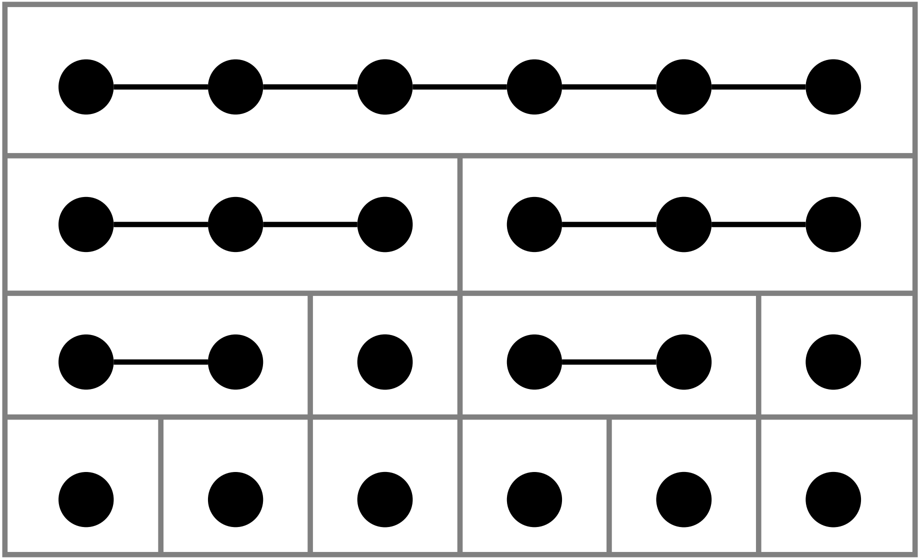
0-GHWT on P_6



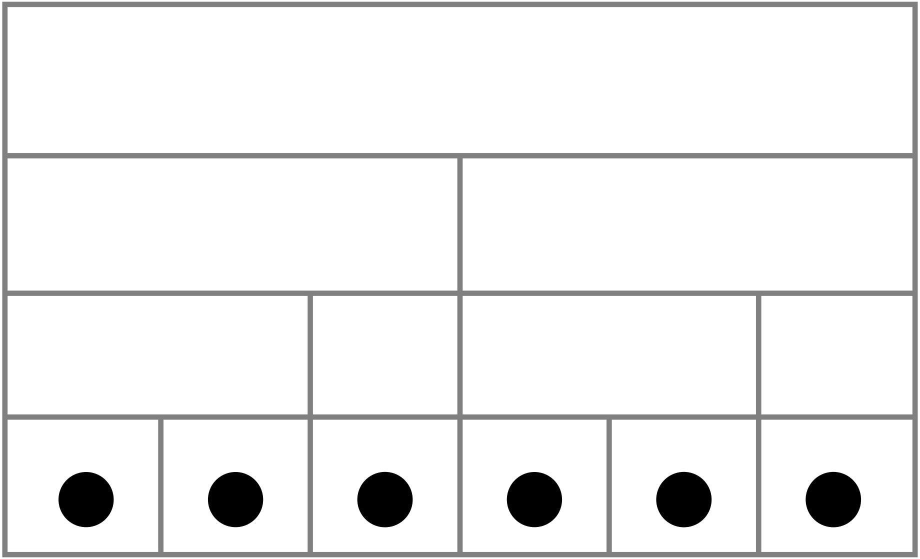
0-GHWT on P_6



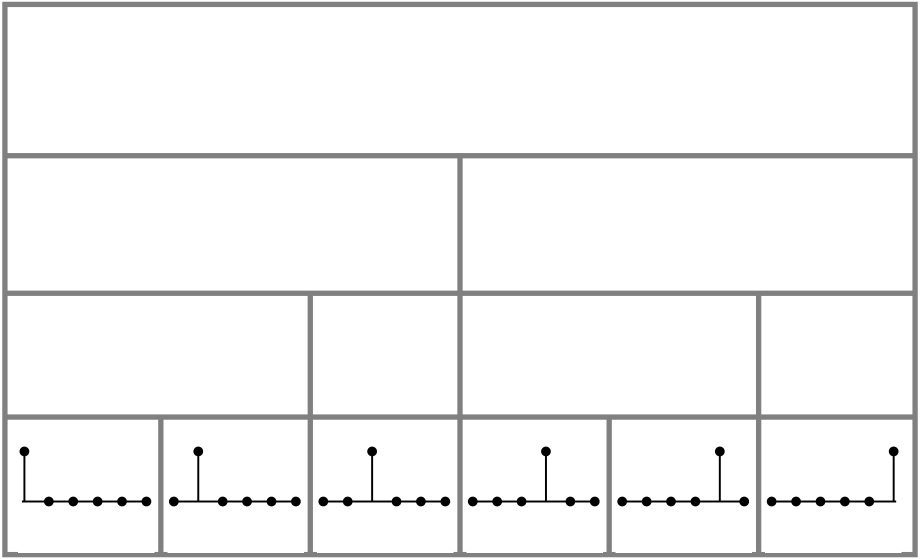
0-GHWT on P_6



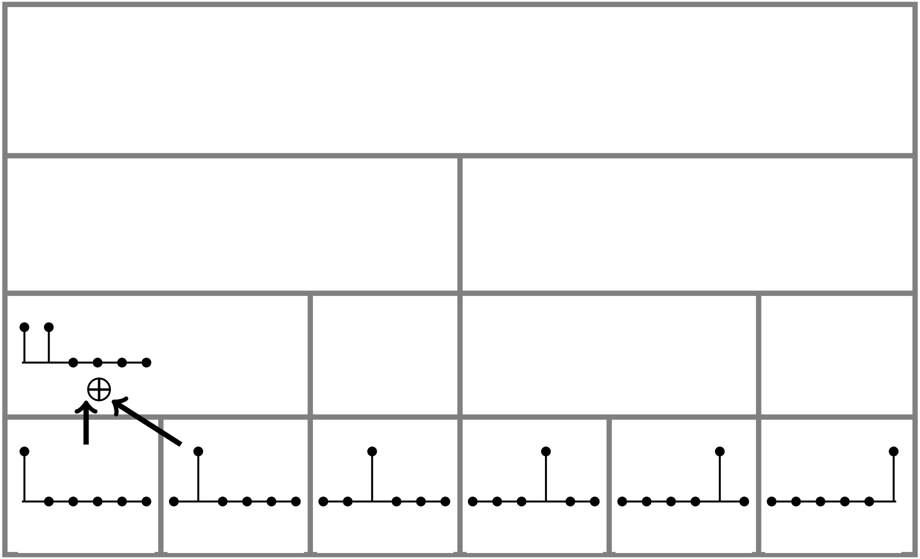
0-GHWT on P_6



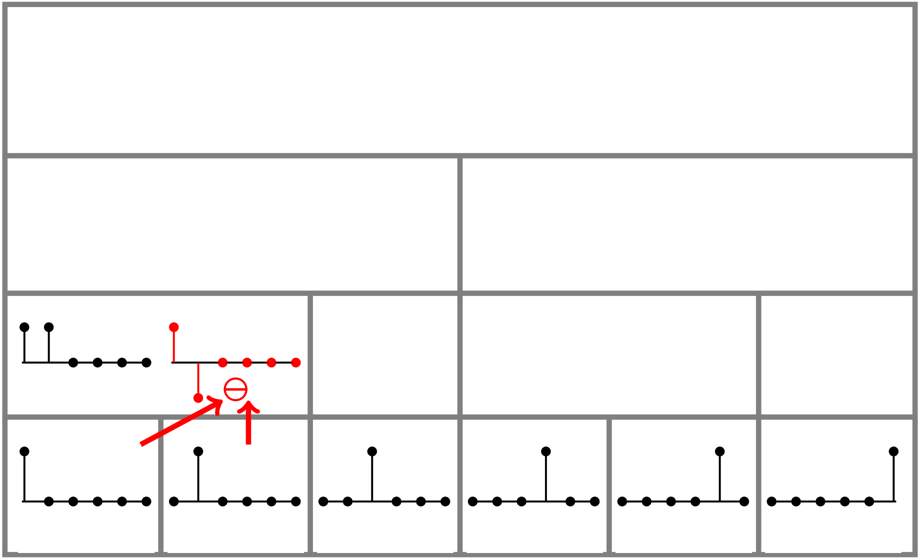
0-GHWT on P_6



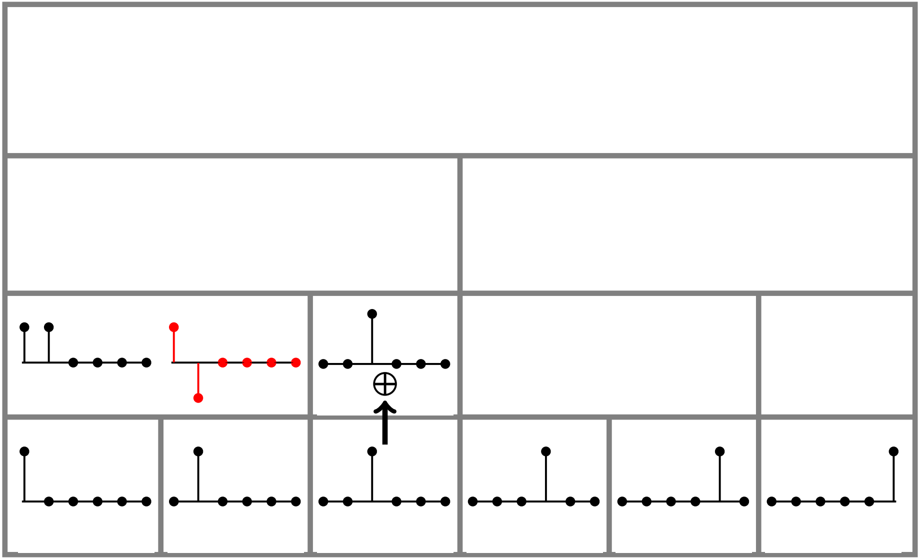
0-GHWT on P_6



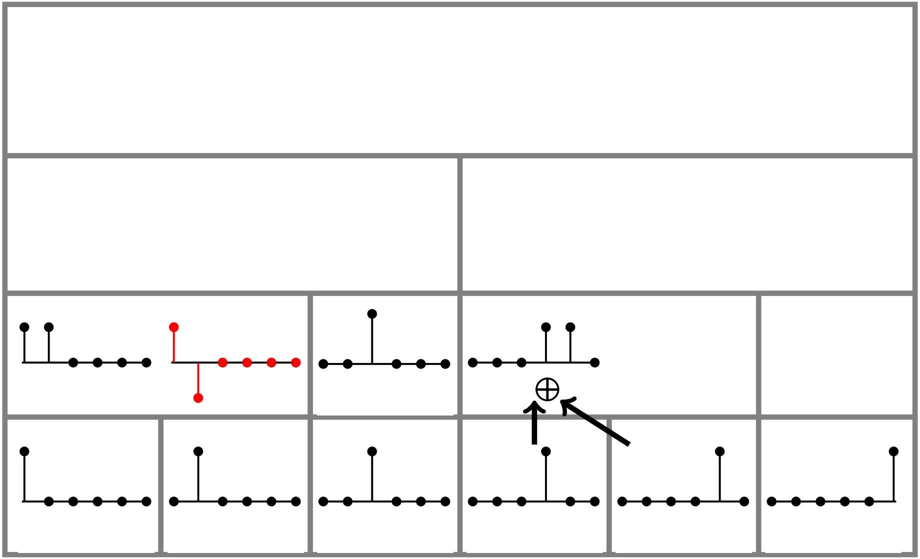
0-GHWT on P_6



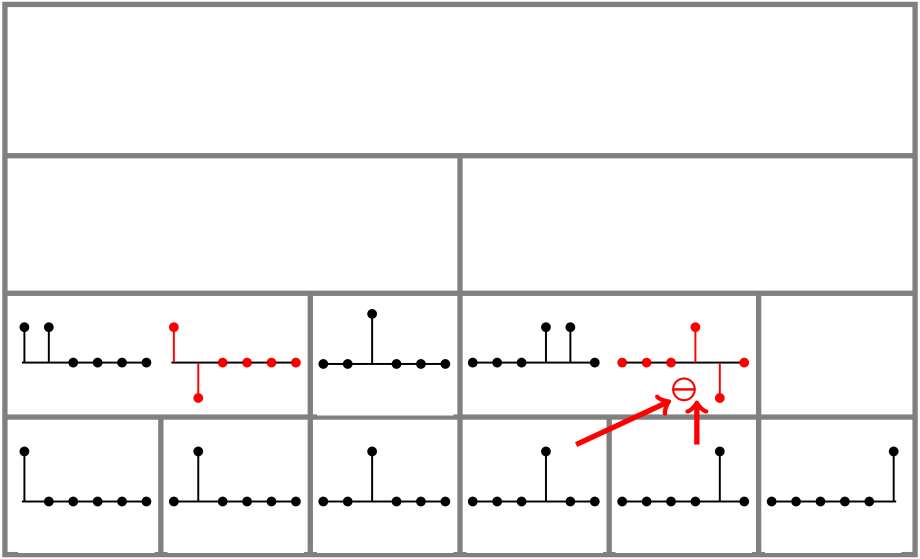
0-GHWT on P_6



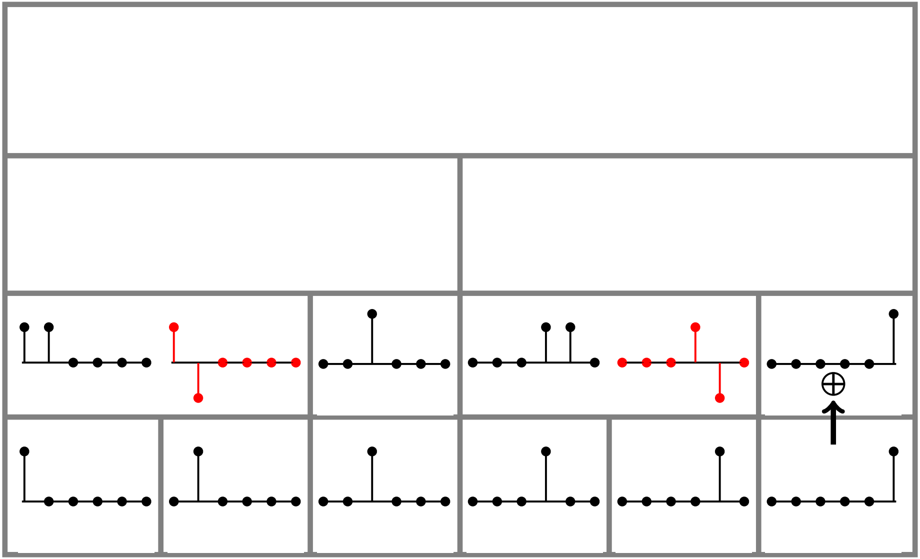
0-GHWT on P_6



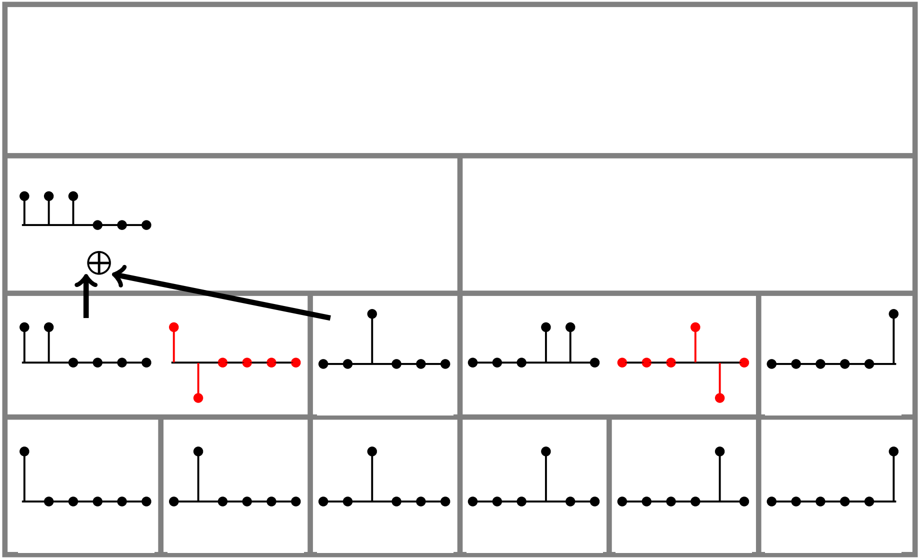
0-GHWT on P_6



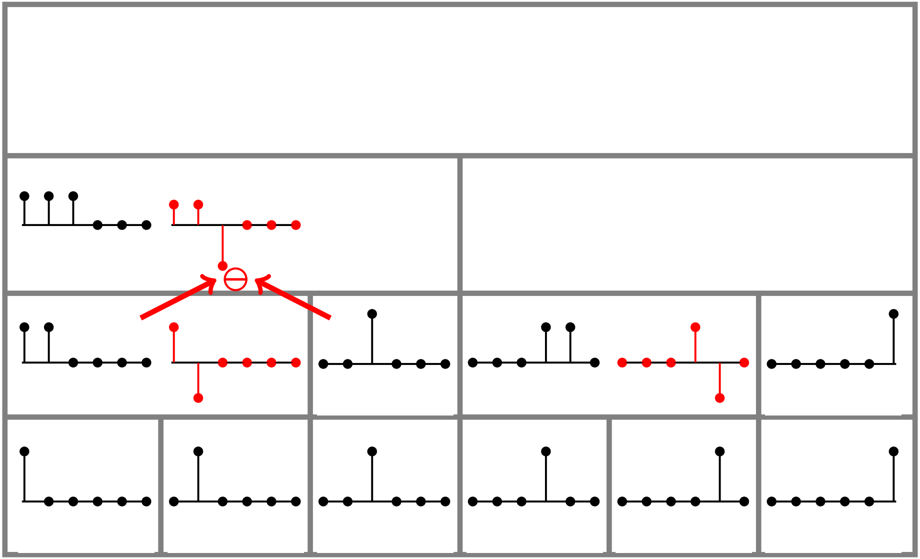
0-GHWT on P_6



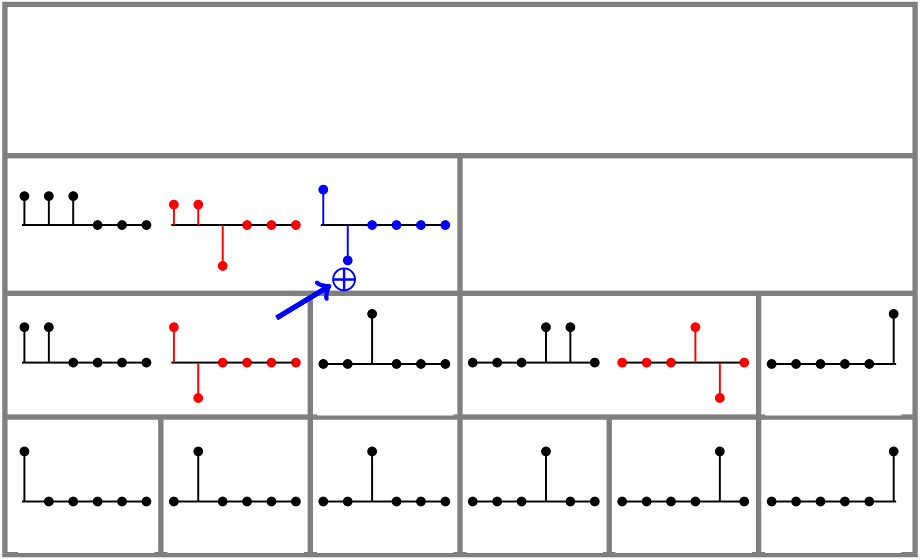
0-GHWT on P_6



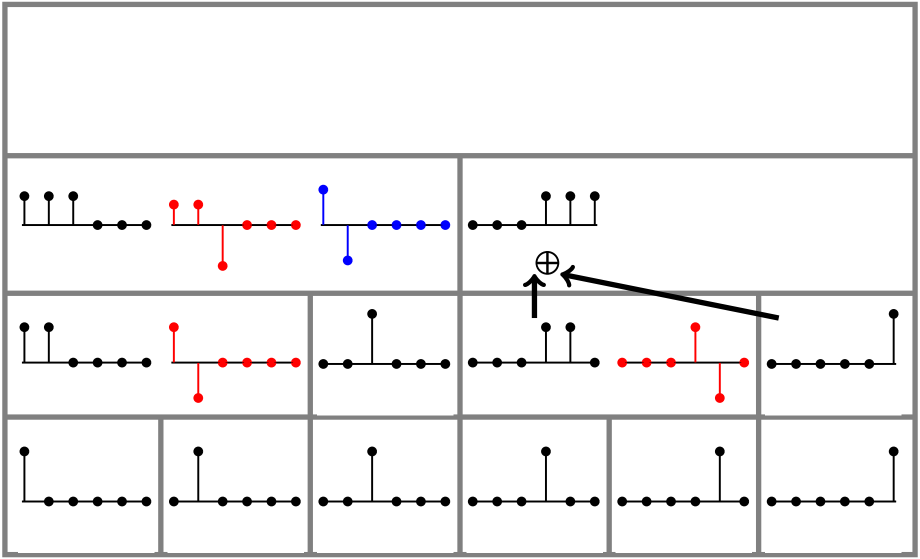
0-GHWT on P_6



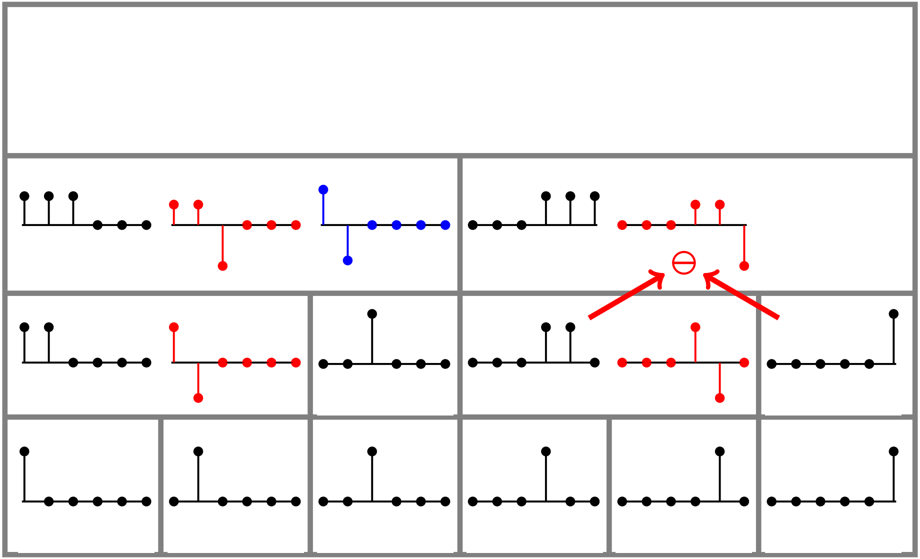
0-GHWT on P_6



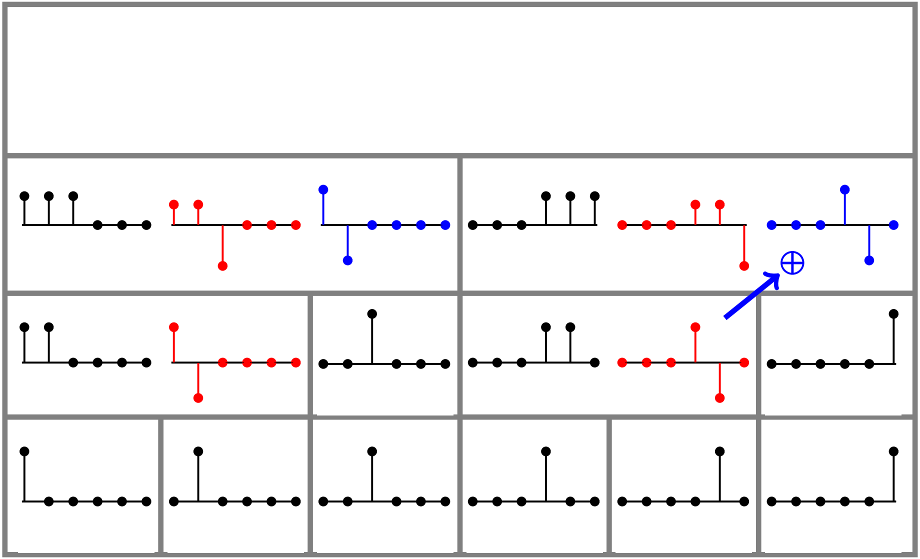
0-GHWT on P_6



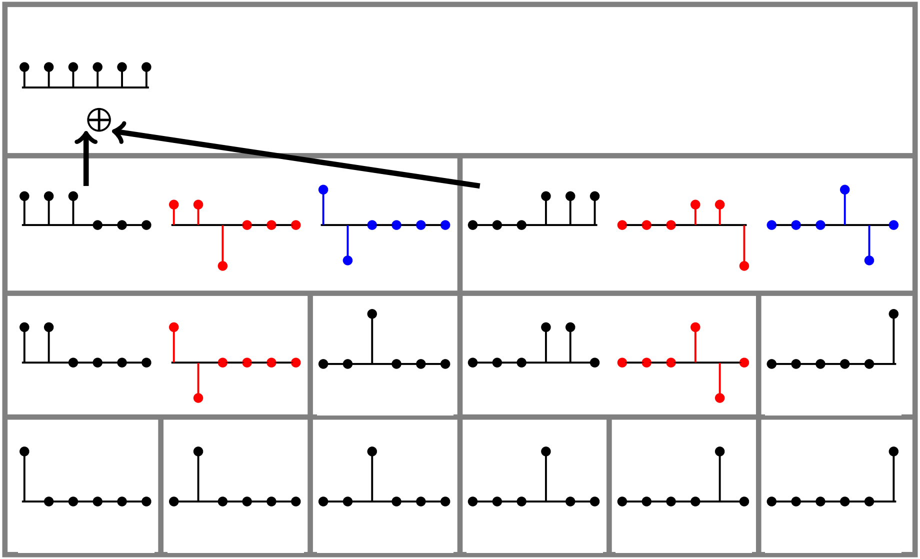
0-GHWT on P_6



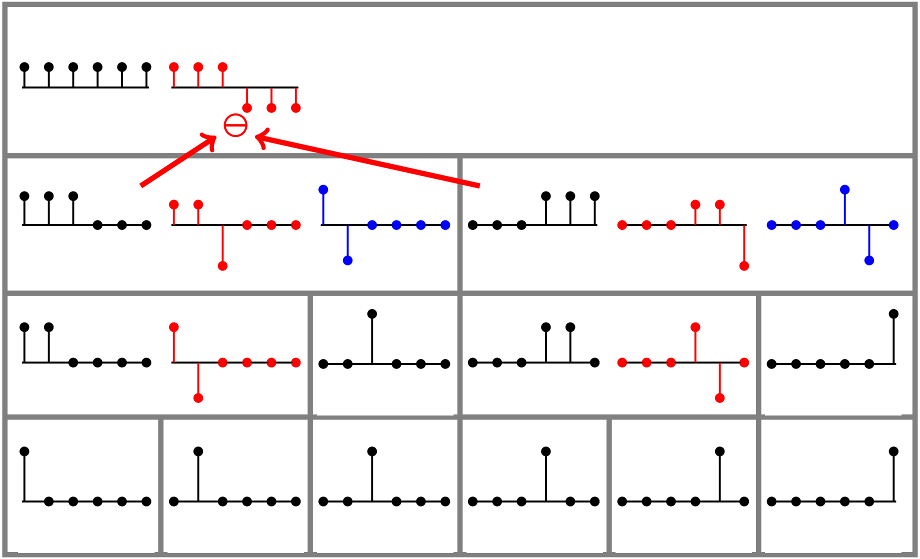
0-GHWT on P_6



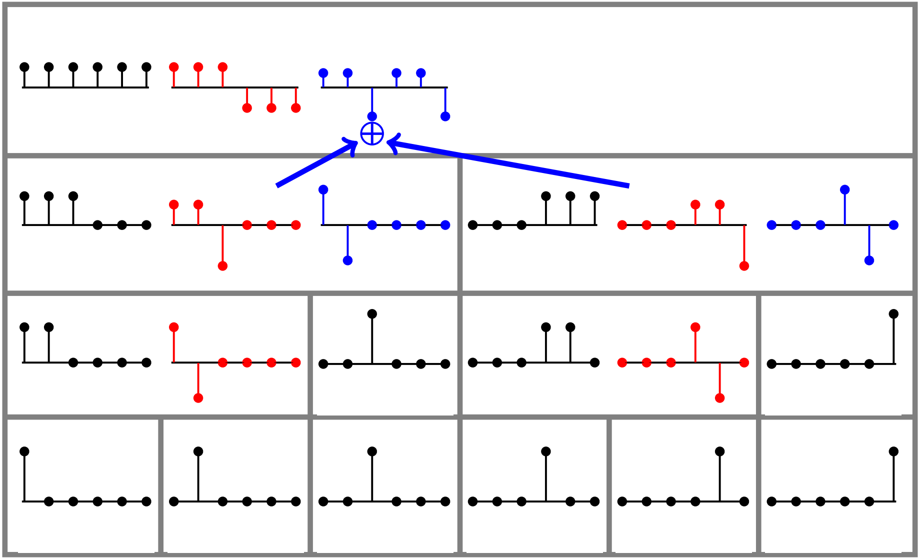
0-GHWT on P_6



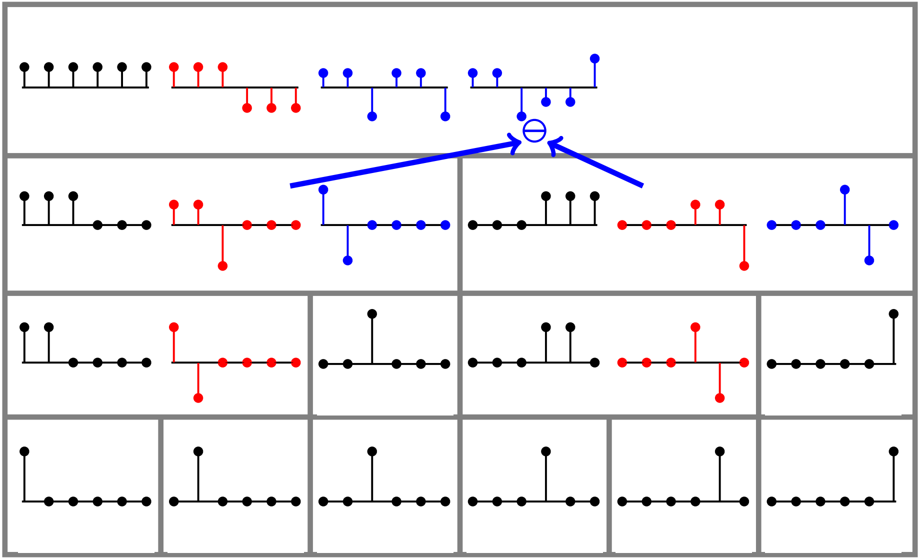
0-GHWT on P_6



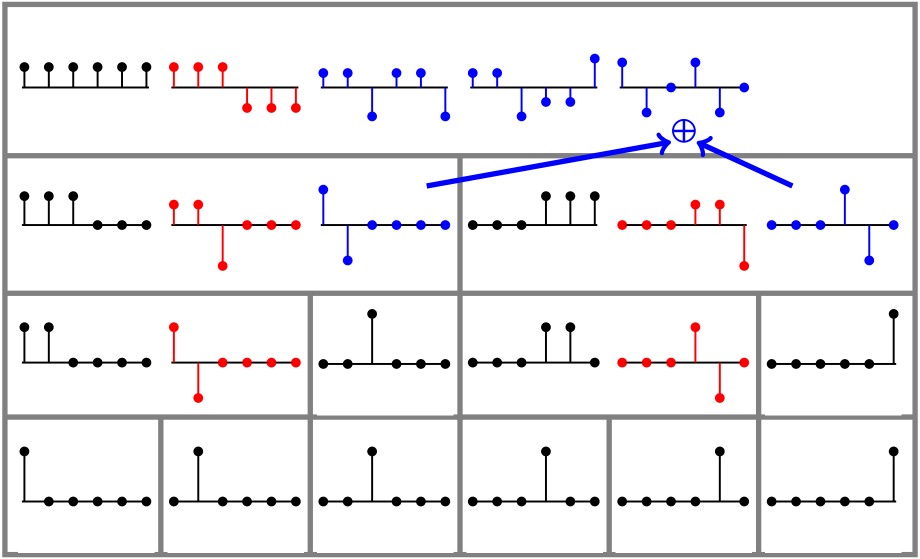
0-GHWT on P_6



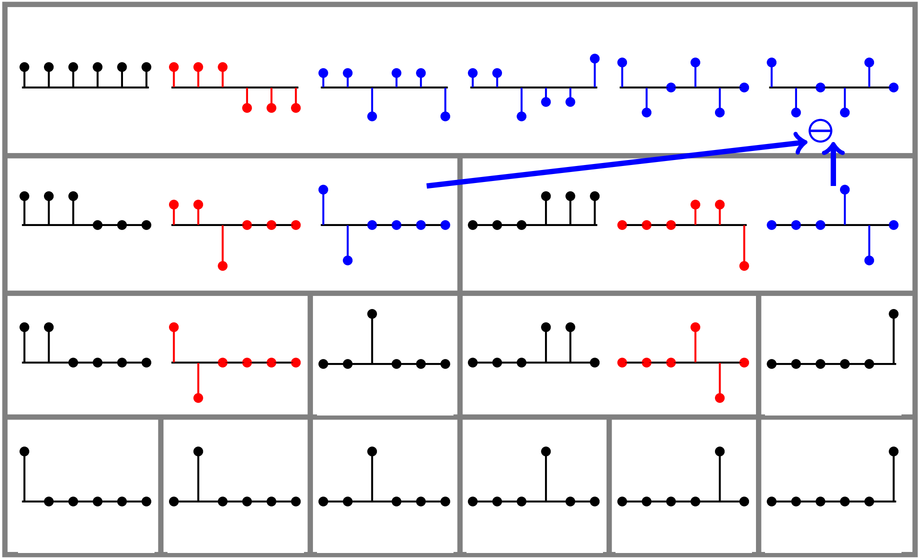
0-GHWT on P_6



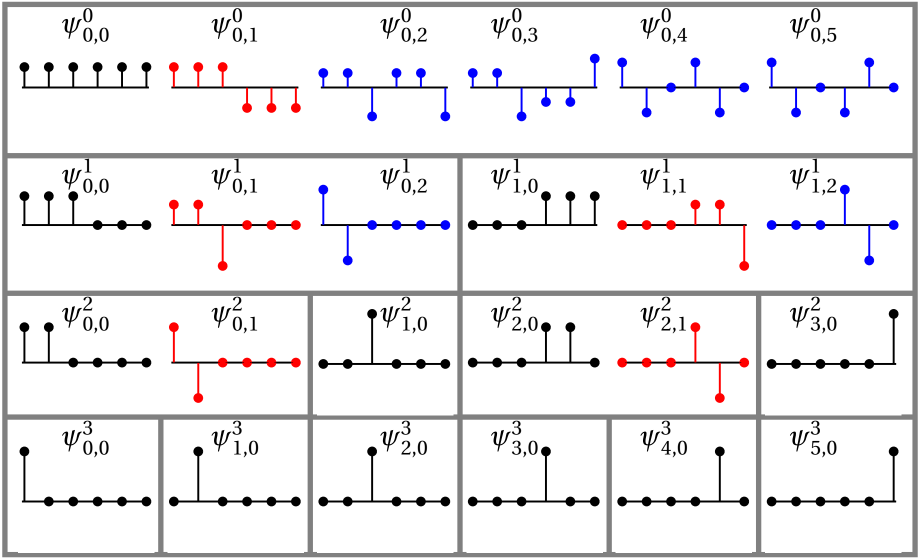
0-GHWT on P_6



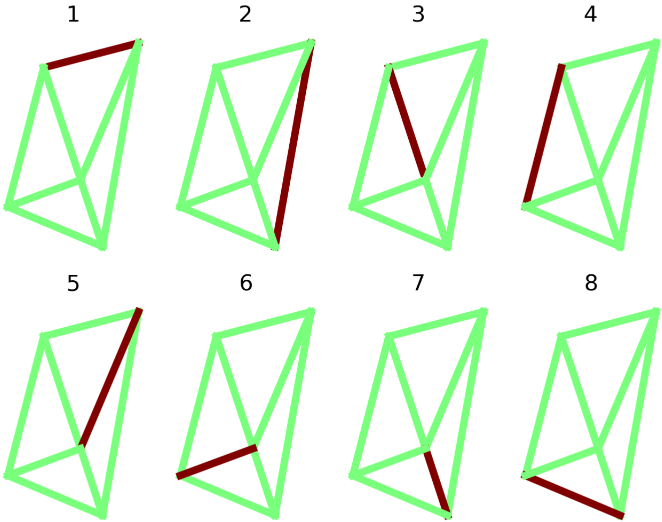
0-GHWT on P_6



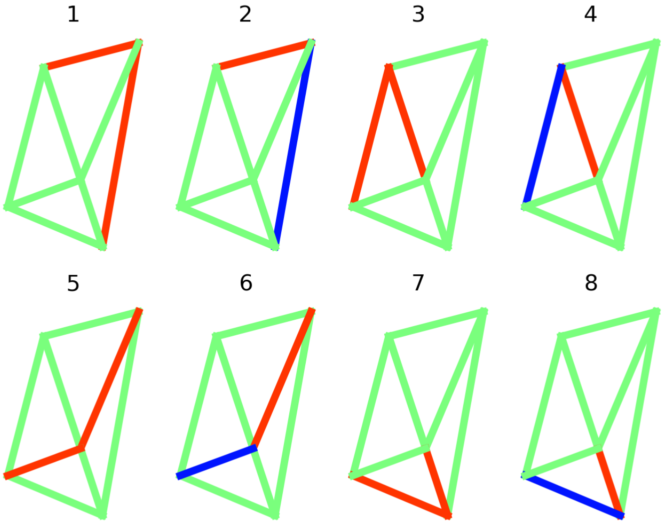
0-GHWT on P_6


$\psi^0_{0,0}$
$\psi^0_{0,1}$
$\psi^0_{0,2}$
$\psi^0_{0,3}$
$\psi^0_{0,4}$
$\psi^0_{0,5}$
$\psi^1_{0,0}$
$\psi^1_{0,1}$
$\psi^1_{0,2}$
$\psi^1_{1,0}$
$\psi^1_{1,1}$
$\psi^1_{1,2}$
$\psi^2_{0,0}$
$\psi^2_{0,1}$
$\psi^2_{1,0}$
$\psi^2_{2,0}$
$\psi^2_{2,1}$
$\psi^2_{3,0}$
$\psi^3_{0,0}$
$\psi^3_{1,0}$
$\psi^3_{2,0}$
$\psi^3_{3,0}$
$\psi^3_{4,0}$
$\psi^3_{5,0}$

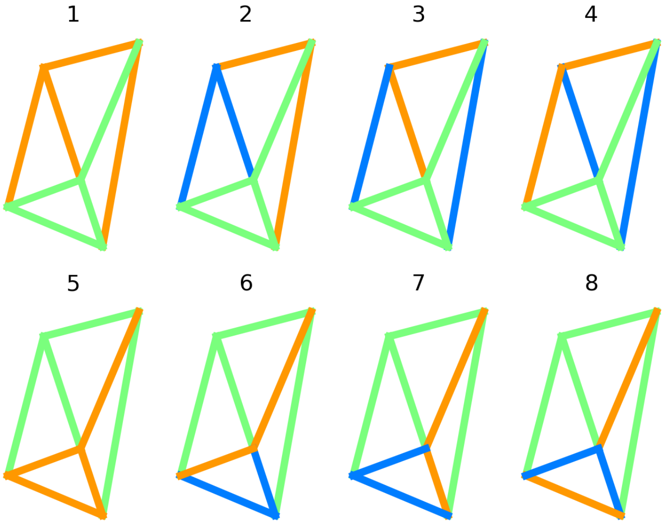
1-GHWT Example: Finest Level



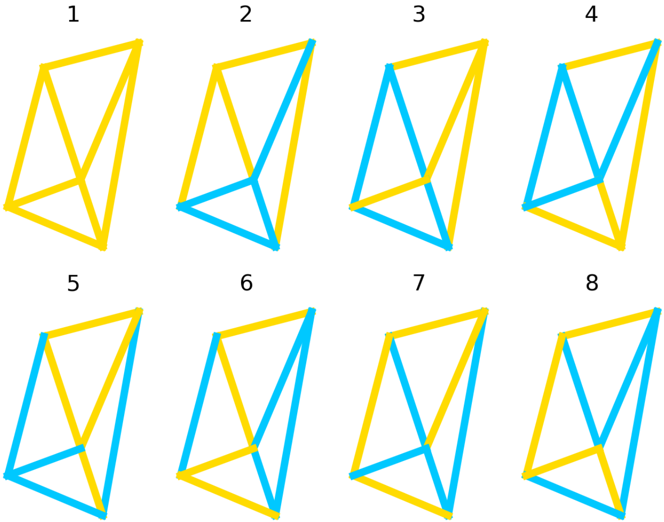
1-GHWT Example: Level 2



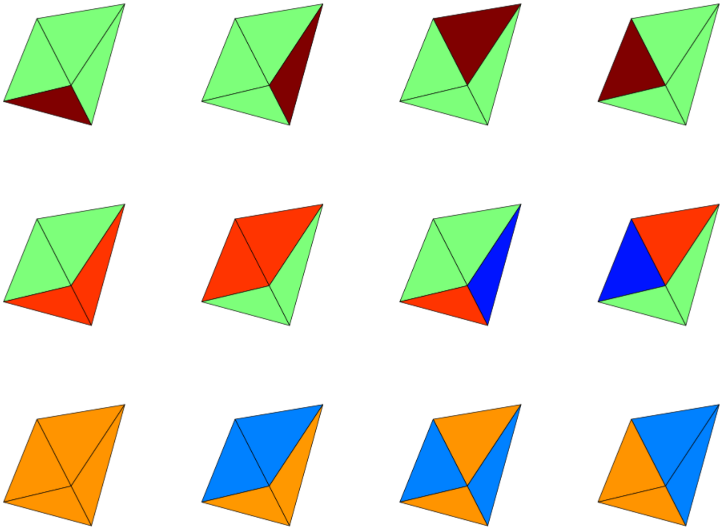
1-GHWT Example: Level 3



1-GHWT Example: Level 4



2-GHWT Example



Hierarchical Graph Laplacian Eigen Transform

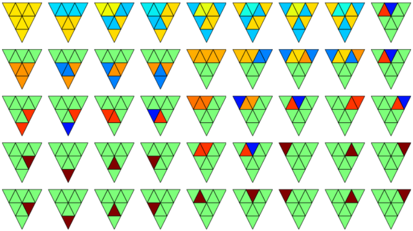
The *Hierarchical Graph Laplacian Eigen Transform (HGLET)* can be viewed as a generalization of the Hierarchical Block DCT dictionary. It can be formed by:

- Forming an ONB for the entire graph via the graph Laplacian eigenvectors
- Partition the graph into two subgraphs
- Compute the graph Laplacian of each subgraph
- Form an ONB for each subgraph via the eigensystem
- Continue the above steps until each subgraph becomes a single node

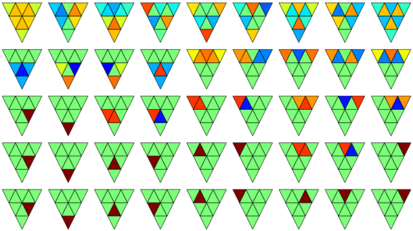
Using the Fiedler vector we can partition k-simplices and compute an analogous dictionary via the Hodge Laplacian!

Comparison: GHWT vs HGLET

GHWT

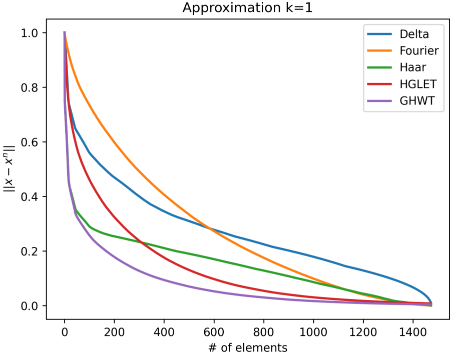


HGLET

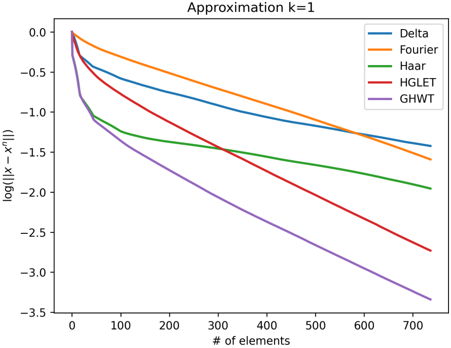


Each row represents one level of the bi-partition

Approximation of Citation Complex with Overcomplete Dictionaries, $k = 1$

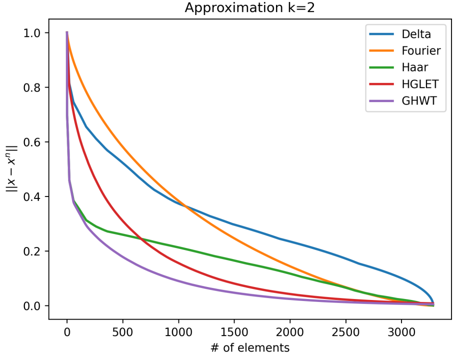


Approximation with entire dictionary

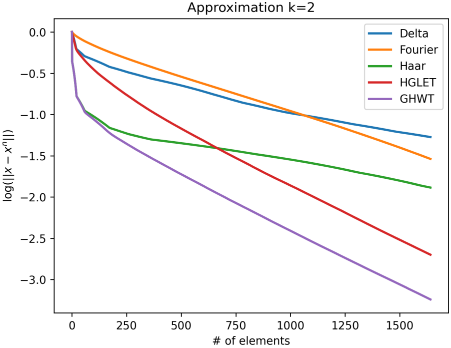


Sparse Approximation

Approximation of Citation Complex with Overcomplete Dictionaries, $k = 2$

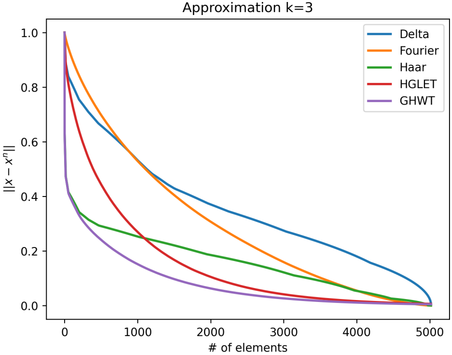


Approximation with entire dictionary

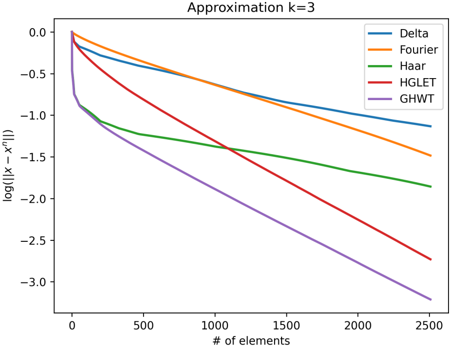


Sparse Approximation

Approximation of Citation Complex with Overcomplete Dictionaries, $k = 3$

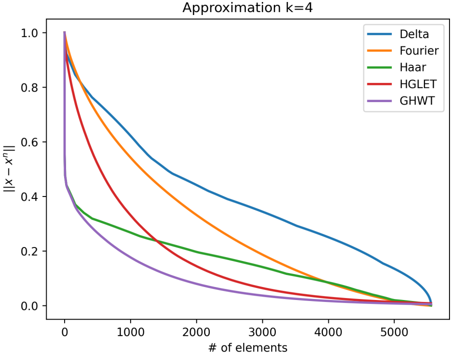


Approximation with entire dictionary

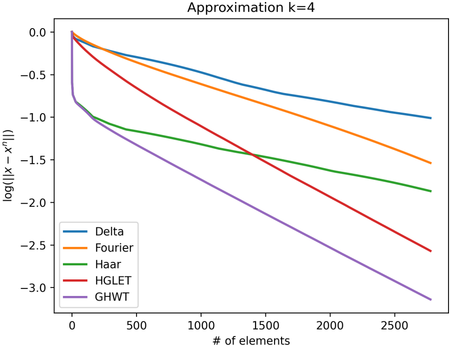


Sparse Approximation

Approximation of Citation Complex with Overcomplete Dictionaries, $k = 4$

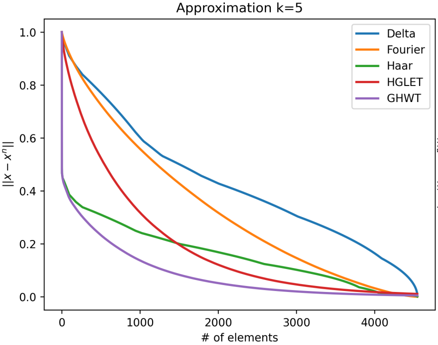


Approximation with entire dictionary

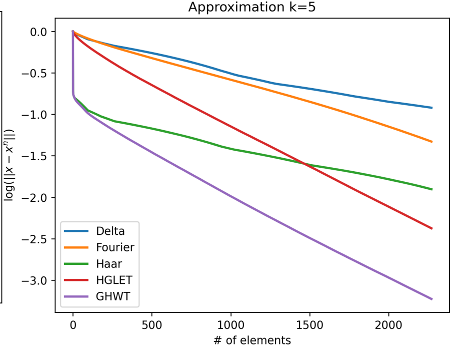


Sparse Approximation

Approximation of Citation Complex with Overcomplete Dictionaries, $k = 5$



Approximation with entire dictionary

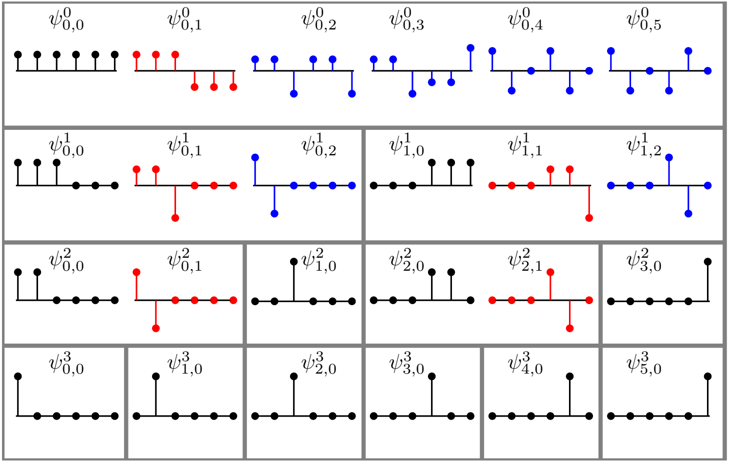


Sparse Approximation

Coarse-to-Fine and Fine-to-Coarse Ordering

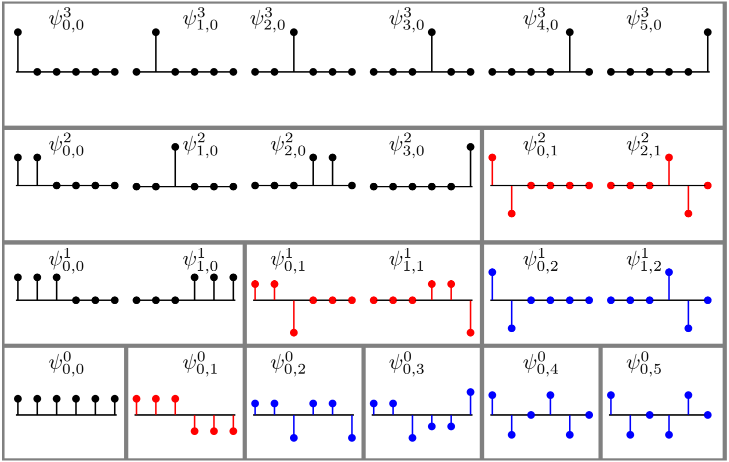
- For many downstream tasks, such as *best-basis selection* [Coifman-Wickerhauser (1992)] and *orthogonal matching pursuit* [Cai-Wang (2011)] it is important to organize the order of these bases.
- In general, the *HGLET* dictionary is naturally ordered in a *Coarse-to-Fine* fashion. In each subgraph, the basis elements are *ordered by "frequency"*. We can also order them by increasing eigenvalue.
- In general, the *GHWT* dictionary is also naturally ordered in a *Coarse-to-Fine* fashion, with increasing "sequency" within each subgraph. This is analogous to the natural ordering of the HGLET.
- Another useful way to order the *GHWT* is in a *Fine-to-Coarse* ordering, which approximates "sequency" domain partitioning.

0-GHWT: Coarse-to-Fine example



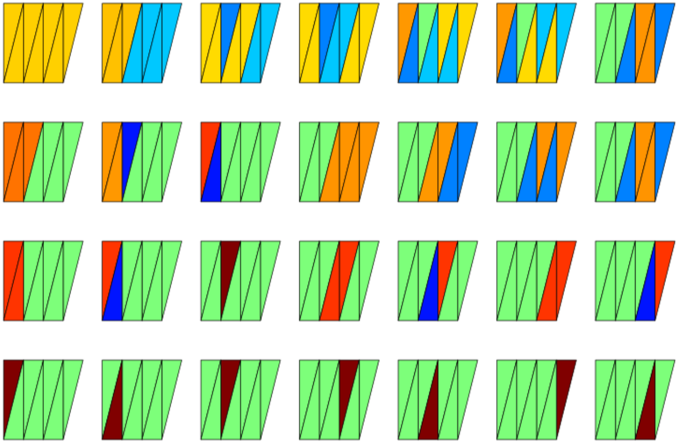
The default *coarse-to-fine* GHWT dictionary on P_6

0-GHWT: Fine-to-Coarse example

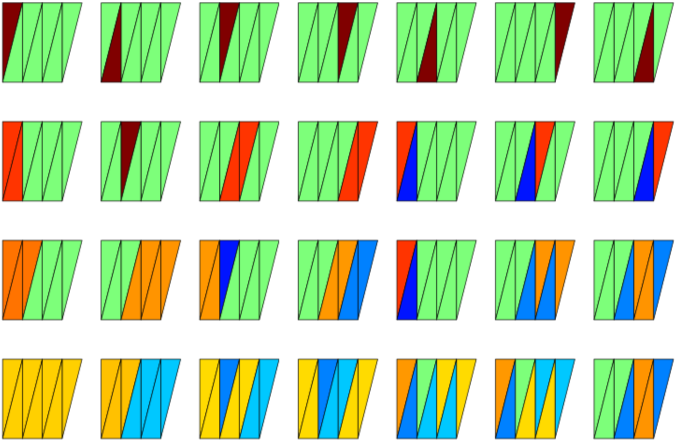


The *fine-to-coarse* GHWT dictionary by reordering & regrouping

2-GHWT: Coarse-to-Fine example

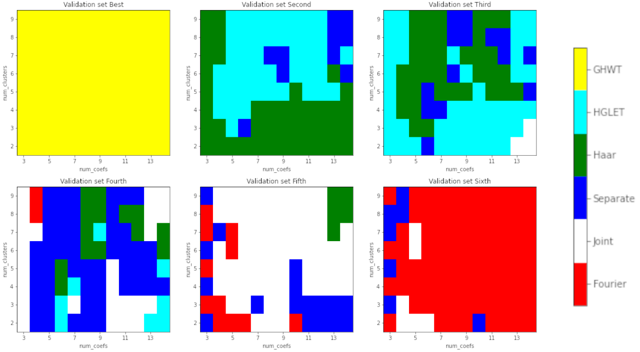


2-GHWT:Fine-to-Coarse example



Clustering of Buoys: HGLET and GHWT

We repeat the bouy test for HGLET and GHWT dictionaries, using (2-9) clusters and a sparse selection of features (3-14) selected through Orthogonal Matching Pursuit (OMP).



Yellow: GHWT, Cyan: HGLET Green: Haar,
Blue: Separate, White: Joint, Red: Fourier

Outline

Summary

- Developed a Fiedler-like vector for solving a relaxed cut problem on simplicial complexes
- Proposed a hierarchical partitioning method for simplicial complexes using *Hodge Laplacians*
- Developed the *k-Haar transform* for signals on simplicial complexes, which is a part of our *multiscale higher-order graph signal basis dictionaries for simplicial complexes*, e.g., signals sampled on edges, faces, etc.
- Extended the k-Haar transform to the *k-GHWT dictionary*
- Developed the *k-HGLET dictionary* using the eigenvectors of the Hodge Laplacians

Future Plan

- Develop *tools to visualize and interpret important basis vectors* for signals on simplicial complexes including *graph embedding methods*.
- Implement *Best Basis* [Coifman-Wickerhauser (1992)], *Local Discriminant Basis (LDB)*, *Local Regression Basis (LRB)*, etc. [Saito et al. (1995; 1997; 2002; . . .)], for signals on simplicial complexes.
- Explore how to reduce computational complexity of $O(N^3)$?
  - For certain problems, one may not need all the GL eigenvectors, in particular, those corresponding to the large eigenvalues.
  - Consider *integral operators* (e.g., *Green's functions*) on graphs, and utilize the *Fast Multipole Method* [Saito (2008); Xue (2007)].
- Truly generalize the *Local Cosine Transform* (LCT) for the graph setting. H. Li (2021) constructed the node version of the *smooth orthogonal projectors involving orthogonal folding and unfolding operators* and the graph basis dictionaries, but we need proper *boundary conditions* at the partition locations.

Outline

References

The following articles (and the other related ones) are available at
https://www.math.ucdavis.edu/~saito/publications/

- J. Irion & N. Saito: "Hierarchical graph Laplacian eigen transforms," *JSIAM Letters*, vol. 6, pp. 21–24, 2014.
- J. Irion & N. Saito: "The generalized Haar-Walsh transform," in *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp. 472–475, 2014.
- J. Irion & N. Saito: "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, Paper # 95971F, 2015.
- J. Irion & N. Saito: "Efficient approximation and denoising of graph signals using the multiscale basis dictionaries,", *IEEE Trans. Signal Inform. Process. Netw.*, vol. 3, no. 3, pp. 607–616, 2017.
- A. Cloninger, H. Li, & N. Saito: "Natural graph wavelet packet dictionaries," *J. Fourier Anal. Appl.*, vol. 27, Article #41, 2021.
- N. Saito & Y. Shao: "eGHWT: The extended Generalized Haar-Walsh Transform," *J. Math. Imaging, Vis.*, vol. 64, no. 3, pp. 261–283, 2022.

References II: Signal Processing on Simplicial Complexes

- Y. C. Chen, M. Meilă, & I. G. Kevrekidis. "Helmholtzian Eigenmap: Topological feature discovery & edge flow learning from point cloud data," *arXiv:2103.07626*, 2021.
- S. Ebli, M. Defferrard, & G. Spreemann, "Simplicial neural networks," *arXiv:2010.03633*, 2022.
- L.-H. Lim, "Hodge Laplacians on graphs," *SIAM Review*, vol.62, no.3, pp.685–715, 2020.
- A. Patania, G. Petri, & F. Vaccarino. "The shape of collaborations," *EPJ Data Sci.*, vol.6, #18, 2017.
- T. M. Roddenberry, F. Frantzen, M. T. Schaub, & S. Segarra,"Hodgelets: Localized spectral representations of flows on simplicial complexes," *2022 IEEE Intern. Conf. on Acoust. Speech Signal Process. (ICASSP)*, pp.5922–5926, 2022.
- M. T. Schaub & S. Segarra "Flow smoothing and denoising: graph signal processing in the edge-space," *IEEE Global Conf. on Signal Info. Process. (GlobalSIP)*, pp.735–739, 2018.
- M. T. Schaub, Y. Zhu, J.-B. Serby, T. M. Roddenberry, & S. Segarra "Signal processing on higher-order networks: Livin' on the edge . . . and beyond," *Signal Process.*, vol.187, #108149, 2021.

Please check our Julia codes on GitHub!!

https://github.com/UCD4IDS/MultiscaleGraphSignalTransforms.jl

Split $\Longrightarrow$ "Organize" $\Longrightarrow$ Merge

Thank you very much for your attention!