Multiscale Graph Basis Dictionaries

Naoki Saito

Department of Mathematics
University of California, Davis

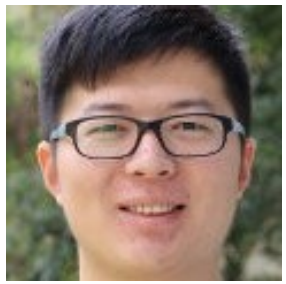
CodEx Virtual Seminar
September 21, 2021

Outline

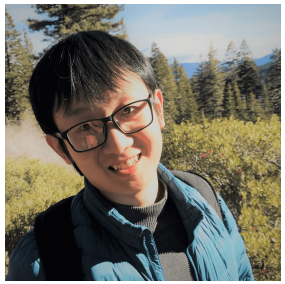
Acknowledgment

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

(a) Jeff Irion (UCD → Zoox)



(b) Yiqun Shao (UCD → Facebook)



(c) Haotian Li (UCD → Facebook)

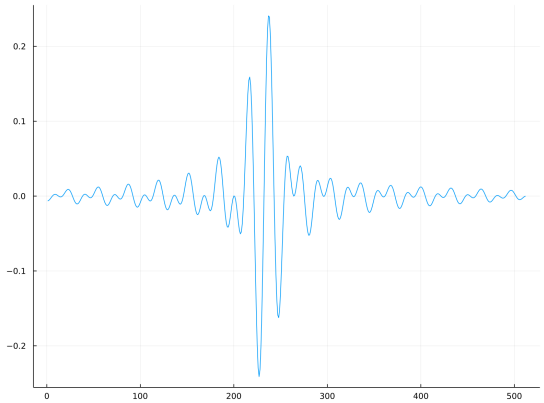
(d) Alex Cloninger (UCSD)

Figure: Collaborators

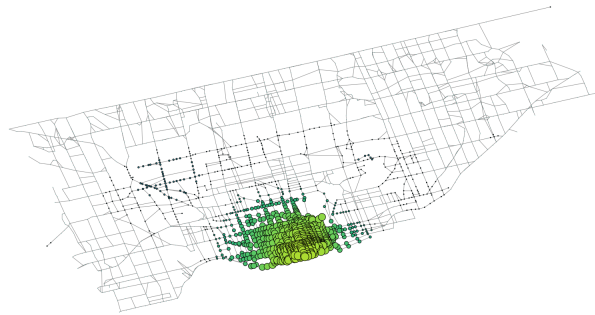
Outline

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*



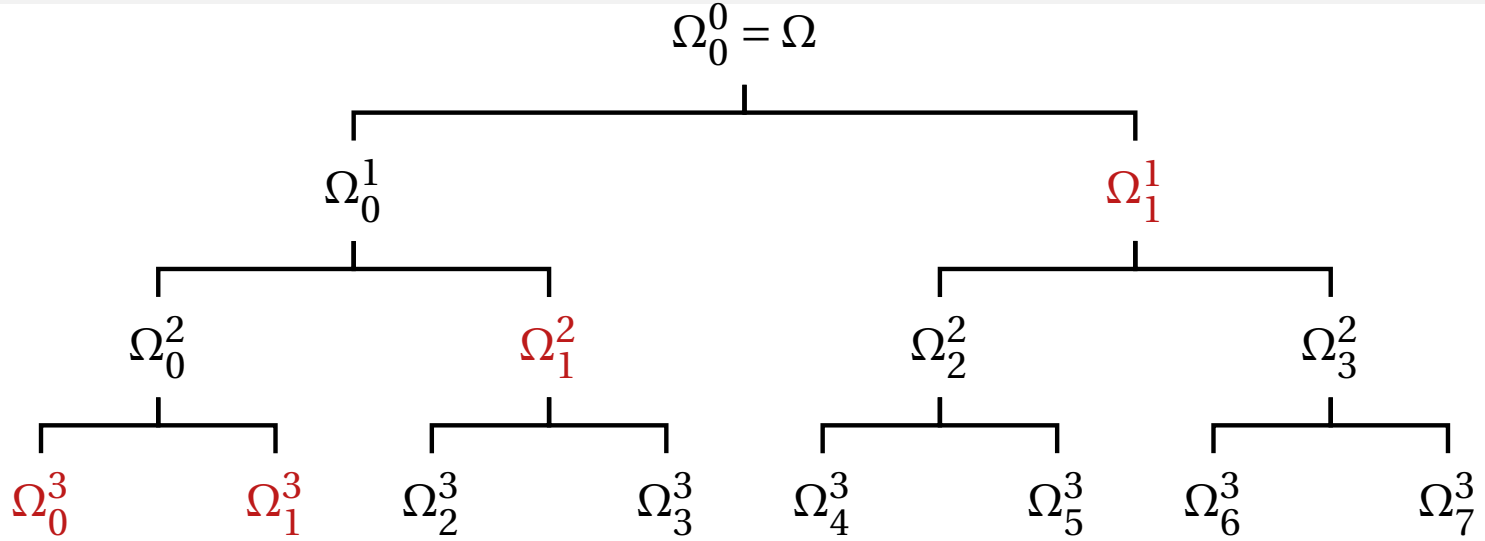
(a) Shannon wavelet on $\mathbb{R}$



(b) Graph wavelet packet vector

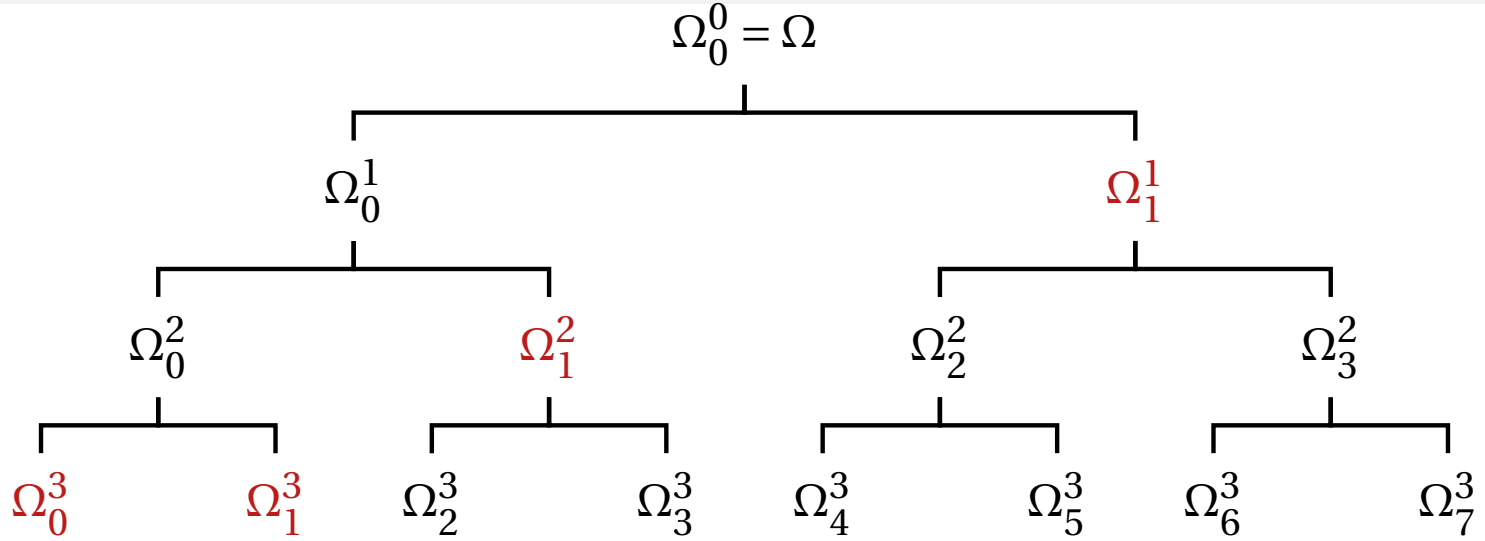
Outline

Classical Dictionaries $\Longleftarrow$ *Rigid* Hierarchical Bipartitioning



- $\Omega = \mathbb{R}^N$ where $N = 2^{j_{\max}}$ is a number of samples of an input signal
- Partitioning is always *rigid* at the middle of the parent subspaces

Classical Dictionaries $\Longleftarrow$ *Rigid* Hierarchical Bipartitioning



- $\Omega = \mathbb{R}^N$ where $N = 2^{j_{\max}}$ is a number of samples of an input signal
- Partitioning is always *rigid* at the middle of the parent subspaces

Dictionary	Ω	Partitioning Mode
Hierarchical Block DCT	time (spatial) axis	hard
Local Cosine Transform (LCT)	time (spatial) axis	soft/overlapping
Haar-Walsh wavelet packets	time/frequency axes	hard
Cmpt.-Supp. wavelet packets	frequency axis	soft/overlapping
Shannon wavelet packets	frequency axis	hard

Classical Multiscale Basis Dictionaries ...

- Let $\dim\left(\Omega_k^j\right) = N/2^j = 2^{j_{\max}-j} =: N^j$
- Let $\left\{\varphi_{k,l}^j\right\}_{l=0}^{N^j-1}$ be a set of orthonormal (ON) vectors of Ω_k^j
- # of ON vectors in the hierarchical bipartition tree is $N(1+\log_2 N)$
- However, it contains more than $O(1.5^N)$ possible orthonormal bases (ONBs): let B_j be the number of choosable ONBs of the hierarchical bipartition tree of depth j. Then $B_{j+1} = B_j^2 + 1$, $B_0 = 1$, $j = 0, 1, \ldots$ [Flajolet-Odlyzko (1984); Thiele-Villemoes (1996)]
- *The best-basis algorithm* [Coifman-Wickerhauser (1992)] can choose the "best" ONB among such a vast number of choosable ONBs very efficiently, i.e., $O(N\log_2 N)$ or $O(N[\log_2 N]^2)$
- What is the "best" depends on the cost functional in the best-basis algorithm (e.g., sparsity $\Longrightarrow$ ℓ^1-norm minimization, ...)
- Can deal with multiple input signals for signal classification, modeling, compression, etc.; see, e.g., *Local Discriminant Basis* [Saito et al. (1995; 2002)]), *Least Statistically-Dependent Basis* [Saito (2001)], ...

Roadmap

Classical Basis Dict.	Ω	Graph Basis Dict.	Ω
HBDCT	time (spatial) axis	*HGLET*	G
LCT	time (spatial) axis	LP-HGLET	G
Haar-Walsh WPs	time/frequency axes	*GHWT/eGHWT*	G
Cmpt-Supp. WPs	frequency axis	LP-NGWPs	?
Shannon WPs	frequency axis	*NGWPs*	?

Underlying Philosophy/Basso Continuo:

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

History

HBDCT Hierarchical Block Discrete Cosine Transform: ?

LCT Local Cosine Transform: Malvar-Staelin (1989), Malvar (1990), Coifman-Meyer (1991)

HWWPs Haar-Walsh Wavelet Packets: Coifman-Meyer (1989)

eHWWPs extended HWWPs (or adapted time-frequency tiling): Thiele-Villemoes (1996)

Cpt-Supp. WPs Compactly-Supported Wavelet Packets: Coifman-Meyer (1990)

Shannon WPs Shannon Wavelet Packets: ?

HGLET *Hierarchical Graph Laplacian Eigen Transform*: Irion-Saito (2014a)

GWHT *Generalized Haar-Walsh Transform*: Irion-Saito (2014b)

eGHWT *extended GHWT*: Saito-Shao (2019, 2021)

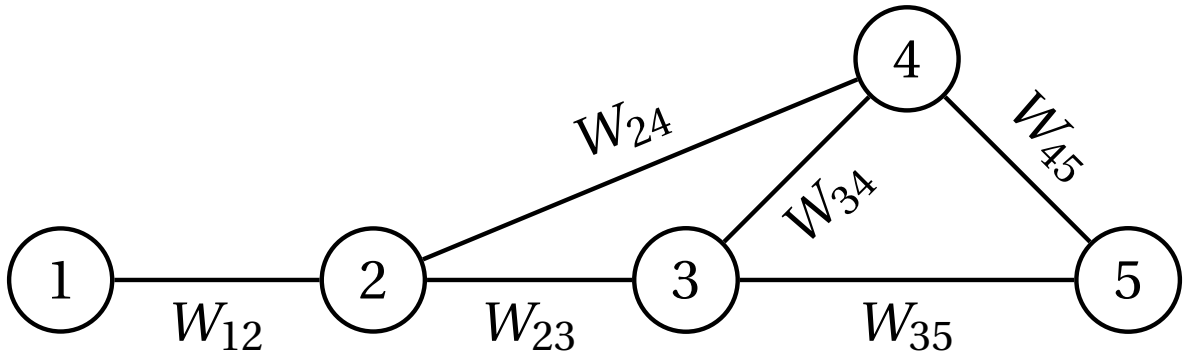
NGWPs *Natural Graph Wavelet Packets*: Cloninger-Li-Saito (2021)

Outline

What is a graph?

Let G be a *graph*.

- $V = V(G) = \{v_1, v_2, \ldots, v_N\}$ is the set of *nodes* (or *vertices*, where $N := |V(G)|$. For simplicity, we usually use i in place of v_i.
- $E = E(G) = \{e_1, e_2, \ldots, e_M\}$ is the set of *edges*, where $e_k = (i, j)$ represents an edge connecting adjacent nodes i and j for some $1 \le i, j \le N$, and $M := |E(G)|$.
- $W = W(G) \in \mathbb{R}^{N \times N}$ is the *edge weight matrix*, where W_{ij} the edge weight between i and j.



How to define W_{ij}?

There are many ways to define W_{ij}.

- For *unweighted* graphs, we use

$$W_{ij} := \begin{cases} 1, & \text{if } i \sim j \text{ (i.e., } i \text{ and } j \text{ are adjacent)}, \\ 0, & \text{otherwise.} \end{cases}$$

  This is often referred to as the *adjacency matrix*.

- For *weighted* graphs, W_{ij} should indicate the *affinity* between $i \sim j$:

$$W_{ij} := 1/\operatorname{dist}(i,j),$$

  where $\operatorname{dist}(\cdot,\cdot)$ is a certain measure of *dissimilarity of information* at i and j, e.g.,
  - the Euclidean distance between the spatial location of i and j;
  - the Wasserstein distance between the vector-valued measurements recorded at i and j;
  - their exponential versions with scale parameters;
  - ...

Our Assumptions

In this talk, we assume that every graph is

Finite: $M, N < \infty$.

Undirected: Any $e_k \in E(G)$ does not specify a direction, i.e., W is symmetric.

Connected: Any two nodes $i, j \in V(G)$ are connected by a sequence of head-tail edges.

Simple: G does not have any *loops* (an edge connecting a node to itself) or *multiple edges* (more than one edge connecting a pair of nodes).

Note that our graphs may be weighted or unweighted.

Graph Laplacians

$$\begin{cases} D(G) := \mathrm{diag}(d_1, d_2, \ldots, d_N), & \text{the } \textit{degree matrix}, \text{ where } d_i := \sum_{j=1}^{N} W_{ij}, \\ L(G) := D(G) - W(G), & \text{the } \textit{(unnormalized) Laplacian matrix}. \end{cases}$$

We have:

- sorted eigenvalues $0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{N-1}$.
- associated eigenvectors $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N-1}$.

The eigenvectors form a basis for $\mathbb{R}^N$. In particular:

- L is *symmetric nonnegative definite* $\Longrightarrow$ the eigenvectors form an ONB.
- $\boldsymbol{\phi}_0 (= \mathbf{1}/\sqrt{N})$ is called the *DC vector*.
- $\boldsymbol{\phi}_1$ is called the *Fiedler vector*.

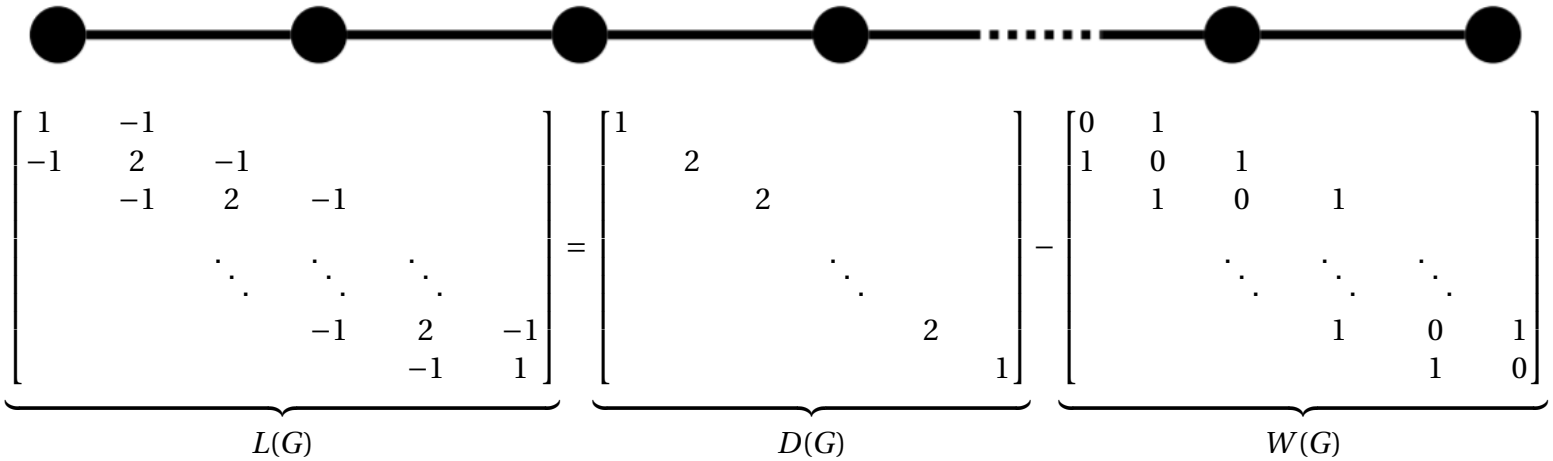
The *random-walk normalized Laplacian matrix* can be obtained by

$$L_{\mathrm{rw}}(G) := D(G)^{-1} L(G),$$

whereas the *symmetrically normalized Laplacian matrix* is defined as

$$L_{\mathrm{sym}}(G) := D(G)^{-1/2} L(G) D(G)^{-1/2}.$$

A Simple Yet Important Example: A Path Graph P_N



- The eigenvalues of L are $\lambda_l = 4\sin^2(\pi l/2N)$, $l = 0 : N-1$.
- The corresponding eigenvectors are $\boldsymbol{\phi}_l[x] = a_{l;N}\cos(\pi l(x+\frac{1}{2})/N)$, $l, x = 0 : N-1$; $a_{l;N}$ is a normalization const. s.t. $\|\boldsymbol{\phi}_l\|_2 = 1$.
- λ_l (eigenvalue) is a *monotonic* function w.r.t. the frequency l, and $\{\boldsymbol{\phi}_l\}_{l=0}^{N-1}$ are the *DCT Type II* basis vectors used in the JPEG standard while those of L_{sym} are the *DCT Type I* basis vectors [Strang (1999)], [Saito-Woei (2009)].
- So, many people view $\{\boldsymbol{\phi}_l\}_{l=0}^{N-1}$ and $\{\lambda_l\}_{l=0}^{N-1}$ as a generalization of the Fourier modes and their corresponding "frequencies" on *general* graphs, and have built graph wavelets based on this viewpoint $\Longrightarrow$ Lead to problems!

Outline

Outline

Graph Cut/Partitioning via Spectral Clustering

- Unlike the regular lattice cases assumed for the classical multiscale basis dictionaries, it is too difficult to apply a fixed and rigid partition (or cut) of a given general graph $\Longrightarrow$ No "center" edge to cut can be easily found a priori!
- We need a good method to *bipartition* a graph (i.e., cut it into two subgraphs of roughly equal sizes).
- In this talk, we use the *Fiedler vector* $\boldsymbol{\phi}_1$ of $L_{\mathrm{rw}}(G)$:

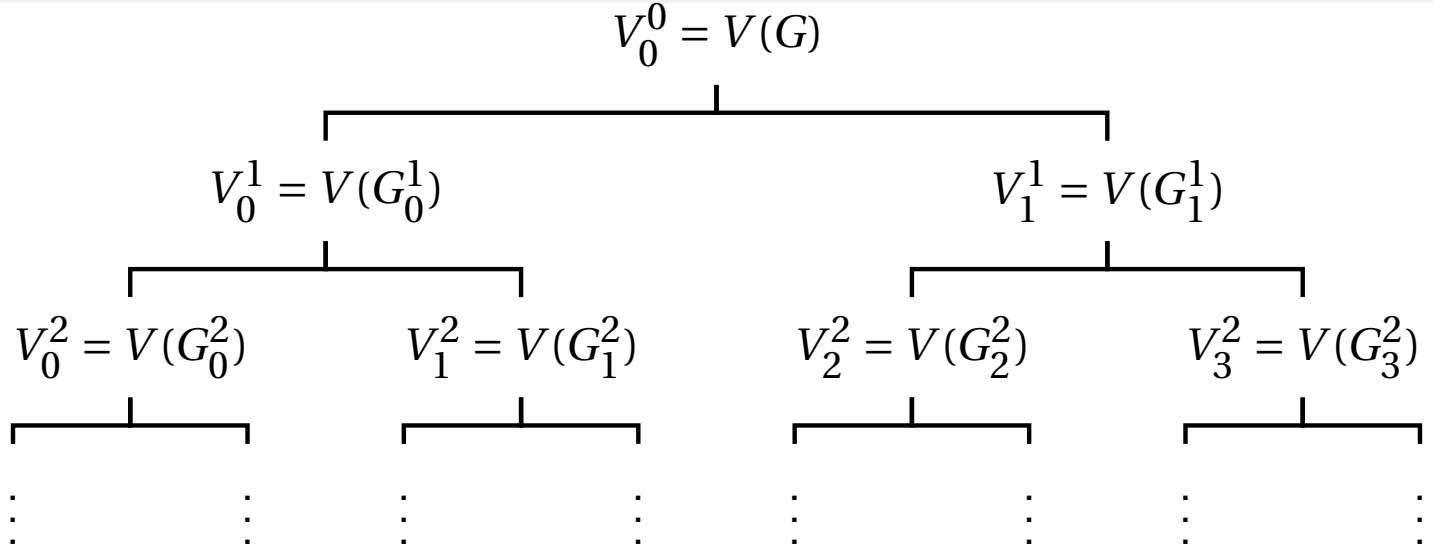
$$V_1(G) = \left\{ i \in V(G) \,|\, \boldsymbol{\phi}_1[i] \geq 0 \right\}, \quad V_2(G) = V(G) \setminus V_1(G),$$

  s.t. the subgraphs induced on V_1 and V_2 by G are both connected.
- This approximately minimizes the *Normalized Cut* (Shi-Malik, 2000):

$$\mathrm{NormalizedCut}(A, A^c) := \frac{\mathrm{cut}(A, A^c)}{\mathrm{vol}(A)} + \frac{\mathrm{cut}(A, A^c)}{\mathrm{vol}(A^c)}, \quad V(G) = A \sqcup A^c,$$

  where $\mathrm{cut}(A, A^c) := \sum_{i \in A, j \in A^c} W_{ij}$ indicates the quality of the cut (the smaller, the better), $\mathrm{vol}(A) := \sum_{i \in A} d_i$ is the so-called *volume* of the set A.
- Note that any other good graph bipartition methods can be used for building our multiscale graph basis dictionaries.

Hierarchical Bipartition Tree of a Graph



1. The root of the tree is the input whole graph, i.e., $G_0^0 = G$;
2. Let $N_k^j := |V_k^j|$, $j = 0 : j_{\max}$, $k = 0 : K_j - 1$. Each G_k^j with $N_k^j > 1$ is partitioned into exactly two subgraphs $G_{k'}^{j+1}$ and $G_{k'+1}^{j+1}$.
3. All regions (i.e., nodes in the subgraphs) on the same level are disjoint, i.e., $V(G_k^j) \sqcup V(G_{k'}^j) = \emptyset$ if $k \neq k'$;
4. The leaves of the tree are single-node graphs, i.e., $N_k^{j_{\max}} = 1$, $k = 0 : N - 1$;
5. (Optional) In practice, $N_{k'}^{j+1} \approx N_{k'+1}^{j+1}$ to reduce inefficiency.

Hierarchical Bipartition Tree of P_6

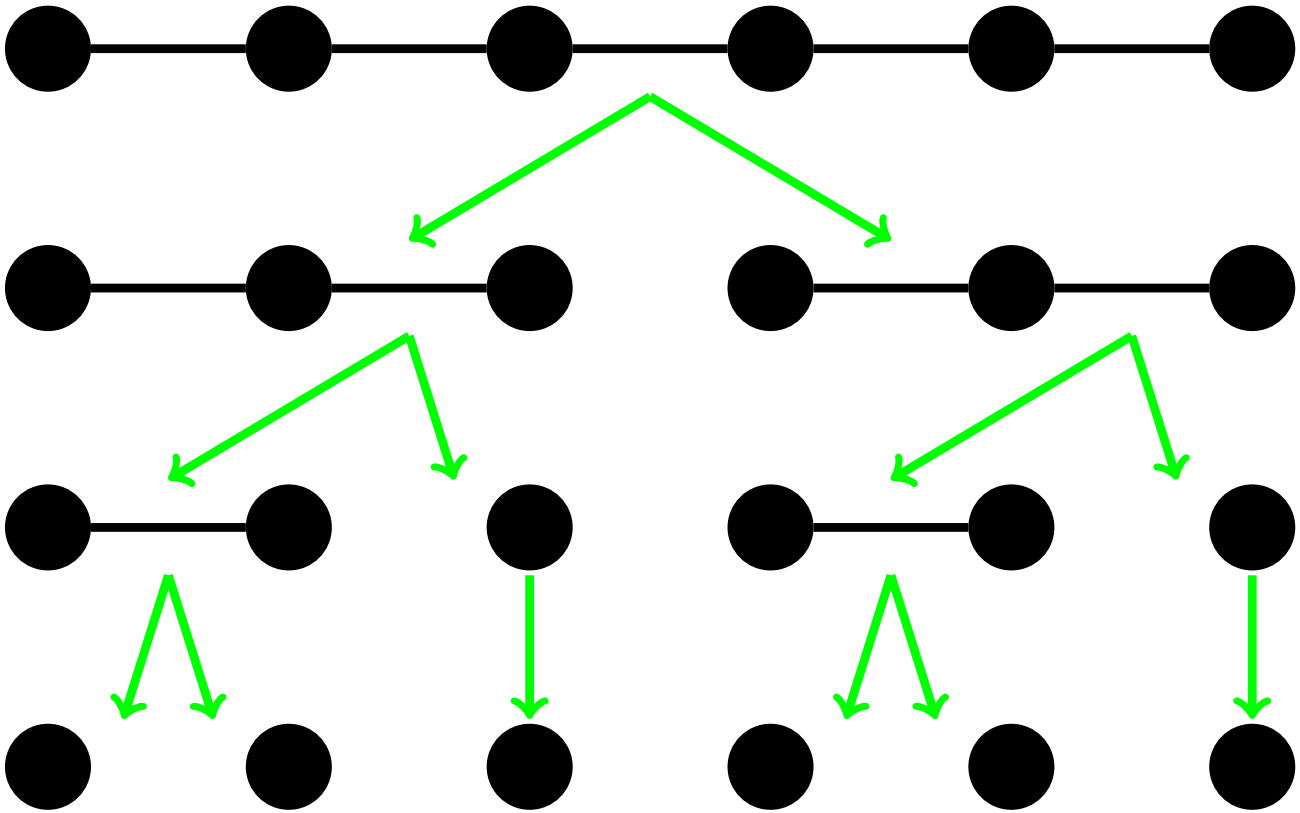


Figure: An example of a hierarchical bipartition tree for a path graph with $N = 6$ nodes, where the edge weights are equal. The root is the whole graph.

Hierarchical Bipartitions of Toronto Street Network ($N = 2275, M = 3381$)

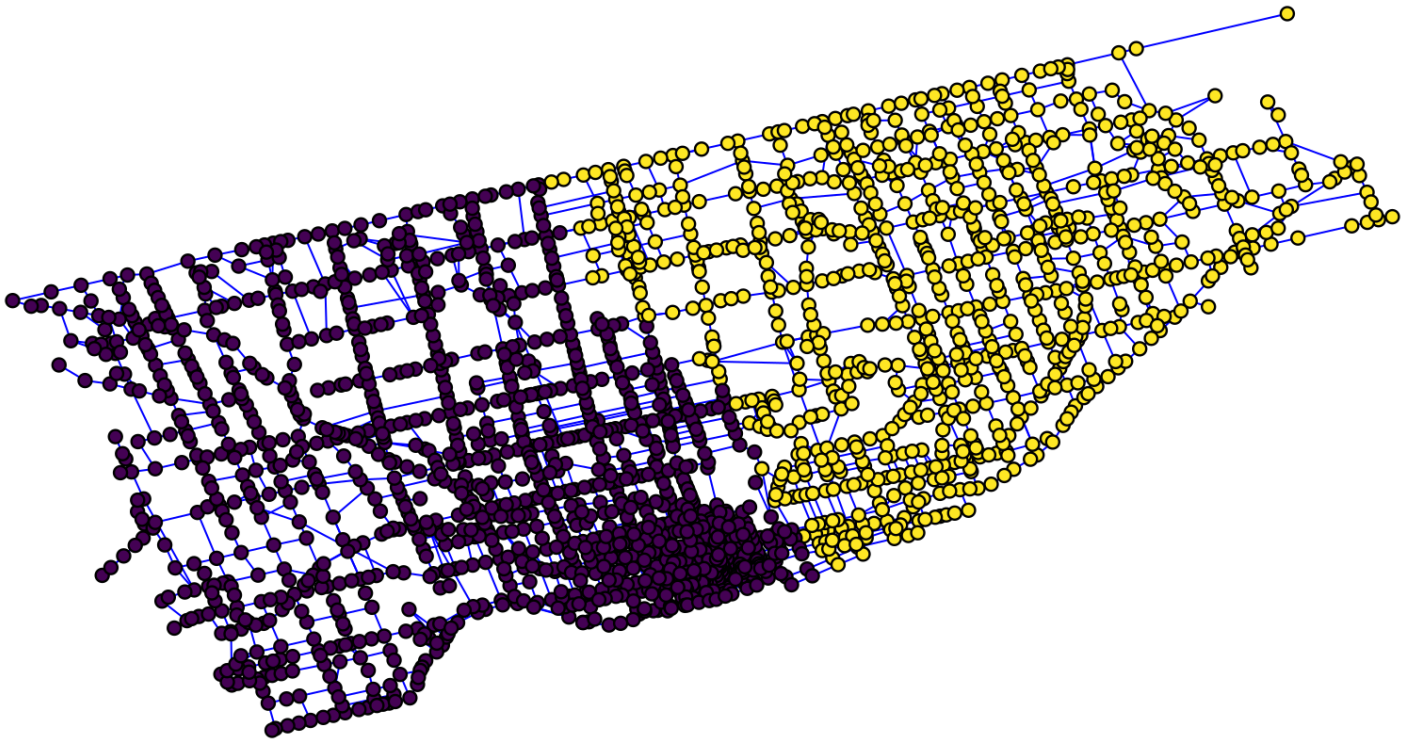


Figure: Level $j = 1$

Hierarchical Bipartitions of Toronto Street Network ($N = 2275, M = 3381$)

Figure: Level $j = 2$

Hierarchical Bipartitions of Toronto Street Network ($N = 2275, M = 3381$)

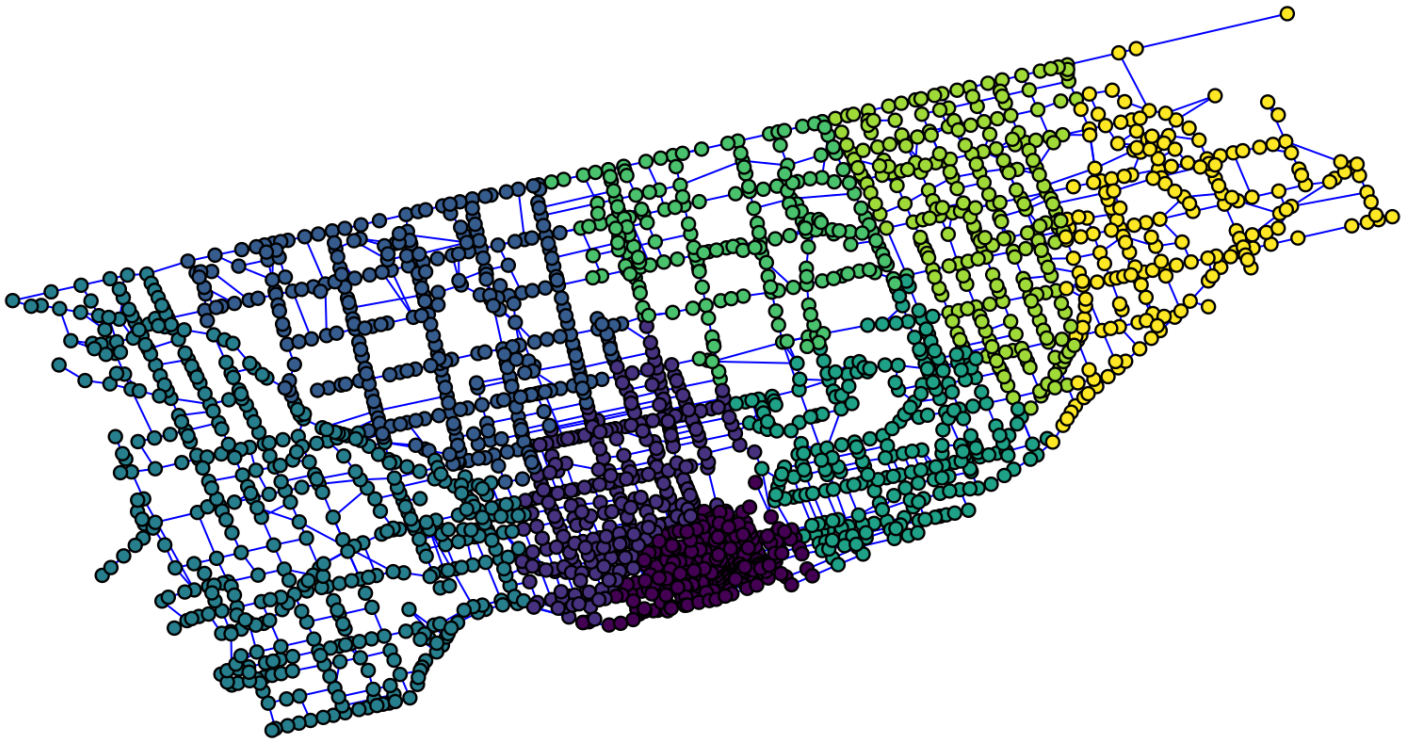


Figure: Level $j = 3$

Outline

Hierarchical Graph Laplacian Eigen Transform (HGLET)

- J. Irion and N. Saito proposed this in 2013, and published it in *JSIAM Letters* in 2014 $\Longrightarrow$ Awarded *the 2016 JSIAM Best Paper Prize*!
- Once a hierarchical bipartition tree of an input graph $G(V,E)$ is given, the idea is quite simple: at each G_k^j, compute the eigenvectors $\left\{\boldsymbol{\phi}_{k,l}^j\right\}_{l=0}^{N_k^j-1}$ of $L(G_k^j)$.
- Since $\operatorname{supp}\boldsymbol{\phi}_{k,\cdot}^j = V_k^j$, we extend its support to the entire V by appending zeros on $V \setminus V_k^j$ so that $\boldsymbol{\phi}_{k,l}^j \in \mathbb{R}^N$, $\forall\, j,k,l$.
- If $G = P_N$, then it *exactly* reduces to the Block Hierarchical DCT of N points. Furthermore, if $N = 2^{j_{\max}}$, the decomposition of a given signal on P_N is $O(N[\log_2 N]^2)$ thanks to the FFT!
- For a general graph G, it takes $O(N^3)$ mainly due to the eigenvector computations.
- One can also construct the HGLET with the eigenvectors of $L_{\mathrm{sym}}(G_k^j)$ (ONBs) or those of $L_{\mathrm{rw}}(G_k^j)$ ("weighted" ONBs) as an option.

HGLET Basis Vectors on the Toronto Street Network

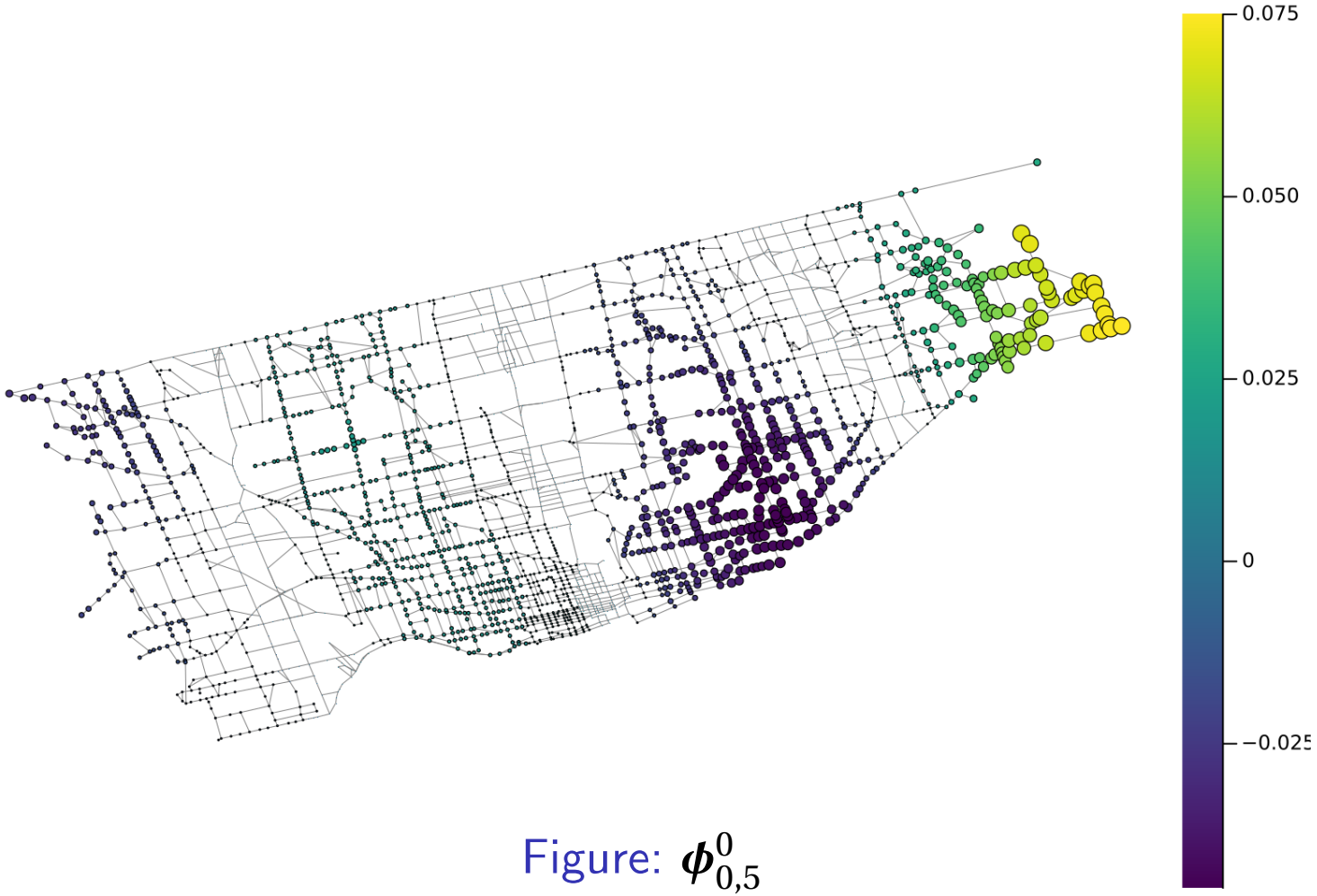


Figure: $\boldsymbol{\phi}_{0,5}^{0}$

HGLET Basis Vectors on the Toronto Street Network

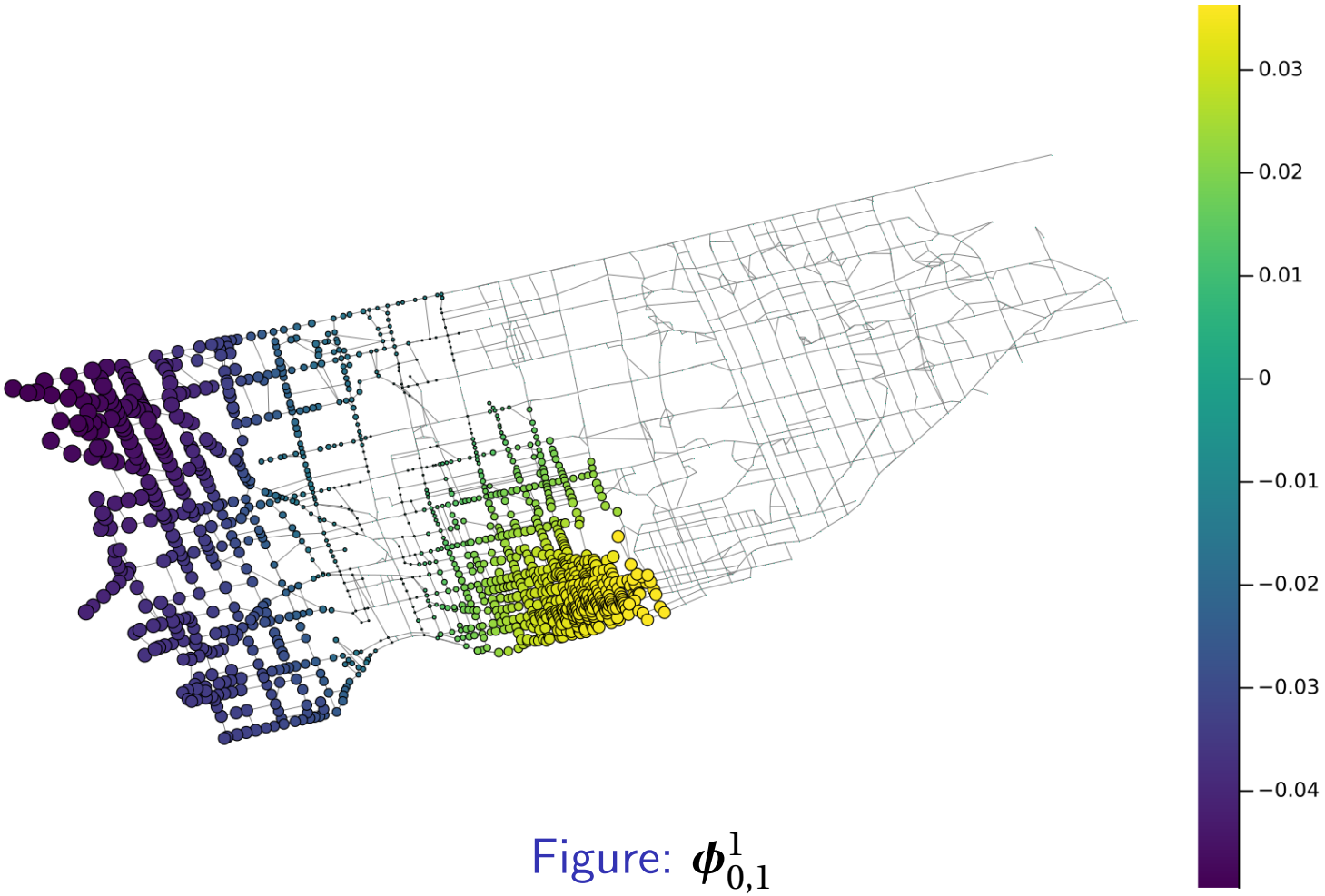


Figure: $\boldsymbol{\phi}_{0,1}^{1}$

HGLET Basis Vectors on the Toronto Street Network

Figure: $\boldsymbol{\phi}_{0,5}^{2}$

HGLET Basis Vectors on the Toronto Street Network

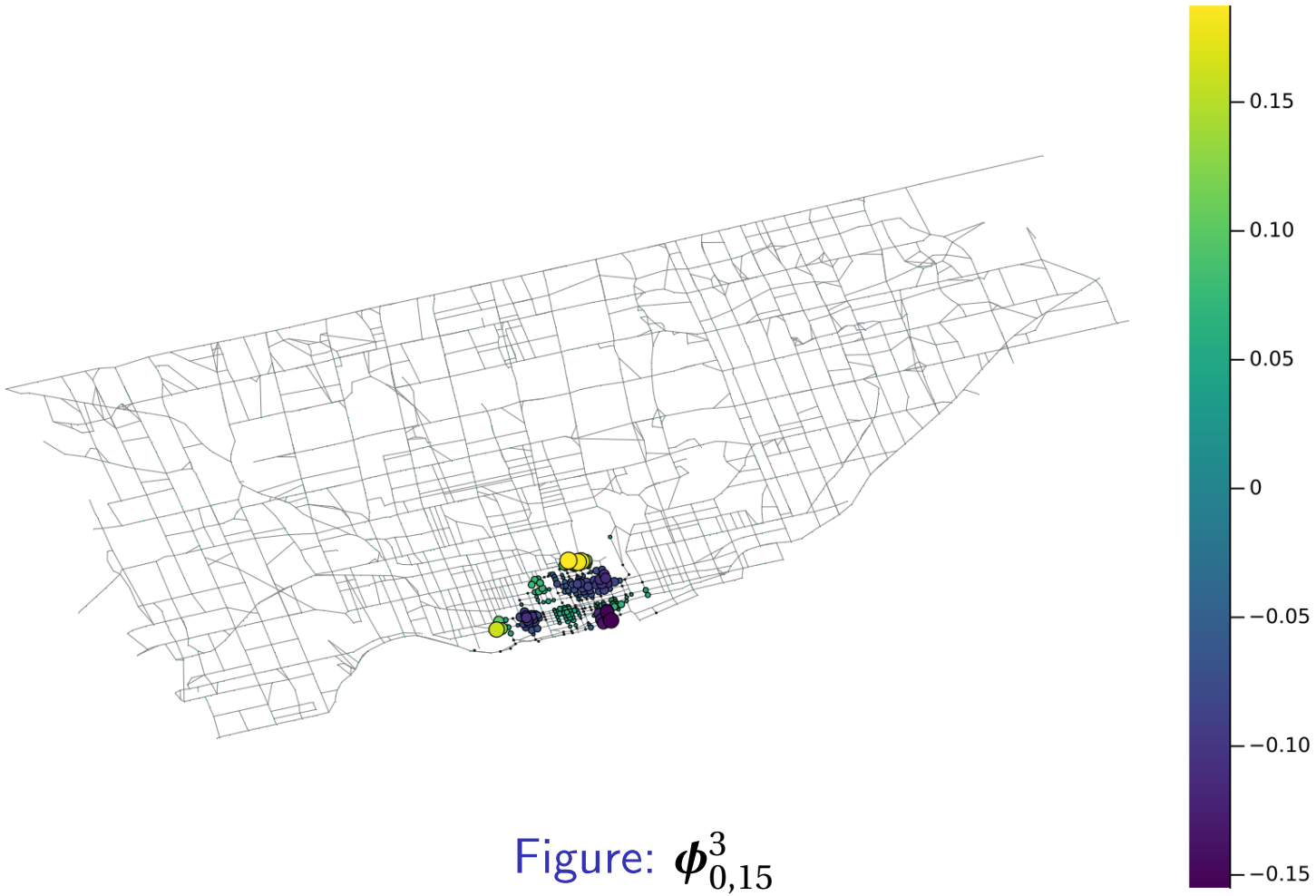


Figure: $\boldsymbol{\phi}^3_{0,15}$

HGLET Basis Vectors on the Toronto Street Network

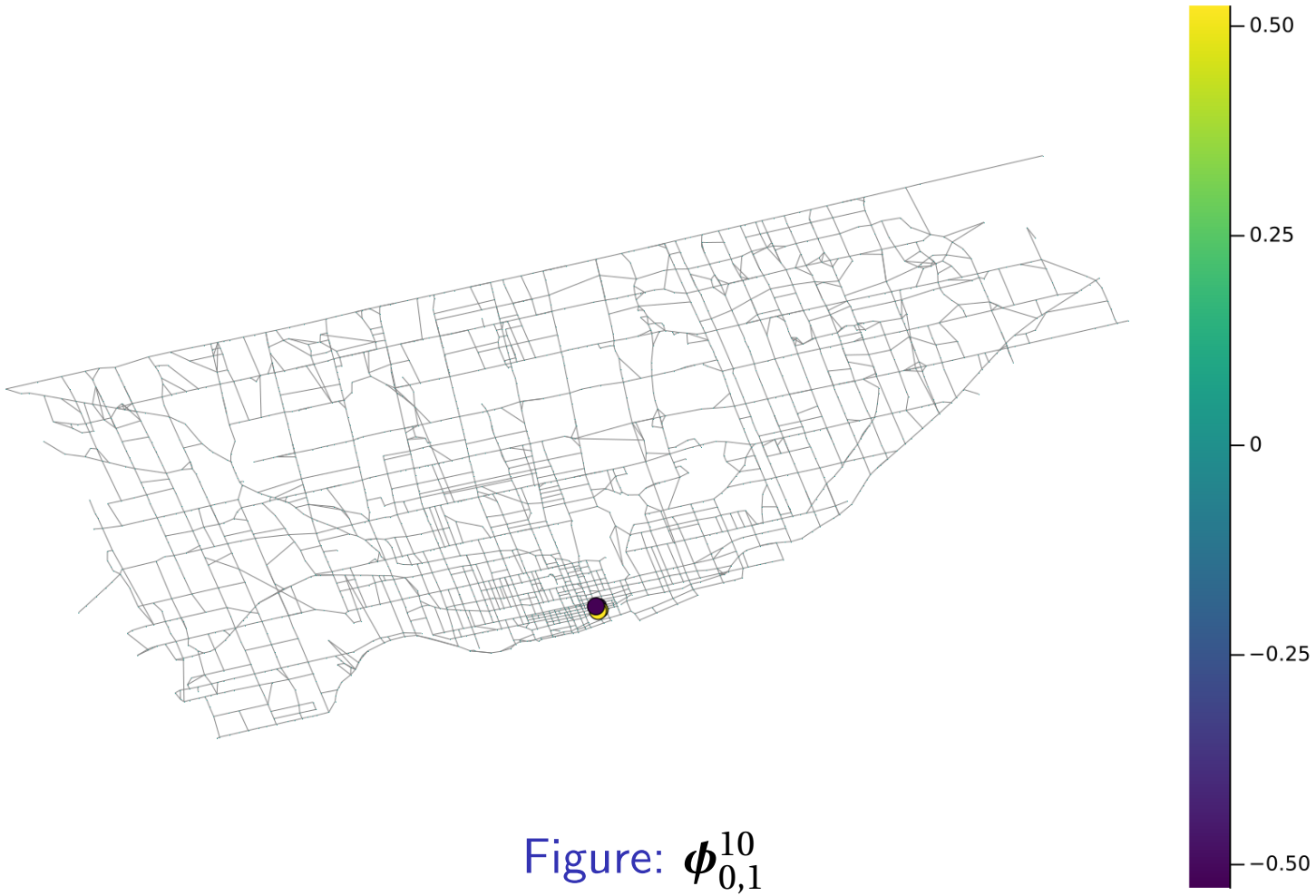


Figure: $\boldsymbol{\phi}_{0,1}^{10}$

Outline

Outline

The *Dual* Domain of a Graph

- Want to build graph wavelet packets that are *smooth* in the primal domain (i.e., on a given graph)
- In the classical setting (e.g., the regular lattices in $\mathbb{R}^d$), the *dual* domain of an input signal is well-defined $\Longrightarrow$ the *reciprocal* regular lattice representing the Fourier modes via the *Discrete Fourier Transform* (DFT).
- In the graph setting, using graph Laplacian eigenvectors as "cosines" or Fourier modes on graphs with eigenvalues as (the square of) their "frequencies" has been quite popular ...
- In fact, if an input graph is a simple path P_N or a simple circle C_N, then the situation is the same as the classical setting.

Problems of Interpreting "eigenvalues ≈ frequencies[2]"

- However, the notion of *frequency* is ill-defined on general graphs; Fourier modes on graphs may be quite different from those on regular lattices $\Longrightarrow$ extremely localized eigenvectors
- Graph Laplacian eigenvectors may also exhibit peculiar behaviors depending on *topology* and *structure* of given graphs!
- Spectral Graph Wavelet Transform (SGWT) [Hammond et al. (2011)] derived wavelets on a graph based on *the Littlewood-Paley theory* that organized the graph Laplacian eigenvectors corresponding to *dyadic partitions of eigenvalues* by viewing the eigenvalues as "frequencies"
- Unfortunately, this view may face difficulty for graphs more complicated than P_N or C_N.

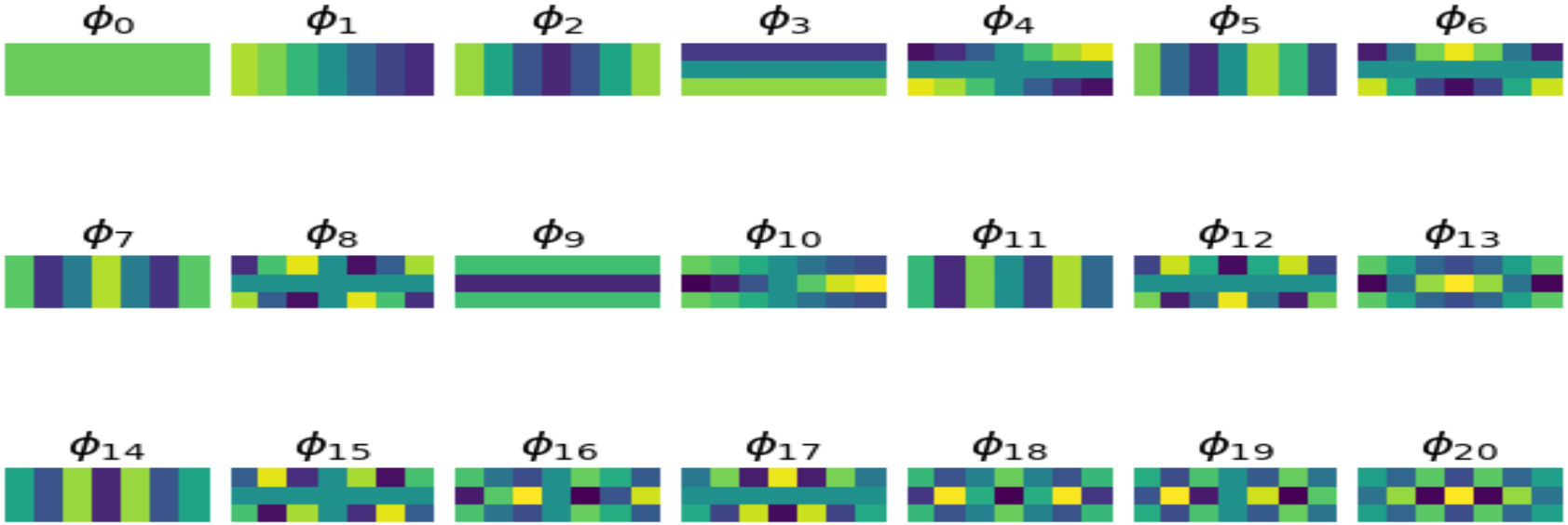
Problem with 2D Lattice Graph

- As soon as the domain becomes *even slightly more complicated than* P_N or C_N, the situation completely changes: *we cannot view the eigenvalues as a simple monotonic function of frequency anymore.*
- For example, consider a thin strip in $\mathbb{R}^2$, and suppose that the domain is discretized as $P_m \times P_n$ $(m > n)$, whose Laplacian eigenpairs are:

$$\lambda_k = 4\left[\sin^2\left(\frac{\pi k_x}{2m}\right) + \sin^2\left(\frac{\pi k_y}{2n}\right)\right],$$

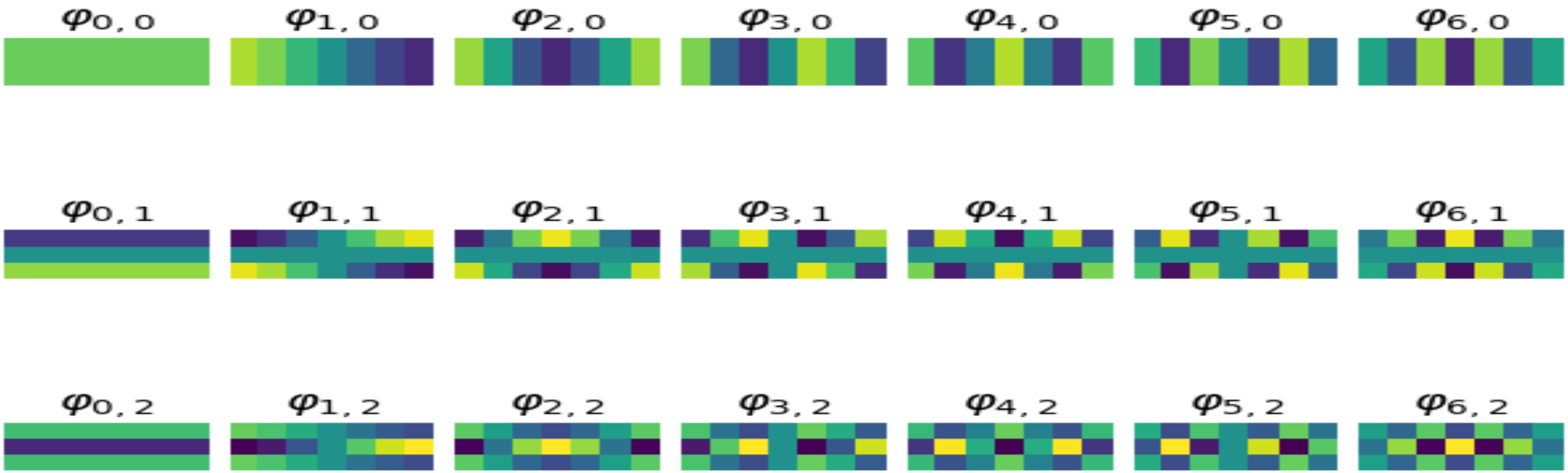
$$\boldsymbol{\phi}_k[x,y] = a_{k_x;m}\, a_{k_y;n} \cos\left(\frac{\pi k_x}{m}\left(x+\frac{1}{2}\right)\right)\cos\left(\frac{\pi k_y}{n}\left(y+\frac{1}{2}\right)\right),$$

  where $k = 0:mn-1$; $k_x = 0:m-1$; $k_y = 0:n-1$; $x = 0:m-1$; and $y = 0:n-1$.
- Any numerical eigenpair solver gives you the sequence of $\{(\lambda_k, \boldsymbol{\phi}_k)\}_{k=0:mn-1}$, but *it doesn't tell you what k_x and k_y are for a given k!!*

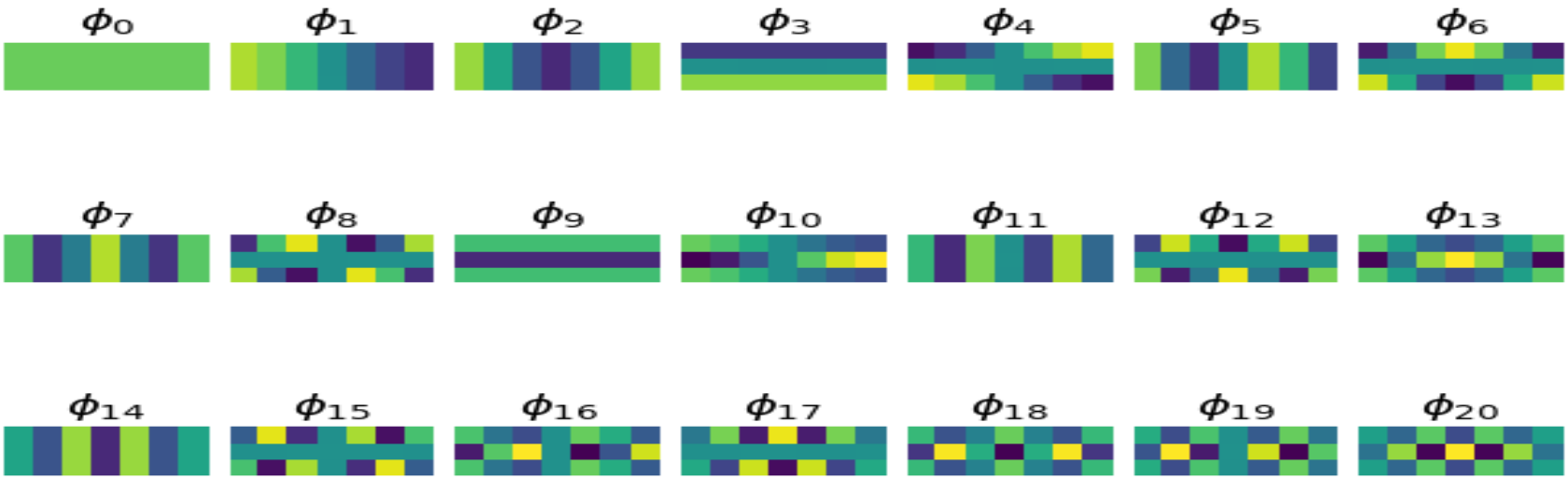


- All of a sudden the eigenvalue of a completely different type of oscillation sneaks into the eigenvalue sequence.
- Hence, on a general domain or a general graph, by simply looking at the Laplacian eigenvalue sequence $\{\lambda_k\}_{k=0,1,\ldots}$, it is *almost impossible to organize the eigenpairs into physically meaningful dyadic blocks* and apply the Littlewood-Paley approach unless the underlying domain is of very simple nature, e.g., P_n or C_n.
- For complicated domains, the notion of *frequency* is not well-defined anymore.

What we want to do is to *organize* those eigenvectors as



instead of



Outline

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}^{N\times N}_{\geq 0}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D^2_{ij}/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}^{N\times N}_{\geq 0}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D^2_{ij}/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}^{N\times N}_{\geq 0}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D^2_{ij}/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}^{N\times N}_{\geq 0}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D^2_{ij}/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}^{N\times N}_{\geq 0}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D_{ij}^2/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Plan

- How can we *quantify the difference between the eigenvectors*?
- The usual ℓ^2-distance doesn't work since $\left\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\right\|_2 = \sqrt{2}\delta_{i\neq j}$.
- Need to come up with a metric that quantifies the "behavioral" differences between any pair of eigenvectors. Having such a metric, we do the following:

1. Choose a metric and compute the "distance" between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ for all $i, j = 0 : N-1$, which results in a "distance" matrix $D \in \mathbb{R}_{\geq 0}^{N\times N}$
2. Construct a *dual graph* $G^\star(V^\star, E^\star)$ where the ith node corresponds to $\boldsymbol{\phi}_i$, and the weight of the edge (i, j) is the *affinity* between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, e.g., $1/D_{ij}$ or $\exp(-D_{ij}^2/\epsilon)$ with some $\epsilon > 0$
3. *Organize* and *group* $V^\star$ to generate *wavelet packet vectors* on G

Various Metrics for Eigenvector Differences

- A *similarity* measure (HAD) based on the *average of local correlations* of eigenvectors [Cloninger-Steinerberger (2018)]
- The *ramified optimal transport* (ROT) method [Saito (2018)]
- The *difference of absolute gradient* (DAG) method [Li-Saito (2019)]
- The *time-stepping diffusion* (TSD) method [Li-Saito (2019)]
- The ROT method seems to work well for transportation networks with hubs (e.g., neuronal dendritic trees) whereas the HAD and DAG methods seem to work well for (ir)regular grids/lattices (e.g., road networks).
- For more pros and cons of these methods, see Li-Saito (2019).

Difference of Absolute Gradient (DAG) Pseudometric

- The basic idea: use the *absolute gradient* of each eigenvector as its feature vector describing its behavior.
- Let $G(V,E)$ be an input graph (connected, undirected, weighted, and simple) with $|V|=N$, $|E|=M$. Let $Q \in \mathbb{R}^{N\times M}$ be its *incidence matrix*. Then, *DAG **pseudo**metric* (*the identity of discernible* is not satisfied) is defined as:

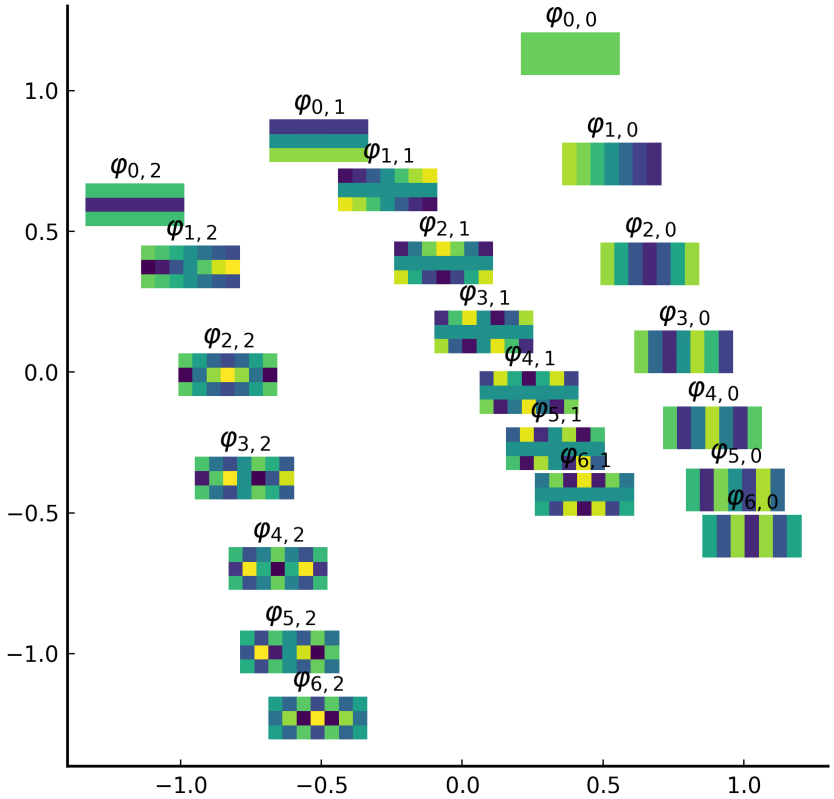
$$d_{\mathrm{DAG}}(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j) := \| |\nabla_G|\boldsymbol{\phi}_i - |\nabla_G|\boldsymbol{\phi}_j \|_2 \quad \text{where } |\nabla_G|\boldsymbol{\phi} := \mathrm{abs}.(Q^{\mathsf{T}}\boldsymbol{\phi}) \in \mathbb{R}^M_{\geq 0}$$

- It is related to the *Hadamard product-based affinity* proposed by Cloninger-Steinerberger (2018) as

$$\begin{aligned} d_{\mathrm{DAG}}(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)^2 &= \left\langle |\nabla_G|\boldsymbol{\phi}_i - |\nabla_G|\boldsymbol{\phi}_j, |\nabla_G|\boldsymbol{\phi}_i - |\nabla_G|\boldsymbol{\phi}_j \right\rangle_E \\ &= \left\langle |\nabla_G|\boldsymbol{\phi}_i, |\nabla_G|\boldsymbol{\phi}_i \right\rangle_E + \left\langle |\nabla_G|\boldsymbol{\phi}_j, |\nabla_G|\boldsymbol{\phi}_j \right\rangle_E - 2\left\langle |\nabla_G|\boldsymbol{\phi}_i, |\nabla_G|\boldsymbol{\phi}_j \right\rangle_E \\ &= \lambda_i + \lambda_j - \sum_{x\in V}\sum_{y\sim x} |\boldsymbol{\phi}_i(x) - \boldsymbol{\phi}_i(y)| \cdot |\boldsymbol{\phi}_j(x) - \boldsymbol{\phi}_j(y)| \quad \text{thanks to } QQ^{\mathsf{T}} = L \end{aligned}$$

  where $\langle\cdot,\cdot\rangle_E$ is the inner product over edges.
- The last term of the formula can be viewed as *a global average of absolute local correlation* between eigenvectors $\Longrightarrow$ the Hadamard-product affinity.
- Given the eigenvectors, *the computational cost is $O(M)$* for each $d_{\mathrm{DAG}}(\cdot,\cdot)$ eval.

2D Lattice $P_7 \times P_3$: d_{DAG} visualized by the classical MDS



- d_{DAG} nicely detects two directions of the oscillations and the eigenvectors are organized naturally in the *2D lattice*.
- For each column of the lattice, the eigenvectors have the same oscillation pattern in the y-direction and the oscillation in the x-direction increases linearly, and vice versa.

Outline

Natural Graph Wavelet Packets: Basic Steps

1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of $G =$ the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:

Natural Graph Wavelet Packets: Basic Steps

1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of $G =$ the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:

Natural Graph Wavelet Packets: Basic Steps

1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of $G =$ the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:

Natural Graph Wavelet Packets: Basic Steps

1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of $G =$ the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:

Natural Graph Wavelet Packets: Basic Steps

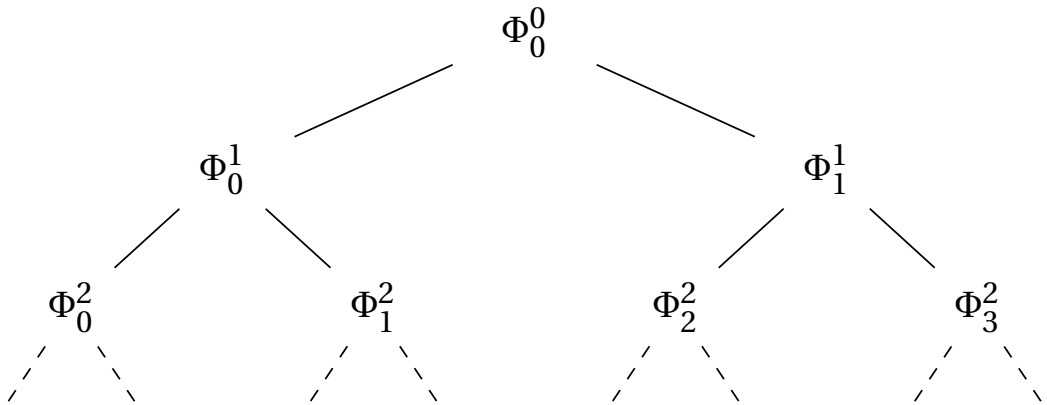
1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of $G =$ the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:

Natural Graph Wavelet Packets: Basic Steps

1. *Bipartition the dual graph $G^\star$ recursively* via any method, e.g., spectral graph bipartition using the *Fiedler vectors*
2. Generate wavelet packet vectors on each subgraph of $G^\star$ that are *well localized on G*

- We refer to the graph wavelet packets generated by the above strategy as *Natural Graph Wavelet Packets* (NGWPs).
- $\exists$ Several possibilities in Step 2; will discuss only one of them today.
- The details were published as part of the *Topical Collection: Harmonic Analysis on Combinatorial Graphs* of the *Journal of Fourier Analysis and Applications* this year.
- Let $\Phi_0^0 := [\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}]$ = a matrix representation of the GL eigenvectors of G = the node set $V^\star$ of $G^\star$, and suppose we get the following *hierarchical bipartition tree* of Φ_0^0:



Hierarchical Bipartitioning of $(P_7 \times P_3)^*$

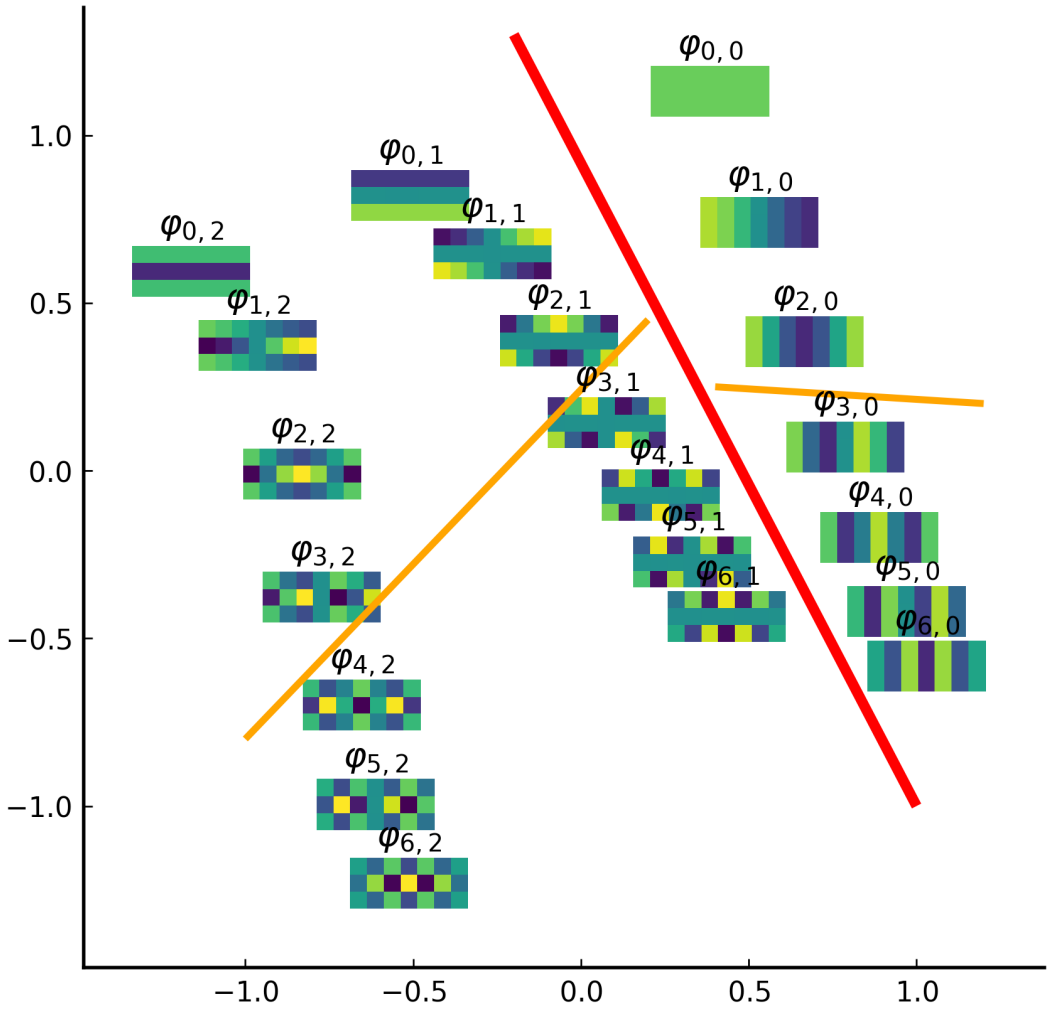


Figure: The hierarchical bipartition algorithm applied to $(P_7 \times P_3)^*$ with $J = 2$. The thick red line: bipartition at $j = 1$; the orange lines: bipartitions at $j = 2$.

NGWPs by Varimax Rotation

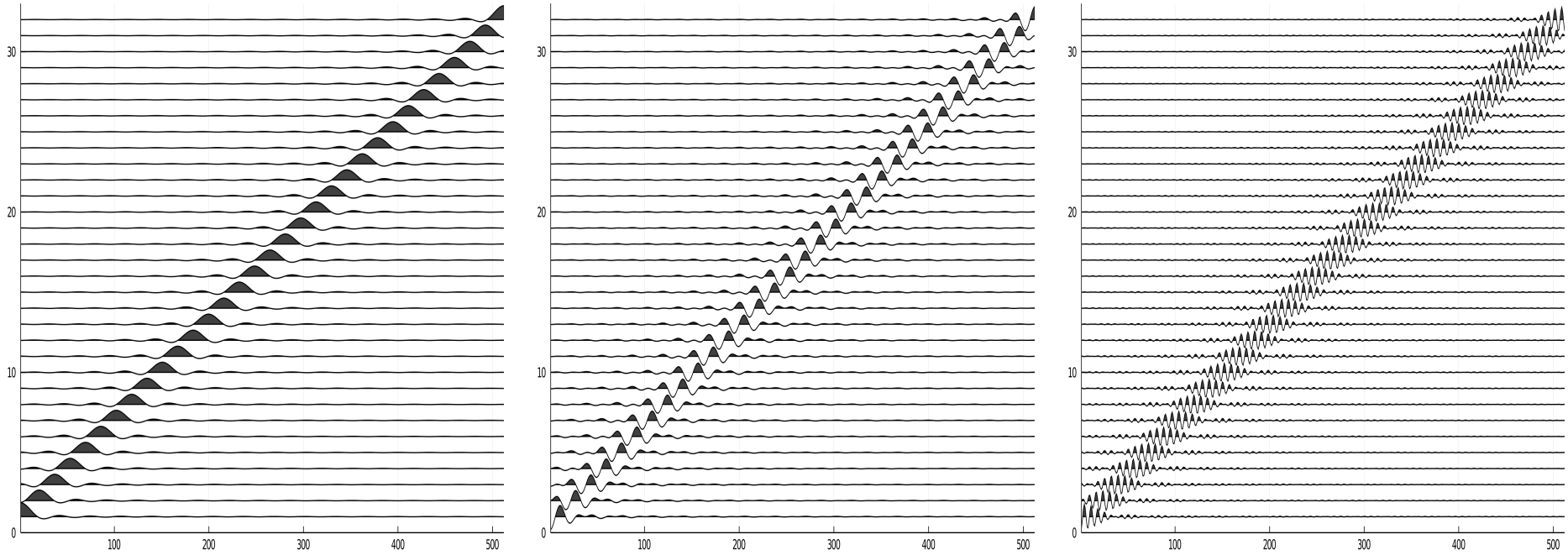
- Given the full binary partition tree of $\Phi_0^0 \in \mathbb{R}^{N \times N}$, perform the *varimax rotation* on $\Phi_k^j \in \mathbb{R}^{N \times N_k^j}$ for each j and k.
- Varimax rotation [Kaiser (1958); Jennrich (2001)] is an orthogonal rotation, often used in *factor analysis*, to maximize the variances of energy distribution (or a scaled version of the *kurtosis*) of the input column vectors.
- Thanks to the orthonormality of columns of Φ_k^j, this is equivalent to finding an orthogonal rotation that maximizes the overall *4th order moments*, i.e.,

$$\Psi_k^j := \Phi_k^j \cdot R_k^j, \quad \text{where } R_k^j = \arg \max_{R \in \mathrm{SO}(N_k^j)} \sum_{p=1}^{N} \sum_{q=1}^{N_k^j} \left[\left(\Phi_k^j \cdot R \right)^4 \right]_{p,q}.$$

- The column vectors of Ψ_k^j are *more "localized" in the primal domain G* than those of Φ_k^j. This type of localization procedure is important since the GL eigenvectors in Φ_k^j may be of *global* nature in general.

The VM-NGWPs on P_N

are essentially the *Shannon wavelet packets !*



(a) Father wavelet vectors Ψ_0^4 (b) Mother wavelet vectors Ψ_1^4 (c) Wavelet packet vectors Ψ_4^4

Figure: Some of the Shannon wavelet packet vectors on P_{512}

The VM-NGWPs on $P_7 \times P_3$

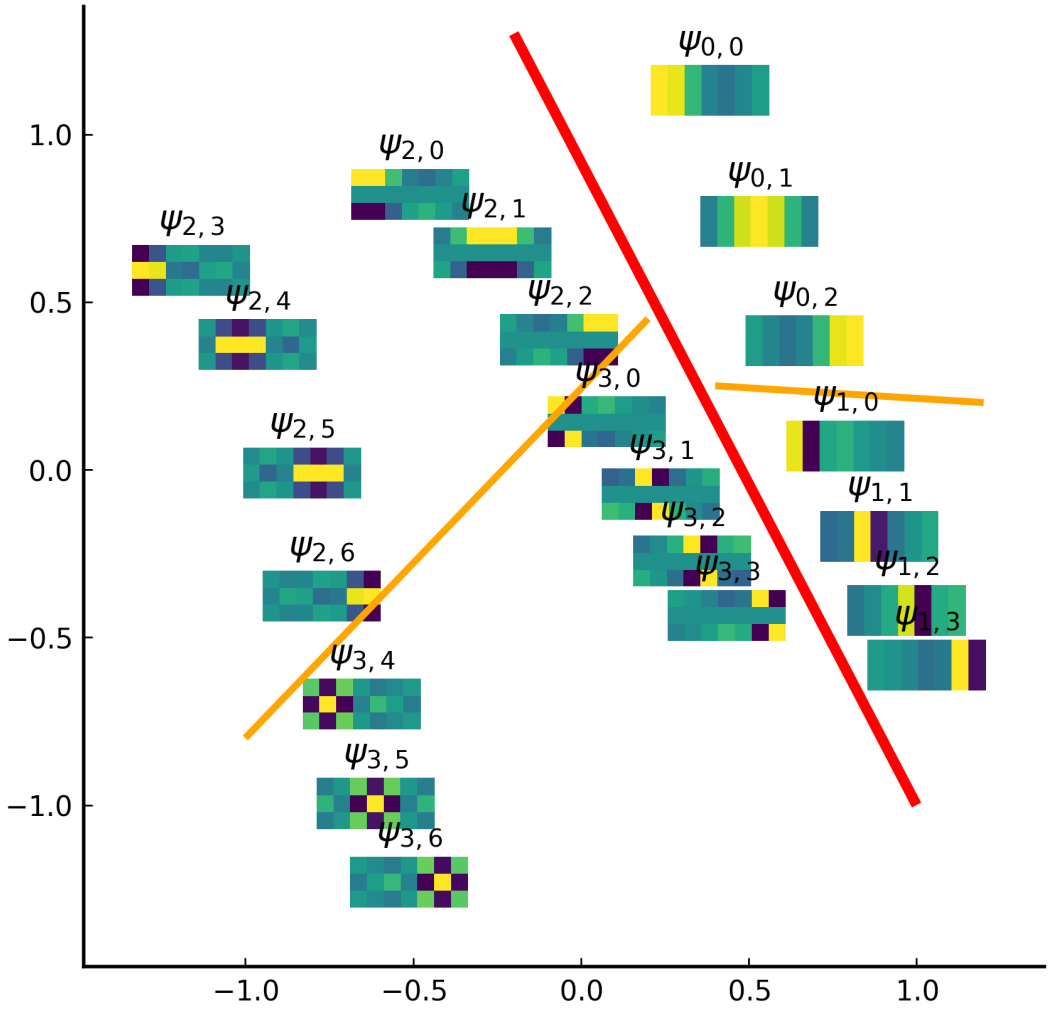
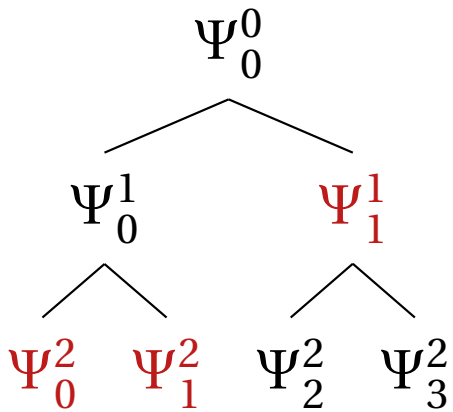


Figure: The VM-NGWP vectors of $P_7 \times P_3$. The column vectors of the basis matrix Ψ_k^2 are denoted as $\psi_{k,l}$, $l = 0, 1, \ldots$

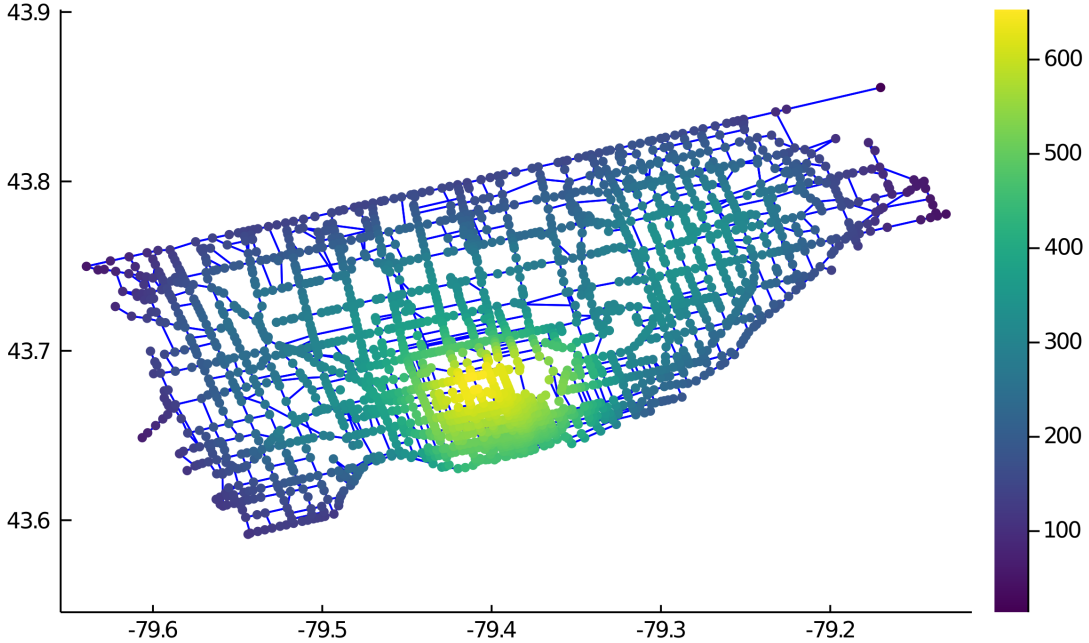
Selecting a Good Natural Graph Wavelet Packet Basis

- Once the NGWP dictionary is built, one can apply the *best-basis selection algorithm* of Coifman-Wickerhauser or its variants developed by the Saito group to choose the most suitable basis for a given task (e.g., efficient approximation, denoising, classification, regression, etc.). Note that the best-basis algorithm searches the best one among more than $(1.5)^N$ possible ONBs from the wavelet packet dictionary.
- For the examples today, we used the *ℓ^1-norm minimization* to select the best (or *sparsest*) basis among the NGWP dictionary.
- Of course, one can also choose the *wavelet basis* explicitly from the NGWP dictionary by selecting the specific subspaces and basis vectors.



Outline

Toronto Street Network ($N = 2275$ nodes; $M = 3381$ edges)

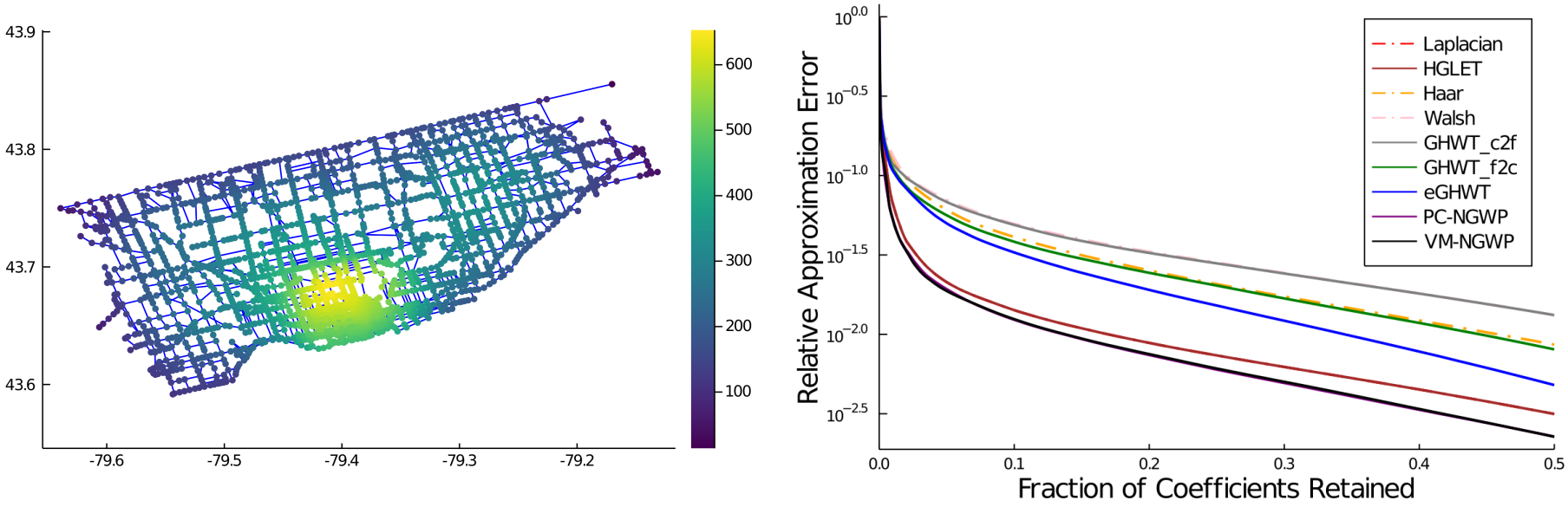


- Obtained from the City of Toronto's open data portal:

`https://open.toronto.ca/dataset/traffic-signal-vehicle-and-pedestrian-volumes`

- Nodes = intersections (with traffic lights); edges = streets
- Edge weights = the inverse Euclidean distances between nodes
- Two graph signals considered: 1) spatial distribution of the street intersections; 2) real pedestrian counts between the hours of 7:30am and 6:00pm on a single day measured during the period 03/22/2004–02/28/2018

Example 1: Smooth Histogram of Street Intersections

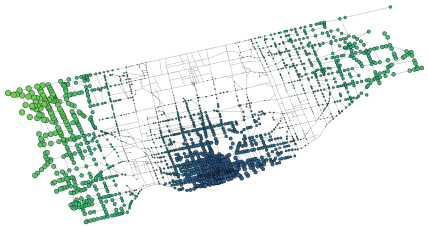


(a) Graph signal

(b) Approximation

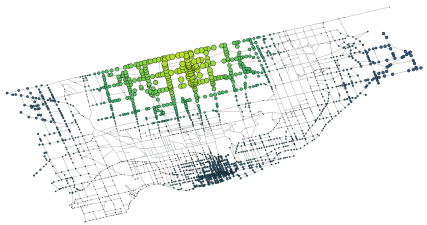
Figure: (a) A graph signal representing the smooth spatial distribution of the street intersections on the Toronto street network Toronto. The horizontal and vertical axes of this plot represent the longitude and latitude geo-coordinates of this area, respectively. (b) The results of our approximation experiments.

Top 12 VM-NGWP best basis vectors (w.o. DC)

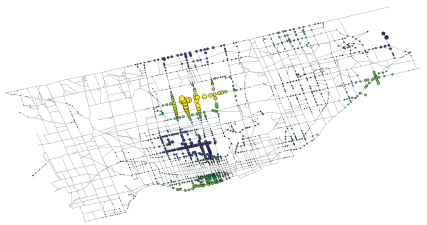


(a) $\boldsymbol{\psi}^{43}_{2,0} \equiv \boldsymbol{\phi}_2$

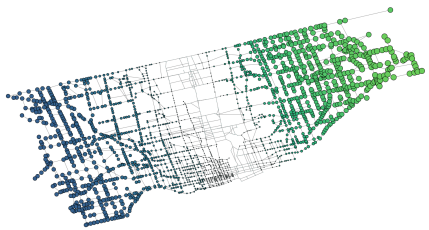
(b) $\boldsymbol{\psi}^{6}_{2,0}$



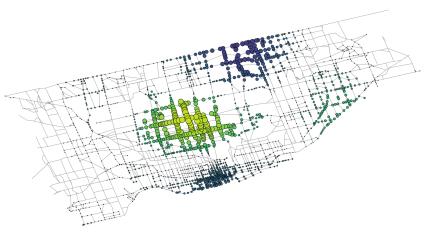
(c) $\boldsymbol{\psi}^{6}_{1,0}$



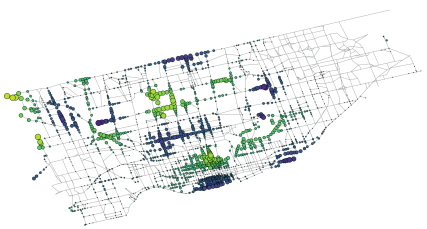
(d) $\boldsymbol{\psi}^{6}_{5,2}$



(e) $\boldsymbol{\psi}^{43}_{1,0} \equiv \boldsymbol{\phi}_1$



(f) $\boldsymbol{\psi}^{6}_{2,4}$

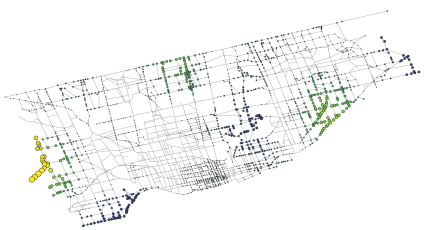


(g) $\boldsymbol{\psi}^{8}_{24,0}$

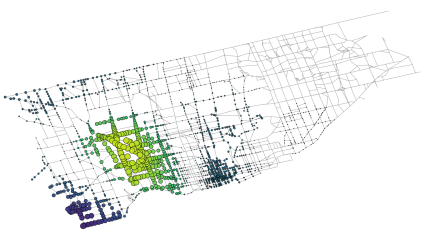
(h) $\boldsymbol{\psi}^{5}_{4,4}$

(i) $\boldsymbol{\psi}^{6}_{5,3}$

(j) $\boldsymbol{\psi}^{6}_{2,2}$

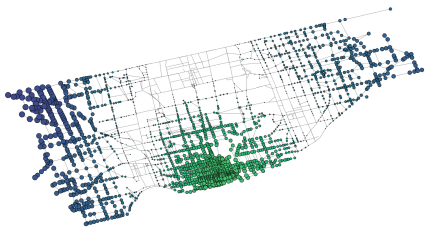


(k) $\boldsymbol{\psi}^{43}_{12,0} \equiv \boldsymbol{\phi}_{10}$

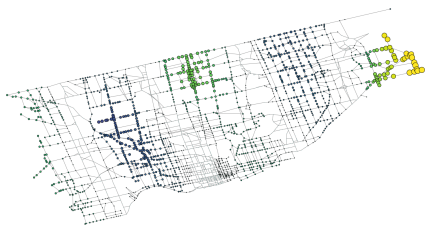


(l) $\boldsymbol{\psi}^{6}_{2,3}$

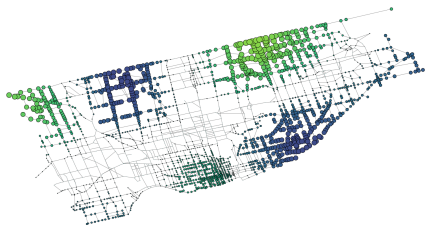
Top 12 HGLET best basis (= the global GL) vectors (w.o. DC)



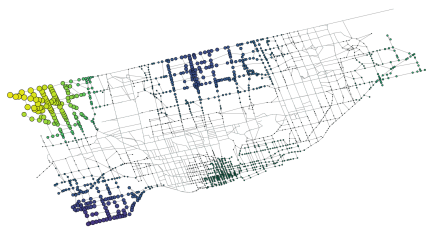
(a) $\boldsymbol{\phi}^0_{0,2}$



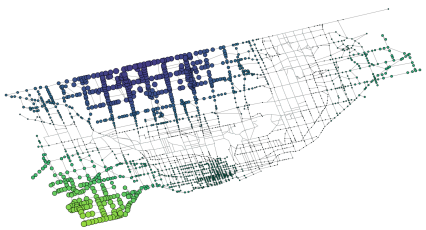
(b) $\boldsymbol{\phi}^0_{0,7}$



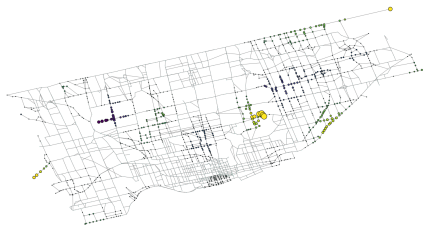
(c) $\boldsymbol{\phi}^0_{0,6}$



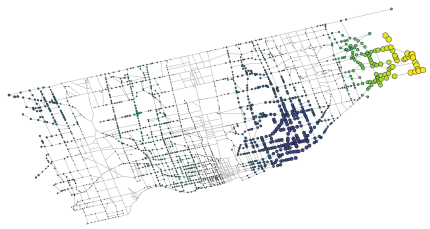
(d) $\boldsymbol{\phi}^0_{0,4}$



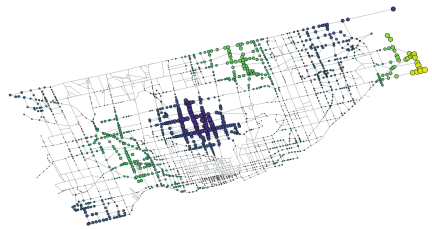
(e) $\boldsymbol{\phi}^0_{0,3}$



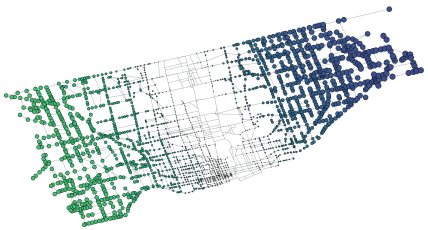
(f) $\boldsymbol{\phi}^0_{0,23}$



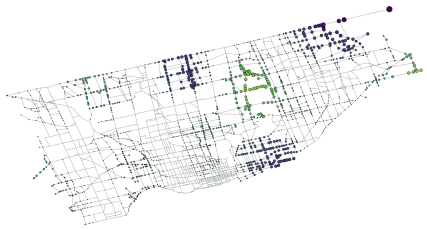
(g) $\boldsymbol{\phi}^0_{0,5}$



(h) $\boldsymbol{\phi}^0_{0,11}$

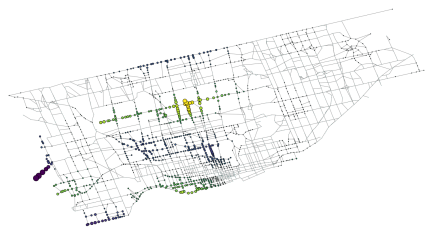


(i) $\boldsymbol{\phi}^0_{0,1}$



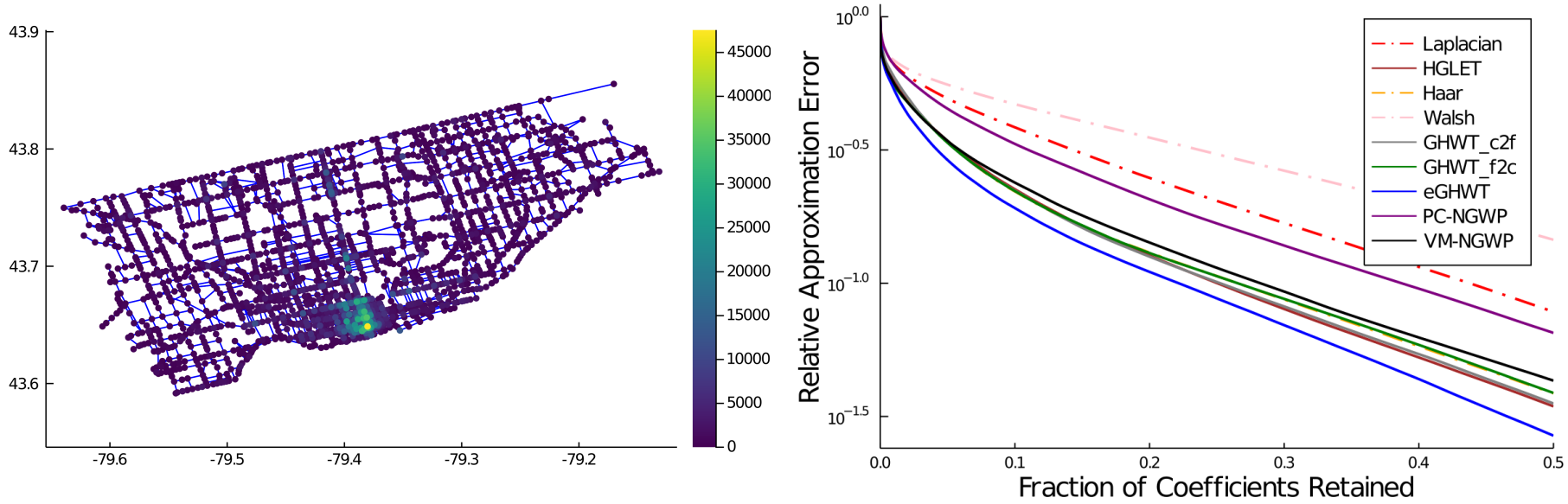
(j) $\boldsymbol{\phi}^0_{0,12}$

(k) $\boldsymbol{\phi}^0_{0,29}$



(l) $\boldsymbol{\phi}^0_{0,22}$

Example 2: Pedestrian Counts at the Intersections



(a) Graph signal

(b) Approximation

Figure: (a) Pedestrian volume data measured at the street intersections. (b) The results of our approximation experiments.

Top 12 HGLET best basis vectors

(a) $\boldsymbol{\phi}^{6}_{1,0}$

(b) $\boldsymbol{\phi}^{5}_{1,0}$

(c) $\boldsymbol{\phi}^{7}_{9,0}$

(d) $\boldsymbol{\phi}^{5}_{1,73}$

(e) $\boldsymbol{\phi}^{10}_{109,0}$

(f) $\boldsymbol{\phi}^{9}_{97,0}$

(g) $\boldsymbol{\phi}^{6}_{1,1}$

(h) $\boldsymbol{\phi}^{7}_{8,0}$

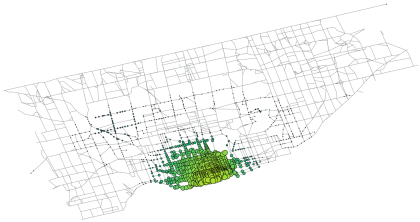
(i) $\boldsymbol{\phi}^{5}_{1,3}$

(j) $\boldsymbol{\phi}^{8}_{0,0}$

(k) $\boldsymbol{\phi}^{6}_{1,10}$

(l) $\boldsymbol{\phi}^{5}_{1,2}$

Top 12 VM-NGWP best basis vectors

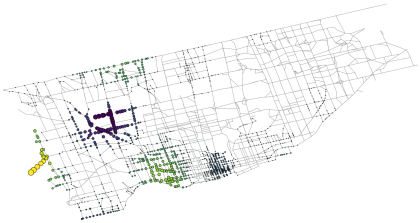


(a) $\boldsymbol{\psi}^4_{0,2}$

(b) $\boldsymbol{\psi}^3_{1,22}$

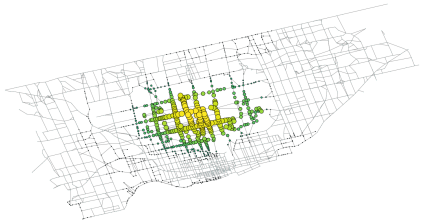
(c) $\boldsymbol{\psi}^3_{1,75}$

(d) $\boldsymbol{\psi}^1_{1,1245}$

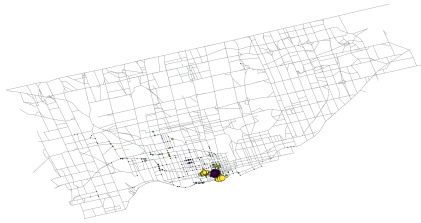


(e) $\boldsymbol{\psi}^7_{8,0}$

(f) $\boldsymbol{\psi}^3_{1,87}$



(g) $\boldsymbol{\psi}^4_{0,11}$



(h) $\boldsymbol{\psi}^2_{1,136}$

(i) $\boldsymbol{\psi}^4_{0,5}$

(j) $\boldsymbol{\psi}^2_{1,16}$

(k) $\boldsymbol{\psi}^3_{1,98}$

(l) $\boldsymbol{\psi}^1_{1,1660}$

Observations from Examples 1 & 2

- Basis vectors that behave like *oriented edge detectors* are *automatically generated* in our dictionaries
- *NGWP* outperformed the other dictionaries on *piecewise smooth* graph signals (e.g., smooth histogram of street intersections)
- *eGHWT* was the best performer on the *non-smooth and localized* graph signals (e.g., pedestrian volume data)
- Potential reason I: NGWPs are a direct generalization of the *Shannon* wavelet packets, i.e., their "frequency" domain support is localized and well controlled while the "time" domain support is not compact.
- Potential reason II: eGHWT tends to have better performance with oscillatory non-smooth signals in general compared to the other transforms and can choose the best one from even larger set of possible ONBs, i.e., $0.618 \cdot (1.84)^N$ [Saito-Shao, 2019; 2021].

Outline

Summary

- Lifted the classical multiscale basis dictionaries (i.e., Hierarchical Block DCT and the Shannon Wavelet Packets) to the graph setting (i.e., HGLET and NGWPs, respectively).
- If an input graph is P_N, then HGLET and NGWPs *exactly* reduce to the HBDCT and the Shannon WPs, respectively.
- Used the *hierarchical bipartitioning* of *the primal graph G for HGLET* and *the dual graph $G^\star$ for NGWPs* where the nodes are the GL eigenvectors and the edge weights represent the affinity between those vectors, e.g., $1/d_{\mathrm{DAG}}(\cdot,\cdot)$
- Used *varimax rotation* to make NGWP vectors well-localized on G.

Summary ...

- Graph signal approximation experiments demonstrate an advantage of a *data-adaptive basis dictionary* from which one can select the most suitable basis for one's task at hand!
- These graph basis dictionaries can also be used for the classical signals and images measured on the regular lattices. And they often outperform the classical ones (with higher computational costs). Note that N does not have to be dyadic unlike the classical cases. Moreover, all the partitions used in the graph basis dictionaries are *data adaptive* and not rigid like the classical basis dictionaries.
- *Separation of Geometry and Statistics of Data* $\Longrightarrow$ PCA cannot do this!!

Future Plan

- How can we 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 graph version of the *smooth orthogonal projectors involving orthogonal folding and unfolding operators* and the graph basis dictionaries *LP-HGLET* and *LP-NGWPs*. To be more effective, however, we need GL eigenvectors associated with different boundary conditions than the usual Neumann conditions.
- Should explore different cost functionals than the sparsity $\Longrightarrow$ *Local Discriminant Basis (LDB)*, *Local Regression Basis (LRB)*, etc. [Saito et al. (1995; 1997; 2002; ...)]
- How can we generalize our multiscale graph basis dictionaries to *directed graphs, higher-order networks, hypergraphs*? $\Longrightarrow$ *Hodge Laplacians!!*

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," arXiv:2107.05121 [eess.SP], 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!