Design natural Graph Wavelet by measuring "Distance" between Graph Laplacian eigenvectors

Haotian Li

Department of Mathematics
University of California, Davis

PhD advisor: Professor Naoki Saito

Jan 24, 2019

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Motivations

- Key of building 1D wavelets by the Littlewood-Paley theory is based on *the ordering of Fourier modes*.
- Using *graph Laplacian eigenvectors* as "cosines" or Fourier modes on graphs with eigenvalues as (the square of) their "frequencies" has been popular.
- **Spectral Graph Wavelet Transform (SGWT)** of Hammond et al. 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, since the notion of *frequency* is not well defined on general graphs, this view is wrong other than for very simple graphs, e.g., undirected unweighted paths and cycles.
- Therefore, we are looking for a *natural ordering* of graph Laplacian eigenvectors.
- Our goal is to construct graph wavelets based on this ordering.

Facts about Graph Laplacian $L(G)$

- Connected undirected graph $G = (V, E, W)$, with $|V| = n$.
- Graph Laplacian is given by $L(G) = D(G) - W(G)$, in which $W(G)$ is the weights matrix of G and $D(G)_{ii} = \sum_j W(G)_{ij}$.
- $L(G)$ is a real symmetric positive semi-definite matrix, so the eigenvalues of L (i.e., $L(G)$) are *nonnegative* and the eigenvectors $\{\boldsymbol{\phi}_l\}_{l=0}^{n-1}$ form *an orthonormal basis*.

$$L\boldsymbol{\phi}_l = \lambda_l \boldsymbol{\phi}_l, \qquad 0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{n-1}$$

- $\lambda_0 = 0$ is always an eigenvalue of L and its corresponding eigenvector $\boldsymbol{\phi}_0$ is a constant vector called *the DC component* (vector).
- The eigenvector $\boldsymbol{\phi}_1$ (with the first nonzero eigenvalue) is called *the Fielder vector* which plays an important role in graph partitioning.
- Also, $\{\boldsymbol{\phi}_l\}_{l=0}^{n-1}$ and $\{\lambda_l\}_{l=0}^{n-1}$ commonly viewed as the Fourier modes on graphs and the corresponding "frequencies".

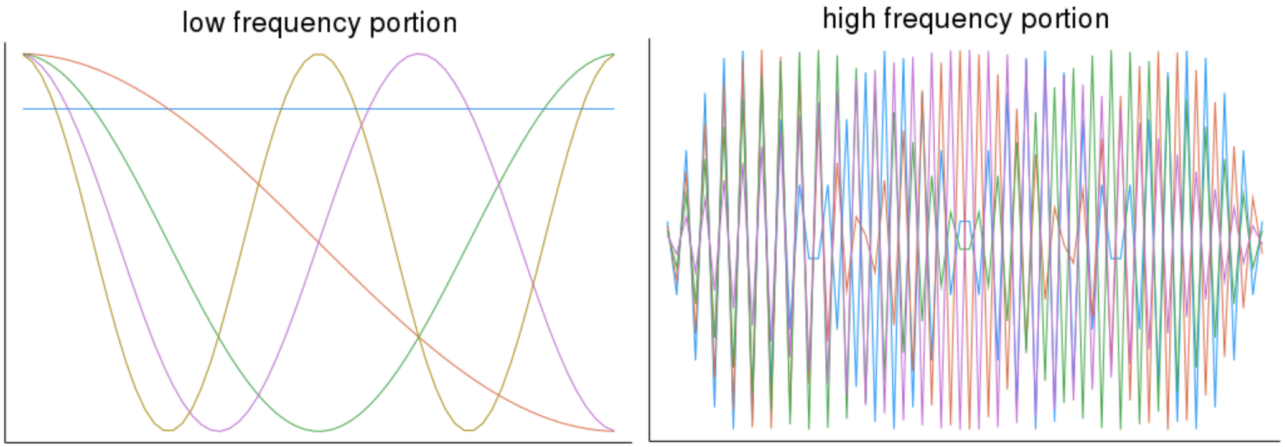
Why view $\boldsymbol{\phi}_l$ as Fourier modes and λ_l as "frequencies"?

$$L(G) = \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & & & & & & \\ & 2 & & & & & \\ & & 2 & & & & \\ & & & \ddots & & & \\ & & & & 2 & & \\ & & & & & 1 \end{bmatrix}}_{D(G)} - \underbrace{\begin{bmatrix} 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 0 & 1 & \\ & & & & 1 & 0 \end{bmatrix}}_{W(G)}$$

- The eigenvalues of $L(G)$ are $\lambda_k = 2 - 2\cos(\pi k/n) = 4\sin^2(\pi k/2n)$, $k = 0 : n-1$.
- The corresponding eigenvectors are $\boldsymbol{\phi}_k(\ell) = a_{k;n}\cos(\pi k(\ell + \frac{1}{2})/n)$, $k, \ell = 0 : n-1$; $a_{k;n}$ is a const. s.t. $\|\boldsymbol{\phi}_k\|_2 = 1$.
- In this 1D path graph, λ (eigenvalue) is a *monotonic* function w.r.t. the frequency, which is the eigenvalue index k and $\{\boldsymbol{\phi}_l\}_{l=0}^{n-1}$ is *the discrete cosine transform type II basis*.
- However, the physical interpretation of *the "frequency" is not clear for a general graph*, since eigenvectors will not always be cosines.

Problem of SGWT

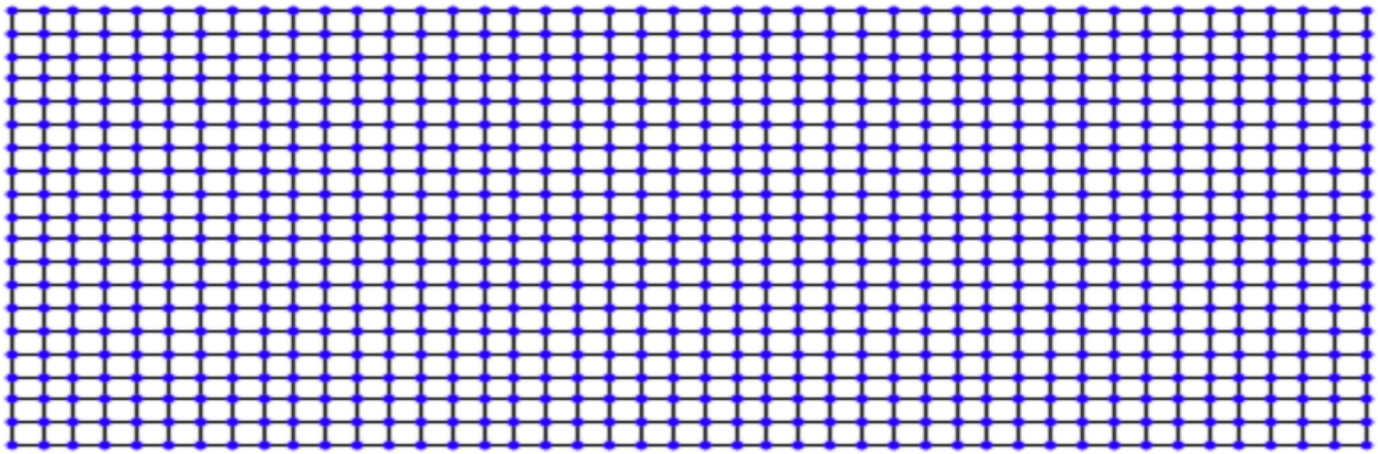
- Construct wavelets in 1D by the Littlewood-Paley theory is to cluster *the Fourier modes* into dyadic blocks (low frequency portion and high frequency portion) based on their corresponding *frequencies*.



- SGWT generalizes this idea to build wavelets on graphs by clustering the *Laplacian eigenvectors* $\{\boldsymbol{\phi}_l\}_{l=0}^{n-1}$ into dyadic blocks based on their corresponding *eigenvalues* $\{\lambda_l\}_{l=0}^{n-1}$.
- For complicated graphs, however, *viewing the eigenvalues as (the square of) frequencies* and constructing wavelet relies on the Littlewood-Paley theory *may lead to unexpected problems.*

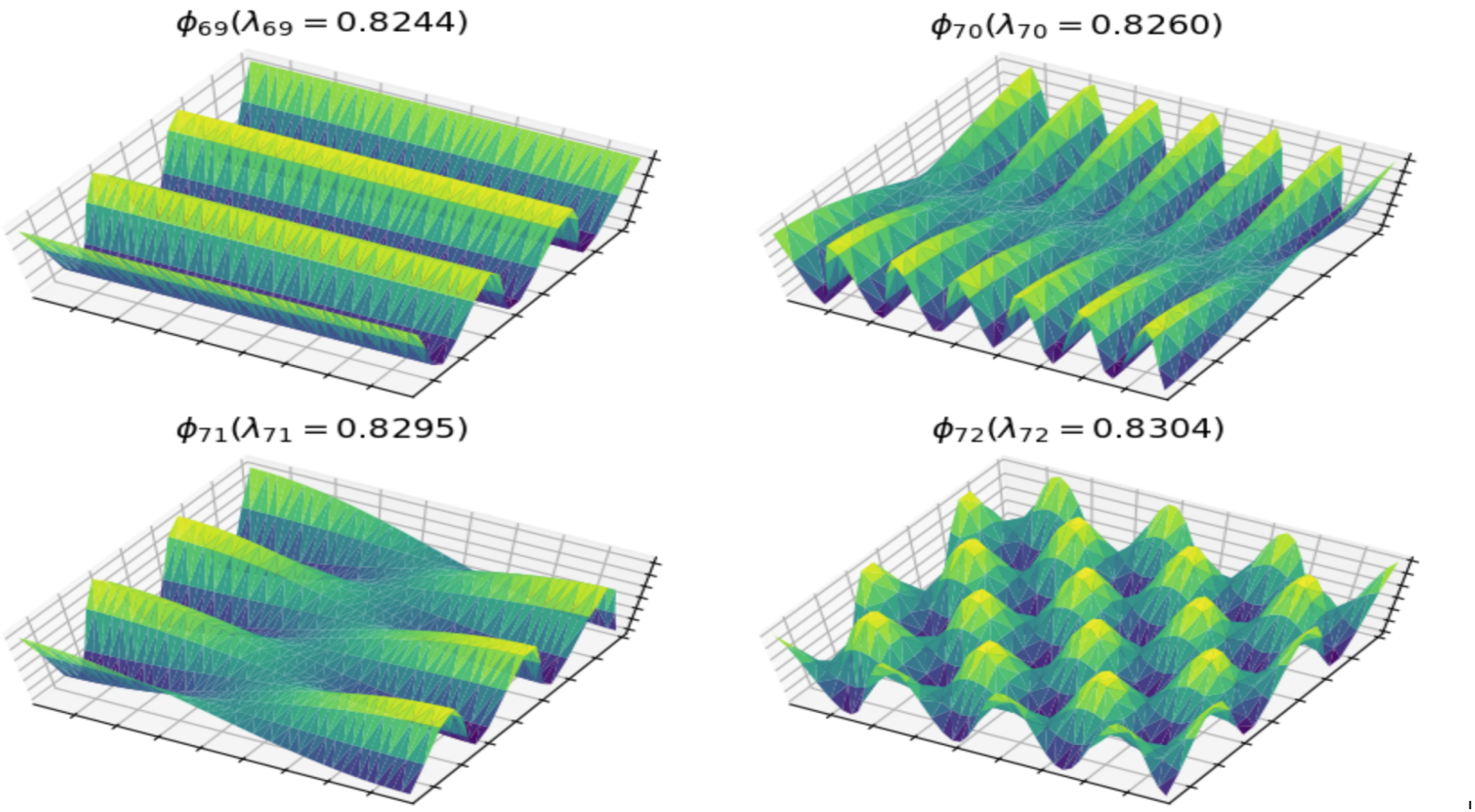
A Counter Example: the 2D Lattice Graph

- Let $G = (V, E, W)$ be a 44×20 *lattice grid graph*, in which $|V| = 880$, $w(e) = 1$ for all $e \in E$.



- We claim that, by simply looking at the Laplacian eigenvalue sequence $\{\lambda_k\}_{k=0,1,\ldots}$, it is *nearly impossible to organize the eigenvectors into physically meaningful dyadic blocks*.

Four Consecutive Eigenvectors of the Lattice Graph



If we order the graph Laplacian eigenvectors by their corresponding eigenvalues in ascending order, we can see the above *four consecutive eigenvectors have very different oscillation structures.*

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Plan

- Given a connected undirected graph $G = (V, E, W)$, with $|V| = n$.
- A natural way to *order and organize the graph Laplacian eigenvectors is based on their 'behaviors' on graphs*, i.e. energy distribution and oscillation patterns.
- Also, order them in n-dim space rather than ordering them in 1D (by eigenvalues).
- Define a proper "distance" between the eigenvectors such that similar behavior ones are close and distinct behavior ones are far apart.
- The usual ℓ^2-distance doesn't work since $\left\| \boldsymbol{\phi}_i - \boldsymbol{\phi}_j \right\|_2 = \sqrt{2}\delta_{ij}$.
- Consider some other distances
  - Compute the dissimilarity between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, for all $i, j = 0 : n-1$, which results in a "distance" matrix $D \in \mathbb{R}^{n \times n}_{\geq 0}$
  - Use *Multidimensional Scaling (MDS)* to embed D into a lower dimensional Euclidean space, say, $\mathbb{R}^m$, $m \ll n$; typically $m = 2$ or $m = 3$

Current distances

- N. Saito (2018) considers using *the ramified optimal transport (ROT) cost* from $\boldsymbol{\phi}_i^2$ to $\boldsymbol{\phi}_j^2$ on graphs to measure the difference.
- A. Cloninger and S. Steinerberger (2018) propose a way to measure the 'similarity' between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$, denoted as $\alpha(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$, based on *the Hadamard product* $\boldsymbol{\phi}_i \odot \boldsymbol{\phi}_j$.
- I come up with two promising methods to measure the distances.
  - *The difference of absolute gradient (DAG) method*
  - *The time-stepping diffusion method (TSDM)*

Outline

1 Motivations

2 Natural ordering: distances between eigenvectors
- The difference of absolute gradient (DAG) method
- The time-stepping diffusion method (TSDM)

3 Design Graph Wavelet

4 Future work

The difference of absolute gradient (DAG) method

- The idea of DAG is that we treat the absolute gradient of each eigenvector as a feature vector, $|d\boldsymbol{\phi}_i| \in \mathbb{R}^{|E|}$.
- Denote the edge $e = (k, l)$, $k, l \in V$, then $|d\boldsymbol{\phi}_i|(e) = |\boldsymbol{\phi}_i(k) - \boldsymbol{\phi}_i(l)|$.
- Then we use *the ℓ^2-norm of the feature vectors' difference* to quantify the distance between eigenvectors.

$$\begin{aligned} d(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)^2 &:= \langle |d\boldsymbol{\phi}_i| - |d\boldsymbol{\phi}_j|, |d\boldsymbol{\phi}_i| - |d\boldsymbol{\phi}_j| \rangle_E \\ &= \langle d\boldsymbol{\phi}_i, d\boldsymbol{\phi}_i \rangle_E + \langle d\boldsymbol{\phi}_j, d\boldsymbol{\phi}_j \rangle_E - 2\langle |d\boldsymbol{\phi}_i|, |d\boldsymbol{\phi}_j| \rangle_E \\ &= \lambda_i + \lambda_j - \sum_{x \in V} \sum_{y \sim x} |\boldsymbol{\phi}_i(x) - \boldsymbol{\phi}_i(y)| \cdot |\boldsymbol{\phi}_j(x) - \boldsymbol{\phi}_j(y)| \end{aligned}$$

  in which $\langle \cdot, \cdot \rangle_E$ is the inner product over edges.
- *Given the eigenvectors, the computational cost is $O(|E|)$.*

MDS Result of 11×5 Lattice Graph: Embedding into $\mathbb{R}^2$

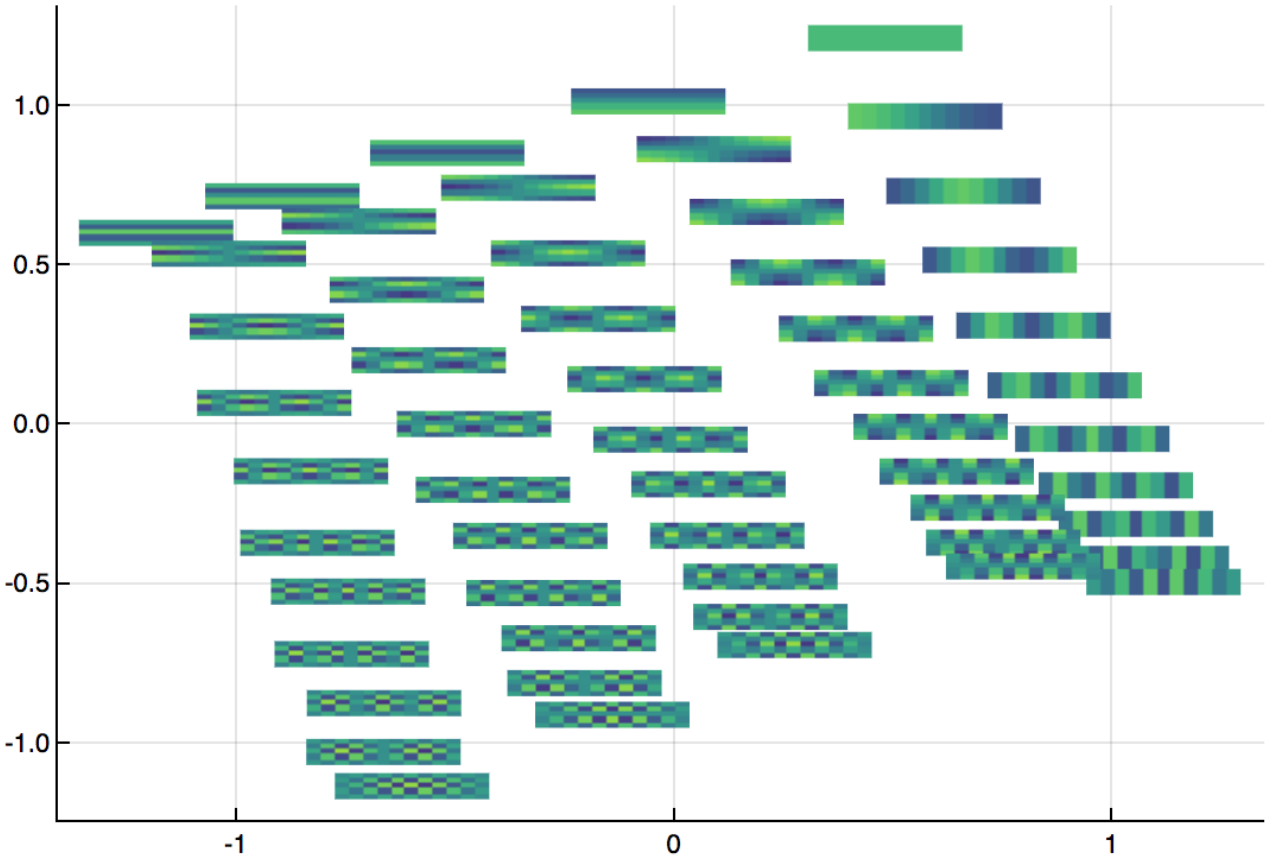
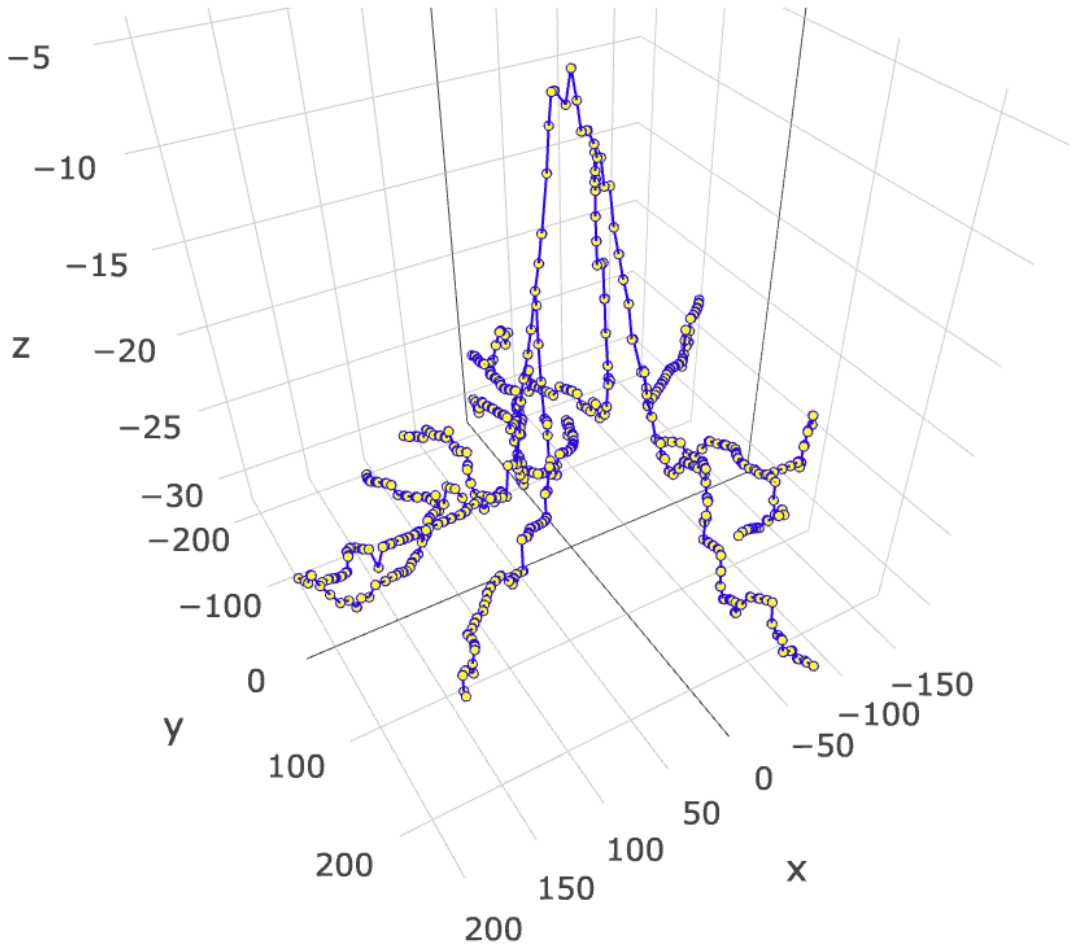


Figure: MDS into $\mathbb{R}^2$: each small heatmap plot in the embedded space describes how the eigenvectors looks like on the lattice graph.

Natural Ordering of the Eigenvectors of Dentritic Tree

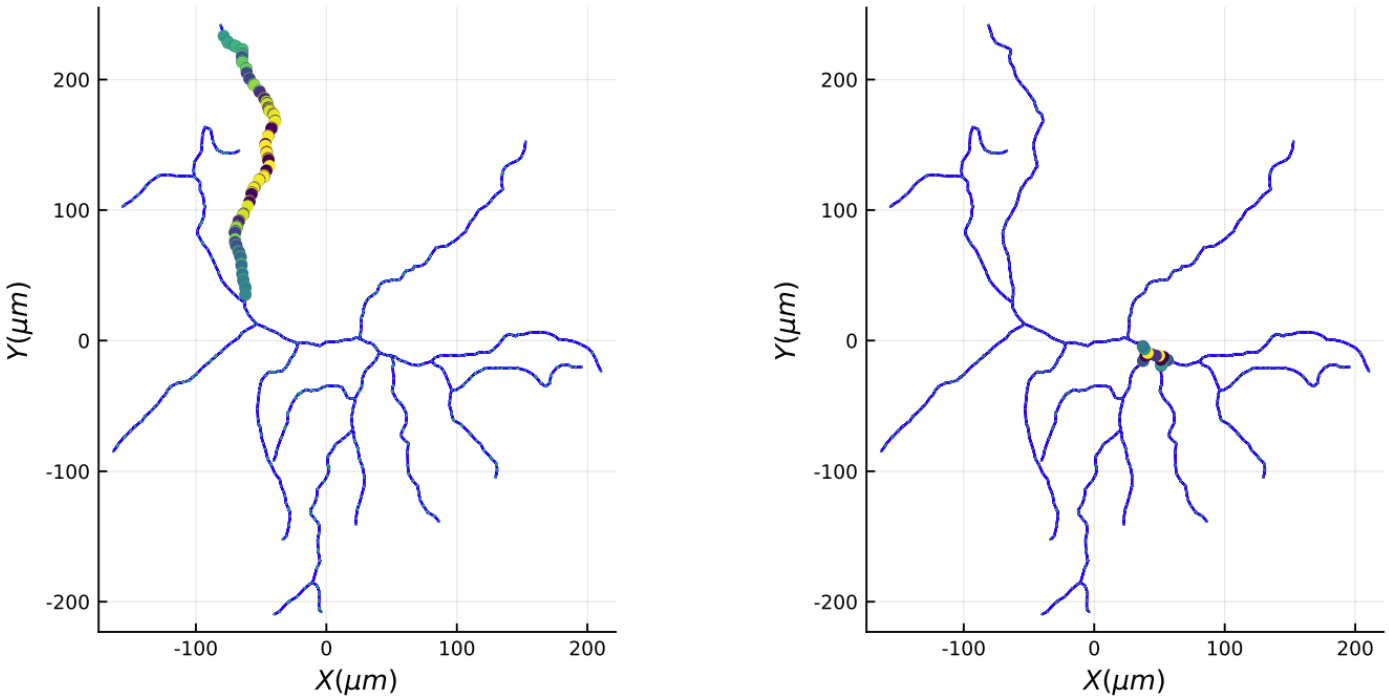
3D RGC #100 (unweighted) graph, $|V| = 1154$:



A Phase Transition Phenomenon of the Eigenvectors

We have observed that the value 4 is critial since:

- The eigenvectors corresponding to the eigenvalues below 4 (left) are semi-oscillations (*like Fourier cosines/sines*) over the dendrites.
- Those corresponding to the eigenvalues above 4 (right) are much more localized (like *wavelets*) around *junctions/bifurcation vertices*.



MDS result in $\mathbb{R}^3$: ROT vs. DAG

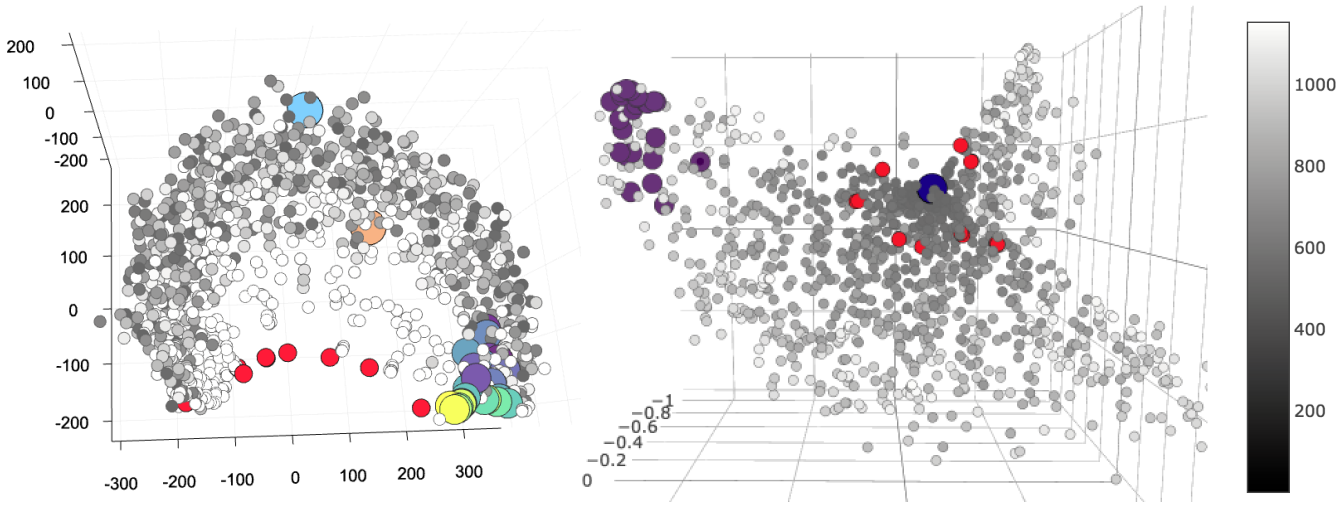


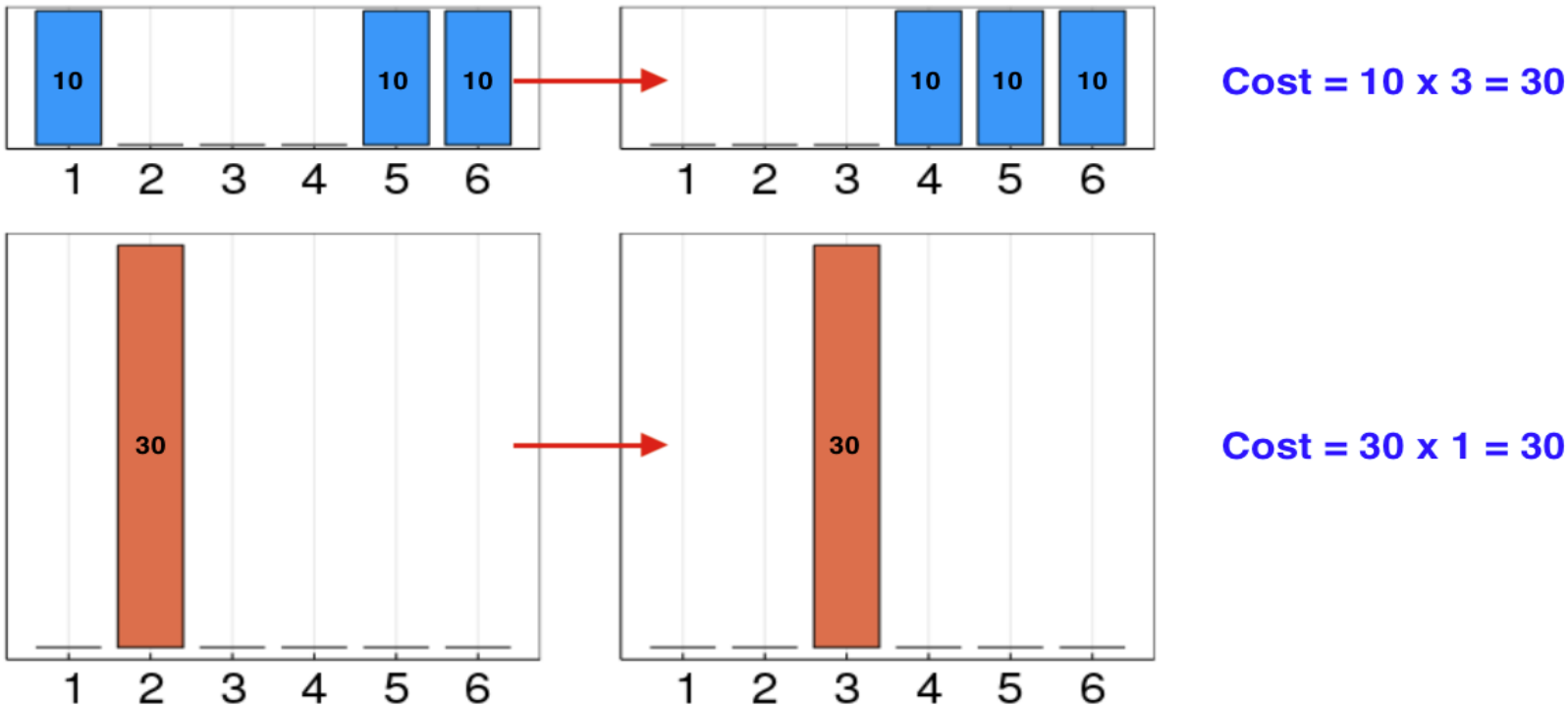
Figure: *ROT (left)*: The light blue point = the DC vector; the light orange = the Fiedler vector; Viridis points = the eigenvectors which concentrate on the upper left branch; red = localized eigenvectors. *DAG (right)*: Two dark blue circles = the DC component and Fiedler vector; the red = localized eigenvectors; the purple = the eigenvectors that concentrated on the upper left branch. Grey scales represent the index of eigenvectors.

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Optimal Transport Cost a.k.a. Earth Mover Distance (EMD)

- EMD is *the minimum cost* of turning one pile of dirt into the other, where the cost is *the amount of dirt moved times the distance by which it is moved* (i.e., cost = mass × distance).



- EMD cannot distinguish these two transport schemes.

The time-stepping diffusion method (TSDM)

- The purpose of TSDM is to design an optimal transport-like method that *depends on time*. In other words, at each given time, we have a cost scheme (or distance method).
- In order to measure *the optimal transport cost* between two vector measures (with the same total mass) on graphs, we need to first take the difference between two vector measures as the initial input, then compute *the minimum effort to flat things out* on the graph.

Motivation of TSDM

- Given a time T, let us consider a diffusion process on the graph. We want to measure *the cost of "flatten" the initial graph signal via diffusion process up to the time T*.
- We expect the graph signal will be flatten out by this process and *the final cost ,as $T \to \infty$, behave similar with the optimal transport cost.*
- Notations: Denote the graph Laplacian matrix as L whose factorization is $L = \Phi \Lambda \Phi^T$, in which $\Phi = [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{n-1}]$ and $\Lambda = \text{diag}([\lambda_0, \lambda_1, \cdots, \lambda_{n-1}])$, $0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{n-1}$. Also, denote the directed incidence matrix of the graph G as $\tilde{Q} \in \mathbb{R}^{|V| \times |E|}$, which served as the graph gradient, i.e., $\tilde{Q}^T = \nabla_G$.

TSDM:

(Heat diffusion) Given initial $\boldsymbol{f}_0$, the governing ODE system which describes the graph signal $\boldsymbol{u}(t)$'s ($\in \mathbb{R}^n$) evolution is following:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) + L \cdot \boldsymbol{u}(t) = \mathbf{0} \qquad \boldsymbol{u}(0) = \boldsymbol{f}_0 \in \mathbb{R}^n \tag{1}$$

Since $\{\boldsymbol{\phi}_0, \cdots, \boldsymbol{\phi}_{n-1}\}$ forms an ONB of $\mathbb{R}^n$, we have $\boldsymbol{u}(t) = \sum_{k=0}^{n-1} C_k(t) \cdot \boldsymbol{\phi}_k$. Then, after plugging it into the above ODE system and solving for $C_k(t)$, we get $C_k(t) = \langle \boldsymbol{f}_0, \boldsymbol{\phi}_k \rangle e^{-\lambda_k t}$. Now, we have the solution:

$$\boldsymbol{u}(t) = \sum_{k=0}^{n-1} \langle \boldsymbol{f}_0, \boldsymbol{\phi}_k \rangle e^{-\lambda_k t} \boldsymbol{\phi}_k \tag{2}$$

At a certain time T, let us *define the cost of the time-stepping diffusion method*, $K(\boldsymbol{f}_0; T)$, as follows:

$$K(\boldsymbol{f}_0; T) := \int_0^T \|\nabla_G \boldsymbol{u}(t)\|_1 \mathrm{d}t \qquad \nabla_G \text{ is the graph gradient.} \tag{3}$$

Convergence of TSDM

Theorem

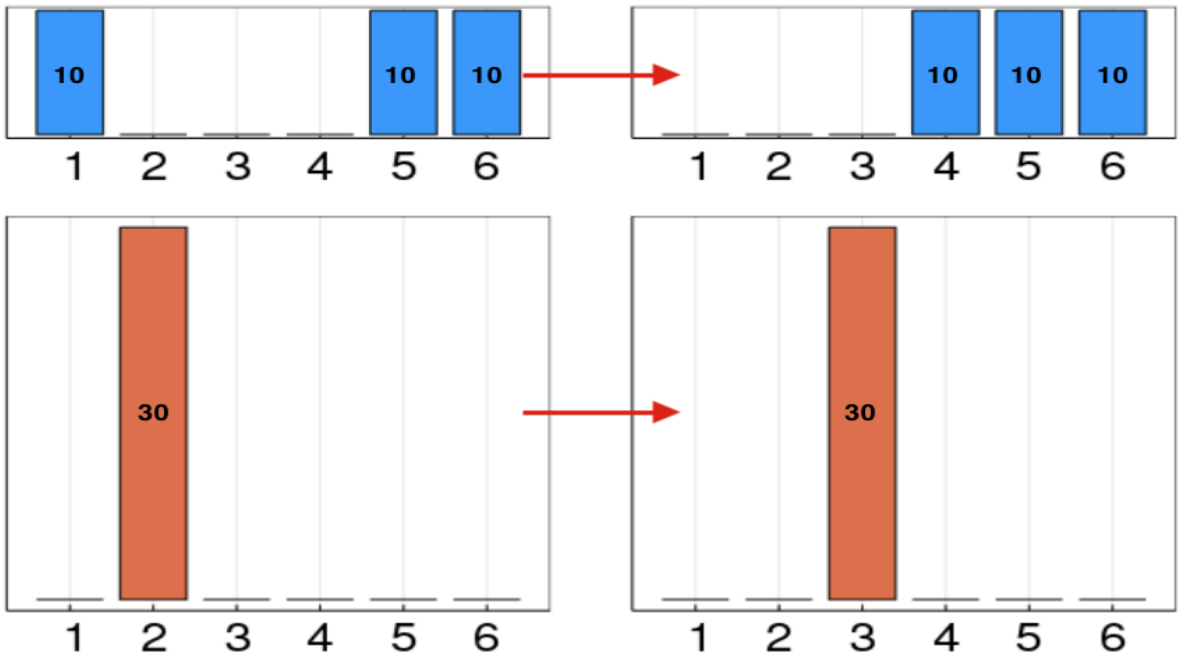
(Convergence of TSDM) *Let $G = (V, E, W)$ be a connected undirected graph and $\boldsymbol{f}_0$ as the initial graph signal. $K(\boldsymbol{f}_0; T)$ converges as $T \to \infty$.*

$$\lim_{T\to\infty} K(\boldsymbol{f}_0; T) = \lim_{T\to\infty} \int_0^T \|\nabla_G \boldsymbol{u}(t)\|_1 \mathrm{d}t < \infty$$

- Furthermore, we can show that for any fixed $T > 0$ (including $T = \infty$), $K(\,\cdot\,; T)$ is a norm on $M = \{f \in L^2(V) : \sum_{x\in V} f(x) = 0\}$.

Result Comparison:

- Optimal transport cost:



- TSDM cost:

time	$T = 0.1$	$T = 1$	$T = 10$	$T = \infty$
blue cost	2.79	16.66	38.30	40.32
orange cost	10.41	38.42	63.65	65.87

The Cost Conjecture

- As time $T \to \infty$, one might expect the *TSDM cost* to *be close to* the optimal transport cost (i.e., *the* 1^{st} *Wasserstein distance*) between any two vector measures with the same total mass defined on the graph.

The Cost Conjecture

Given any two probability distributions p, q *on a connected graph* $G = (V, E, W)$ *with graph geodesic distance metric* $d : V \times V \to \mathbb{R}_{\geq 0}$,

$$W_1(p, q) \leq K(p - q; \infty) \leq C \cdot W_1(p, q)$$

in which $W_1(p, q) := \inf_{\gamma \in \Gamma(p,q)} \int_{V \times V} d(x, y) \mathrm{d}\gamma(x, y)$, *where* $\Gamma(p, q)$ *denotes the collection of all measures on* $V \times V$ *with marginals* p *and* q *in the first and second factors respectively and* C *is a constant depends on* G.

- There is also a manifold version of this conjecture and if the underlying manifold is $[0, 1]$ or $\mathbb{T}$, we can show $W_1(p, q) \leq K(p - q; \infty)$.

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Graph Wavelet Frame

- Given a graph $G = \{V, E, W\}$ with $|V| = n$ and the distance matrix D of its eigenvectors, we can get a partition of all the eigenvectors based on some clustering method, $\mathscr{P} = \{\mathscr{C}_1, \mathscr{C}_2, \cdots, \mathscr{C}_N\}$, $1 \le N \le n$, in which

$$\cup_{j=1}^{N} \mathscr{C}_j = \{1, 2, \cdots, n\} \quad \text{and} \quad \mathscr{C}_i \cap \mathscr{C}_j = \emptyset, i \neq j.$$

- In the following notation, the subindex j stands for the cluster and the subindex k represents the localization.

$$\psi_{k,j} = \Phi \overbrace{F_j \Phi^T e_k}^{\text{Filtering}} \quad \text{for } j = 1, 2, \cdots, N \text{ and } k = 1, 2, \cdots, n$$

  in which, the diagonal matrix $F_j \in \mathbb{R}^{n \times n}$ satisfies $F_j(l, l) = \chi_{\mathscr{C}_j}(l)$ for $l = 1, 2, \cdots, n$, Φ stores all the graph Laplacian eigenvectors, and e_k is the spike vector at vertex v_k.
- We can show that $\{\psi_{k,j}\}_{k=1,\cdots,n;j=1,\cdots,N}$ *is a N times redundant wavelet frame*.

Graph Wavelet Basis

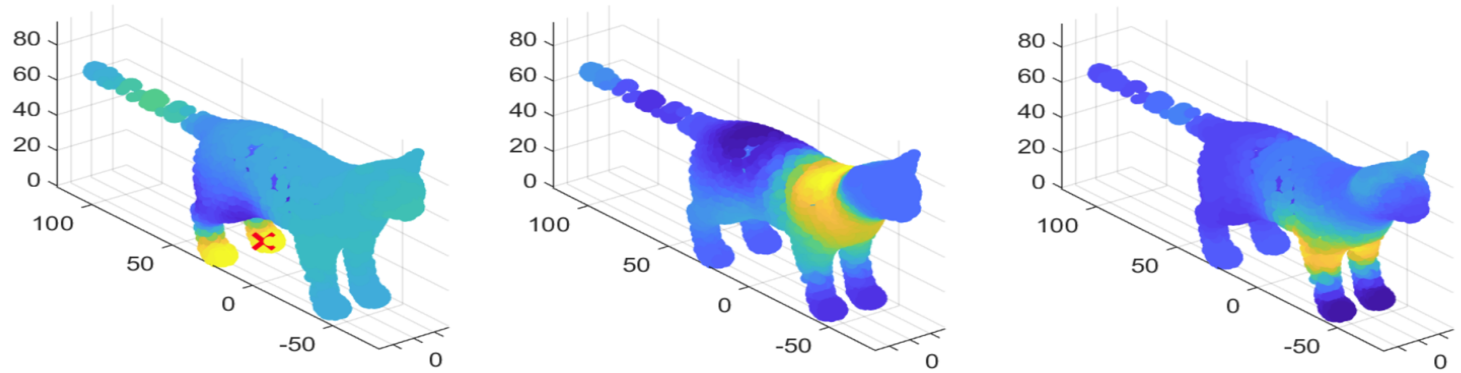
- One way is *to pick* n vectors out of nN vectors in the frame $\{\psi_{k,j}\}$ by using $\{e_k\}_{k\in T_j}$, in which $T_j \subset \{1,2,\cdots,n\}$, instead of all $\{e_k\}_{k=1,\cdots,n}$ for each $\mathscr{C}_j$, so that $\sum_{j=1}^{N} |T_j| = n$. These n vectors may not be mutually orthogonal, so we may need some orthogonalization procedure.
- Another way is by *sparsifying rotation*. First, we permute Φ into $\hat{\Phi}$ based on $\mathscr{P}$.

$$\hat{\Phi} = \left[\phi_{\mathscr{C}_1}, \phi_{\mathscr{C}_2}, \cdots, \phi_{\mathscr{C}_N}\right]$$

in which $\mathscr{C}_j = \{j_1, j_2, \cdots, j_l\}$ and $\phi_{\mathscr{C}_j} = [\phi_{j_1}, \phi_{j_2}, \cdots, \phi_{j_l}]$. Then, we rotate $\hat{\Phi}$ within each cluster $\mathscr{C}_j$ for $j = 1,2,\cdots,N$ to get an sparse orthonormal wavelet basis $\Psi \in \mathbb{R}^{n\times n}$.

Cat Example by Dr. Cloninger

- Ongoing collaboration work on natural graph wavelets and its applications with N. Saito and A. Cloninger (UCSD).
- Consider the graph G to be a *triangulated surface of a cat with* 3000 *3D-points*. Use Cloninger and Steinerberger's method to obtain the *affinity matrix*. Apply *greedy clustering* by iteratively $\text{argmax}_{j\neq 1}\alpha(\phi_1,\phi_j)$ to get a cluster of the eigenvectors $\mathscr{C}_1$.



Graph wavelets on $\mathscr{C}_1$. Red cross indicates the location of e_k.

Outline

1. Motivations
2. Natural ordering: distances between eigenvectors
   - The difference of absolute gradient (DAG) method
   - The time-stepping diffusion method (TSDM)
3. Design Graph Wavelet
4. Future work

Future work

- Give *a better mathematical explanation* why the difference of absolute gradient method works well on the Cartesian product graphs.
- Since the TSDM is a time-involving optimal transport method, it contains more information than the ROT method about the "behavior" of graph Laplacian eigenvectors and the geometry information of the underlying graph. Therefore, we may have *better ways to reveal the geometry of the eigenvectors* after considering the time factor.
- Try to *find an automatic and reasonable way to do the clustering* of the eigenvectors based on the distance matrix D. Also, based on the results, building different kinds of wavelets on graph (e.g., Shannon wavelet and Meyer wavelet) or wavelet packet trees on graph. Test them on different graphs and compare the results with the other graph wavelets, e.g., SGWT [10].

Future work

- Working on the cost conjecture, see if there is something to *bridge the optimal transport and the heat diffusion equation* when the underlying graph is more general than a 1D analytic path.
- What we have been doing is to design a wavelet basis on graphs, so in order to test this basis we need to apply them to real data or graphs, e.g., triangular FEM-like meshes of objects or Facebook data, etc. Therefore, *realizations* on computer will also be very important for the future work.

References

- N. Saito. *"How can we naturally order and organize graph Laplacian eigenvectors?"* In: *Proc. 2018 IEEE Workshop on Statistical Signal Processing*. (**2018**, pp. 483-487).
- Cloninger, Alexander, and Stefan Steinerberger. *"On the Dual Geometry of Laplacian Eigenfunctions."* arXiv preprint arXiv:1804.09816 (**2018**).
- Q. Xia, *"Motivations, ideas and applications of ramified optimal transportation,"* ESAIM: Mathematical Modelling and Numerical Analysis, vol. 49, no. 6, pp. 1791-1832, **2015**, Special Issue Optimal Transport.
- Shuman, David I., Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. *"The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains."* IEEE Signal Processing Magazine 30, no. 3 (**2013**): 83-98.
- Y. Nakatsukasa, N. Saito, and E. Woei, *"Mysteries around the graph Laplacian eigenvalue 4,"* Linear Algebra Appl., vol. 438, no. 8, pp. 3231-3246, **2013**.
- Hammond, David K., Pierre Vandergheynst, and Remi Gribonval. *"Wavelets on graphs via spectral graph theory."* Applied and Computational Harmonic Analysis 30, no. 2 (**2011**): 129-150.
- I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, 2nd edition, **2005**.

Thanks for your attention!
Any questions?