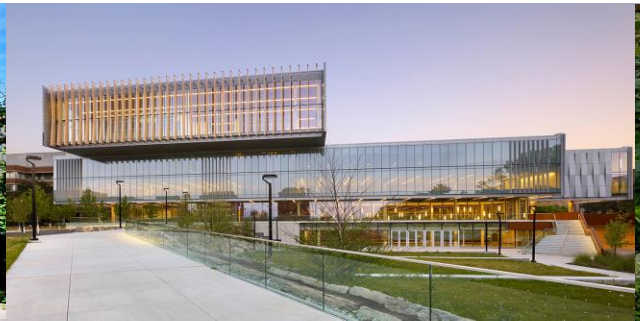
Graph Learning, Sampling & Filtering for Image & Signal Estimation

Gene Cheung
York University
Toronto, Canada

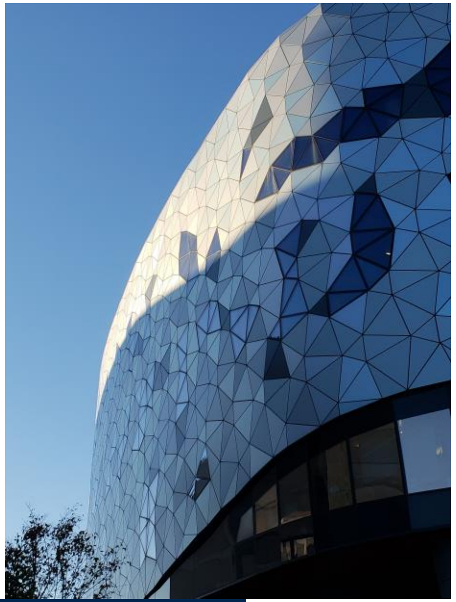
March 29, 2021

Acknowledgement

- **Graph and Image Signal Processing (GISP) Lab** (York University, Toronto, Canada)
  - Post-docs: Cheng Yang*, Xue Zhang
  - Grad students: Saghar Bagheri, Fengbo Lan, Huy Vu
  - Visiting researchers: Weng-tai Su (NTHU), Chinthaka Dinesh (SFU), Fen Wang (Xidian)

- **Collaborators**
  - Richard Wildes, Michael Brown (York Univ., Canada)
  - Ivan V. Bajic (Simon Fraser Univ., Canada)
  - Antonio Ortega (Univ. of Southern California, USA)
  - Stanley Chan (Purdue Univ., USA)
  - Wai-Tian Tan (Cisco, USA)
  - Jiahao Pang, Dong Tian (InterDigital, USA)
  - Yuji Nakatsukasa (Oxford Univ., UK)
  - Vladimir Stankovic (Univ. of Strathclyde, UK)
  - Wei Hu, Wen Gao (Peking Univ., China)
  - Chia-Wen Lin (NTHU, Taiwan)
  - Yonina C. Eldar (Weizmann Inst. of Science, Israel)



* September 2018 to June 2020.

Outline

- **What is Graph Signal Processing?**
  - Graph spectrum
  - Graph Fourier transform (GFT), graph Laplacian regularizer (GLR)
- **Graph Learning**
  - Precision / Graph Laplacian Matrix Estimation (w/ eigen-structure constraint)
  - Feature Graph Learning: Gershgorin Disc Perfect Alignment (GDPA)
  - **Application**: Semi-supervised classifier learning

- **Graph Sampling**
  - Gershgorin Disc Alignment Sampling (GDAS)
  - **Application**: Sampling for matrix completion, 3D point cloud sub-sampling

- **Graph Filtering**
  - Signal-dependent GLR, GTV
  - **Application**: Image denoising

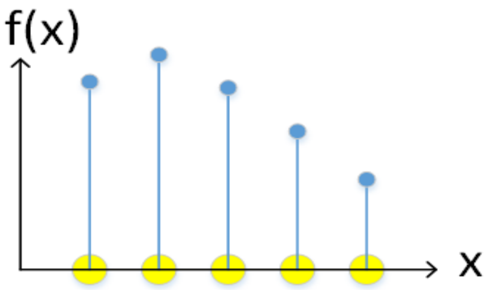
Gene Cheung (genec@yorku.ca)

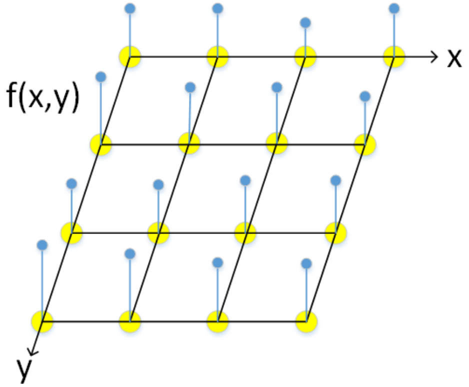
Outline

- **What is Graph Signal Processing?**
  - Graph spectrum
  - Graph Fourier transform (GFT), graph Laplacian regularizer (GLR)
- **Graph Learning**
  - Precision / Graph Laplacian Matrix Estimation (w/ eigen-structure constraint)
  - Feature Graph Learning: Gershgorin Disc Perfect Alignment (GDPA)
  - **Application**: Semi-supervised classifier learning
- **Graph Sampling**
  - Gershgorin Disc Alignment Sampling (GDAS)
  - **Application**: Sampling for matrix completion, 3D point cloud sub-sampling
- **Graph Filtering**
  - Signal-dependent GLR, GTV
  - **Application**: Image denoising

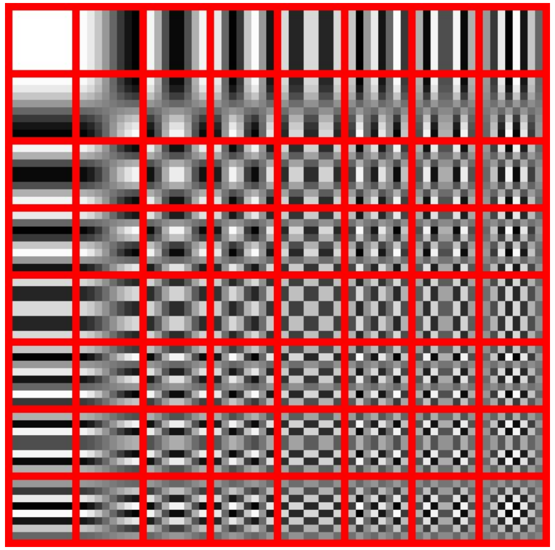
Digital Signal Processing

- Discrete signals on ***regular*** data kernels.
  - Ex.1: audio on regularly sampled timeline.
  - Ex.2: image on 2D grid.





- **Harmonic analysis** tools (transforms, wavelets):
  - Compression, restoration, segmentation, etc.



2D DCT basis

$$a = \Phi\, x$$

sparse
transform coeff.

DCT transform

desired
signal

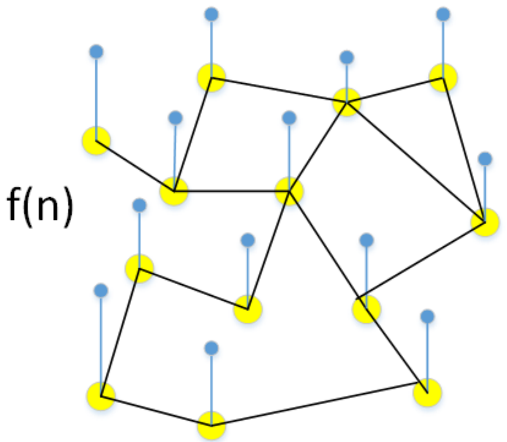
Gene Cheung (genec@yorku.ca)

Graph Signal Processing

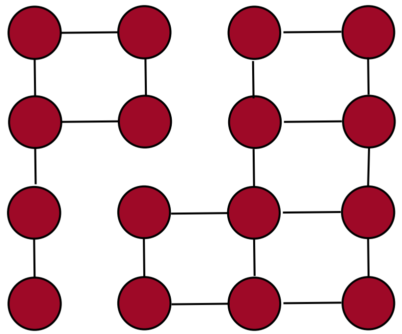
- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.

1. **Harmonic Analysis** of graph signals.

2. Embed pairwise (dis)similarity info into edge weights.
   - **Eigenvectors provide global info aggregated from local info.**



signal on graph kernel



signal on graph kernel

[1] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "**Graph signal processing: Overview, challenges, and applications**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "**Graph spectral image processing**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.

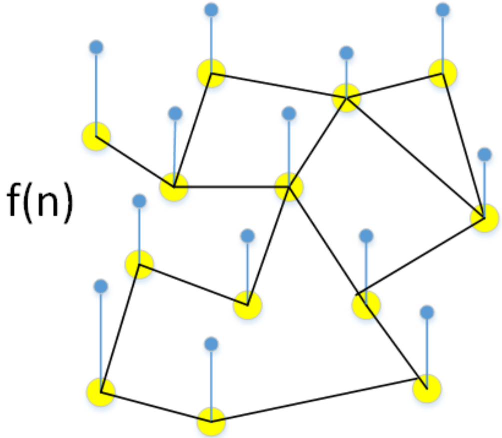
Graph Signal Processing

- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.

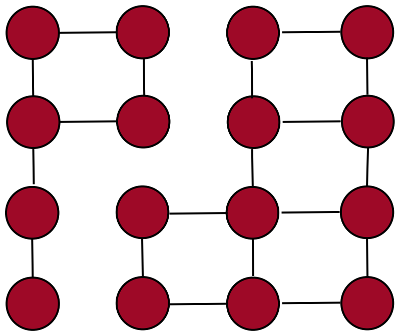
1. **Harmonic Analysis** of graph signals.

2. Embed pairwise (dis)similarity info into edge weights.
   - **Eigenvectors provide global info aggregated from local info.**

> Graph Signal Processing (GSP) provides spectral analysis tools for signals residing on graphs.



signal on graph kernel



signal on graph kernel

[1] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "**Graph signal processing: Overview, challenges, and applications**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "**Graph spectral image processing**,*" Proceedings of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.

Graph Fourier Transform (GFT)

Graph Laplacian:

- *Adjacency Matrix* **W**: entry $W_{i,j}$ has *non-negative* edge weight $w_{i,j}$ connecting nodes *i* and *j*.
- *Degree Matrix* **D**: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row *i* of **W**.

$$D_{i,i} = \sum_{i} W_{i,j}$$

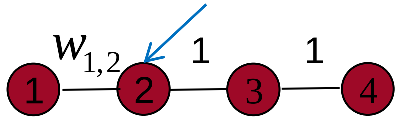
- *Combinatorial Graph Laplacian* **L**: **L = D - W**
  - **L** is related to *2nd derivative*.

$$L_{3,:}\,\mathrm{x} = -x_2 + 2x_3 - x_4$$

$$f''(x) = \lim_{h\to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

  - **L** is a differential operator on graph.

undirected graph



$$\mathbf{W} = \begin{bmatrix} 0 & w_{1,2} & 0 & 0 \\ w_{1,2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} w_{1,2} & 0 & 0 & 0 \\ 0 & w_{1,2}+1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

*https://en.wikipedia.org/wiki/Second_derivative

Graph Fourier Transform (GFT)

Graph Laplacian:

- *Adjacency Matrix* **W**: entry $W_{i,j}$ has *non-negative* edge weight $w_{i,j}$ connecting nodes *i* and *j*.

- *Degree Matrix* **D**: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row *i* of **W**.

$$D_{i,i} = \sum_{i} W_{i,j}$$

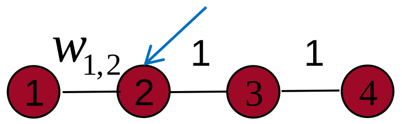
- *Combinatorial Graph Laplacian* **L**: **L = D - W**
  - **L** is related to *2nd derivative*.

$$L_{3,:}\,\mathrm{x} = -x_2 + 2x_3 - x_4$$

$$f''(x) = \lim_{h \to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

  - **L** is a differential operator on graph.

undirected graph



$$\mathbf{W} = \begin{bmatrix} 0 & w_{1,2} & 0 & 0 \\ w_{1,2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} w_{1,2} & 0 & 0 & 0 \\ 0 & w_{1,2}+1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

*https://en.wikipedia.org/wiki/Second_derivative

Graph Spectrum from GFT

Graph Fourier Transform (GFT) is eigen-matrix of graph Laplacian **L**.

eigenvalues along diagonal

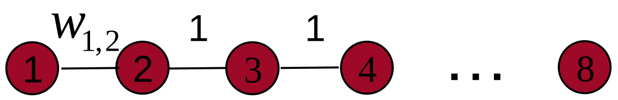
$$L = V \Sigma V^T$$

GFT

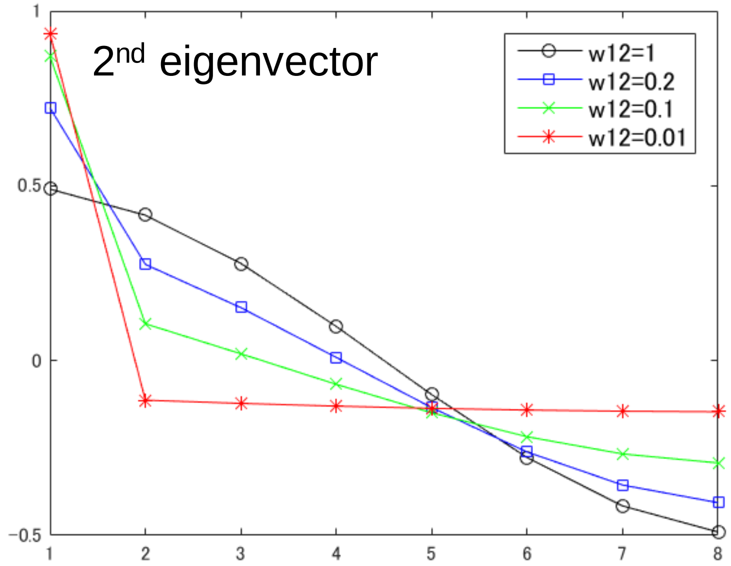
eigenvectors in columns

$$\tilde{x} = V^T x$$

GFT coefficients



1. **Eigenvectors** aggregate info from edge weights.
   - Constant 1st eigenvector is DC.
   - \# *zero-crossings* increases as λ increases.
2. **Eigenvalues** (≥ 0) as *graph frequencies*.



GFT defaults to ***DCT*** for un-weighted connected line.

GFT defaults to ***DFT*** for un-weighted connected circle.

Gene Cheung (genec@yorku.ca)

Graph Spectrum from GFT

Graph Fourier Transform (GFT) is eigen-matrix of graph Laplacian **L**.

eigenvalues along diagonal

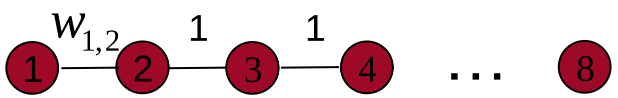
$$\mathrm{L} = \mathrm{V}\,\Sigma\,\mathrm{V}^T$$

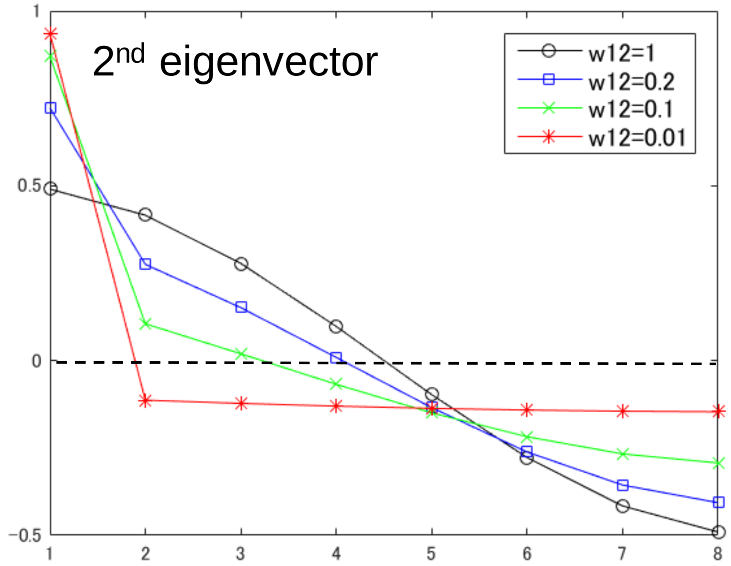
GFT

eigenvectors in columns

$$\tilde{\mathrm{x}} = \mathrm{V}^T \mathrm{x}$$

GFT coefficients





1. **Eigenvectors** aggregate info from edge weights.
   - Constant 1st eigenvector is DC.
   - # *zero-crossings* increases as λ increases.
2. **Eigenvalues** (≥ 0) as *graph frequencies*.

GFT defaults to ***DCT*** for un-weighted connected line.

GFT defaults to ***DFT*** for un-weighted connected circle.

Gene Cheung (genec@yorku.ca)

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

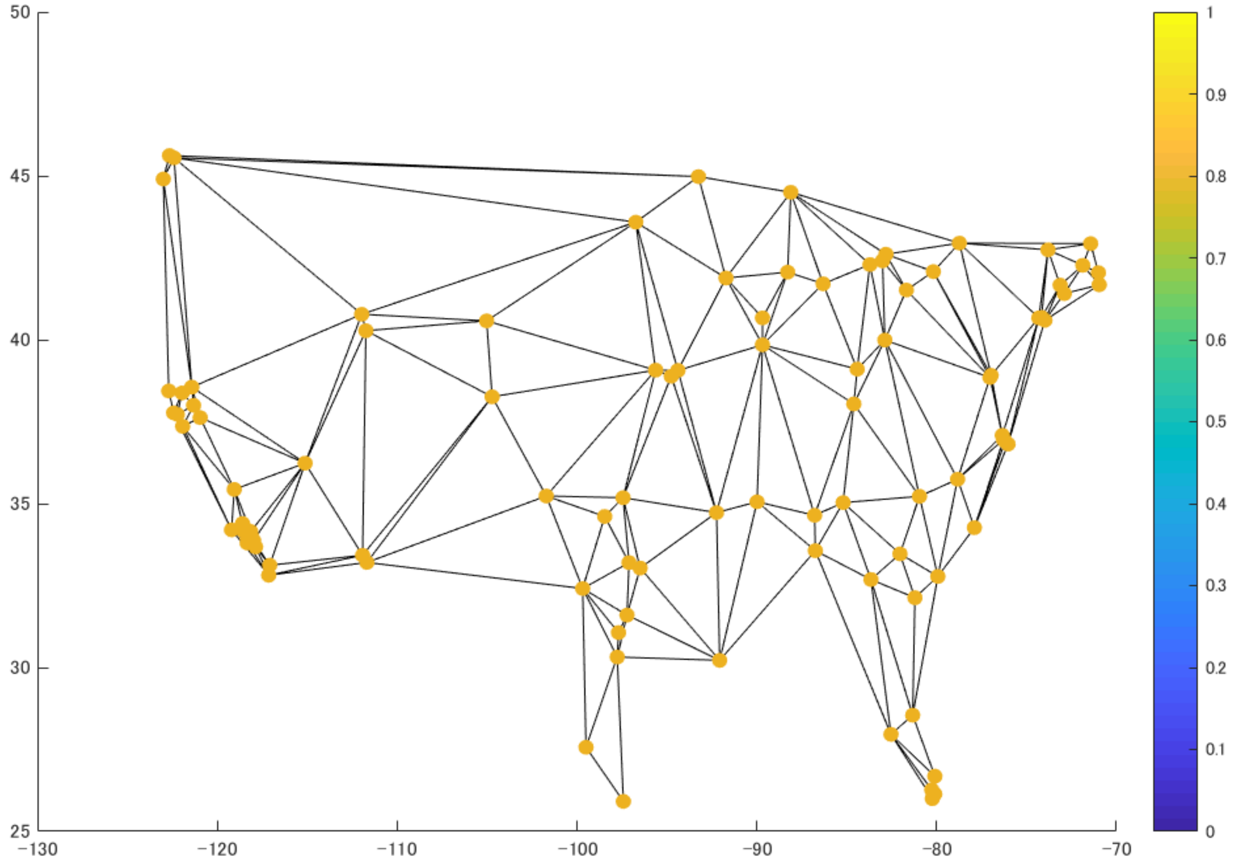
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma^2} \right)$$

Edge weights



V1: DC component

*https://en.wikipedia.org/wiki/Delaunay triangulation

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

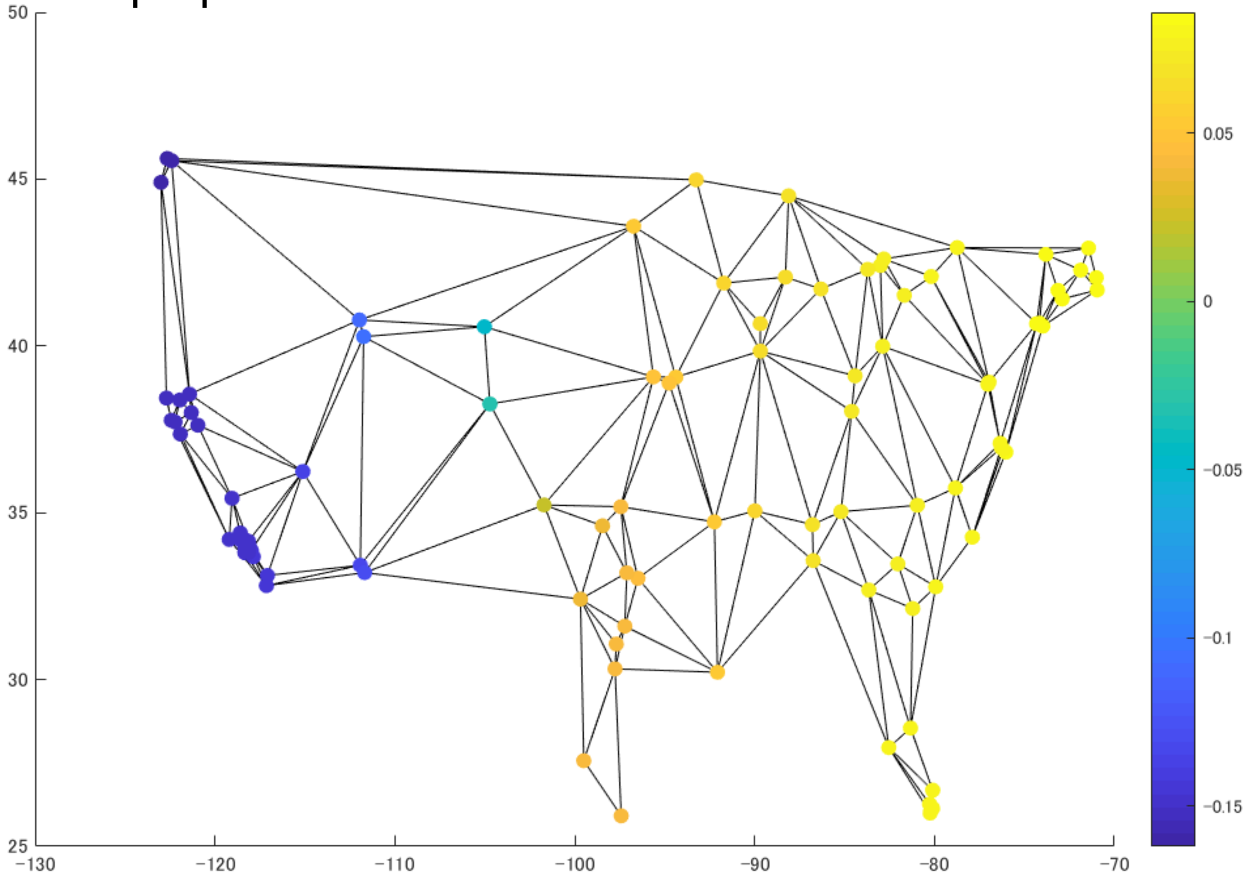
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

$$w_{i,j} = \exp\left(\frac{-\|l_i - l_j\|_2^2}{\sigma^2} \right)$$

Edge weights



V2: 1st AC component

Gene Cheung (genec@yorku.ca)

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

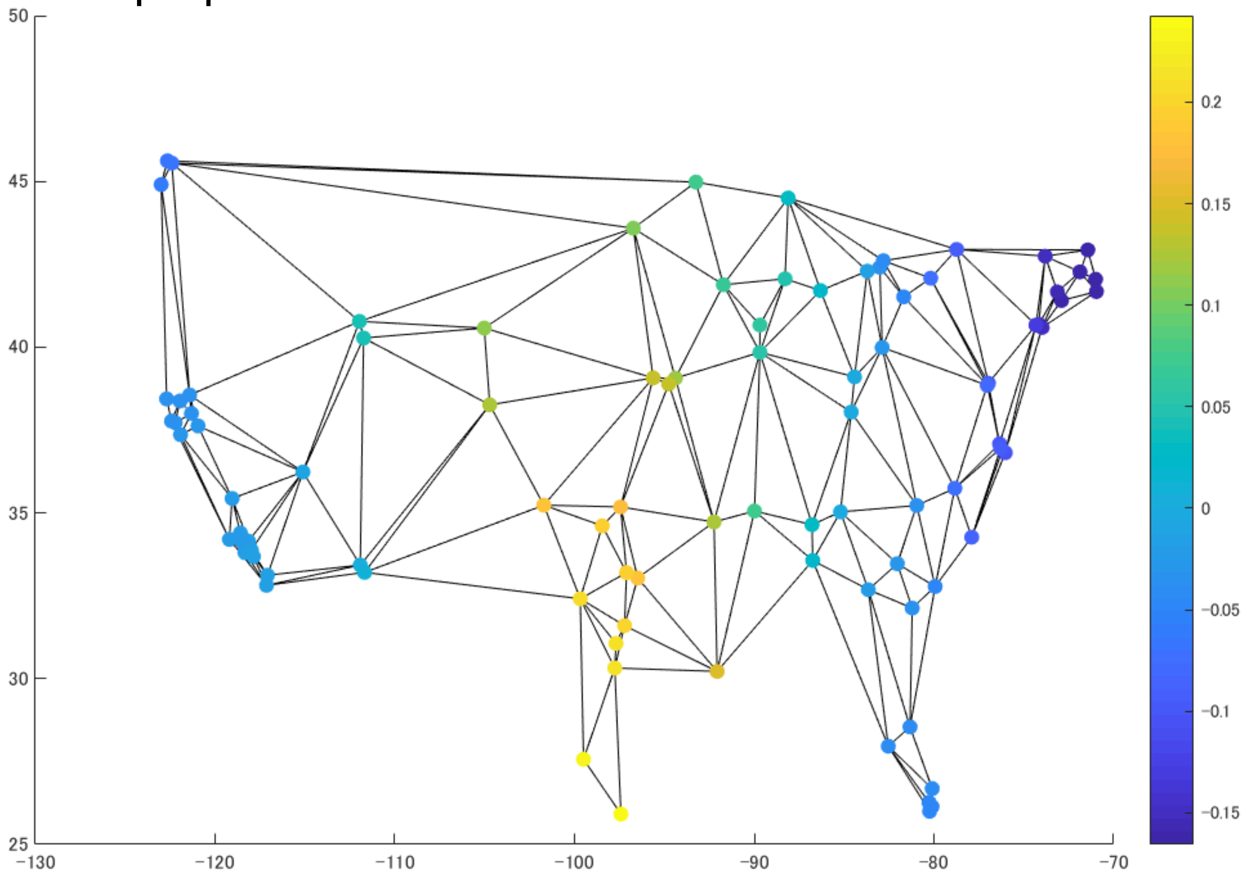
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma^2} \right)$$

Edge weights



V3: 2^{nd} AC component

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

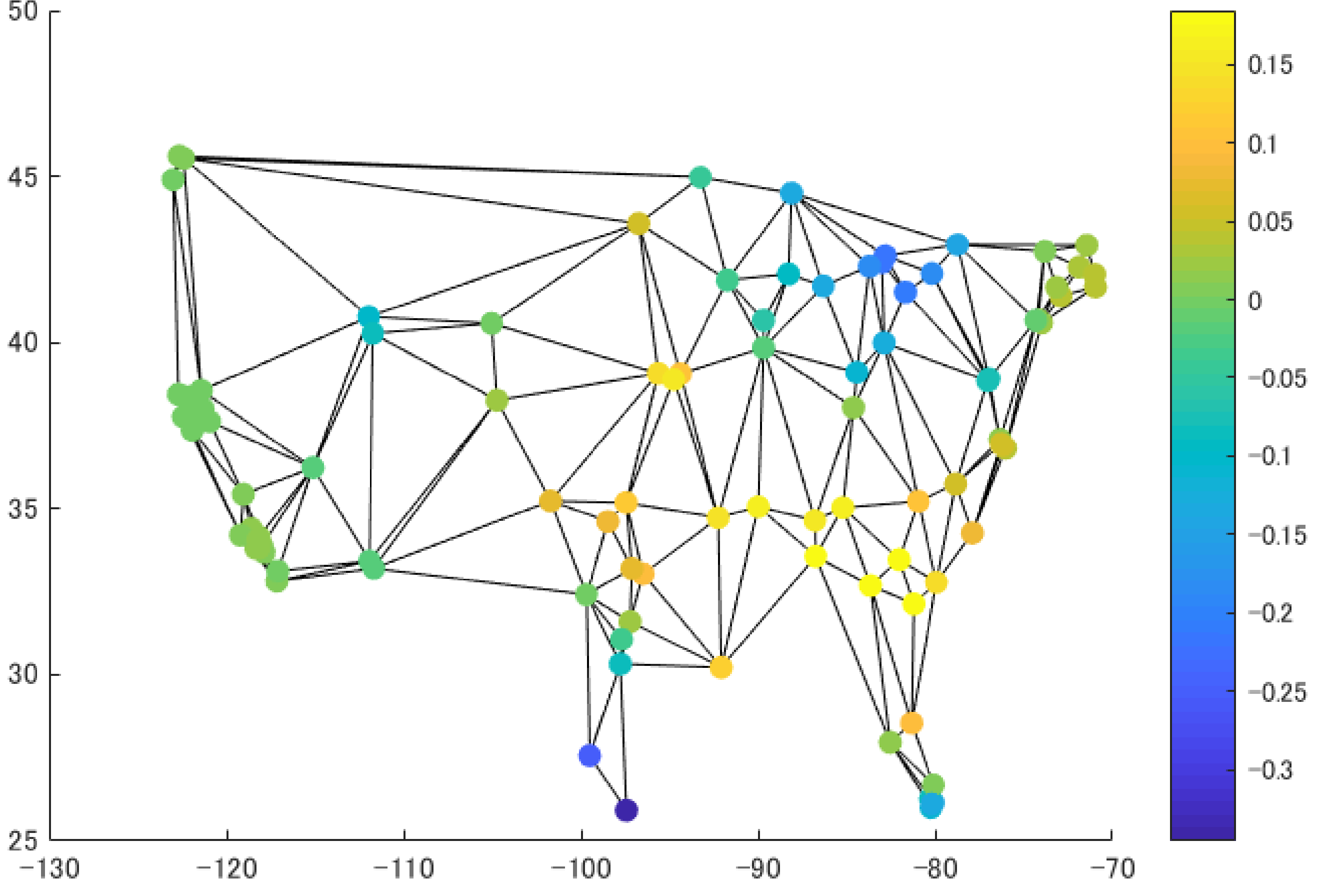
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma^2} \right)$$

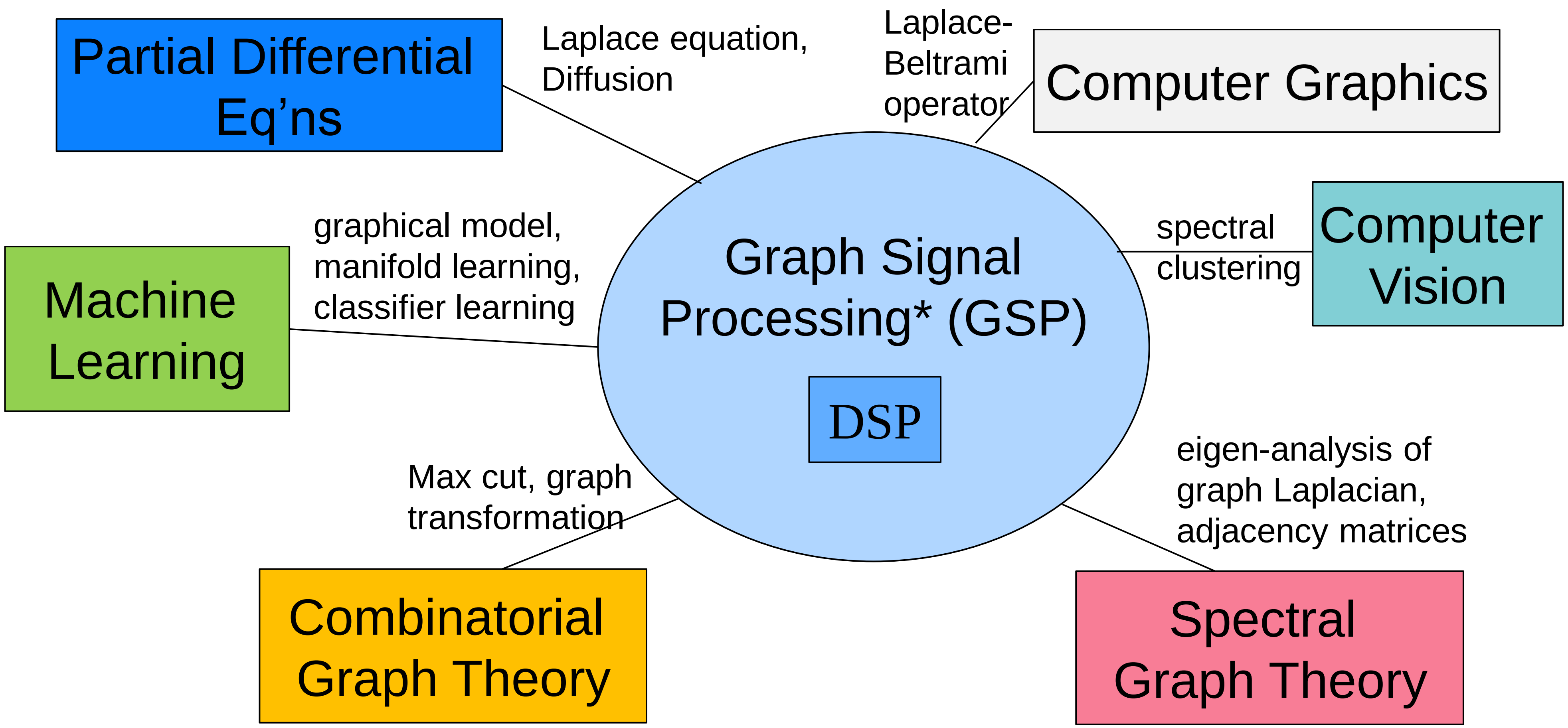
Edge weights



V4: 9th AC component

GSP and Graph-related Research

GSP: SP framework that unifies concepts from multiple fields.



Gene Cheung (genec@yorku.ca)

Outline

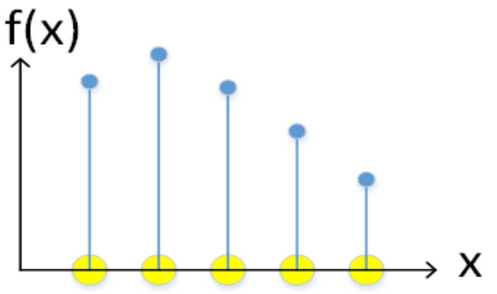
- **What is Graph Signal Processing?**
  - Graph spectrum
  - Graph Fourier transform (GFT), graph Laplacian regularizer (GLR)
- **Graph Learning**
  - Precision / Graph Laplacian Matrix Estimation (w/ eigen-structure constraint)
  - Feature Graph Learning: Gershgorin Disc Perfect Alignment (GDPA)
  - **Application**: Semi-supervised classifier learning

- **Graph Sampling**
  - Gershgorin Disc Alignment Sampling (GDAS)
  - **Application**: Sampling for matrix completion, 3D point cloud sub-sampling

- **Graph Filtering**
  - Signal-dependent GLR, GTV
  - **Application**: Image denoising

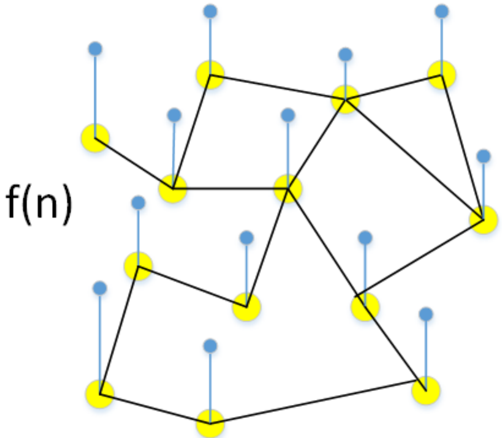
What is a good graph?

- **Graph Signal Processing** (GSP) provides **spectral analysis** tools for signals on <u>*fixed*</u> graphs.



signal on line kernel

- Graph captures *pairwise relationships*.
  1. Domain knowledge.
  2. Correlations.
  3. Feature distance.



signal on graph kernel

- **Goal**:
  1. Learn **inverse covariance matrix** from limited data.
  2. Learn metric to determine **feature distance**.

[1] X. Dong et al., Learning graphs from data: A signal representation perspective," *IEEE SPM*, vol. 36, no. 3, pp. 44-63, 2019.

Sparse Precision Matrix Estimation: GLASSO

- Given *empirical covariance matrix* Σ, **Graphical Lasso** computes positive-definite (PD) *precision matrix* Θ:

$$\max_{\Theta} \quad \log\det\Theta - \mathrm{Tr}(\Sigma\Theta) - \rho\,\|\Theta\|_1$$

- 1st and 2nd terms are likelihood.
- 3rd term promotes **sparsity**.

- Solved via **block-coordinate descent** (BCD) algorithm.

[1] Friedman J, Hastie T, Tibshirani R. "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*. 2008; 9(3): 432-441.

Sparse Precision Matrix Estimation: GLASSO

- Given *empirical covariance matrix* Σ, **Graphical Lasso** computes positive-definite (PD) *precision matrix* Θ:

$$\max_{\Theta} \quad \log\det\Theta - \mathrm{Tr}(\Sigma\Theta) - \rho\,\|\Theta\|_1$$

- 1^{st} and 2^{nd} terms are likelihood.
- 3^{rd} term promotes **sparsity**.

- Solved via **block-coordinate descent** (BCD) algorithm.

α-incoherence condition

[1] Friedman J, Hastie T, Tibshirani R. "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*. 2008; 9(3): 432-441.

Graph Laplacian Estimation

- Assume precision matrix is:
  - **Generalized graph Laplacian** (GGLs),
  - **Diagonally dominant generalized graph Laplacian** (DDGLs), or
  - **Combinatorial graph Laplacian** (CGLs).

- Given *empirical covariance matrix* S, computes *Laplacian* Θ:

$$\min_{\Theta} \quad \mathrm{Tr}(\Theta \mathbf{K}) - \log\det\Theta \quad \text{subject to} \quad \Theta \in \mathcal{L}_g(A)$$

- **K** = **S** + **H**, **H** is regularization matrix.
- $L_g(A)$ ensures Θ is GGL.
- Solved via **block-coordinate descent** (BCD) algorithm.

[1] H. E. Egilmez, E. Pavez and A. Ortega, "Graph Learning From Data Under Laplacian and Structural Constraints," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825-841, Sept. 2017

Graph Laplacian Estimation w/ Eigen-Structure Constraint

- Assume graph Laplacian matrix **L** has:

Pre-determined first *K* eigenvectors.

- Define **convex cone** $\mathcal{H}_{\mathbf{u}}^{+}$ of PSD matrices with same first *K* eigenvectors.
- Design **projection operator** to $\mathcal{H}_{\mathbf{u}}^{+}$ inspired by **Gram-Schmidt procedure**.
- Given *empirical covariance matrix* S, computes *Laplacian* **L**:

$$\min_{\mathbf{L} \in \mathcal{H}_{\mathbf{u}}^{+}} \quad \mathrm{Tr}(\mathbf{L}\bar{\mathbf{C}}) - \log\det\mathbf{L} + \rho \, \|\mathbf{L}\|_1$$

- Solve via alternating BCD and projection algorithm.

[1] S. Bagheri, G. Cheung, A. Ortega, F. Wang, "Learning Sparse Graph Laplacian with *K* Eigenvector Prior via Iterative GLASSO and Projection," accepted to *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, June 2021.

Graph Laplacian Estimation w/ Eigen-Structure Constraint

- Assume graph Laplacian matrix **L** has:

  Pre-determined first *K* eigenvectors.

Ex:
1. 1st e-vector is constant for image coding.
2. 1st e-vector is PWC for voting in Senate.
3. Sparse first *K* e-vectors for transform coding.

- Define **convex cone** $\mathcal{H}_{\mathbf{u}}^{+}$ of PSD matrices with same first *K* eigenvectors.
- Design **projection operator** to $\mathcal{H}_{\mathbf{u}}^{+}$ inspired by **Gram-Schmidt procedure**.
- Given *empirical covariance matrix* S, computes *Laplacian* **L**:

$$\min_{\mathbf{L} \in \mathcal{H}_{\mathbf{u}}^{+}} \; \mathrm{Tr}(\mathbf{L}\bar{\mathbf{C}}) - \log\det\mathbf{L} + \rho \, \|\mathbf{L}\|_1$$

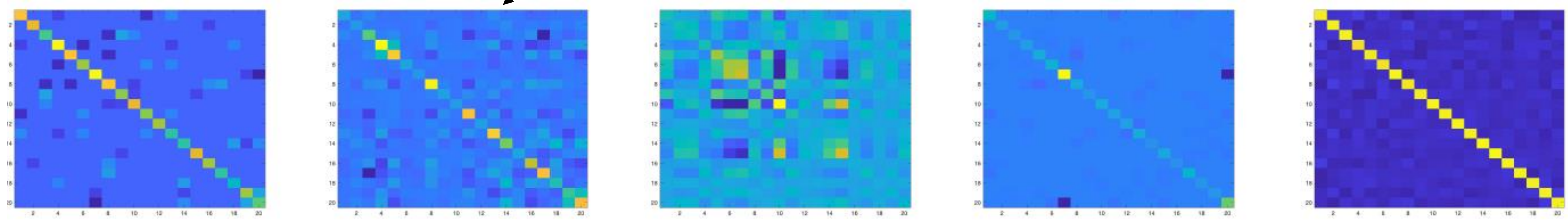
- Solve via alternating BCD and projection algorithm.

[1] S. Bagheri, G. Cheung, A. Ortega, F. Wang, "Learning Sparse Graph Laplacian with *K* Eigenvector Prior via Iterative GLASSO and Projection," accepted to *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, June 2021.

Graph Laplacian Estimation: results

- Randomly located 20 nodes in 2D space. Use the Erdos-Renyi model to determine connectivity with probability 0.6. Compute edge weights using a Gaussian kernel. Remove weights <0.75. Flip sign of each edge with probability 0.5. K=1.

- (a) Ground Truth Laplacian L , (b) Proposed Proj-Lasso with K = 1, (c) GLASSO, (d) DDGL and (e) GL-SigRep .



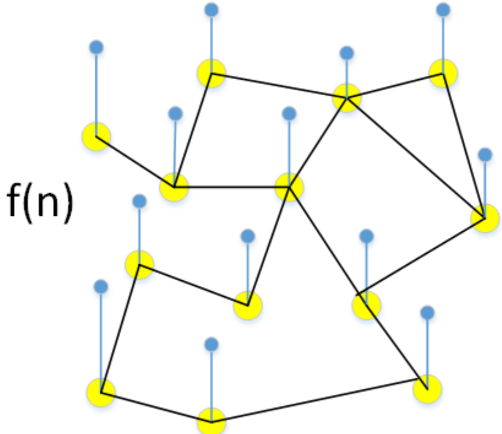
(a) (b) (c) (d) (e)

[1] S. Bagheri, G. Cheung, A. Ortega, F. Wang, "Learning Sparse Graph Laplacian with *K* Eigenvector Prior via Iterative GLASSO and Projection," accepted to *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, June 2021.

Metric Learning for Graph Construction

- Construct graph when ≤ 1 signal observation, but

  Each node has *K*-dimension feature vector.



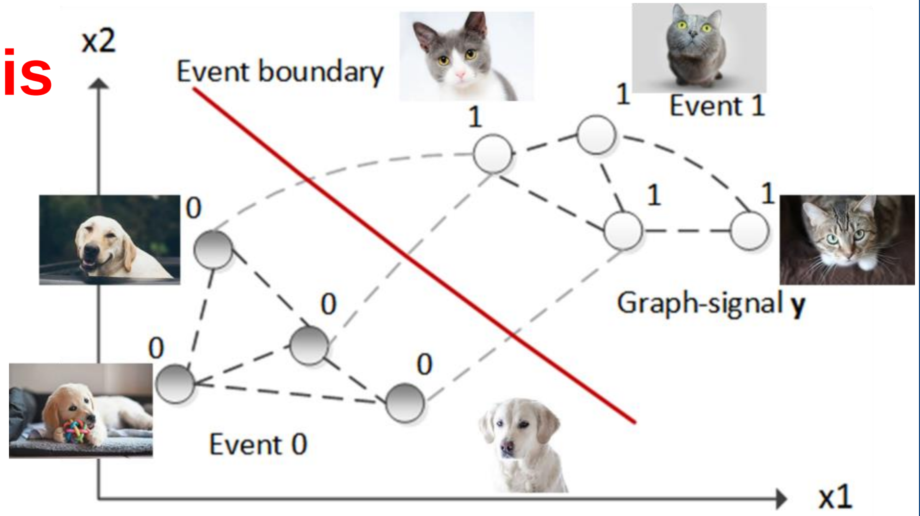
signal on graph kernel

- **Example**: *semi-supervised graph classifier*
  - Each node *i* has **feature vector** $\mathbf{f}_i \in \mathbb{R}^K$
  - Use PSD **metric matrix** **M**, establish **Mahalanobis distance**:

$$\delta_{ij} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)$$

  - Compute positive edge weight using exp:

$$w_{ij} = \exp\left(-\delta_{ij}\right)$$



[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Signal Reconstruction using GLR

- **Signal Model**:

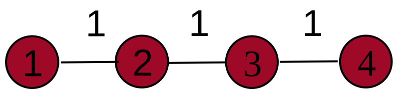
observation

sampling matrix

desired signal

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

noise



Sample set {2, 4}

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Signal prior is **graph Laplacian regularizer** (GLR):

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i,j} w_{i,j} \left(x_i - x_j\right)^2 = \sum_k \lambda_k \tilde{x}_k^2$$

signal smooth w.r.t. graph

signal contains mostly low graph freq.

$$\mathbf{H}^{\mathrm{T}}\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **MAP Formulation**:

fidelity term

signal prior

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu \ \mathbf{x}^T \mathbf{L} \mathbf{x}$$

$$(\mathbf{H}^T\mathbf{H} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

linear system of eqn's solved using *conjugate gradient*

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.

[2] C. Yang, G. Cheung, V. Stankovic, "Alternating Binary Classifier and Graph Learning from Partial Labels," *APSIPA ASC 2018*, Hawaii, USA, November 2018.

Metric Learning for Graph Construction

- Optimal **metric matrix M**:

upper bound on distance

$$\min_{\mathbf{M}} Q(\{\delta_{ij}(\mathbf{M})\}) \;\; \text{s.t.} \;\; \begin{cases} \operatorname{tr}(\mathbf{M}) \leq C \\ \mathbf{M} \succ 0 \;\; \text{or} \;\; \mathbf{M} \succeq 0 \end{cases}$$

for convex, differentiable $Q(\mathbf{M})$.

- For example, **Graph Laplacian Regularizer** (GLR):

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

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Gene Cheung (genec@yorku.ca)

Metric Learning for Graph Construction

- Optimal **metric matrix M**:

upper bound on distance

$$\min_{\mathbf{M}} Q(\{\delta_{ij}(\mathbf{M})\}) \quad \text{s.t.} \begin{cases} \mathrm{tr}(\mathbf{M}) \leq C \\ \mathbf{M} \succ 0 \ \text{ or } \ \mathbf{M} \succeq 0 \end{cases}$$

for convex, differentiable $Q(\mathbf{M})$.

- For example, **Graph Laplacian Regularizer** (GLR):

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

PSD cone constraint is **hard**!

Naïve Approach:
- Gradient descent via $-\nabla Q(\mathbf{M})$
- Projection to PSD cone.
- Repeat.

Our Approach:
- Convert PSD cone to K adaptive linear constraints via **Gershgorin Disc Alignment (GDA)**.
- Min $Q(\mathbf{M})$ w/ linear constraints.
- Repeat.

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Gershgorin Circle Theorem

Gershgorin Circle Theorem:

- Row i of **M** maps to a **Gershgorin disc** w/ centre M_{ii} and radius R_i

$$R_i = \sum_{j \neq i} |M_{ij}|$$

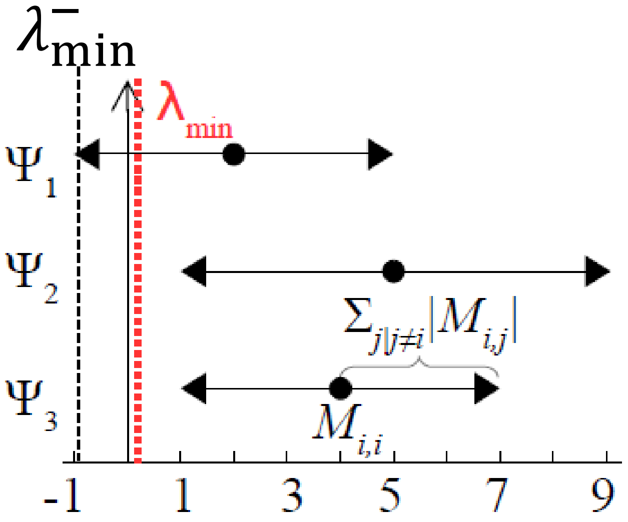
- $\lambda_{\min}$ is lower-bounded by *smallest disc left-end*:

$$\bar{\lambda}_{\min}(\mathbf{M}) \triangleq \min_i M_{i,i} - R_i \;\leq \lambda_{\min}$$

- To ensure PSDness, apply linear constr's

$$M_{i,i} - \sum_{j \neq i} |M_{ij}| \geq 0, \qquad \forall i$$

$$\mathbf{M} = \begin{bmatrix} 2 & -2 & -1 \\ -2 & 5 & -2 \\ -1 & -2 & 4 \end{bmatrix}$$



[1] R. S. Varga, *Gershgorin and His Circles*, Springer, Dec 2004.

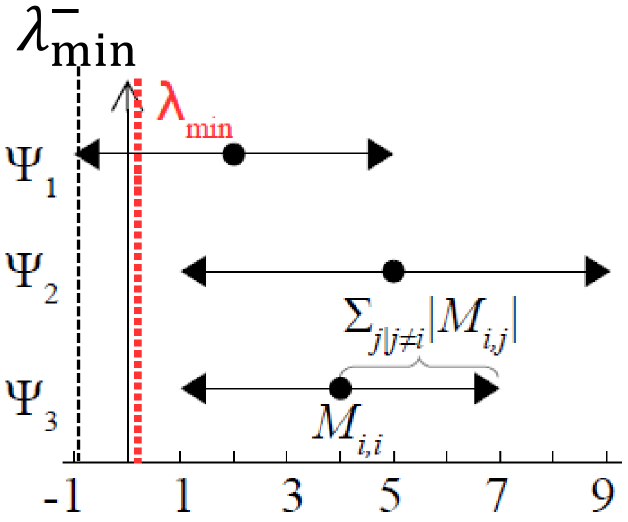
Gershgorin Circle Theorem

Gershgorin Circle Theorem:

- Row i of $\mathbf{M}$ maps to a **Gershgorin disc** w/ centre M_{ii} and radius R_i

$$R_i = \sum_{j \neq i} |M_{ij}|$$

$$\mathbf{M} = \begin{bmatrix} 2 & -2 & -1 \\ -2 & 5 & -2 \\ -1 & -2 & 4 \end{bmatrix}$$

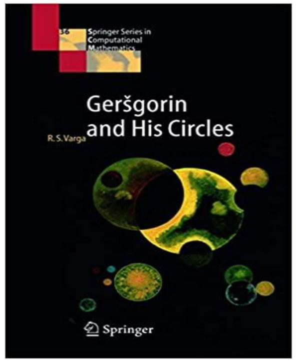


- $\lambda_{\min}$ is lower-bounded by *smallest disc left-end*:

$$\lambda^-_{\min}(\mathbf{M}) \triangleq \min_i M_{i,i} - R_i \leq \lambda_{\min}$$

- To ensure PSDness, apply linear constr's

$$M_{i,i} - \sum_{j \neq i} |M_{ij}| \geq 0, \qquad \forall i$$



[1] R. S. Varga, *Gershgorin and His Circles*, Springer, Dec 2004.

Gershgorin Disc Perfect Alignment (GDPA)

- Consider **similarity transform** of **M** (same eigenvalues!):

$$\mathbf{M} = \begin{bmatrix} 2 & -2 & -1 \\ -2 & 5 & -2 \\ -1 & -2 & 4 \end{bmatrix}$$

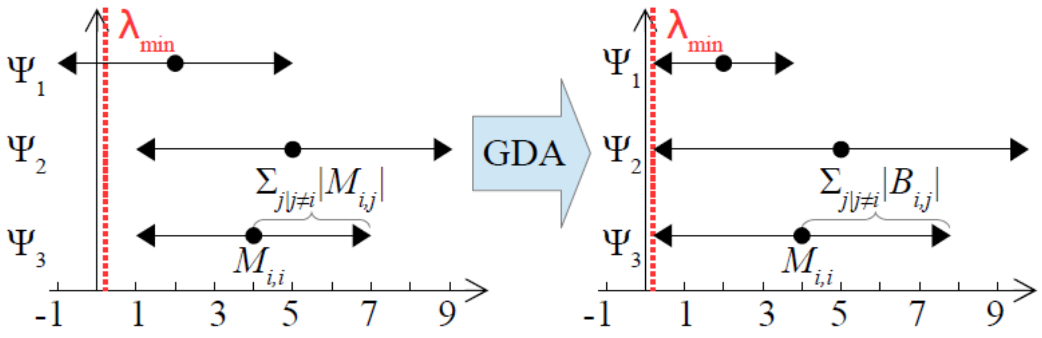
$$\mathbf{B} = \mathbf{S}\,\mathbf{M}\,\mathbf{S}^{-1}$$

similarity transform

diagonal matrix w/ scale factors s_i

- Different **S**'s induce different lower bounds $\lambda^{-}_{\min}(\mathbf{B})$!

- Which **S** do we to use??



[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Gershgorin Disc Perfect Alignment (GDPA)

- Consider **similarity transform** of **M** (same eigenvalues!):

$$\mathbf{B} = \mathbf{S}\,\mathbf{M}\,\mathbf{S}^{-1}$$

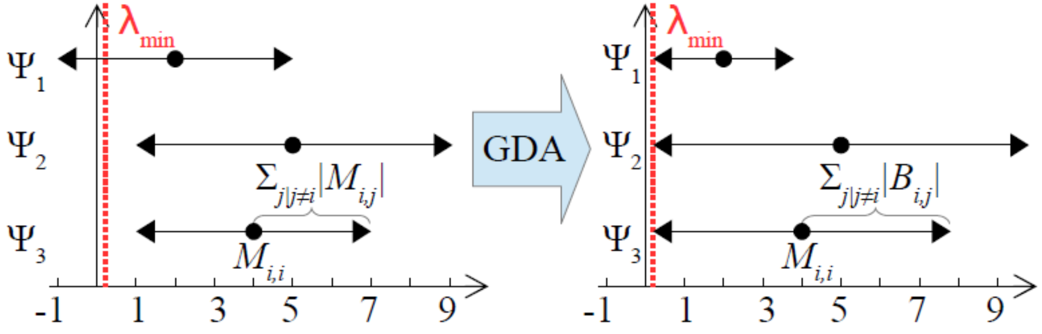
similarity transform

diagonal matrix w/ scale factors s_i

$$\mathbf{M} = \begin{bmatrix} 2 & -2 & -1 \\ -2 & 5 & -2 \\ -1 & -2 & 4 \end{bmatrix}$$

- Different **S**'s induce different lower bounds $\lambda^{-}_{\min}(\mathbf{B})$!

- Which **S** do we to use??



Theorem 1: Let **M** be a generalized graph Laplacian matrix corresponding to an irreducible, positive graph **G**. Denote by **v** the first eigenvector of **M** corresponding to the smallest eigenvalue $\lambda_{\min}$. Then by computing scalars $s_i = \frac{1}{v_i}, \forall i$, all Gershgorin disc left-ends of $\mathbf{B} = \mathbf{S}\,\mathbf{M}\,\mathbf{S}^{-1}$, $\mathbf{S} = diag(s_1, \ldots, s_N)$, are aligned at $\lambda_{\min}$.

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Metric Optimization via GDPA

$$\min_{\mathbf{M}} Q(\{\delta_{ij}(\mathbf{M})\}) \quad \text{s.t.} \begin{cases} \text{tr}(\mathbf{M}) \leq C \\ \mathbf{M} \succ 0 \quad \text{or} \quad \mathbf{M} \succeq 0 \end{cases}$$

original metric optimization

- Original **diagonal** opt w/ *PSD cone constraint*:

$$\min_{\{M_{ii}\}} \quad Q(\mathbf{M})$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \sum_i M_{ii} \leq C; \quad M_{ii} > 0, \forall i$$

- Revised **diagonal** opt w/ *linear constraints*:

$$\min_{\{M_{ii}\}} Q(\mathbf{M})$$

$$\text{s.t.} \quad M_{ii} \geq \sum_{j \mid j \neq i} \left| \frac{s_i^t M_{ij}}{s_j^t} \right| + \rho, \forall i; \quad \sum_i M_{ii} \leq C$$

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Metric Optimization via GDPA

- Original **diagonal** opt w/ *PSD cone constraint*:

$$\min_{\mathbf{M}} Q(\{\delta_{ij}(\mathbf{M})\}) \;\; \text{s.t.} \;\; \begin{cases} \text{tr}(\mathbf{M}) \leq C \\ \mathbf{M} \succ 0 \;\; \text{or} \;\; \mathbf{M} \succeq 0 \end{cases}$$

original metric optimization

$$\min_{\{M_{ii}\}} \quad Q(\mathbf{M})$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \sum_i M_{ii} \leq C; \quad M_{ii} > 0, \forall i$$

- Revised **diagonal** opt w/ *linear constraints*:

scalars s_i computed from 1st e-vector of last sol'n **M**

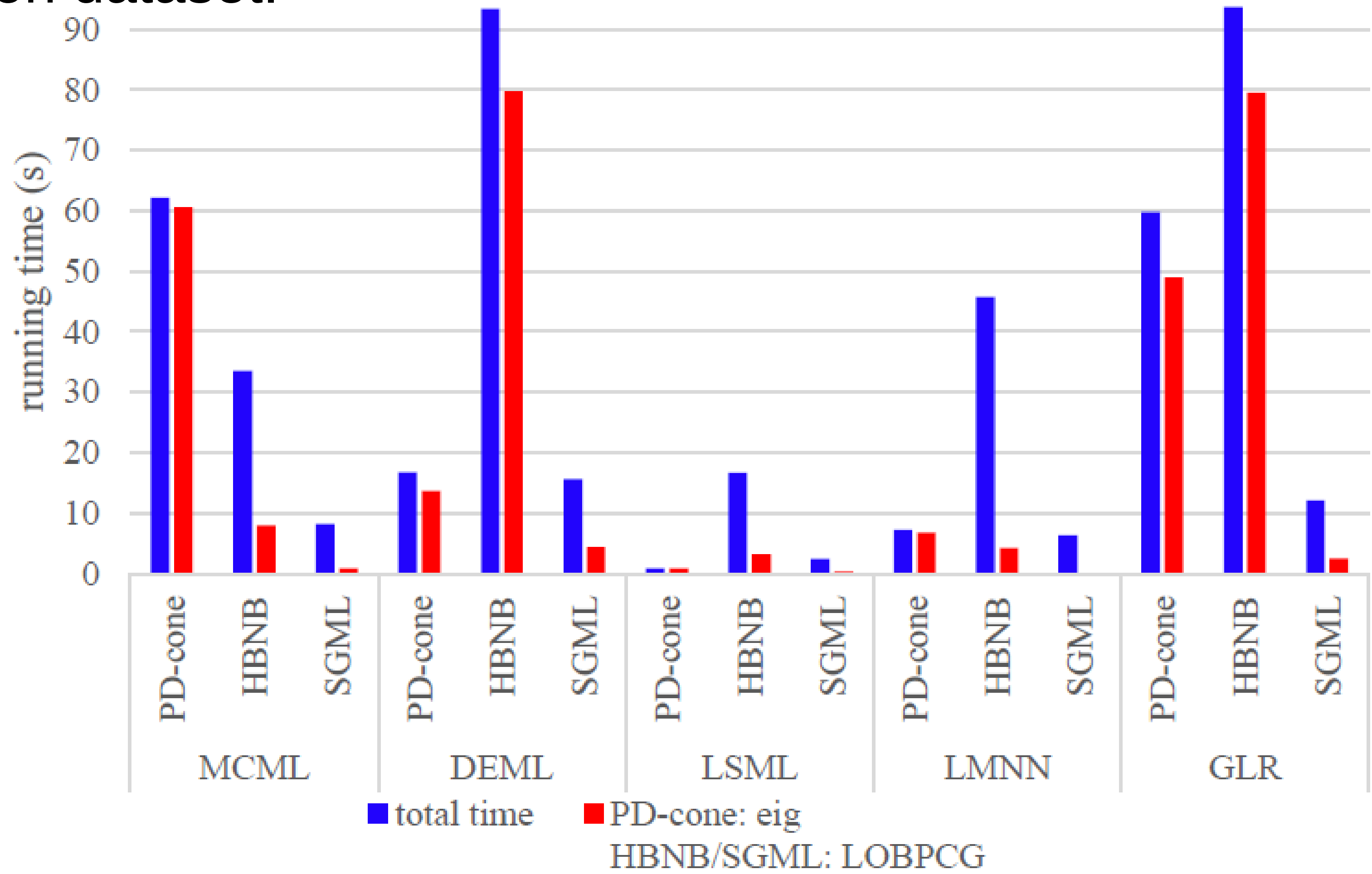
$$\min_{\{M_{ii}\}} Q\left(\mathbf{M}\right)$$

$$\text{s.t.} \;\; M_{ii} \geq \sum_{j \,|\, j \neq i} \left| \frac{s_i^t M_{ij}}{s_j^t} \right| + \rho, \forall i; \quad \sum_i M_{ii} \leq C$$

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Metric Learning Results (speed)

- Running time comparison against PD-cone and HBNB[1], for different metrics, using Madelon dataset.



[1] W. Hu, X. Gao, G. Cheung, and Z. Guo, "Feature graph learning for 3D point cloud denoising," *IEEE TSP*, vol. 68, pp. 2841-2856, 2020.

Metric Learning Results (accuracy)

- Using a GLR objective, SGML achieved the best classification results in 7 out of 14 datasets and remained competitive for 12 out of 14 datasets.

Datasets	RVML [50]	PLML [51]	mmLMNN [1]	GMML [33]	DMLMJ [52]	SCML [53]	DMLE [32]	R2LML [54]	LMLIR [49]	SGML (prop.)		
										3-NN	Mahalanobis	Graph
australian	83.0±1.6	80.5±1.1	82.5±2.6	84.4±1.0	83.9±1.3	82.3±1.4	82.6±1.5	84.7±1.3	85.1±1.9	83.3±1.2	84.8±1.3	**85.3±1.7**
breastcancer	95.8±1.1	96.4±0.9	96.7±1.0	97.3±0.8	96.6±0.8	97.0±0.9	97.0±1.1	97.0±0.7	96.4±2.1	97.6±1.0	**98.0±0.6**	97.6±0.7
diabetes	71.0±2.6	68.5±2.0	72.2±1.9	74.2±2.6	71.5±3.1	71.5±2.2	72.6±2.0	73.8±1.4	**75.9±1.9**	71.6±1.8	70.5±2.5	70.3±1.4
fourclass	70.5±1.4	72.4±2.4	75.6±1.4	76.1±1.9	76.1±1.9	75.5±1.4	75.6±1.4	76.1±1.9	**79.9±0.9**	74.5±2.4	71.1±1.6	78.0±1.2
german	71.7±1.8	70.0±2.9	68.9±1.8	71.6±1.1	69.3±2.7	70.9±2.7	72.0±2.1	72.9±1.8	**73.7±1.6**	71.6±1.7	70.9±1.3	70.0±0.0
haberman	66.7±2.3	67.1±3.1	69.0±2.7	71.2±3.4	68.5±3.2	69.2±2.5	70.8±3.5	71.1±3.4	**74.4±3.7**	68.8±3.9	66.6±6.3	73.6±0.3
heart	77.7±4.1	75.1±3.2	79.4±3.7	81.2±2.7	80.6±2.8	79.0±3.2	77.9±3.1	82.0±3.8	83.1±3.2	81.0±3.4	83.2±3.6	**83.6±3.5**
ILPD	68.0±2.9	67.4±3.0	66.8±2.1	67.1±2.2	68.0±1.6	68.0±2.9	68.8±2.7	65.9±2.2	69.6±2.7	65.2±2.4	59.1±2.4	**71.3±0.2**
liverdisorders	64.6±3.9	62.2±2.5	62.0±3.5	63.8±5.4	60.9±3.8	61.7±4.6	61.8±2.7	66.8±3.7	66.7±3.6	69.5±3.3	68.8±5.9	**72.1±3.0**
monk1	89.2±2.7	96.6±2.7	90.3±2.6	75.0±2.6	87.7±3.8	97.5±0.9	**99.9±0.3**	89.2±1.5	95.0±7.2	84.6±5.1	66.3±3.0	71.1±3.7
pima	69.5±1.7	68.4±2.2	72.5±2.7	73.0±1.8	71.1±2.8	71.1±2.6	72.1±2.4	72.3±1.5	**74.6±2.0**	73.4±1.3	73.6±2.0	69.2±1.5
planning	55.1±7.4	60.8±5.5	54.7±0.9	65.2±5.5	64.3±2.9	61.9±5.0	60.1±5.5	63.9±3.4	67.5±6.5	62.8±4.1	48.8±4.8	**71.3±0.7**
voting	95.8±1.3	95.5±1.0	95.4±0.9	95.2±1.9	95.3±1.1	95.0±1.3	93.1±1.9	96.3±1.2	93.2±3.9	**96.4±1.4**	94.3±2.0	94.8±1.6
WDBC	96.6±1.3	96.4±0.9	**97.4±1.0**	96.7±0.8	97.3±1.9	97.0±0.9	96.7±0.5	96.9±1.7	96.6±1.0	96.6±0.9	94.8±1.2	96.2±1.1
Average	76.7	76.9	77.3	77.9	77.9	78.4	78.6	79.2	80.8	78.4	75.1	78.9
# of best	0	0	1	0	0	0	1	0	5	1	1	5

[1] C. Yang, G. Cheung, W. Hu, "Signed Graph Metric Learning via Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2020.

Outline

- **What is Graph Signal Processing?**
  - Graph spectrum
  - Graph Fourier transform (GFT), graph Laplacian regularizer (GLR)
- **Graph Learning**
  - Precision / Graph Laplacian Matrix Estimation (w/ eigen-structure constraint)
  - Feature Graph Learning: Gershgorin Disc Perfect Alignment (GDPA)
  - **Application**: Semi-supervised classifier learning

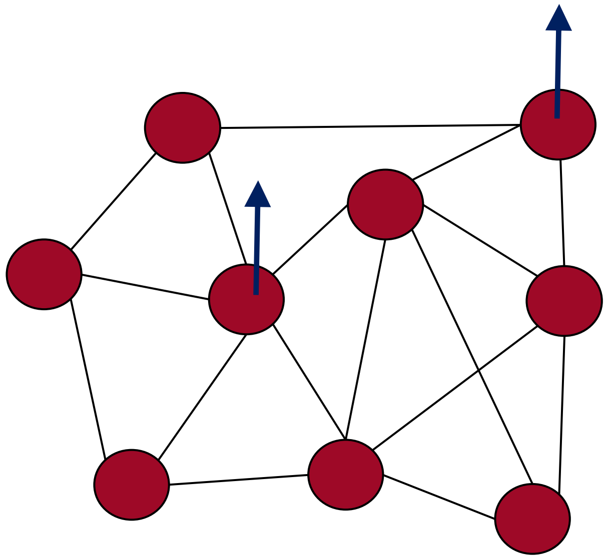
- **Graph Sampling**
  - Gershgorin Disc Alignment Sampling (GDAS)
  - **Application**: Sampling for matrix completion, 3D point cloud sub-sampling

- **Graph Filtering**
  - Signal-dependent GLR, GTV
  - **Application**: Image denoising

Graph Sampling (with and without noise)

Q: How to choose best samples for graph-based reconstruction?

- Existing graph sampling strategies extend **Nyquist sampling** to graph data kernels:
  - Assume ***bandlimited*** signal.
  - Greedily select most "informative" samples by computing extreme eigenvectors of sub-matrix.
  - Computation-expensive.



[1] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.

[2] Y. Tanaka, Y. C. Eldar, A. Ortega, G. Cheung, "Sampling on Graphs: From Theory to Applications," *IEEE Signal Processing Magazine*, vol. 37, no.6, pp.14-30, November 2020.

Signal Reconstruction using GLR

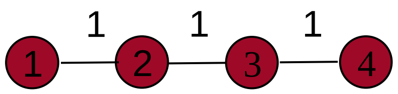
- **Signal Model**:

observation　　sampling matrix　　desired signal

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

noise

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Sample set {2, 4}

- Signal prior is **graph Laplacian regularizer** (GLR):

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j} w_{i,j}\left(x_i - x_j\right)^2 = \sum_k \lambda_k \tilde{x}_k^2$$

signal smooth w.r.t. graph

signal contains mostly low graph freq.

- **MAP Formulation**:

fidelity term　　signal prior

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu\ \mathbf{x}^T\mathbf{L}\mathbf{x}$$

$$(\mathbf{H}^T\mathbf{H} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

linear system of eqn's solved using *conjugate gradient*

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.

Stability of Linear System

- Examine solution's linear system:

$$(\mathbf{H}^T\mathbf{H} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

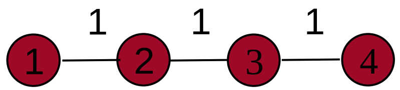
coefficient matrix **B**

- Stability depends on **condition number** (λ_{max}/ λ_{min}) of **B**.
- λ_{max} is upper-bounded by $1+\mu 2^* d_{max}$.

Goal: select **H** to maximize $\lambda_{\min}(\mathbf{B})$ (w/o computing eigen-pairs)!

Also minimizes **worst-case MSE**:

$$\|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq \mu \left\| \frac{1}{\lambda_{min}(\mathbf{B})} \right\|_2 \|\mathbf{L}(\mathbf{x} + \tilde{\mathbf{n}})\|_2 + \|\tilde{\mathbf{n}}\|_2$$



$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$\mathbf{H}^{\mathrm{T}}\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sample set {2, 4}

[1] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, "Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment," vol. 68, pp. 2419-2434, *IEEE Transactions on Signal Processing*, March 2020.

Gershgorin Circle Theorem

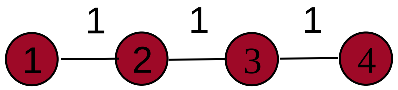
Gershgorin Circle Theorem:

- Row *i* of **L** maps to a **Gershgorin disc** w/ centre L_{ii} and radius R_i

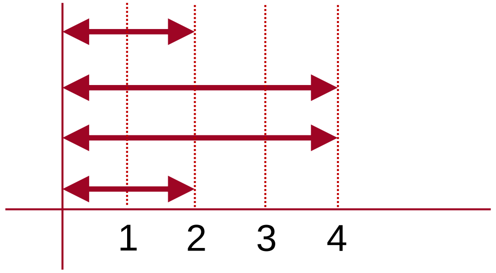
$$R_i = \sum_{j \neq i} |L_{ij}|$$

- $\lambda_{\min}$ is lower-bounded by smallest left-ends of Gershgorin discs:

$$\min_i \; L_{i,i} - R_i \leq \lambda_{\min}$$



$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$



<u>*Graph Laplacian L has all Gershgorin disc left-ends at 0*</u>

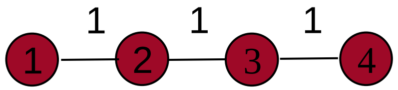
→ **L** is PSD.

Gershgorin Disc Alignment Sampling (GDAS)

Main Idea: Select samples to max smallest disc left-end of coefficient matrix **B**:

$$\mathbf{B} = \mathbf{H}^T \mathbf{H} + \mu\, \mathbf{L}$$

coeff. matrix



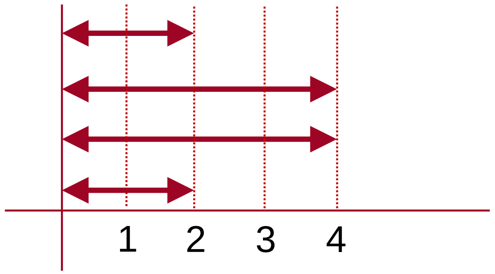
$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

- Sample node → shift disc.
- Consider similarity transform of **B** (same eigenvalues!):

$$\mathbf{C} = \mathbf{S}\,\mathbf{B}\,\mathbf{S}^{-1}$$

similarity transform

diagonal matrix w/ scale factors



- Scale row → **expand** disc radius.
  → **shrink** neighbors' disc radius.

Sample set { }

Scale factor {1,1,1,1}

[1] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, "Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment," vol. 68, pp. 2419-2434, *IEEE Transactions on Signal Processing*, March 2020.

Gershgorin Disc Alignment Sampling (GDAS)

Main Idea: Select samples to max smallest disc left-end of coefficient matrix **B**:

$$\mathbf{B} = \mathbf{H}^T \mathbf{H} + \mu \mathbf{L} \quad \longleftarrow \text{ coeff. matrix}$$

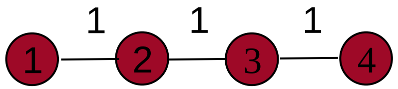
- Sample node → shift disc.
- Consider similarity transform of **B** (same eigenvalues!):

$$\mathbf{C} = \mathbf{S}\,\mathbf{B}\,\mathbf{S}^{-1} \quad \longleftarrow \text{ similarity transform}$$

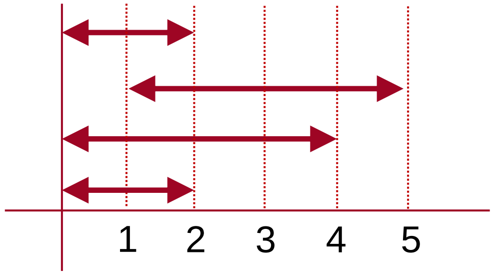
diagonal matrix w/ scale factors

- Scale row → **expand** disc radius.

  → **shrink** neighbors' disc radius.



$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$



Sample set {2}

Scale factor {1,1,1,1}

[1] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, "Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment," vol. 68, pp. 2419-2434, *IEEE Transactions on Signal Processing*, March 2020.

Gershgorin Disc Alignment Sampling (GDAS)

Main Idea: Select samples to max smallest disc left-end of coefficient matrix **B**:

$$\mathbf{B} = \mathbf{H}^T\,\mathbf{H} + \mu\,\mathbf{L}$$ ⟵ coeff. matrix

- Sample node → shift disc.

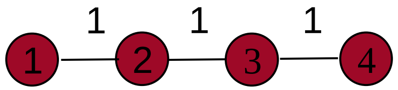
- Consider similarity transform of **B** (same eigenvalues!):

$$\mathbf{C} = \mathbf{S}\,\mathbf{B}\,\mathbf{S}^{-1}$$ ⟵ similarity transform

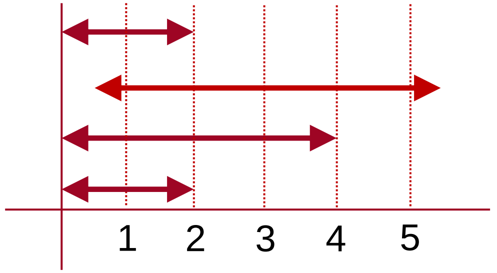
diagonal matrix w/ scale factors

- Scale row → **expand** disc radius.

  → **shrink** neighbors' disc radius.



$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$



Sample set {2}

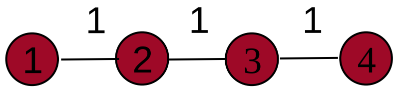
Scale factor $\{1, s_2, 1, 1\}$

[1] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, "Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment," vol. 68, pp. 2419-2434, *IEEE Transactions on Signal Processing*, March 2020.

Gershgorin Disc Alignment Sampling (GDAS)

Main Idea: Select samples to max smallest disc left-end of coefficient matrix **B**:

$$\mathbf{B} = \mathbf{H}^T \mathbf{H} + \mu \mathbf{L} \quad \longleftarrow \text{ coeff. matrix}$$



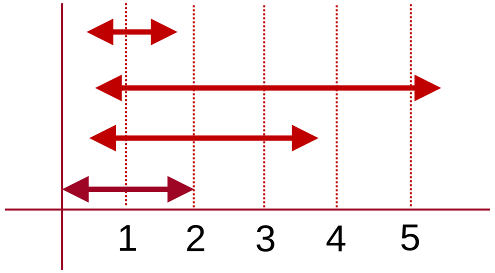
$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

- Sample node → shift disc.

- Consider similarity transform of **B** (same eigenvalues!):

$$\mathbf{C} = \mathbf{S}\,\mathbf{B}\,\mathbf{S}^{-1} \quad \longleftarrow \text{ similarity transform}$$

diagonal matrix w/ scale factors



- Scale row → **expand** disc radius.
  → **shrink** neighbors' disc radius.

Sample set {2}

Scale factor $\{1, s_2, 1, 1\}$

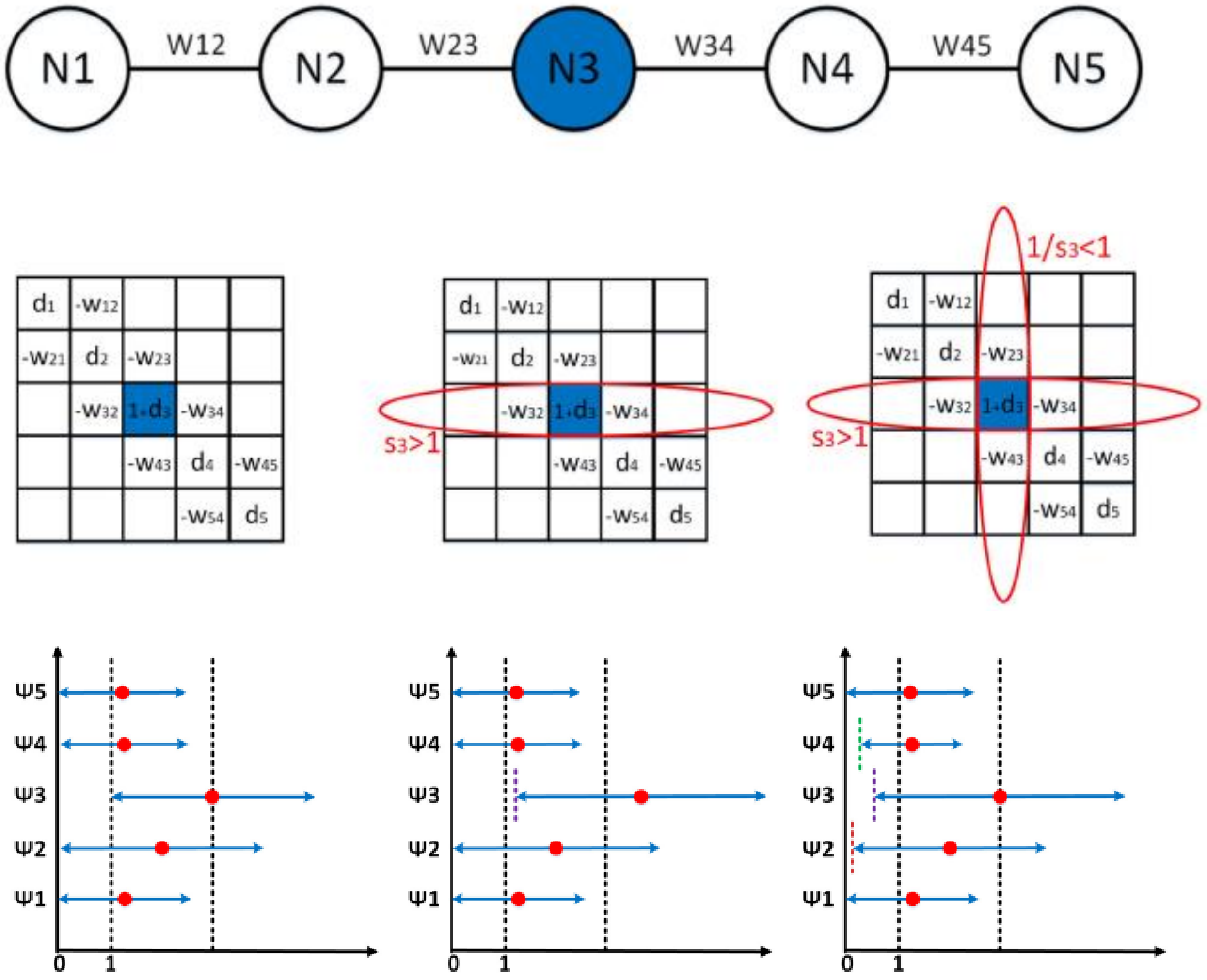
[1] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, "Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment," vol. 68, pp. 2419-2434, *IEEE Transactions on Signal Processing*, March 2020.

Solving Dual Sampling Problem: align discs @ *T*

Breadth First Iterative Sampling (BFIS):

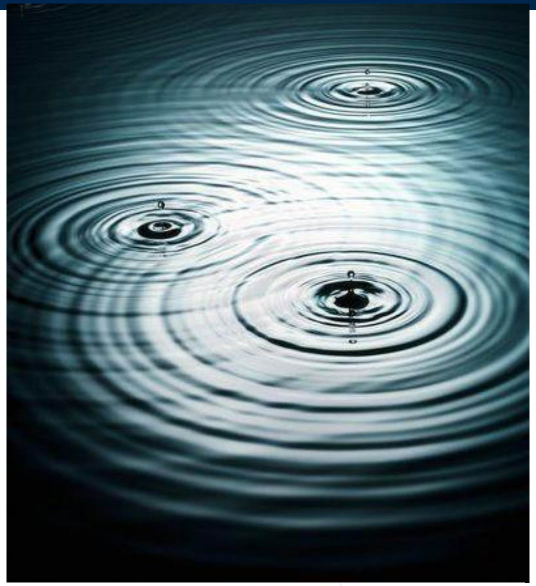
- Given initial node set, threshold *T*.

1. Sample chosen node *i* (shift disc)
2. Scale row *i* (expand disc radius *i* to *T*)
3. If disc left-end of connected node *j* >*T*,
   Scale row *j* (expand disc radius *j* to *T*)
   Else,
   Add node *j* to node set.
4. Goto step 1 if node set not empty.
5. Output sample set and count *K*.



[1] Y. Bai, G. Cheung, F. Wang, X. Liu, W. Gao, "Reconstruction-Cognizant Graph Sampling Using Gershgorin Disc Alignment," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, May 2019.

Disc-based Sampling (Intuition)

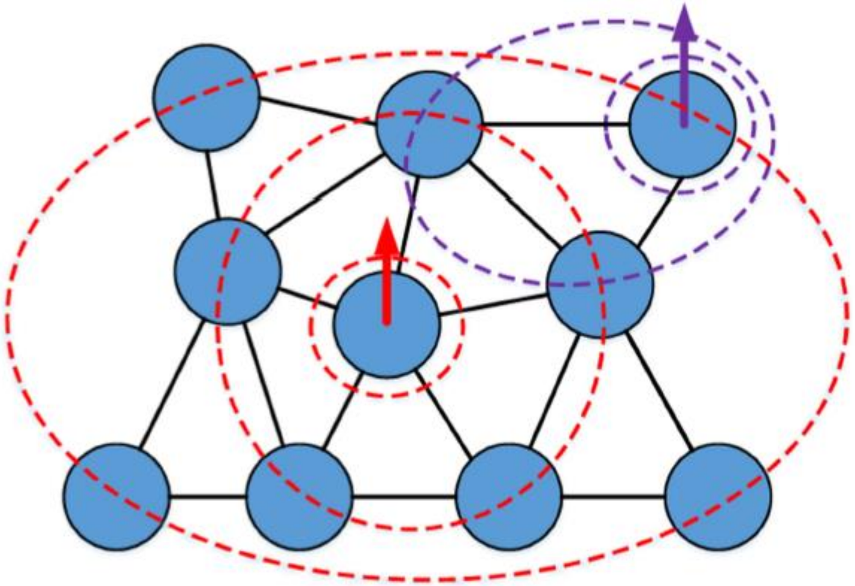


Analogy: throw pebbles into a pond.

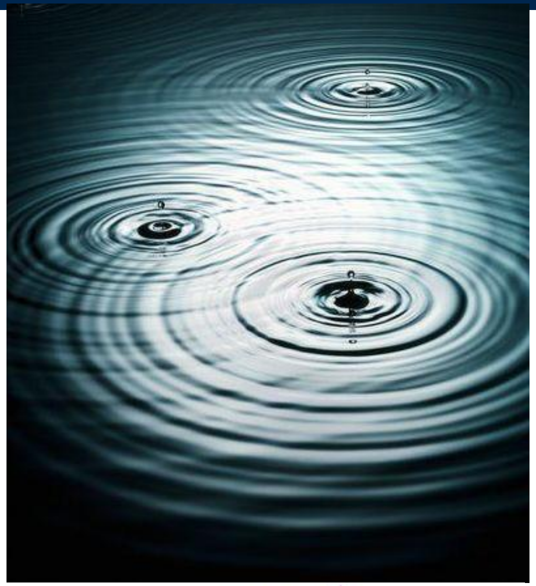
Disc Shifting: throw pebble at sample node *i*.

Disc Scaling: ripple to neighbors of node *i*.

Goal: Select min # of samples so ripple at each node is at least *T*.



Disc-based Sampling (Intuition)



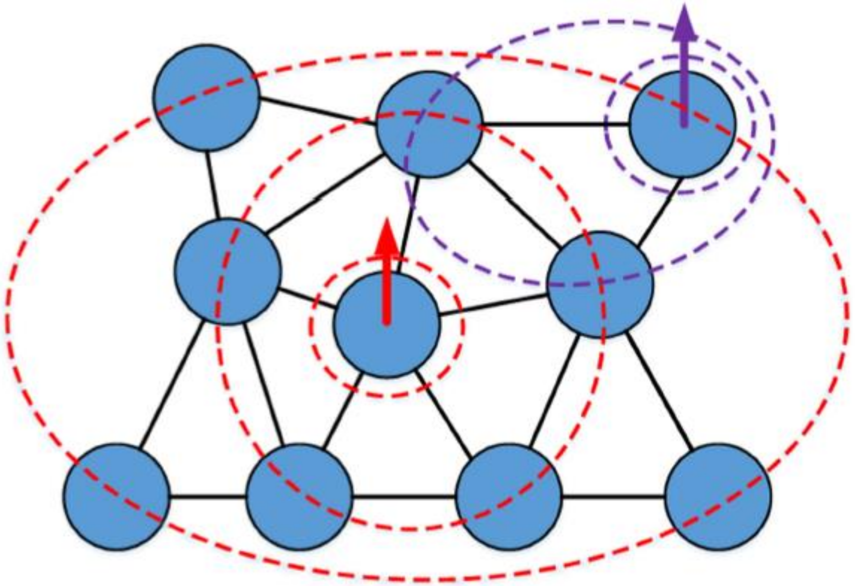
Analogy: throw pebbles into a pond.

Disc Shifting: throw pebble at sample node *i*.

Disc Scaling: ripple to neighbors of node *i*.

Goal: Select min # of samples so ripple at each node is at least *T*.

Takeaway Message: roughly linear time graph sampling algorithm minimizing a global error obj.



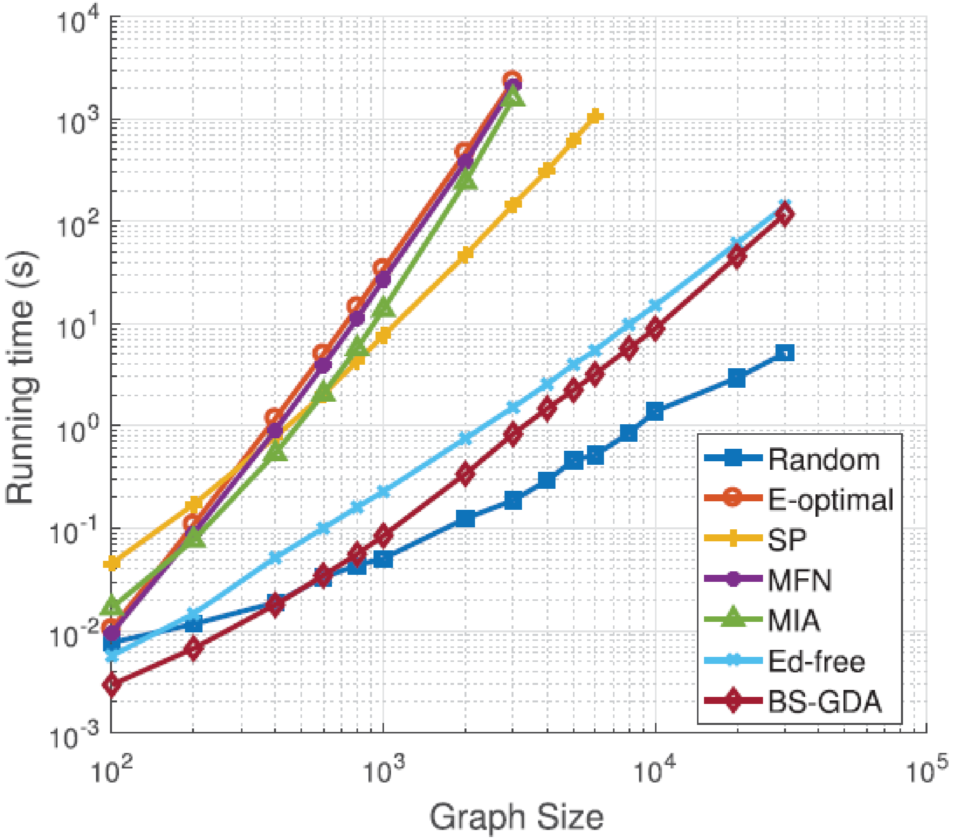
Gene Cheung (genec@yorku.ca)

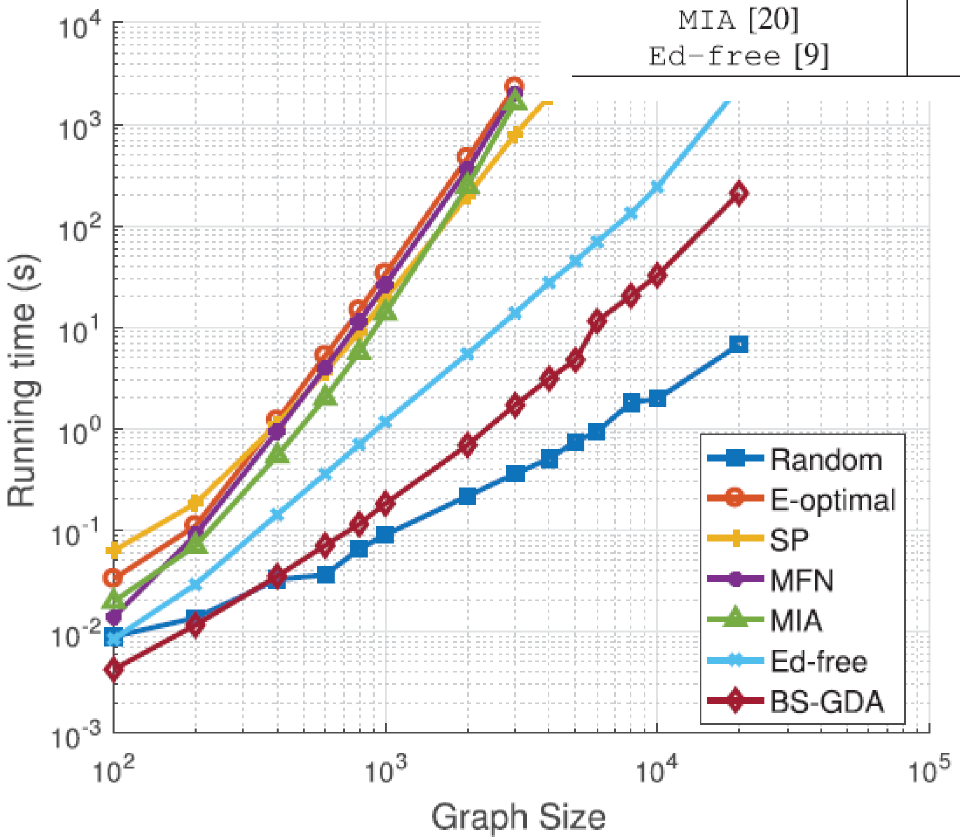
Graph Sampling Results: speed

- Running time comparisons on two different graphs. (a) Random sensor raph. (b) Community graph.

TABLE II
SPEEDUP FACTORS OF OUR ALGORITHM WITH RESPECT TO OTHER SAMPLING ALGORITHMS FOR $N = 3000$

Sampling Algorithms	Sensor	Community
Random [27]	0.22	0.21
E-optimal [24]	2812.77	1360.76
SP [16]	174.09	466.18
MFN [22]	2532.91	1184.23
MIA [20]	1896.19	964.65
Ed-free [9]	1.82	8.11

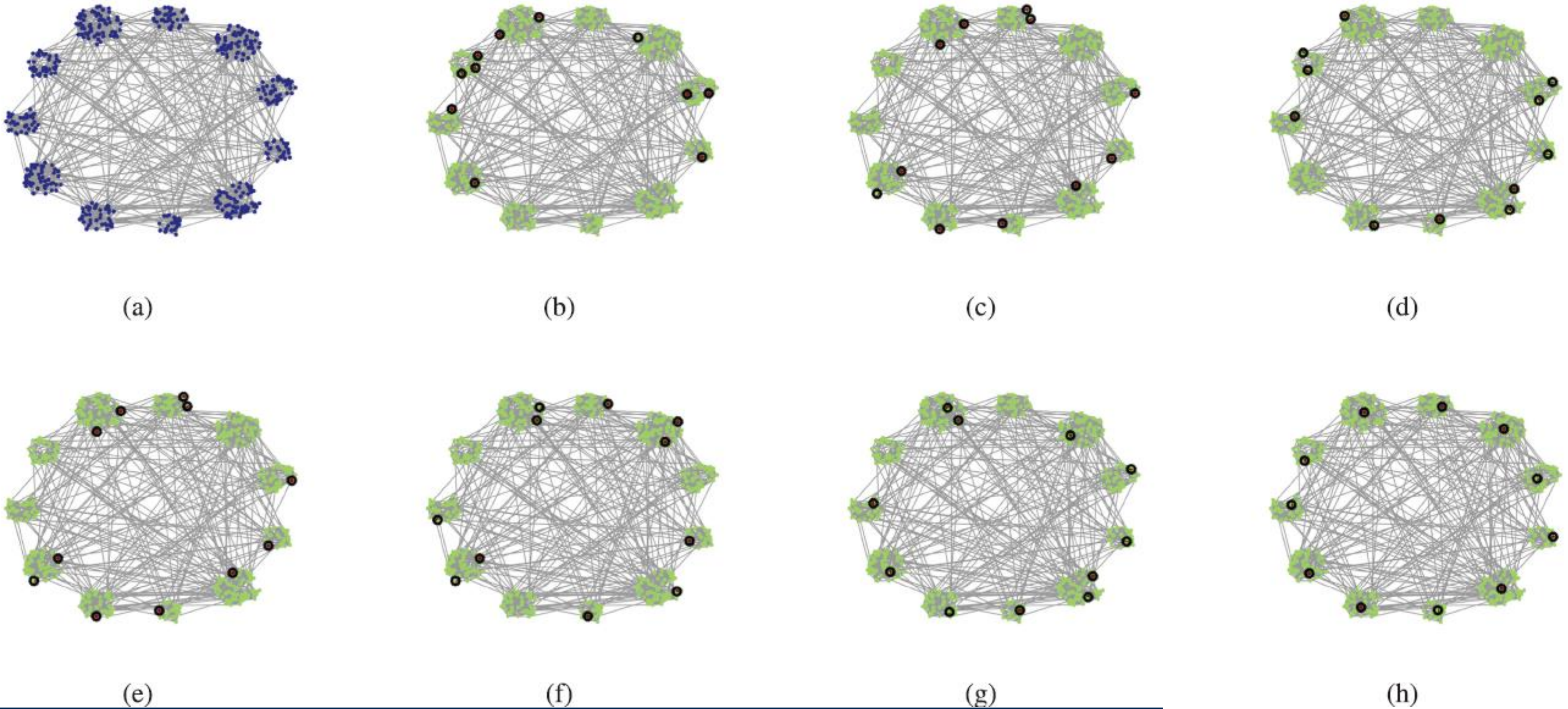




Gene Cheung (genec@yorku.ca)

Graph Sampling Results: community graph

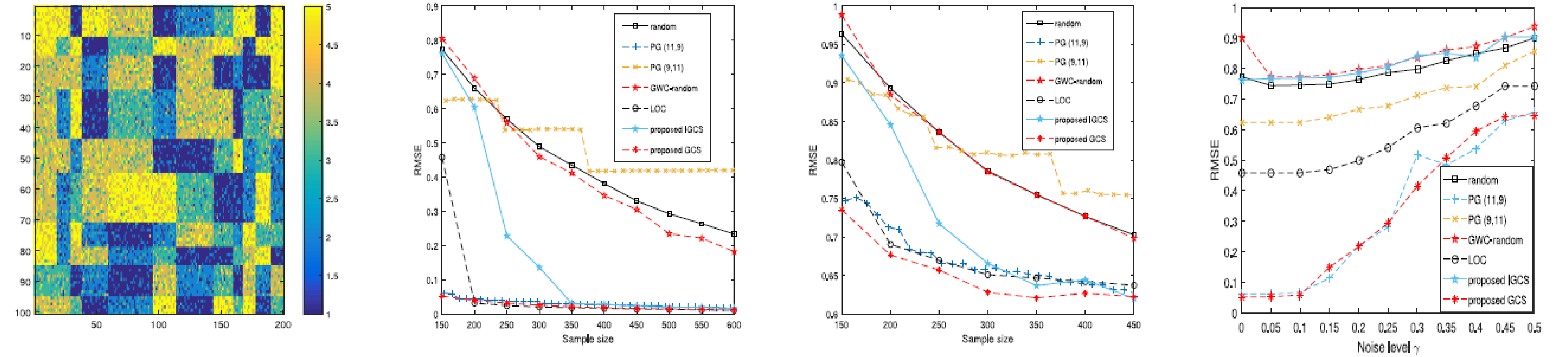
- Visualization of selected nodes on the community graph ($N = 500, K = 11$). Black circles denote sampled nodes. (a) Original graph. (b) Random [28].(c) E-optimal [25]. (d) SP [16]. (e) MFN [23]. (f) MIA [20]. (g) Ed-free [9]. (h) The proposed BS-GDA.



Graph Sampling Results: matrix completion

- Pre-select a subset of matrix entries for sampling to maximize **matrix completion** fidelity.
- **Challenge**: select sampling set Ω to maximize λ_{min} of $\tilde{\mathbf{A}}_\Omega + \alpha \mathbf{I}_n \otimes \mathbf{L}_r + \beta \mathbf{L}_c \otimes \mathbf{I}_m$

- RMSE of different sampling methods for MC on Synthetic Netflix. The matrix was completed using the *double graph smoothness* based method.



(a) Noisy synthetic Netflix signal (b) RMSE on noiseless signal (c) RMSE on noisy signal with $\gamma = 0.6$ (d) RMSE on different noise level γ

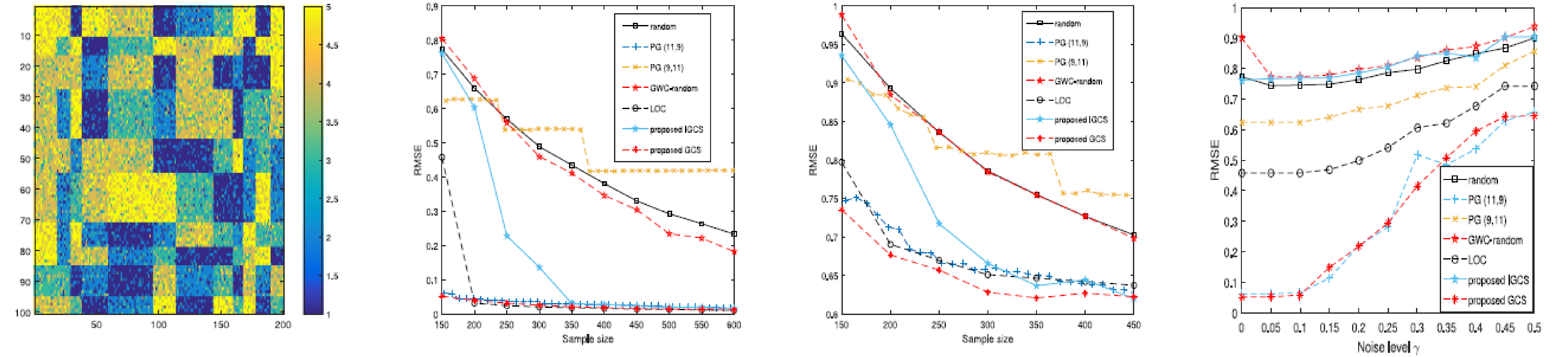
[1] F. Wang, Y. Wang, G. Cheung, C. Yang, "Graph Sampling for Matrix Completion Using Recurrent Gershgorin Disc Shift," vol. 68, pp. 1814-2829, *IEEE Transactions on Signal Processing*, April 2020.

Graph Sampling Results: matrix completion

- Pre-select a subset of matrix entries for sampling to maximize **matrix completion** fidelity.
- **Challenge**: select sampling set Ω to maximize λ_{min} of $\tilde{\mathbf{A}}_\Omega + \alpha \mathbf{I}_n \otimes \mathbf{L}_r + \beta \mathbf{L}_c \otimes \mathbf{I}_m$

graph Laplacians for row / column graphs

- RMSE of different sampling methods for MC on Synthetic Netflix. The matrix was completed using the *double graph smoothness* based method.

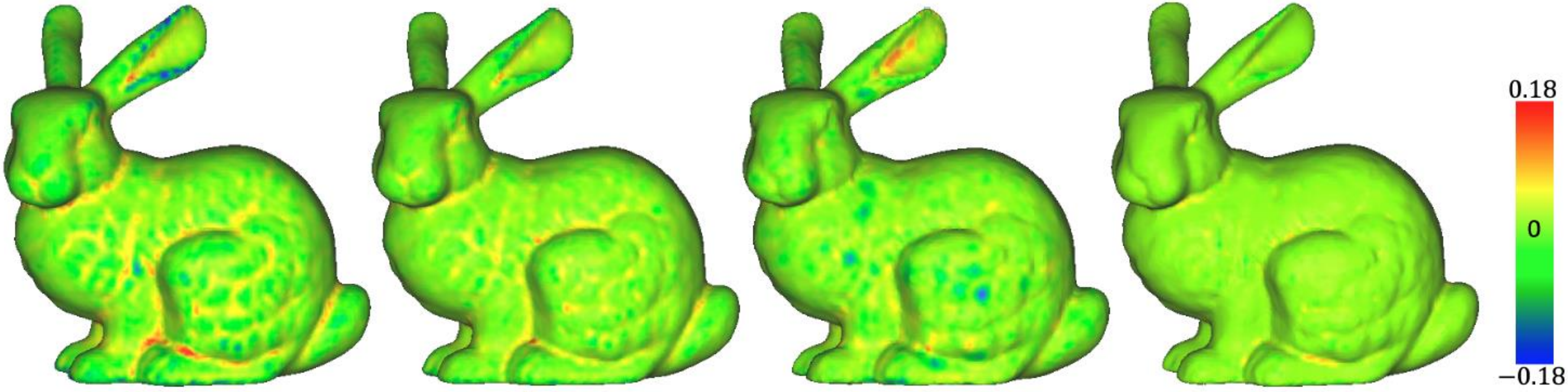


(a) Noisy synthetic Netflix signal (b) RMSE on noiseless signal (c) RMSE on noisy signal with $\gamma = 0.6$ (d) RMSE on different noise level γ

[1] F. Wang, Y. Wang, G. Cheung, C. Yang, "Graph Sampling for Matrix Completion Using Recurrent Gershgorin Disc Shift," vol. 68, pp. 1814-2829, *IEEE Transactions on Signal Processing*, April 2020.

Graph Sampling Results: 3D point cloud sub-sampling

- Reduce 3D point cloud size by sub-sampling while preserving the overall object shape.
- **Challenge**: select sampling matrix **H** to maximize λ_{min} of $\mathbf{H}^\top \mathbf{H} + \mu \mathcal{L}$
- SR reconstruction results from diff. methods of sub-sampled Bunny under 0.2 sub-sampling ratio.



(a) BGFS (b) PDS (c) FPS (d) proposed

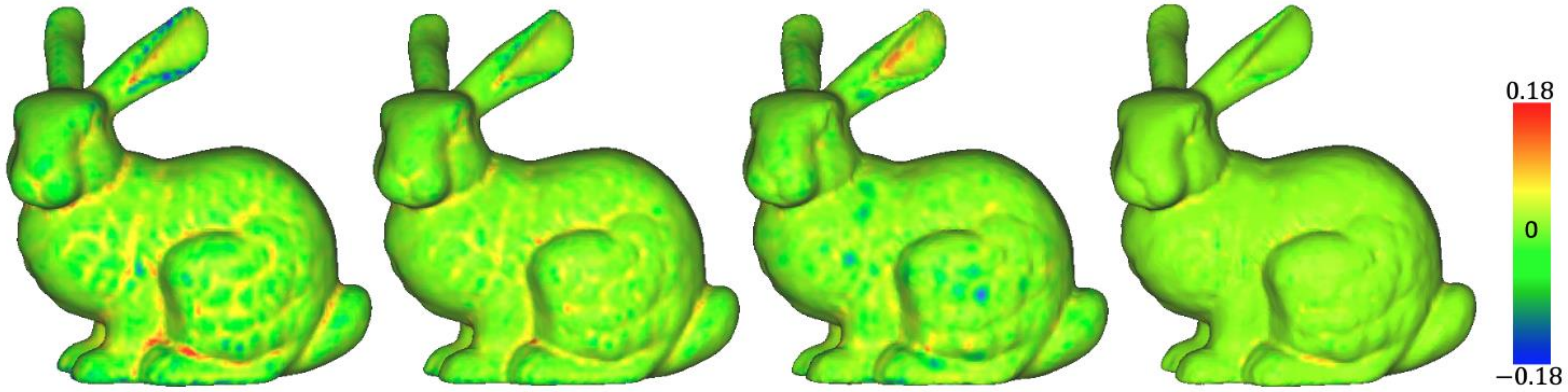
[1] C. Dinesh, G. Cheung, I. V. Bajic, "Point Cloud Sampling via Graph Balancing and Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, January 2021.

Graph Sampling Results: 3D point cloud sub-sampling

- Reduce 3D point cloud size by sub-sampling while preserving the overall object shape.
- **Challenge**: select sampling matrix **H** to maximize λ_{min} of $\mathbf{H}^\top \mathbf{H} + \mu \mathcal{L}$

generalized graph Laplacian

- SR reconstruction results from diff. methods of sub-sampled Bunny under 0.2 sub-sampling ratio.



(a) BGFS (b) PDS (c) FPS (d) proposed

[1] C. Dinesh, G. Cheung, I. V. Bajic, "Point Cloud Sampling via Graph Balancing and Gershgorin Disc Alignment," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, January 2021.

Outline

- **What is Graph Signal Processing?**
  - Graph spectrum
  - Graph Fourier transform (GFT), graph Laplacian regularizer (GLR)
- **Graph Learning**
  - Precision / Graph Laplacian Matrix Estimation (w/ eigen-structure constraint)
  - Feature Graph Learning: Gershgorin Disc Perfect Alignment (GDPA)
  - **Application**: Semi-supervised classifier learning

- **Graph Sampling**
  - Gershgorin Disc Alignment Sampling (GDAS)
  - **Application**: Sampling for matrix completion, 3D point cloud sub-sampling

- **Graph Filtering**
  - Signal-dependent GLR, GTV
  - **Application**: Image denoising

GLR for Image Denoising: motivation

- **Graph Laplacian Regularizer (GLR)** $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is a smoothness measure.

- Denoising has simplest formation model $\mathbf{y} = \mathbf{x} + \mathbf{z}$, thus formulation

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu\ \mathbf{x}^T\mathbf{L}\mathbf{x}$$

$$(\mathbf{I} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

- To promote **Piecewise Smoothness** (PWS), $\mathbf{L}(\mathbf{x})$ is ***signal-dependent***:
  - Fix **L** and solve unconstrained QP each iteration.

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu\ \mathbf{x}^T\mathbf{L}(\mathbf{x})\mathbf{x}$$

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.

[2] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” *IEEE ICCV*, 1998.

GLR for Image Denoising: motivation

- **Graph Laplacian Regularizer (GLR)** $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is a smoothness measure.

- Denoising has simplest formation model $\mathbf{y} = \mathbf{x} + \mathbf{z}$, thus formulation

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{x}||_2^2 + \mu \ \mathbf{x}^T\mathbf{L}\mathbf{x}$$

$$(\mathbf{I} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

pixel intensity diff. pixel location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| x_i - x_j \right\|_2^2}{\sigma_1^2} \right) \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma_2^2} \right)$$

Bilateral filter weights

- To promote **Piecewise Smoothness** (PWS), $\mathbf{L}(\mathbf{x})$ is ***signal-dependent***:
  - Fix **L** and solve unconstrained QP each iteration.

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{x}||_2^2 + \mu \ \mathbf{x}^T\mathbf{L}(\mathbf{x})\mathbf{x}$$

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.
[2] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” *IEEE ICCV*, 1998.

GLR for Image Denoising: motivation

- **Graph Laplacian Regularizer (GLR)** $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is a smoothness measure.

- Denoising has simplest formation model $\mathbf{y} = \mathbf{x} + \mathbf{z}$, thus formulation

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \; \mathbf{x}^T\mathbf{L}\mathbf{x}$$

$$(\mathbf{I} + \mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

pixel intensity diff. pixel location diff.

$$w_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2^2}{\sigma_1^2}\right)\exp\left(\frac{-\|l_i - l_j\|_2^2}{\sigma_2^2}\right)$$

Bilateral filter weights

- To promote **Piecewise Smoothness** (PWS), $\mathbf{L}(\mathbf{x})$ is ***signal-dependent***:
  - Fix **L** and solve unconstrained QP each iteration.

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \; \mathbf{x}^T\mathbf{L}(\mathbf{x})\mathbf{x} \longleftarrow \boxed{\text{Signal-dependent GLR}}$$

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.
[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE ICCV*, 1998.

OGLR Denoising Results: visual comparison

- Subjective comparisons ($\sigma_{\mathrm{I}} = 40$)

Original

Noisy, 16.48 dB

K-SVD, 26.84 dB

BM3D, 27.99 dB

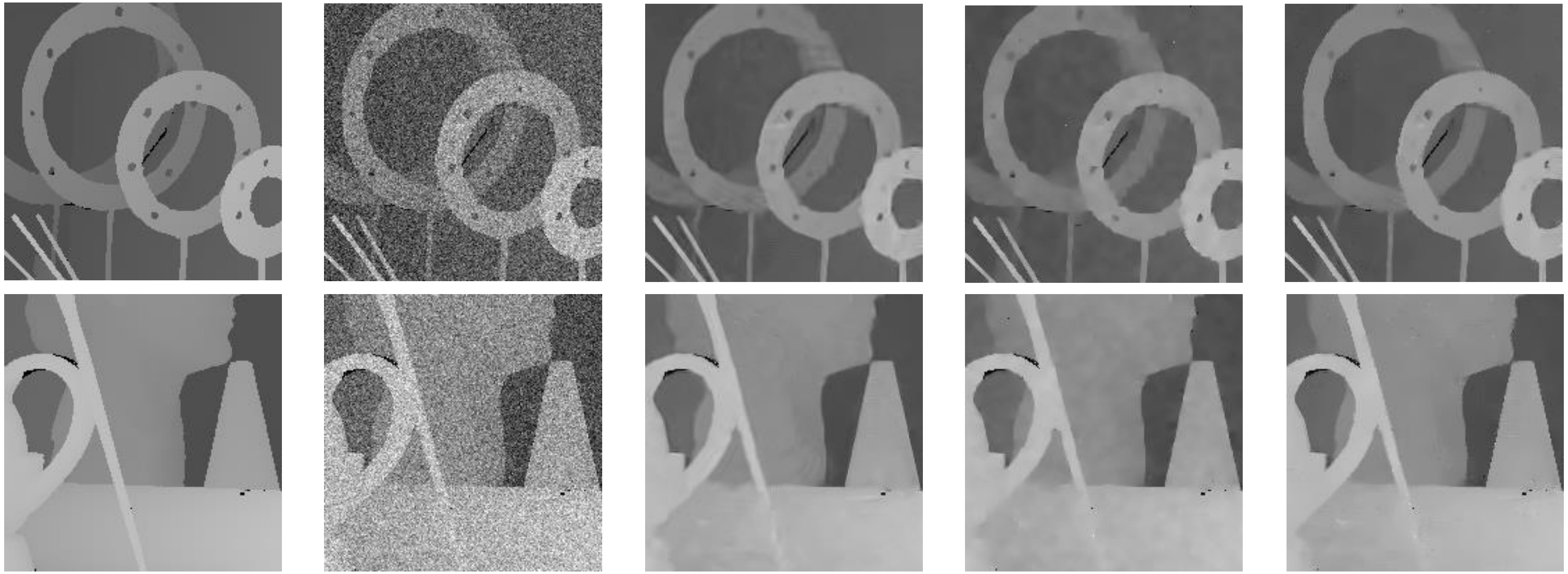
PLOW, 28.11 dB

OGLR, 28.35 dB

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.

OGLR Denoising Results: visual comparison

- Subjective comparisons ($\sigma_I = 30$)



Original Noisy, 18.66 dB BM3D, 33.26 dB NLGBT, 33.41dB OGLR, 34.32 dB

[1] J. Pang, G. Cheung, "Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain," *IEEE TIP*, vol. 26, no.4, pp.1770-1785, April 2017.

Deep GLR: motivation

- Recall MAP formulation of denoising w/ GLR:

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu\, \mathrm{x}^T \mathrm{L}\mathrm{x}$$

fidelity term

smoothness prior

- Solution is system of linear equations:

Sparse PD

$$(\mathbf{I}+\mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

LP graph filter

$$\mathbf{x}^* = (\mathbf{I}+\mu\mathbf{L})^{-1}\mathbf{y}$$

- *Interpretable filter*.

[1] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” *IEEE ICCV*, 1998.

Deep GLR: motivation

- Recall MAP formulation of denoising w/ GLR:

$$\min_{x} \underbrace{\|y - x\|_2^2}_{\text{fidelity term}} + \underbrace{\mu\, x^T L x}_{\text{smoothness prior}}$$

- Solution is system of linear equations:

$$\underbrace{(\mathbf{I} + \mu\mathbf{L})}_{\text{Sparse PD}}\mathbf{x}^* = \mathbf{y} \qquad \mathbf{x}^* = \underbrace{(\mathbf{I} + \mu\mathbf{L})^{-1}}_{\text{LP graph filter}}\mathbf{y}$$

- *Interpretable filter*.

Q: what is the "most appropriate" graph?

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE ICCV*, 1998.

Deep GLR: motivation

- Recall MAP formulation of denoising w/ GLR:

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu\, \mathrm{x}^T \mathrm{L} \mathrm{x}$$

fidelity term

smoothness prior

- Solution is system of linear equations:

Sparse PD

LP graph filter

$$(\mathbf{I}+\mu\mathbf{L})\mathbf{x}^* = \mathbf{y} \qquad \mathbf{x}^* = (\mathbf{I}+\mu\mathbf{L})^{-1}\mathbf{y}$$

- *Interpretable filter*.

Bilateral weights:

$$w_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2^2}{\sigma_1^2}\right)\exp\left(\frac{-\|l_i - l_j\|_2^2}{\sigma_2^2}\right)$$

Q: what is the "most appropriate" graph?

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE ICCV*, 1998.

Deep GLR: unrolling

- **Deep GLR**:
  1. Learn features **f**'s using CNN.
  2. Compute distance from features.
  3. Compute edge weights using Gaussian kernel.
  4. Construct graph, solve QP.

$$w_{ij} = \exp\left(-\frac{\text{dist}(i,j)}{2\epsilon^2}\right),$$

$$\text{dist}(i,j) = \sum_{n=1}^{N} \left(\mathbf{f}_n(i) - \mathbf{f}_n(j)\right)^2.$$

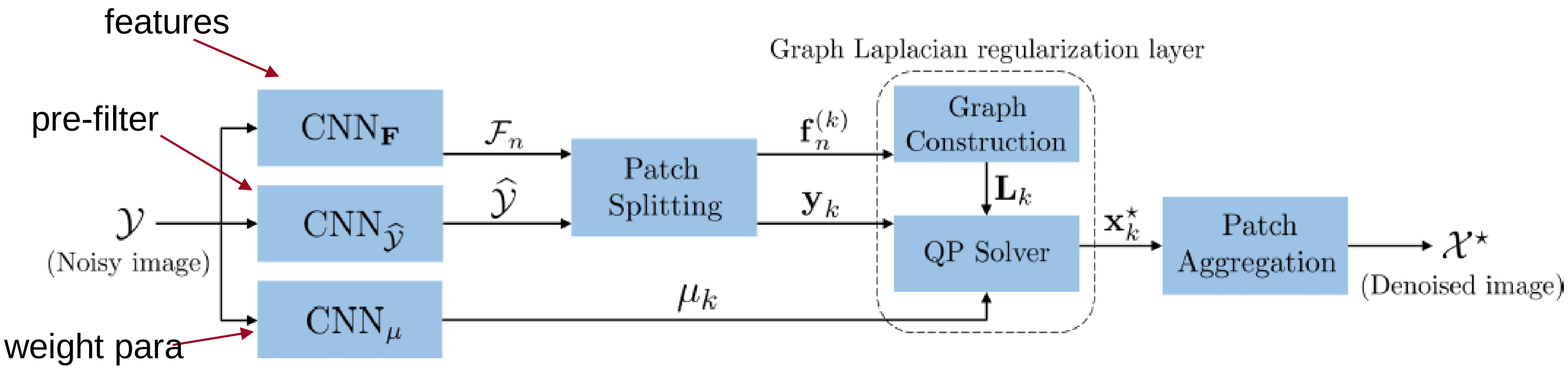


Fig. 1. Block diagram of the proposed GLRNet which employs a graph Laplacian regularization layer for image denoising.

[1] K. Gregor and Y. LeCun, "**Learning fast approximations of sparse coding,**" in *Proc. 27th Int. Conf. Machine Learning*, 2010..

Deep GLR: CNN implementation

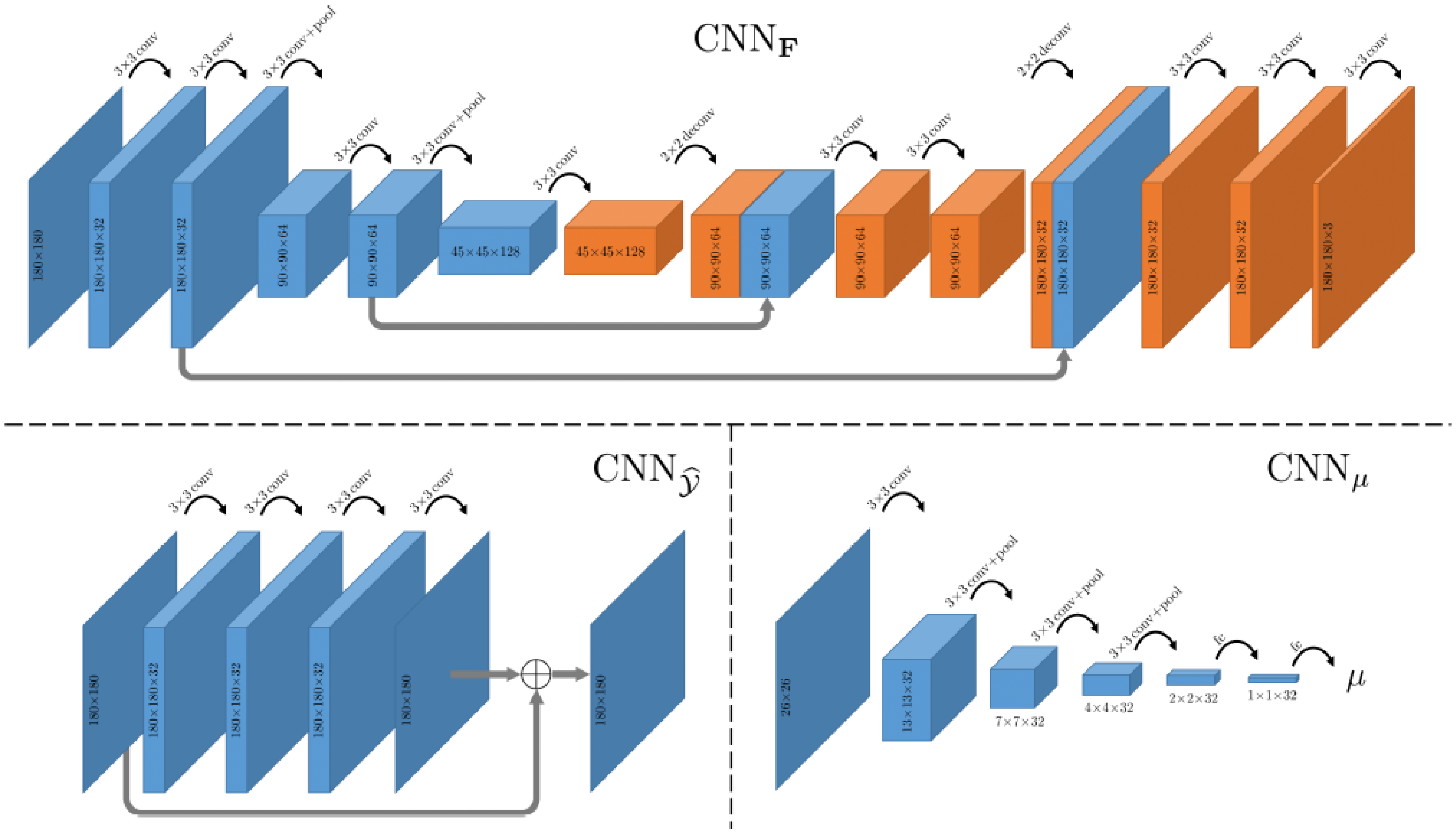


Fig. 3. Network architectures of CNN$_{\mathbf{F}}$, CNN$_{\hat{\mathcal{Y}}}$ and CNN$_{\mu}$ in the experiments. Data produced by the decoder of CNN$_{\mathbf{F}}$ is colored in orange.

[1] J. Zeng et al., "Deep Graph Laplacian Regularization for Robust Denoising of Images," *NTIRE Workshop, CVPR 2019*.

Deep GLR: unrolling

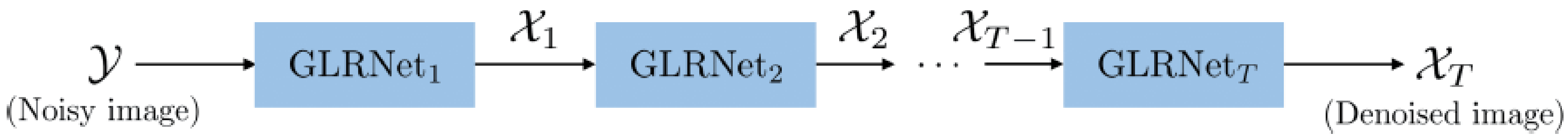


Fig. 2. Block diagram of the overall DeepGLR framework.

- Model *guarantees numerical stability* of solution:

$$(\mathrm{I} + \mu \mathrm{L})\, \mathrm{x}^* = \mathrm{y}$$

- **Thm 1**: *condition number* κ of matrix satisfies [1]:

maximum node degree

$$\kappa \leq 1 + 2\, \mu\, d_{\max},$$

- **Observation**: Restricting CNN search space → achieve robust learning.

[1] J. Zeng et al., "Deep Graph Laplacian Regularization for Robust Denoising of Images," *NTIRE Workshop, CVPR 2019*.

Deep GLR: numerical comparison

- Trained on *AWGN* on 5 images, patches of size 26-by-26.
- Batch size is 4, model is trained for 200 epochs.
- Trained for both known and blind noise variance.

Table 3. Average PSNR (dB) and SSIM values for Gaussian noise removal.

Noise	Method (PSNR/SSIM)		
	CBM3D	CDnCNN	DeepGLR
15	33.49/ 0.9216	33.80/ 0.9268	33.65/ 0.9259
25	30.68/ 0.8675	31.13/ 0.8799	31.03/ 0.8797
50	27.35/ 0.7627	27.91/ 0.7886	27.86/ 0.7924

[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Deep GLR: numerical comparison

- **Cross-domain generalization**.
- Trained on *Gaussian noise*, tested *on low-light images* in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- Outperformed DnCNN by 5.74 dB, and noise clinic by 1.87 dB.

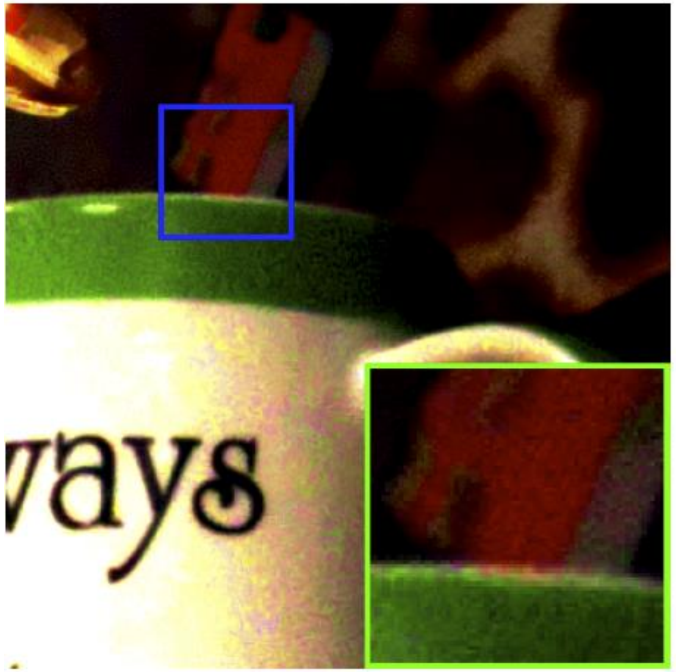
Table 4. Evaluation of cross-domain generalization for real image denoising. The best results are highlighted in boldface.

Metric	Noisy	Method		
		Noise Clinic	CDnCNN	DeepGLR
PSNR	20.36	27.43	24.36	**30.10**
SSIM	0.1823	0.6040	0.5206	**0.8028**

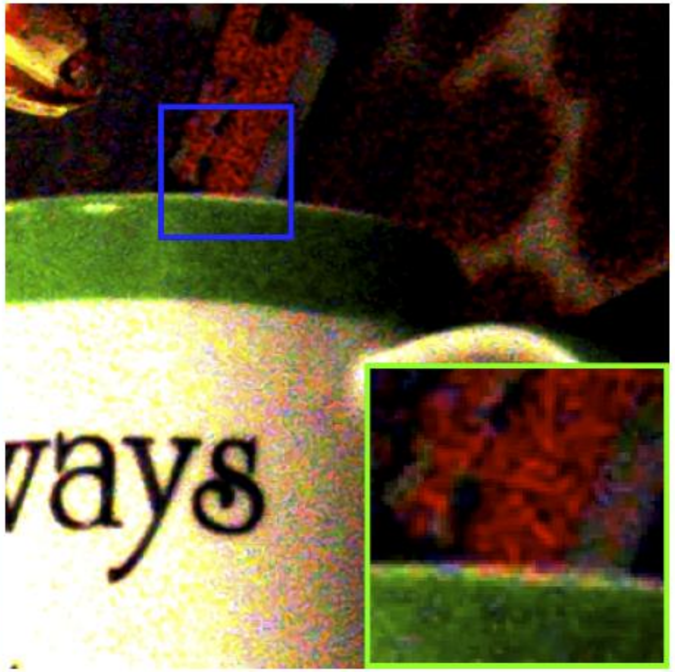
[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Deep GLR: visual comparison

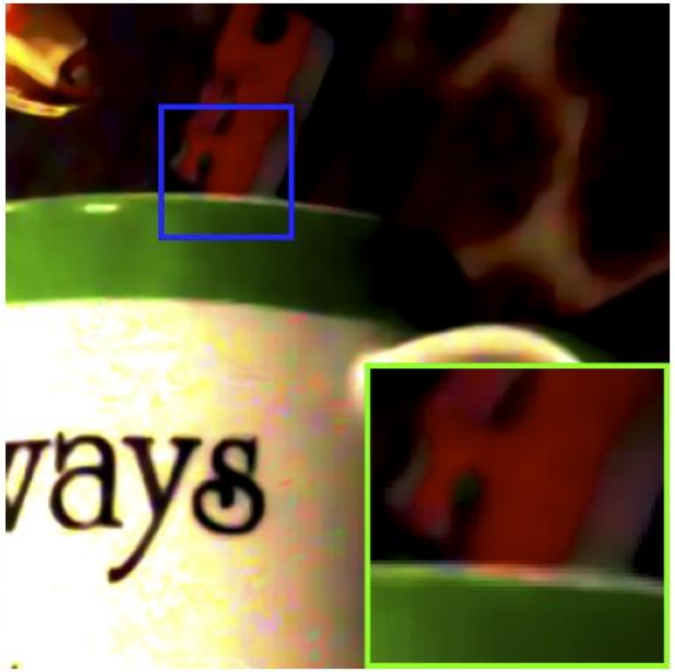
- Trained on *Gaussian noise*, tested *on low-light images* in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- Outperformed DnCNN by 5.74 dB, and noise clinic by 1.87 dB.



Noise Clinic



CDnCNN



DeepGLR

[1] Kai Zhang et al, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” *TIP* 2017.
[2] Marc Lebrun et al, “The noise clinic: a blind image denoising algorithm,” *IPOL* 2015.

Deep GTV: motivation

- **GTV** promotes PWS faster than **GLR**.

$$\min_x \|\mathbf{y}-\mathbf{x}\|_2^2 + \mu \|\mathbf{x}\|_{GTV} \qquad \|\mathbf{x}\|_{GTV} = \sum_{i,j} w_{i,j} |x_i - x_j|$$

- Solve as QP via **L_1-Laplacian**: $\Gamma_{i,j} = \frac{w_{i,j}}{\max\{|x_i - x_j|, \epsilon\}}$

$$\min_x \|\mathbf{y}-\mathbf{x}\|_2^2 + \mu \; \mathbf{x}^T \mathbf{L}_\Gamma \mathbf{x} \qquad \mathbf{x}^* = (\mathbf{I} + \mu \, \mathbf{L}_\Gamma)^{-1} \mathbf{y}$$

- Still *interpretable LP graph filter*.

[1] Y. Bai, G. Cheung, X. Liu, W. Gao, "Graph-Based Blind Image Deblurring from a Single Photograph," *IEEE TIP*, vol. 28, no.3, pp.1404-1418, March 2019.
[2] H. Vu, G. Cheung, Y. C. Eldar, "Unrolling of Deep Graph Total Variation for Image Denoising," accepted to *IEEE ICASSP*, Toronto, Canada, June 2021.

Deep GTV: algorithm

- Learn feature via CNN for **graph construction**.
- Obtain **graph filter response**:

$$\mathbf{x}^* = (\mathbf{I} + \mu \mathbf{L}_\Gamma)^{-1}\mathbf{y} = \mathbf{U}\mathrm{diag}(1+\mu\lambda_1, \dots, 1+\mu\lambda_N)^{-1}\mathbf{U}^T\mathbf{y}$$

- Fast filter implementation via **Lanczos approx.**:
  1. Compute tri-diagonal matrix $H_M \in \mathbb{R}^{M \times M}$
  2. Compute approx. filter:

$$g(\mathbf{L})\mathbf{y} \approx \|\mathbf{y}\|_2 \mathbf{V}_M g(\mathbf{H}_M)\mathbf{e}_1$$

where $g(\mathcal{L}) := U g(\Lambda) U^*$

$$V_M^* \mathcal{L} V_M = H_M = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_M \\ & & & \beta_M & \alpha_M \end{bmatrix}$$

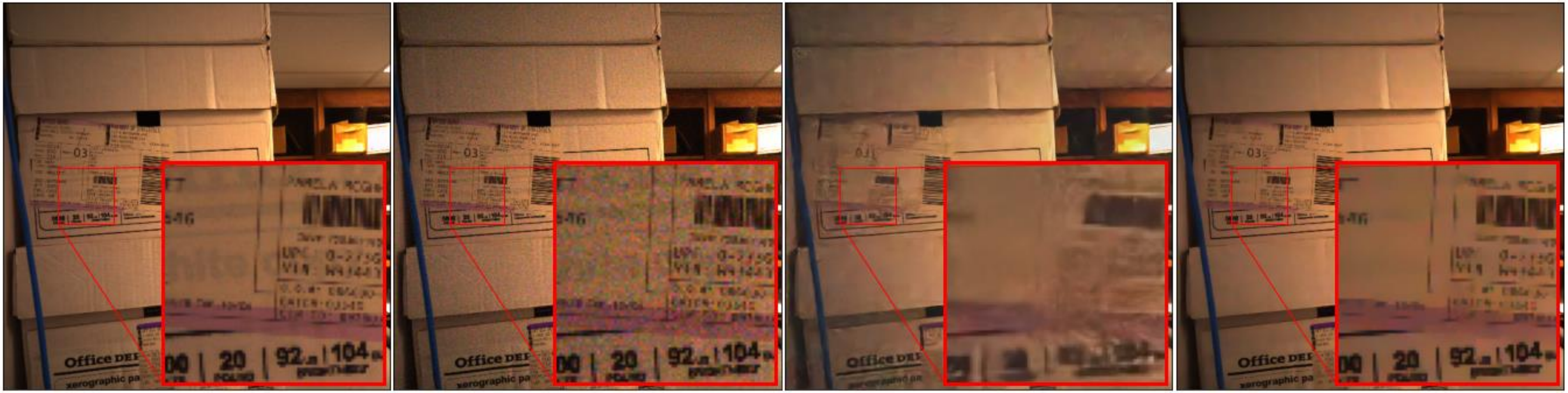
- *Interpretable graph filter → fast implementation*.

[1] J. Zeng et al., "Deep Graph Laplacian Regularization for Robust Denoising of Images," *NTIRE Workshop, CVPR 2019*.

[2] A. Susnjara, N. Perraudin, D. Kressner1, and P. Vandergheynst, "Accelerated filtering on graphs using Lanczos method," in unpublished, arXiv:1509.04537, 2015.

Deep GTV: experimental comparison

- Train on Gaussian (σ=50) and test on captured noise



(a) ground-truth (b) noisy (PSNR: 23.56) (c) CDnCNN-S (PSNR: 26.83) (d) DeepGTV (PSNR: 28.82)

	DnCNN-S	DeepAGF	DeepGTV
# Parameters	0.55M	0.32M	0.12M

Table 3: Number of trainable parameters

save ≥ 80% parameters!

Gene Cheung (genec@yorku.ca)

Conclusion

- Graph is flexible abstraction to convey pairwise similarities.
  - Similarity defined as correlation or feature distance.
  - Graph frequencies contains global notions.
  - Graph is an expression of domain knowledge.
- GSP leverages on mature understanding in SP and linear algebra.
- GSP tools are excellent for building hybrid model-based / data-driven systems.

Applications:
Image coding, denoising, deblurring, interpolation, contrast enhancement, light field image coding, 3D point cloud denoising, enhancement, sub-sampling, super-resolution, inpainting, matrix completion, semi-supervised classifier learning, video summarization

[1] X. Dong*, D. Thanou*, L. Toni, M. Bronstein, P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” *IEEE Signal Processing Magazine*, vol.37, no.6, pp.117-127, Nov., 2020.

Contact Info

- **Homepage**:

https://www.eecs.yorku.ca/~genec/index.html

- **E-mail**:

genec@yorku.ca

- **Forthcoming book**:

G. Cheung, E. Magli, (edited) ***Graph Spectral Image Processing***, ISTE/Wiley, June 2021.