Gene Cheung
Associate Professor, York University
16th October, 2019

YORK UNIVERSITÉ UNIVERSITY

Graph Signal Analysis: Imaging, Learning, Sampling

.

Acknowledgement

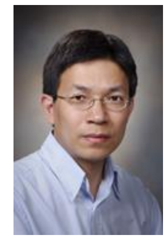
Collaborators:

- X. Liu (HIT, China)
- **W. Hu**, W. Gao (Peking U., China)
- L. Fang (Tsinghua, China)
- **C.-W. Lin** (National Tsing Hua University, Taiwan)
- **A. Ortega** (USC, USA)
- D. Florencio (MSR, USA)
- J. Liang, **I. Bajic** (SFU, Canada)
- X. Wu (McMaster U, Canada)
- P. Frossard (EPFL, Switzerland)
- **V. Stankovic** (U of Strathclyde, UK)
- **Y. Nakatsukasa** (Oxford, UK)
- **P. Le Callet** (U of Nantes, France)

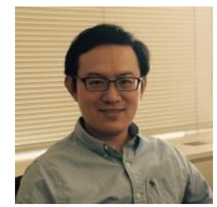


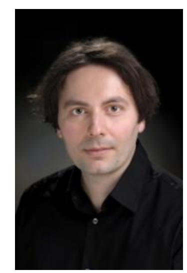






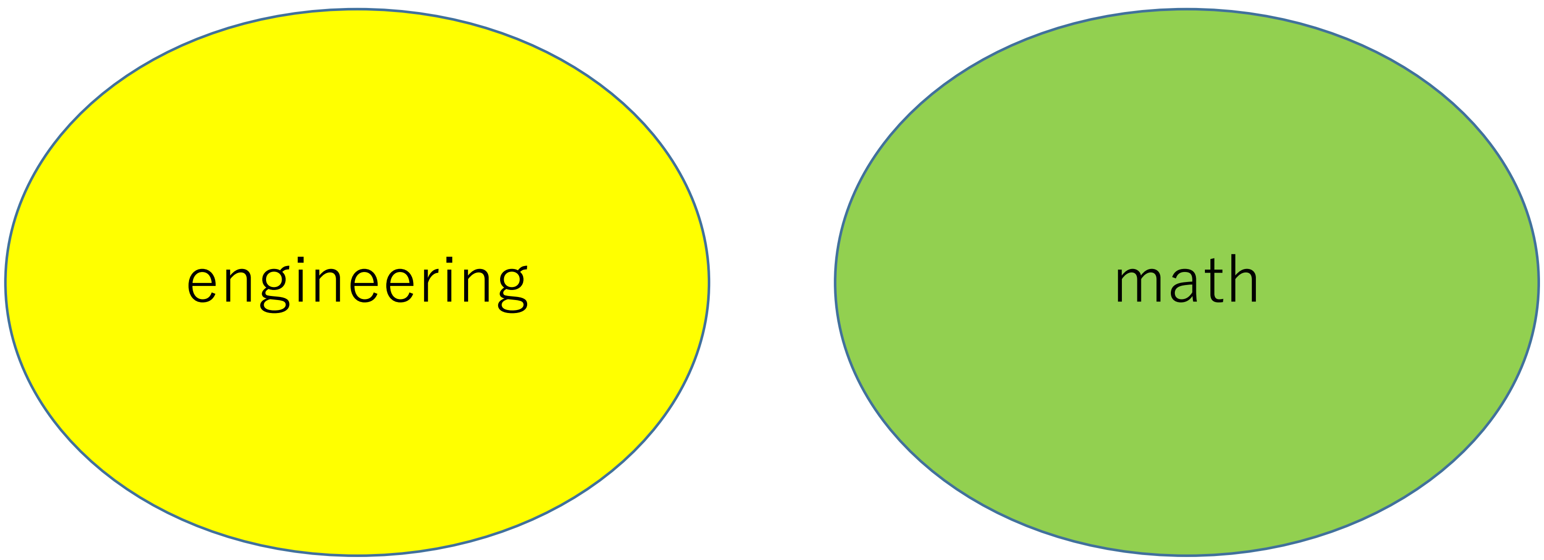






Introducing math tools

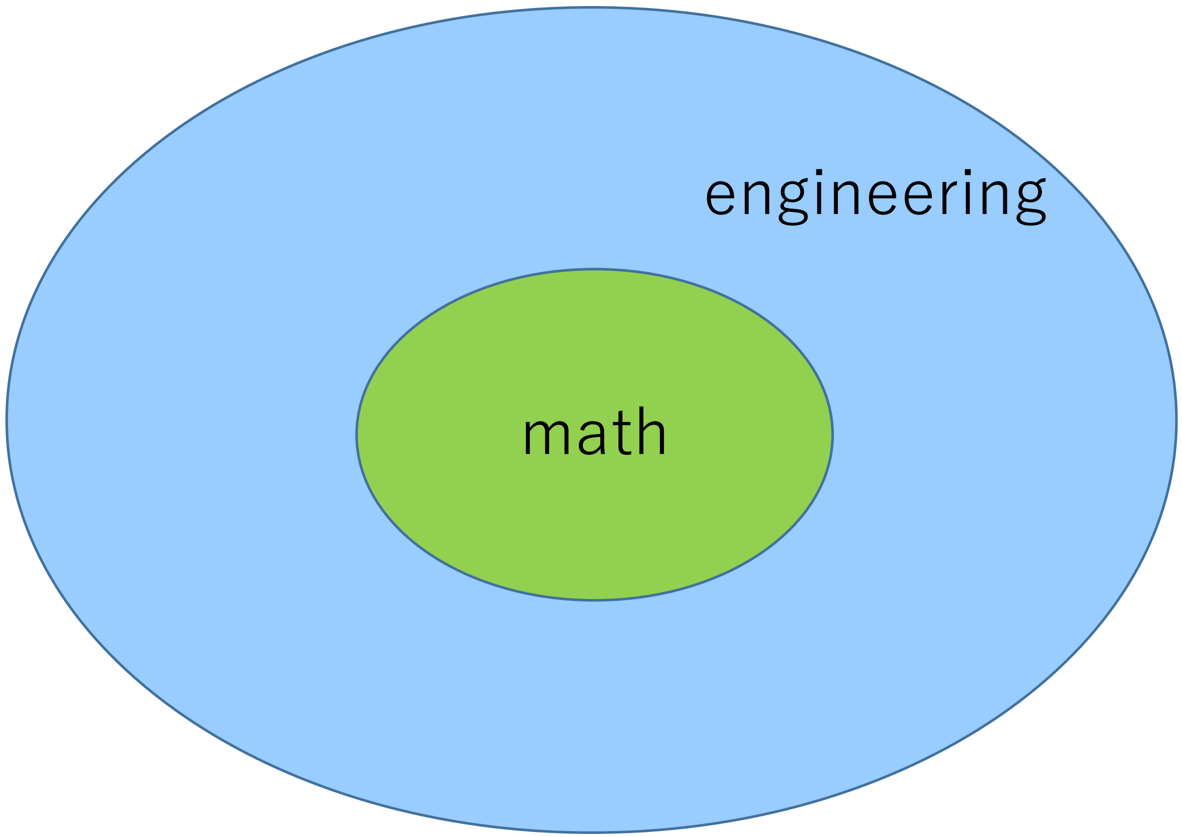
Students in EECS4452: “This is math, not engineering!”



Introducing math tools

Students in EECS4452: “This is math, not engineering!”

Me: “Math is the heart of engineering!”



Outline

- Defining Graph frequencies

- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution

- Deep GLR

- Semi-Supervised Learning

- Graph Sampling
  - Matrix completion

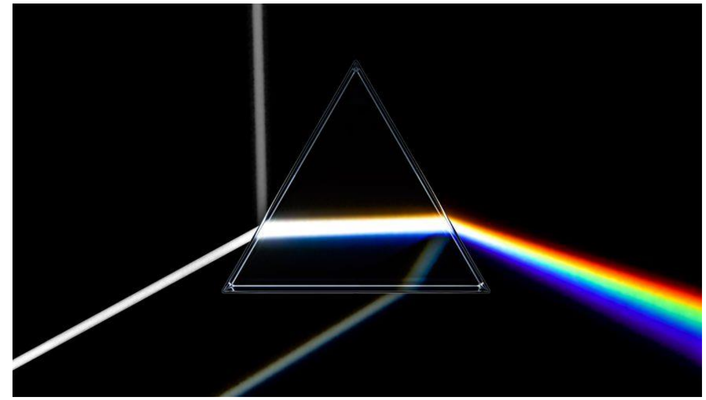
Outline

- Defining Graph frequencies
- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution
- Deep GLR
- Semi-Supervised Learning
- Graph Sampling
  - Matrix completion

Signal Decomposition

- Decompose signal into basic components:

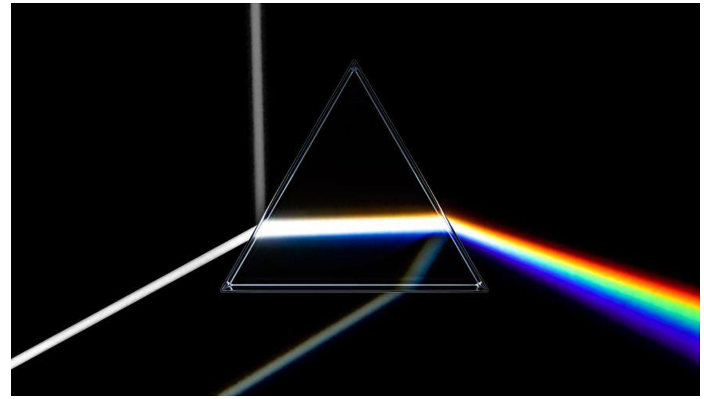
$$\mathrm{x} = \sum_{k \in \mathrm{Z}} X_k \varphi_k$$



  - Newton decomposed white light into color components (1730).

Signal Decomposition

- Decompose signal into basic components:



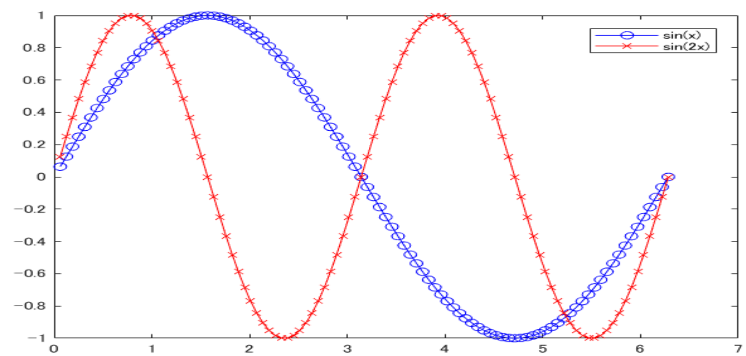
$$x = \sum_{k \in Z} X_k \varphi_k$$

  - Newton decomposed white light into color components (1730).

- “Basic” components can be complex exponentials:

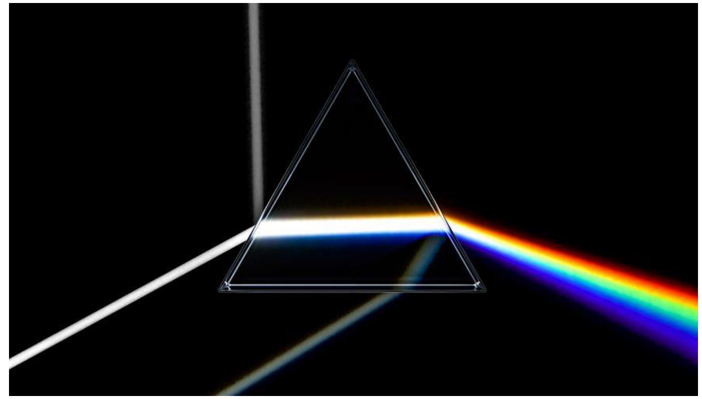
$$x = \sum_{k \in Z} X_k e^{j 2\pi k t}$$

$$X_k = \int x(t) e^{-j 2\pi k t} dt$$



Signal Decomposition

- Decompose signal into basic components:



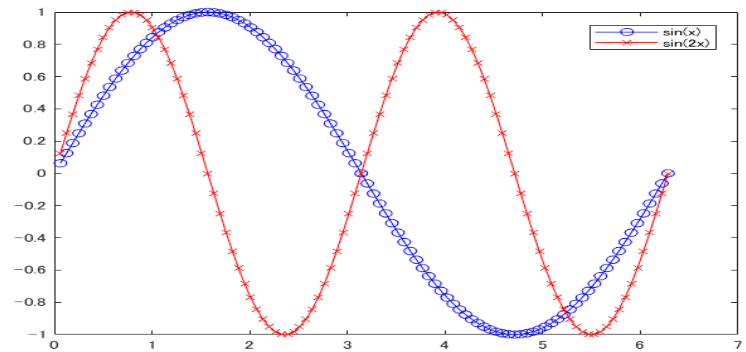
$$\mathrm{x} = \sum_{k \in Z} X_k \varphi_k$$

  - Newton decomposed white light into color components (1730).

- "Basic" components can be complex exponentials:

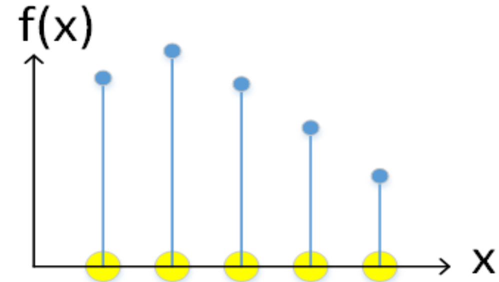
$$x = \sum_{k \in Z} X_k e^{j2\pi kt}$$

$$X_k = \int x(t) e^{-j2\pi kt} dt$$

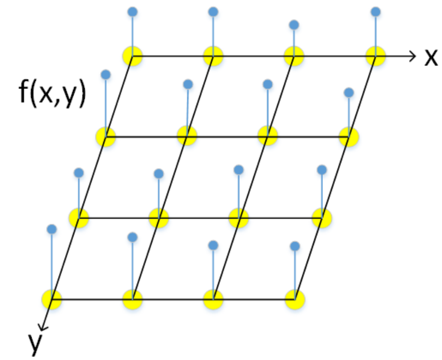


- Complex exponentials are eigenfunctions of 2nd derivative operator.

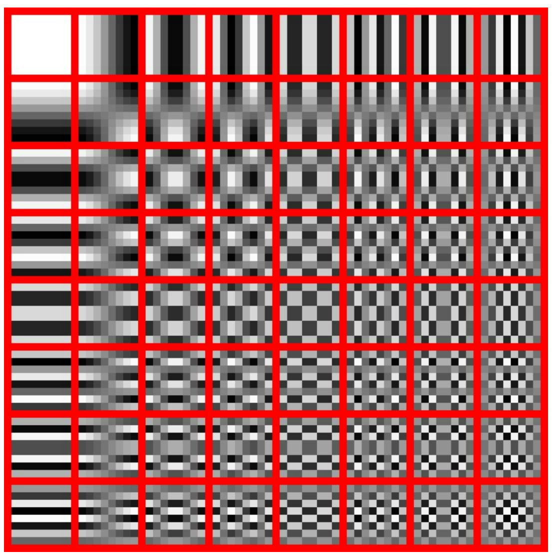
Digital Signal Processing



- 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

$$a = \Phi\, x$$

desired signal

transform

sparse
transform coeff.

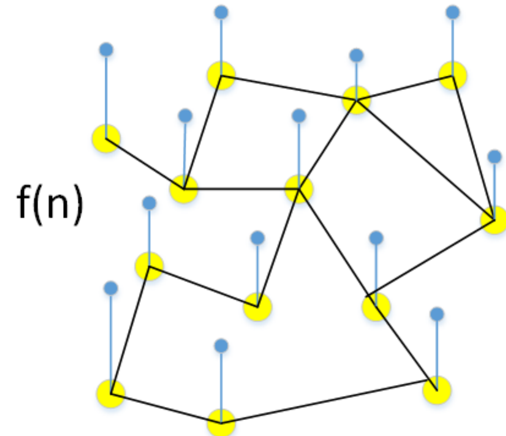
Graph Signal Processing

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

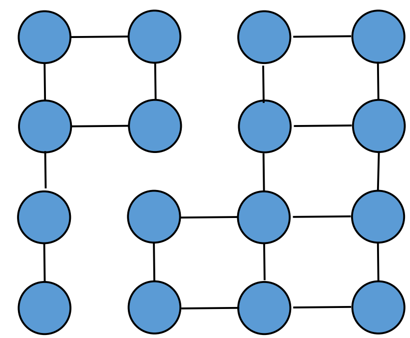
1. Harmonic Analysis of graph signals.

2. Embed pairwise similarity info into graph.
   - **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



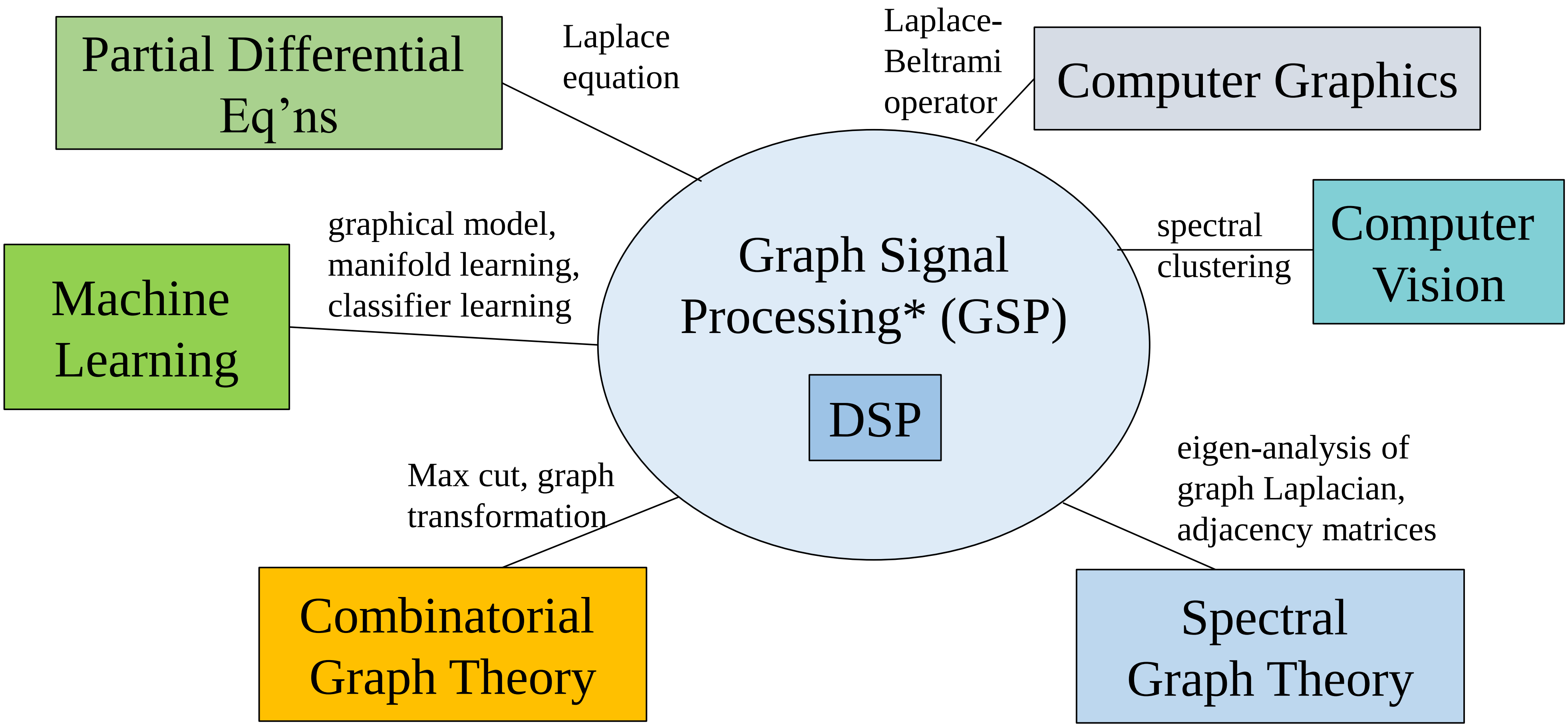
signal on graph kernel

[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.

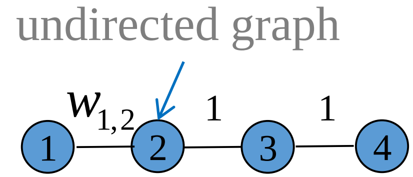
[2] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "**Graph spectral image processing**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.

GSP and Graph-related Research

GSP: SP framework that unifies concepts from multiple fields.



Graph Fourier Transform (GFT)



Graph Laplacian:

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

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

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

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

$$D_{i,i} = \sum_j A_{i,j}$$

- *Combinatorial Graph Laplacian* **L**: **L = D-A**
  - **L** is related to *2nd derivative*.
  - **L** is a differential operator on graph.

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

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

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

*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

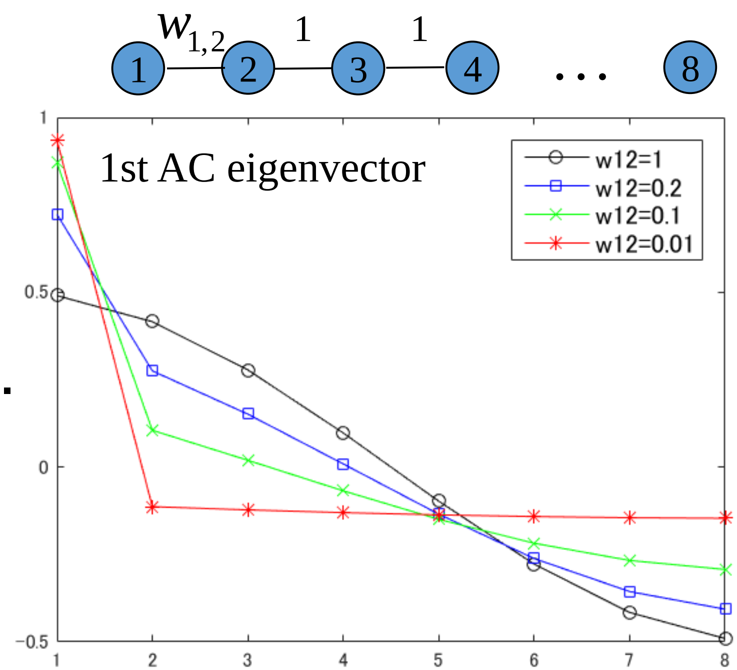
$$\mathrm{L} = \mathrm{V}\,\Sigma\,\mathrm{V}^T$$

GFT

eigenvectors in columns

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

GFT coefficients



1. Eigenvectors aggregates info from weights.
   - Constant 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.

Graph Spectrum from GFT

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

eigenvalues along diagonal

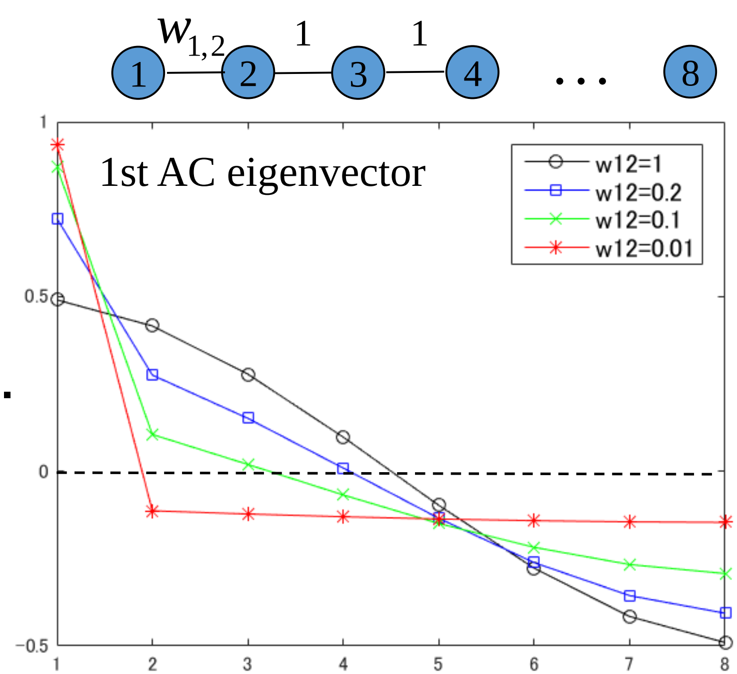
$$L = V \Sigma V^T$$

GFT

eigenvectors in columns

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

GFT coefficients



1. Eigenvectors aggregates info from weights.
   - Constant 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.

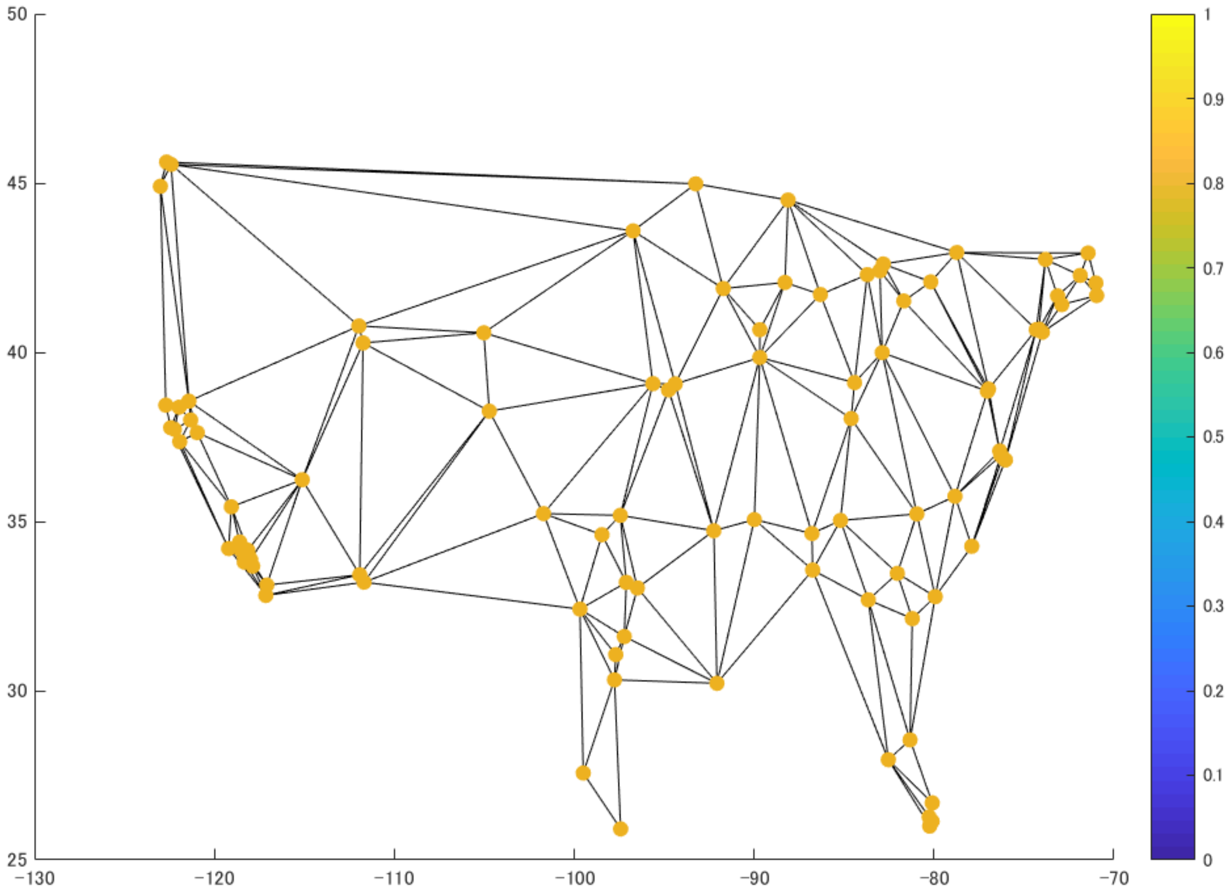
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.
- Edge weights inverse proportion 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

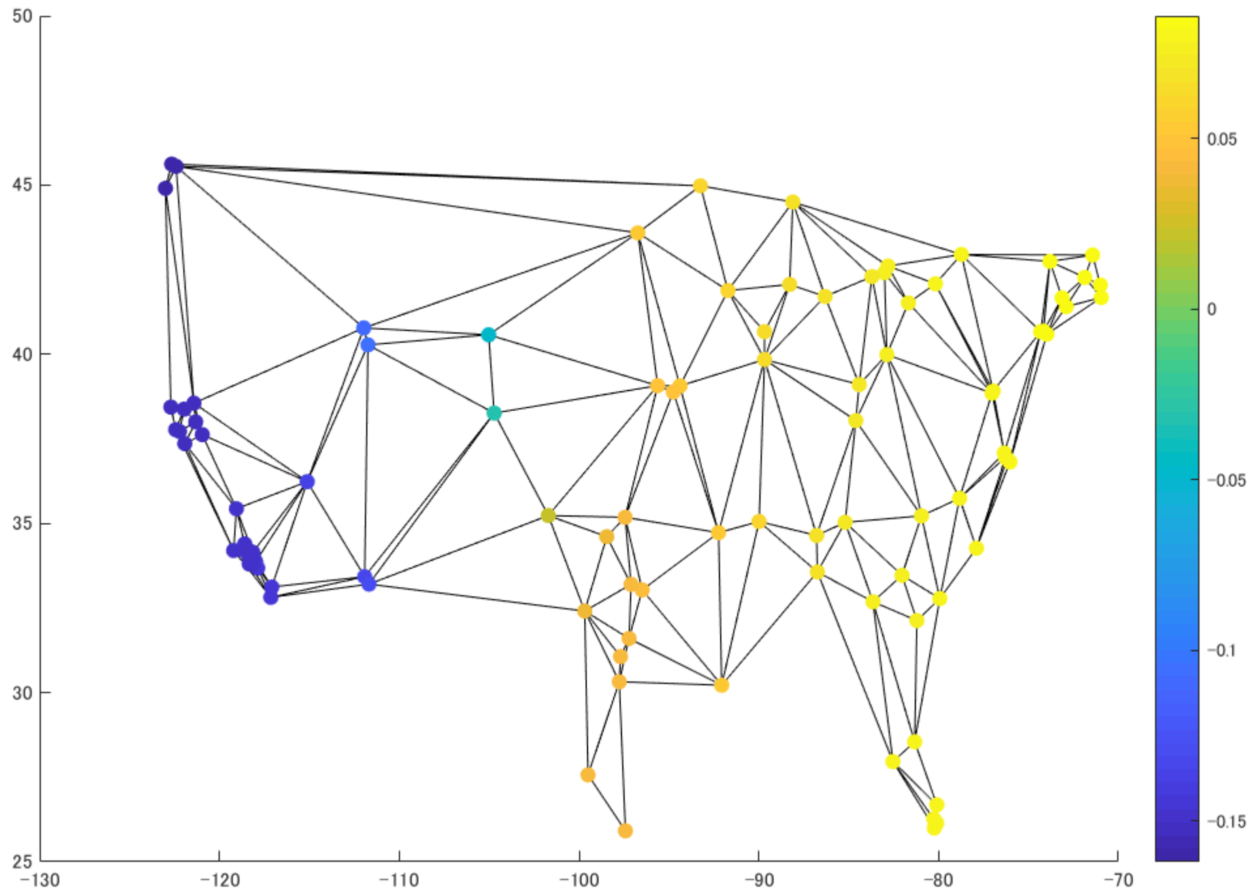
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

location diff.

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

Edge weights



V2: 1st AC component

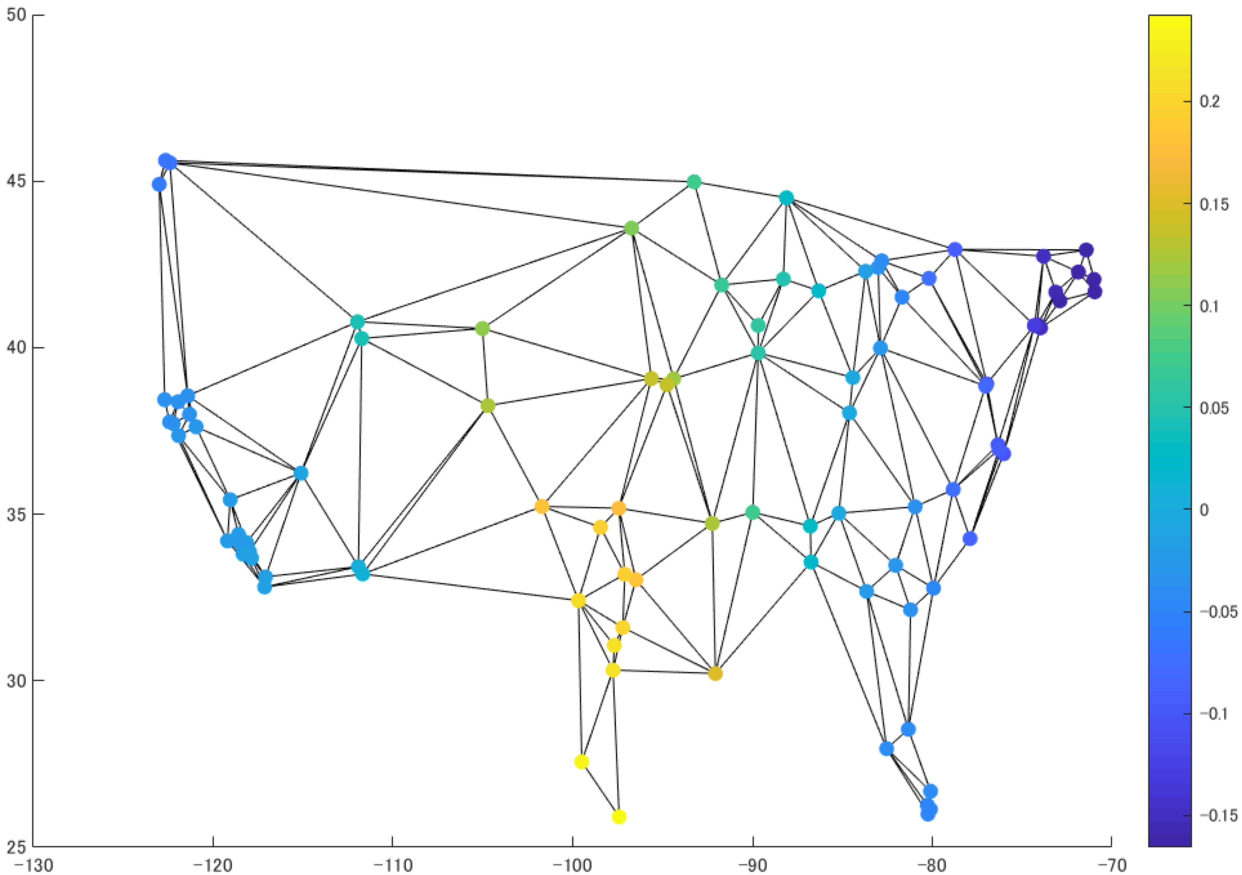
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

location diff.

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

Edge weights



V3: 2nd AC component

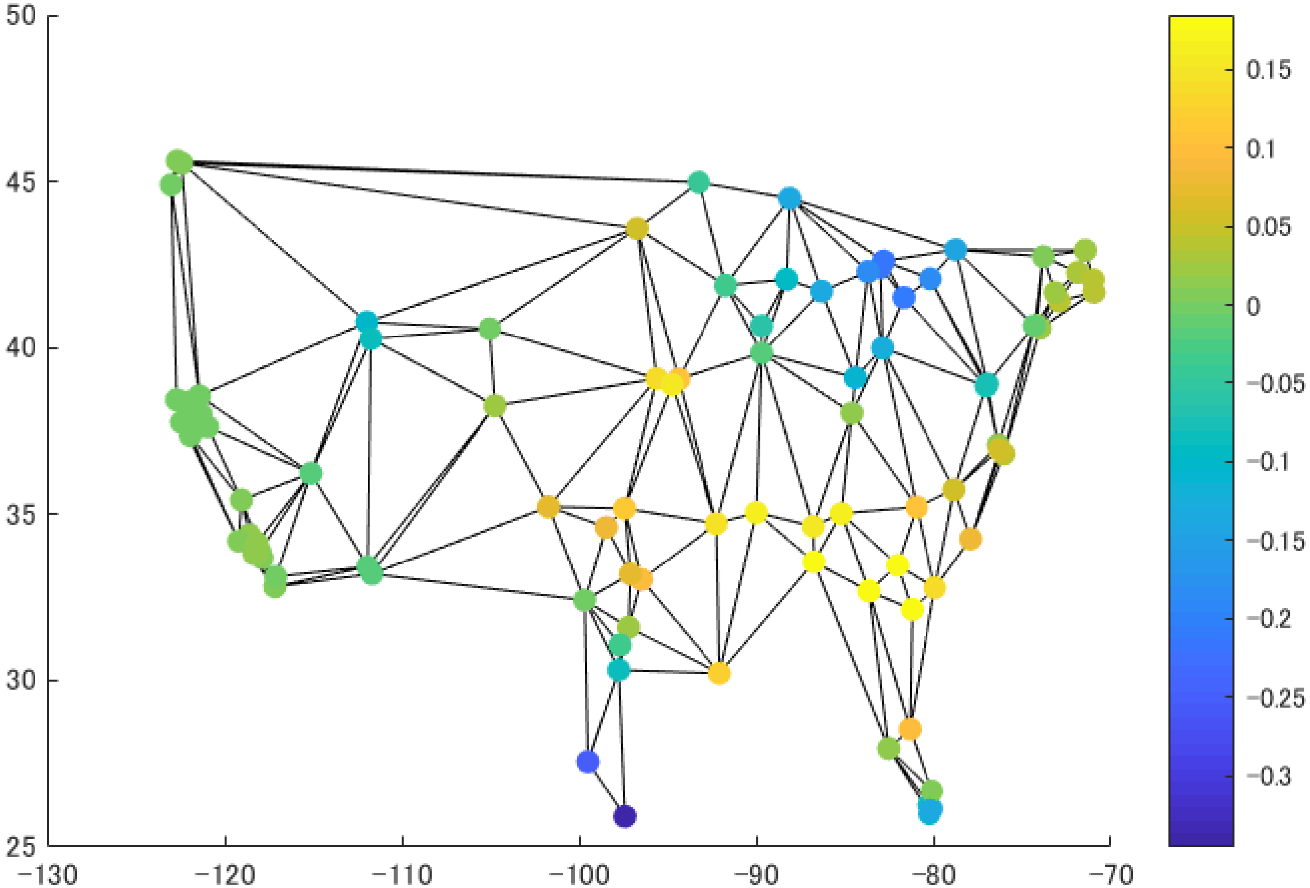
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

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

Outline

- Defining Graph frequencies
- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution
- Deep GLR
- Semi-Supervised Learning
- Graph Sampling
  - Matrix completion

Graph Laplacian Regularizer

- $\mathbf{x}^T \mathbf{L} \mathbf{x}$ (graph Laplacian regularizer) [1]) is one smoothness measure.

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

signal smooth in nodal domain

signal contains mostly low graph freq.

- **Signal Denoising**:

observation

desired signal

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

noise

- **MAP Formulation**:

fidelity term

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

smoothness prior

update edge weights

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

linear system of eqn's w/ sparse, symmetric PD matrix

[1] P. Milanfar, "**A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical**," *IEEE Signal Processing Magazine*, vol.30, no.1, pp.106-128, January 2013.

Graph Laplacian Regularizer

- $\mathbf{x}^T \mathbf{L}\, \mathbf{x}$ (graph Laplacian regularizer) [1]) is one smoothness measure.

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

signal smooth in nodal domain

signal contains mostly low graph freq.

- **Signal Denoising**:

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

- **MAP Formulation**:

$$\min_x \left\| \mathbf{y} - \mathbf{x} \right\|_2^2 + \mu\, \mathbf{x}^T \mathbf{L} \mathbf{x}$$

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

update edge weights

linear system of eqn's w/ sparse, symmetric PD matrix

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

[1] P. Milanfar, "**A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical**," *IEEE Signal Processing Magazine*, vol.30, no.1, pp.106-128, January 2013.

Results: natural image denoising

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

Original

Noisy, 16.48 dB

K-SVD, 26.84 dB

BM3D, 27.99 dB

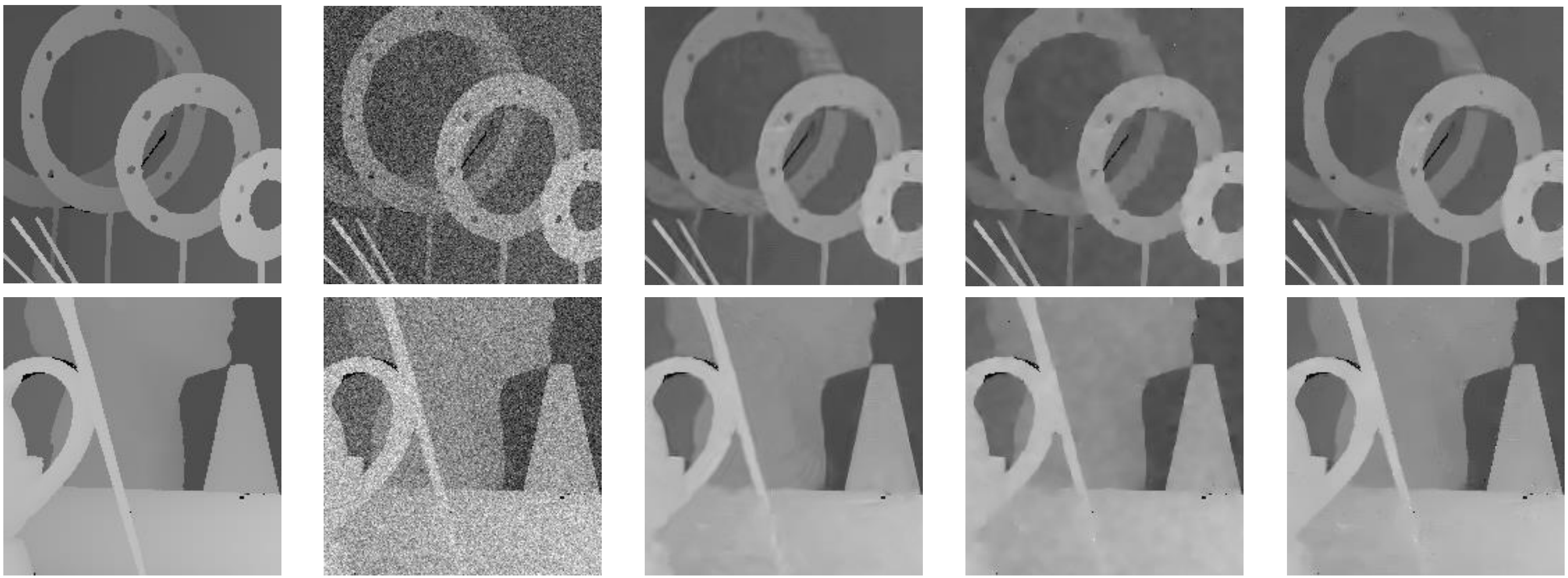
PLOW, 28.11 dB

OGLR, 28.35 dB

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

Results: depth image denoising

- Subjective comparisons ($\sigma_{\mathrm{I}} = 30$)



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

[1] W. Hu et al., "**Depth Map Denoising using Graph-based Transform and Group Sparsity**," *IEEE International Workshop on Multimedia Signal Processing*, Pula (Sardinia), Italy, October, 2013.

GLR for Joint Dequantization / Contrast Enhancement

- Retinex decomposition model:

$$\mathbf{y} = \underbrace{\tau}_{\text{scalar}}\ \underbrace{\mathbf{l}}_{\text{illumination}} \odot \overbrace{\mathbf{r}}^{\text{reflectance}} + \underbrace{\mathbf{z}}_{\text{noise}}$$

- **Objective**: general smoothness for luminance, smoothness w/ negative edges for reflectance.

$$\begin{aligned} \min_{\mathbf{l},\mathbf{r}} \quad & \mathbf{l}^\top \overbrace{\left(\mathbf{L}_l + \alpha \mathbf{L}_l^2\right)}^{\text{generalized smooth}} \mathbf{l} + \mu\, \mathbf{r}^\top \overbrace{\mathcal{L}_r}^{\text{piecewise smooth}} \mathbf{r} \\ \text{s.t.} \quad & \left(\mathbf{q} - \tfrac{1}{2}\right) \mathbf{Q} \preceq \mathbf{T} \tau\, \mathbf{l} \odot \mathbf{r} \prec \left(\mathbf{q} + \tfrac{1}{2}\right) \mathbf{Q} \end{aligned}$$

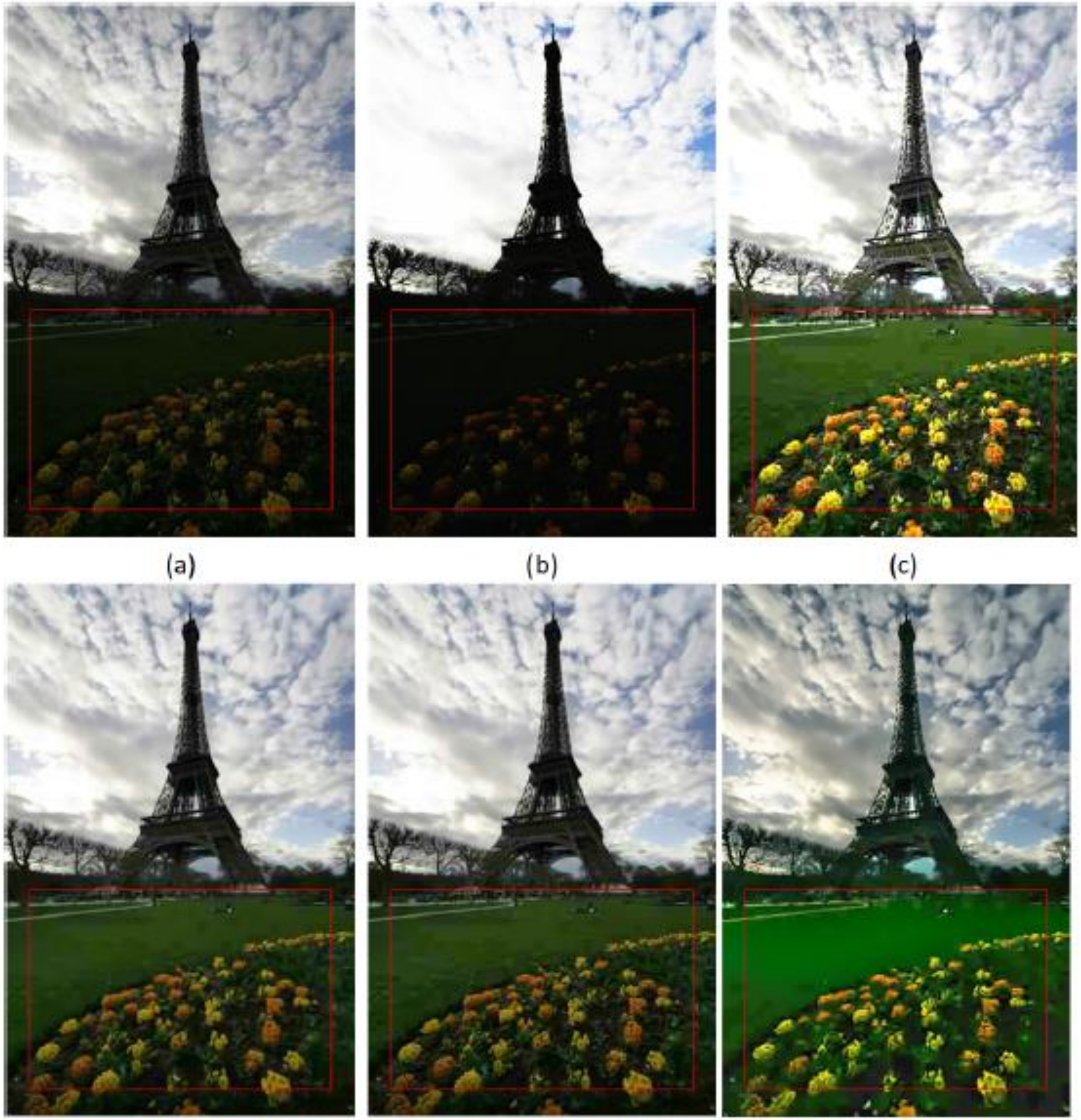
- **Constraints:** quantization bin constraints

- **Solution**: Alternating accelerated proximal gradient alg [1].

[1] X. Liu, G. Cheung, X. Ji, D. Zhao, W. Gao, "**Graph-based Joint Dequantization and Contrast Enhancement of Poorly Lit JPEG Images**," *IEEE Transactions on Image Processing*, vol. 28, no.3, pp.1205-1219, March 2019.

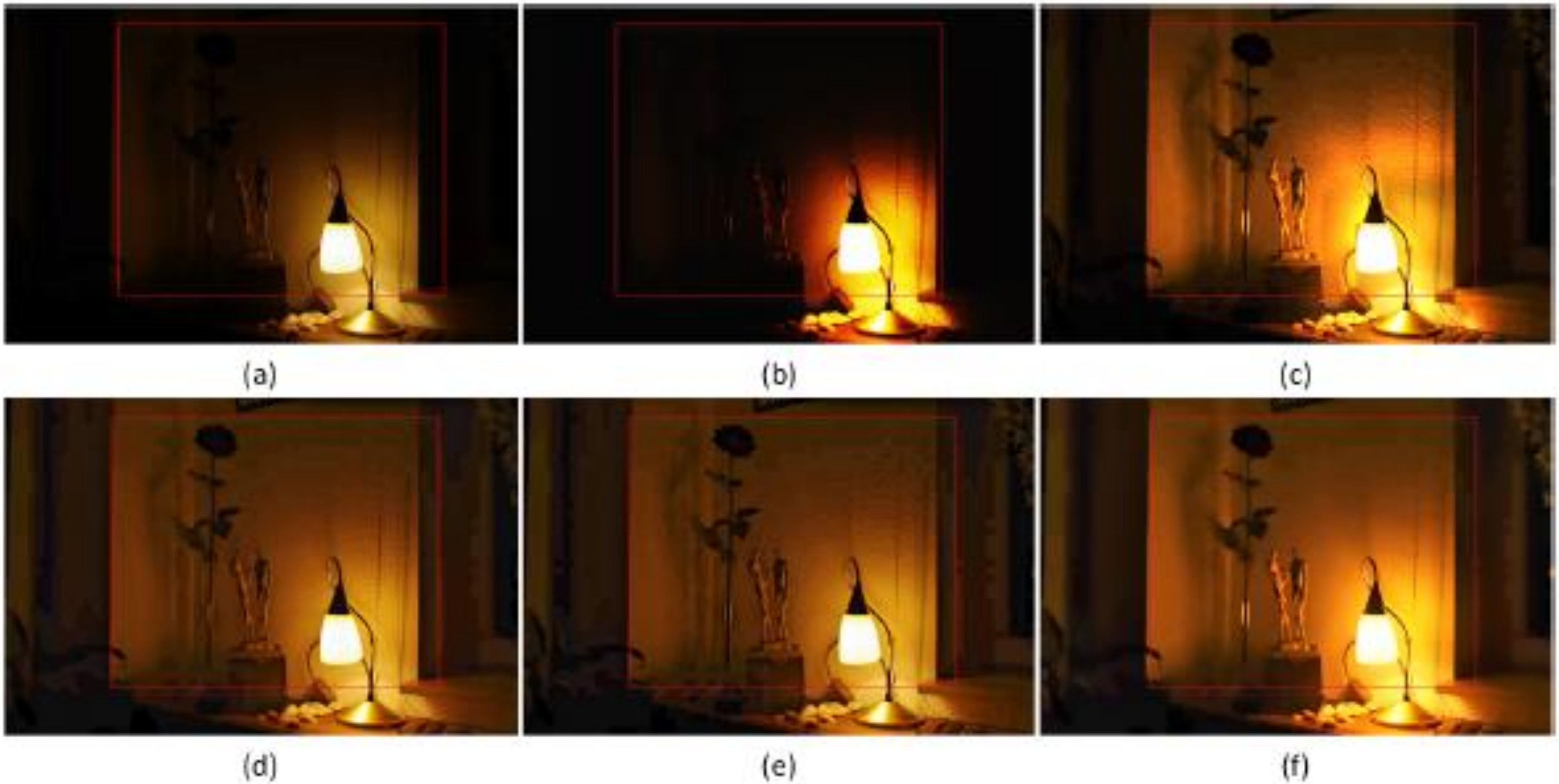
Results: Contrast Enhancement

Results: Contrast Enhancement



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

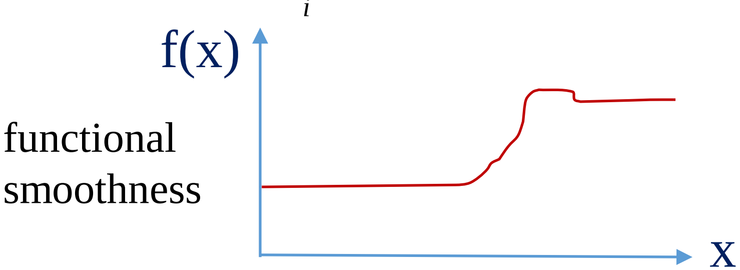
Results: Contrast Enhancement



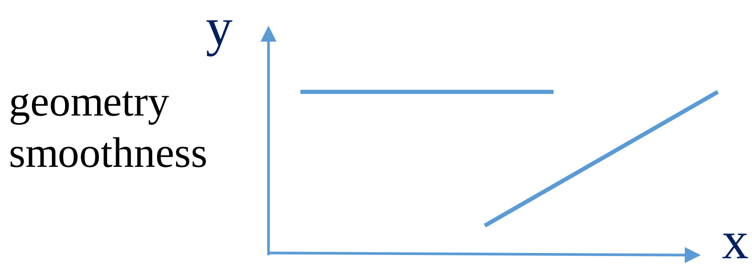
GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

$$\sum_i |f(x_i) - f(x_{i-1})|$$



$$\sum_i |y_i - y_{i-1}|$$



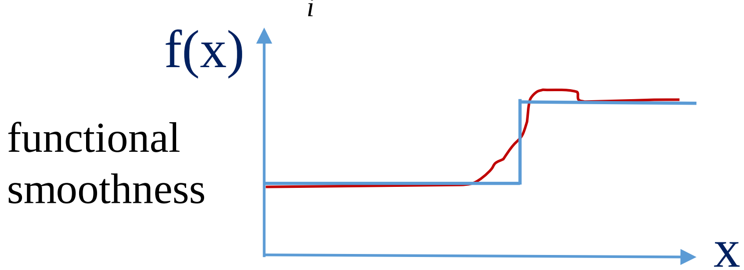
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

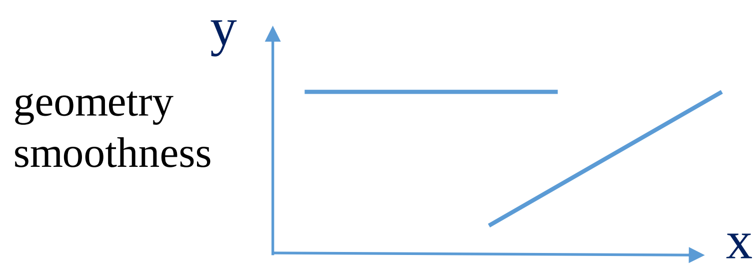
GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

$$\sum_i |f(x_i) - f(x_{i-1})|$$



$$\sum_i |y_i - y_{i-1}|$$



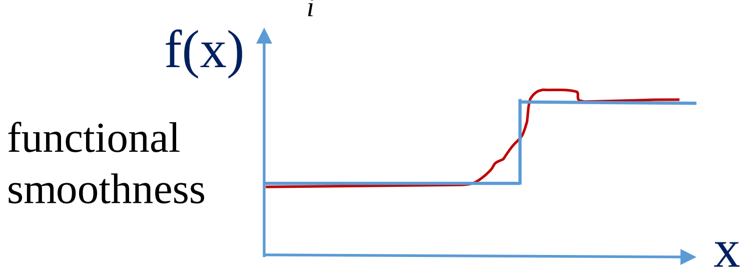
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

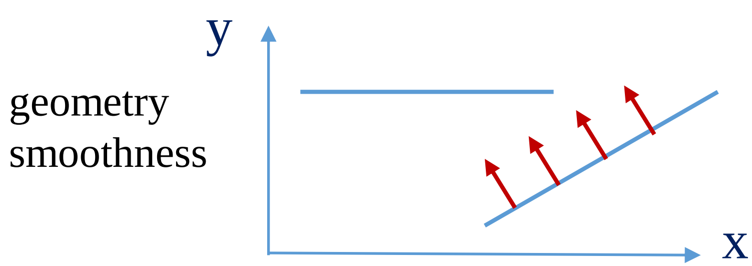
GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

$$\sum_i |f(x_i) - f(x_{i-1})|$$



$$\sum_i |y_i - y_{i-1}| \longrightarrow \sum_i \|\mathrm{n}_i - \mathrm{n}_{i-1}\|_1$$



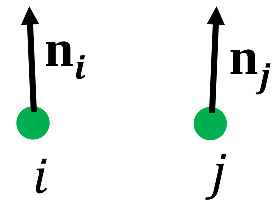
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

PC Denoising Algorithm

- Use GTV of surface normals over the K-NN graph:

$$||\mathbf{n}||_{\text{GTV}} = \sum_{i,j \in \mathcal{E}} w_{i,j} ||\mathbf{n}_i - \mathbf{n}_j||_1$$



$$w_{i,j} = \exp\left(-\frac{||\mathbf{p}_i - \mathbf{p}_j||_2^2}{\sigma_p^2}\right)$$

- Denoising problem as l2-norm fidelity plus GTV of surface normals:

$$\min_{\mathrm{p},\mathrm{n}} \|\mathrm{q} - \mathrm{p}\|_2^2 + \gamma \sum_{i,j \in E} w_{i,j} \|\mathrm{n}_i - \mathrm{n}_j\|_1$$

smoothness on surface normals

- Surface normal estimation of $\mathbf{n}_i$ is a nonlinear function of $\mathbf{p}_i$ and neighbors.

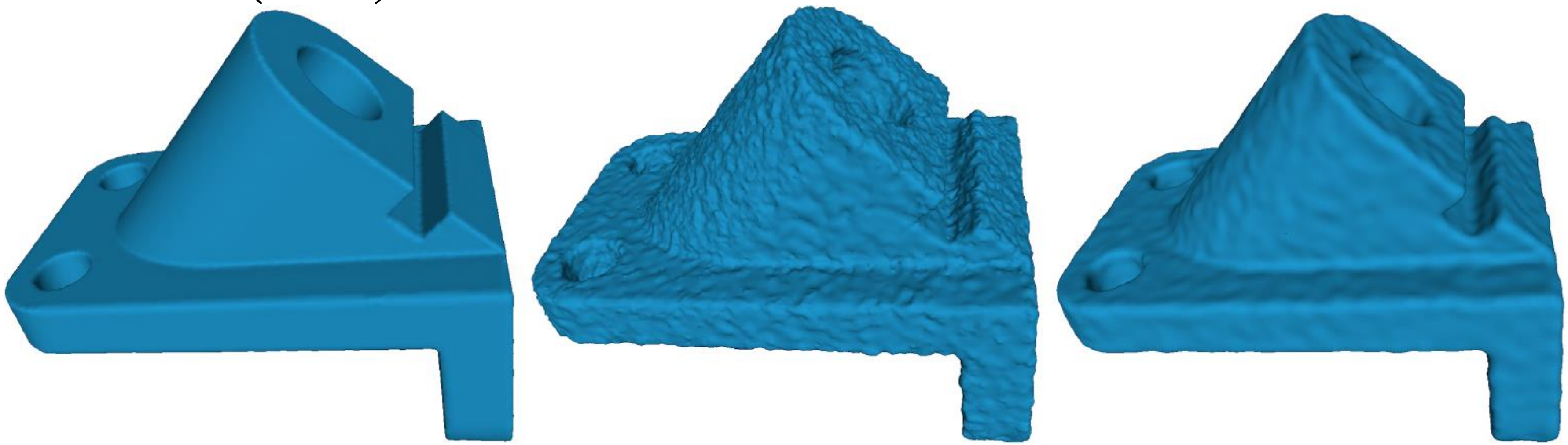
Proposal:

1. Partition point cloud into **two independent classes** (say red and blue).
2. When computing surface normal for a red node, use only neighboring blue points.
3. Solve convex optimization for red (blue) nodes alternately.

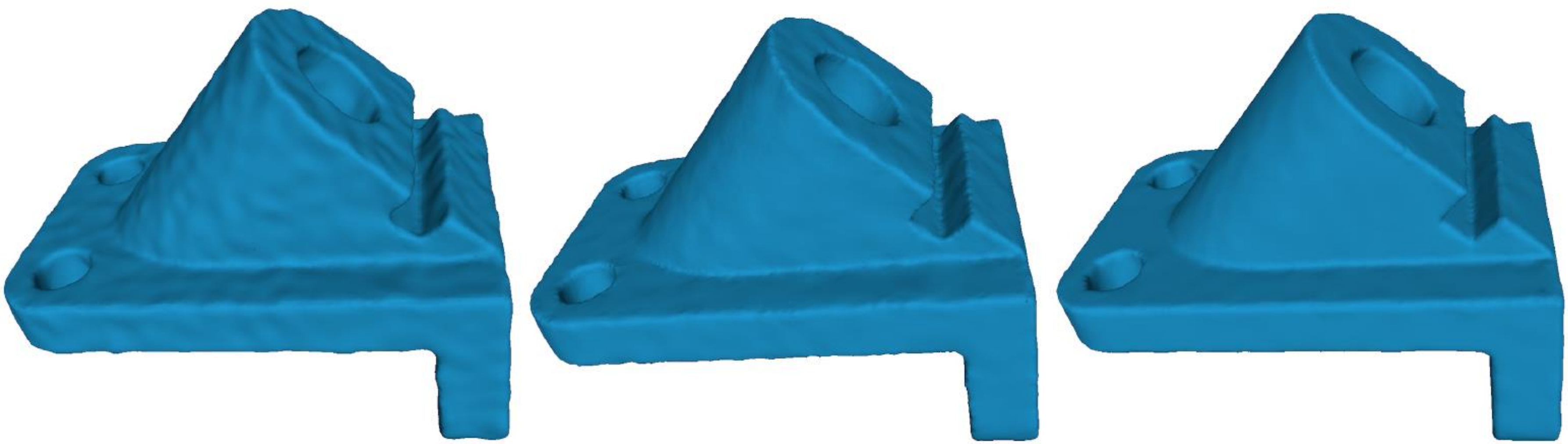
[1] C. Dinesh, G. Cheung, I. V. Bajic, C. Yang, "**Fast 3D Point Cloud Denoising via Bipartite Graph Approximation & Total Variation**," *IEEE 20th International Workshop on Multimedia Signal Processing*, Vancouver, Canada, August 2018.

Results: Point Cloud Denoising

Anchor model (σ=0.3)



(a) ground truth (b) noisy input (c) APSS

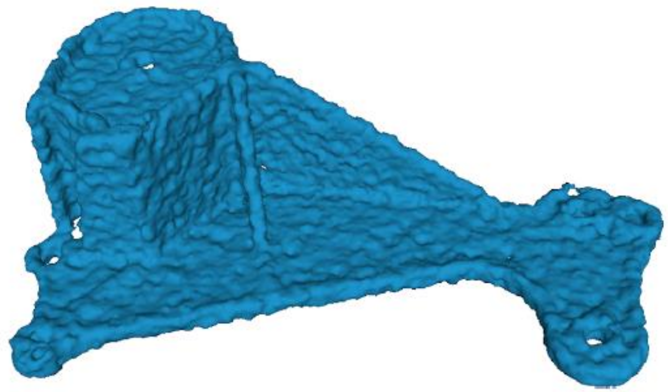


(d) RIMLS (e) MRPCA (f) proposed

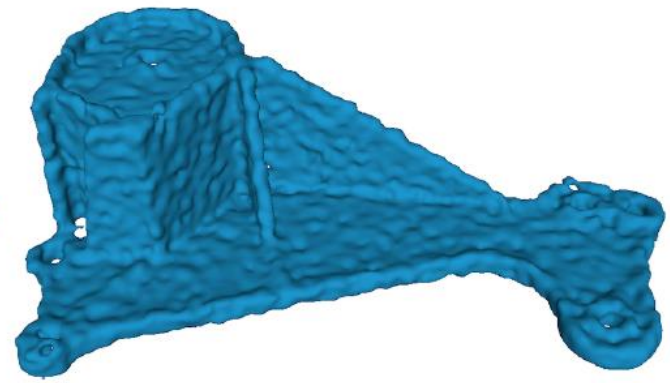
Results: Point Cloud Denoising

Daratech model (σ=0.3)

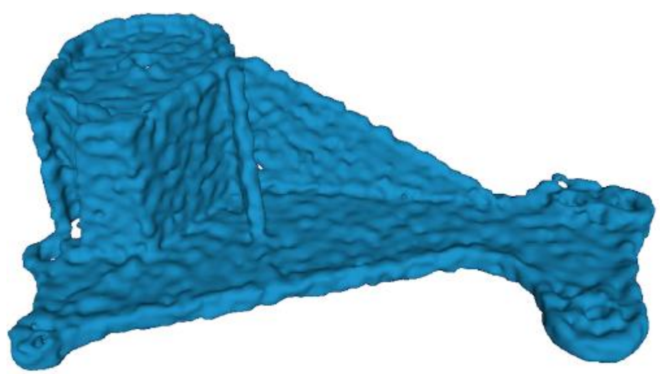
(a) ground truth



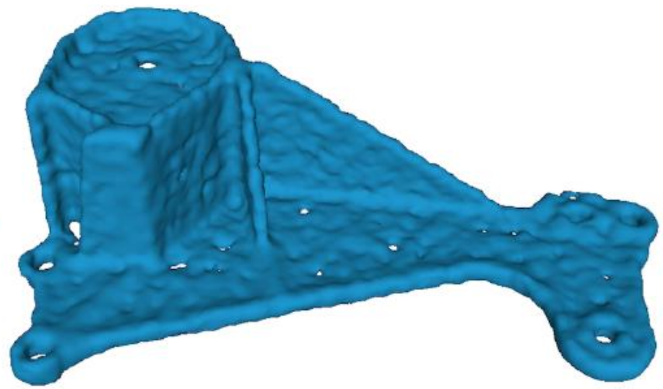
(b) noisy input



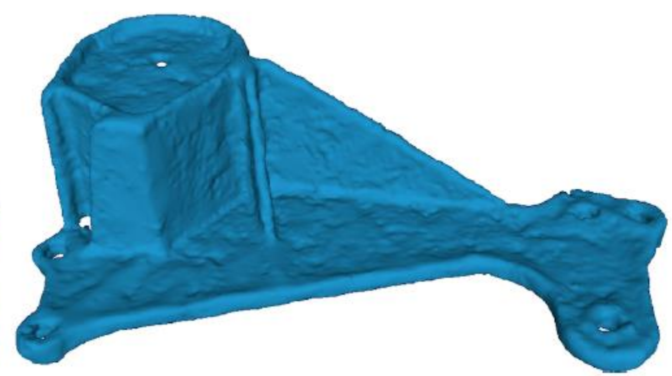
(c) APSS



(d) RIMLS



(e) MRPCA



(f) proposed

PC Super-Res Algorithm

- Add new interior points to low-res point cloud.
  1. Construct triangular mesh using Delaunay triangulation using known points **q**.
  2. Insert new points at the centroids of triangles.

- Partition point cloud into **two independent classes** (say **red** and **blue**).
- When computing normal for a red node, use only neighboring blue points.

- Use graph total variation (GTV) of surface normals over the K-NN graph:

smoothness on surface normals

$$\min_{\mathbf{p},\mathbf{n}} \sum_{i,j \in E} w_{i,j} \left\| \mathbf{n}_i - \mathbf{n}_j \right\|_1$$

$\mathbf{m}_{i,j} = \mathbf{n}_i - \mathbf{n}_j$

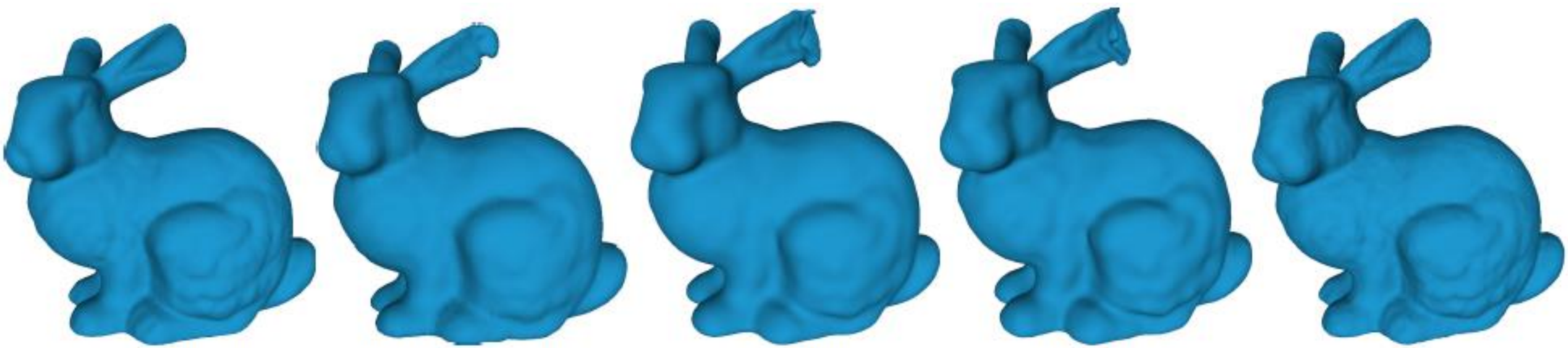
$$\begin{bmatrix} \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{q} \end{bmatrix}$$

- Solved via augmented Lagrangian + ADMM.

[1] C. Dinesh, G. Cheung, I. V. Bajic, C. Yang, "**3D Point Cloud Super-Resolution via Graph Total Variation on Surface Normals**," *IEEE International Conference on Image Processing*, Taiwan, October 2019.

Results: Point Cloud Super-Resolution

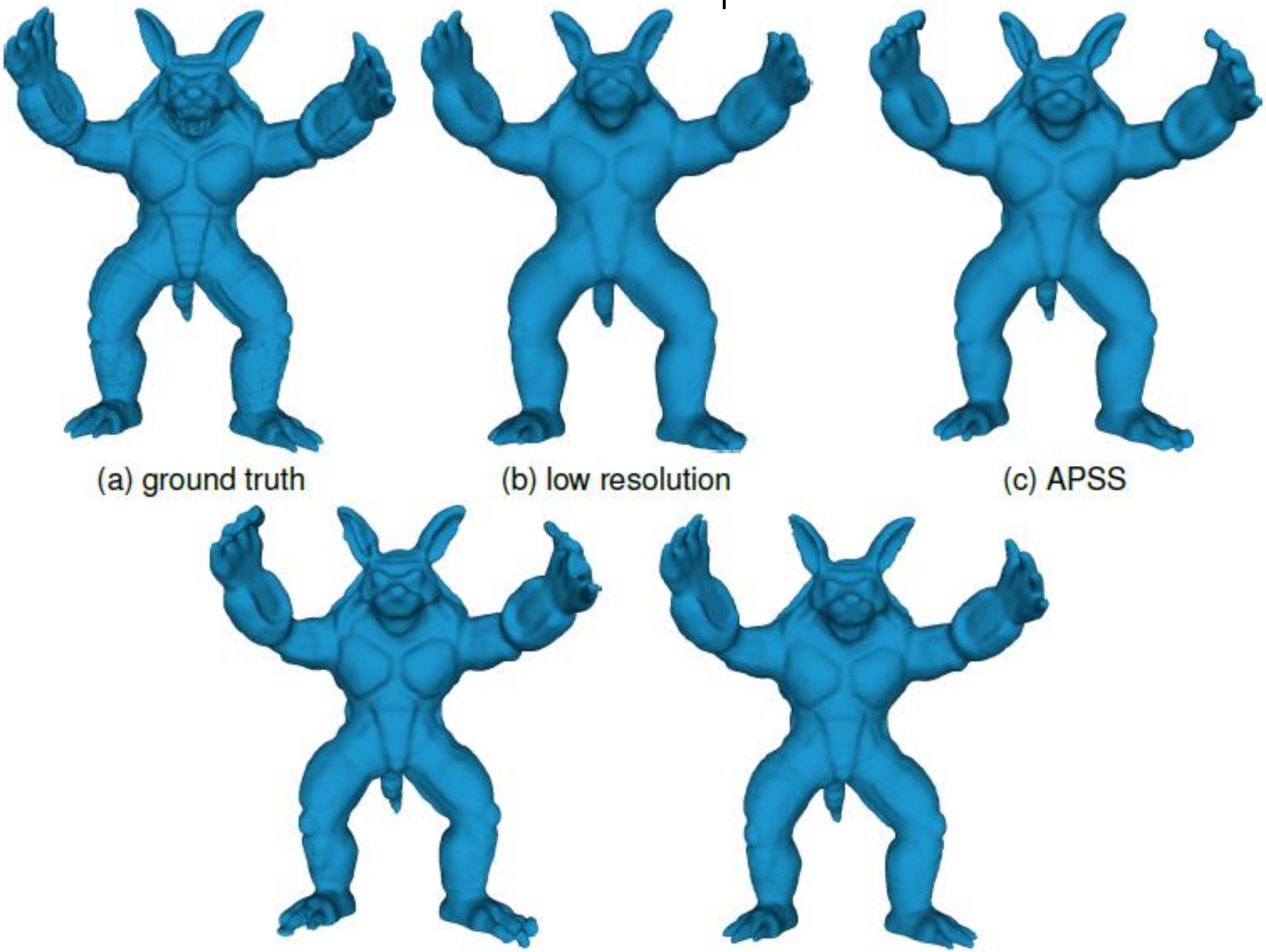
- APSS and RIMLS schemes generate overly smooth models.
- Existing methods result in distorted surfaces with some details lost.



(a) ground truth (b) low resolution (c) APSS (d) RIMLS (e) proposed

[1] C. Dinesh, G. Cheung, I. V. Bajic, "**3D Point Cloud Super-Resolution via Graph Total Variation on Surface Normals**," *IEEE International Conference on Image Processing*, October 2019.

Results: Point Cloud Super-Resolution



(a) ground truth (b) low resolution (c) APSS

(d) RIMLS (e) proposed

Outline

- Defining Graph frequencies
- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution
- Deep GLR
- Semi-Supervised Learning
- Graph Sampling
  - Matrix completion

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

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

fidelity term

smoothness prior

- Solution is system of linear equations:

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

linear system of eqn's w/ sparse, symmetric PD matrix

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

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

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

fidelity term

smoothness prior

- Solution is system of linear equations:

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

linear system of eqn's w/ sparse, symmetric PD matrix

Q: what is the "most appropriate" graph?

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

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

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

fidelity term

smoothness prior

- Solution is system of linear equations:

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

linear system of eqn's w/ sparse, symmetric PD matrix

Q: what is the "most appropriate" graph?

Bilateral weights:

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

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

Unrolling Graph Laplacian Regularizer

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

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

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

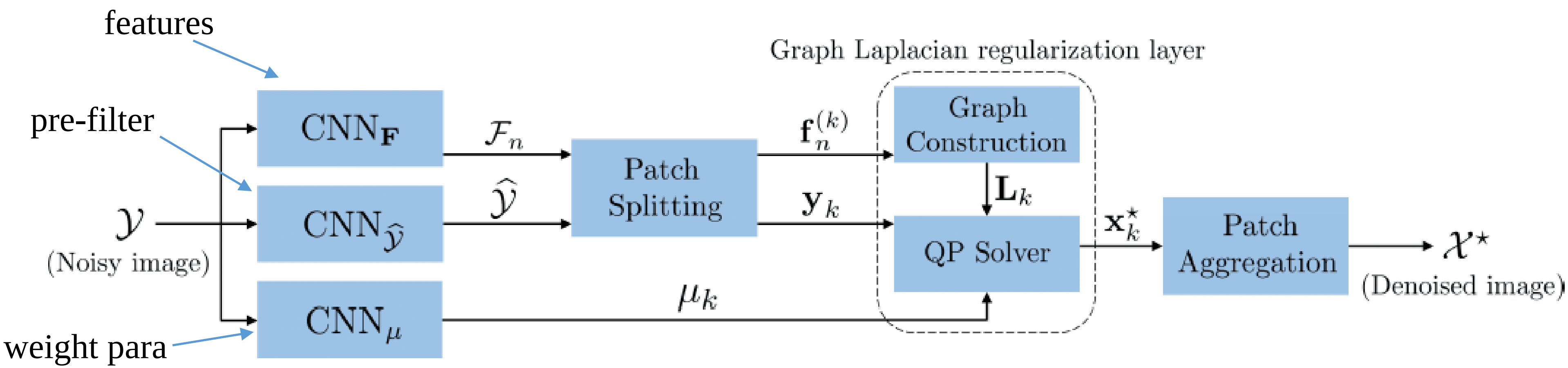


Fig. 1. Block diagram of the proposed GLRNet which employs a graph Laplacian regularization layer for image denoising.

[1] M. McCann et al., "**Convolutional Neural Networks for Inverse Problems in Imaging**," *IEEE SPM*, Nov. 2017.

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

Unrolling Graph Laplacian Regularizer

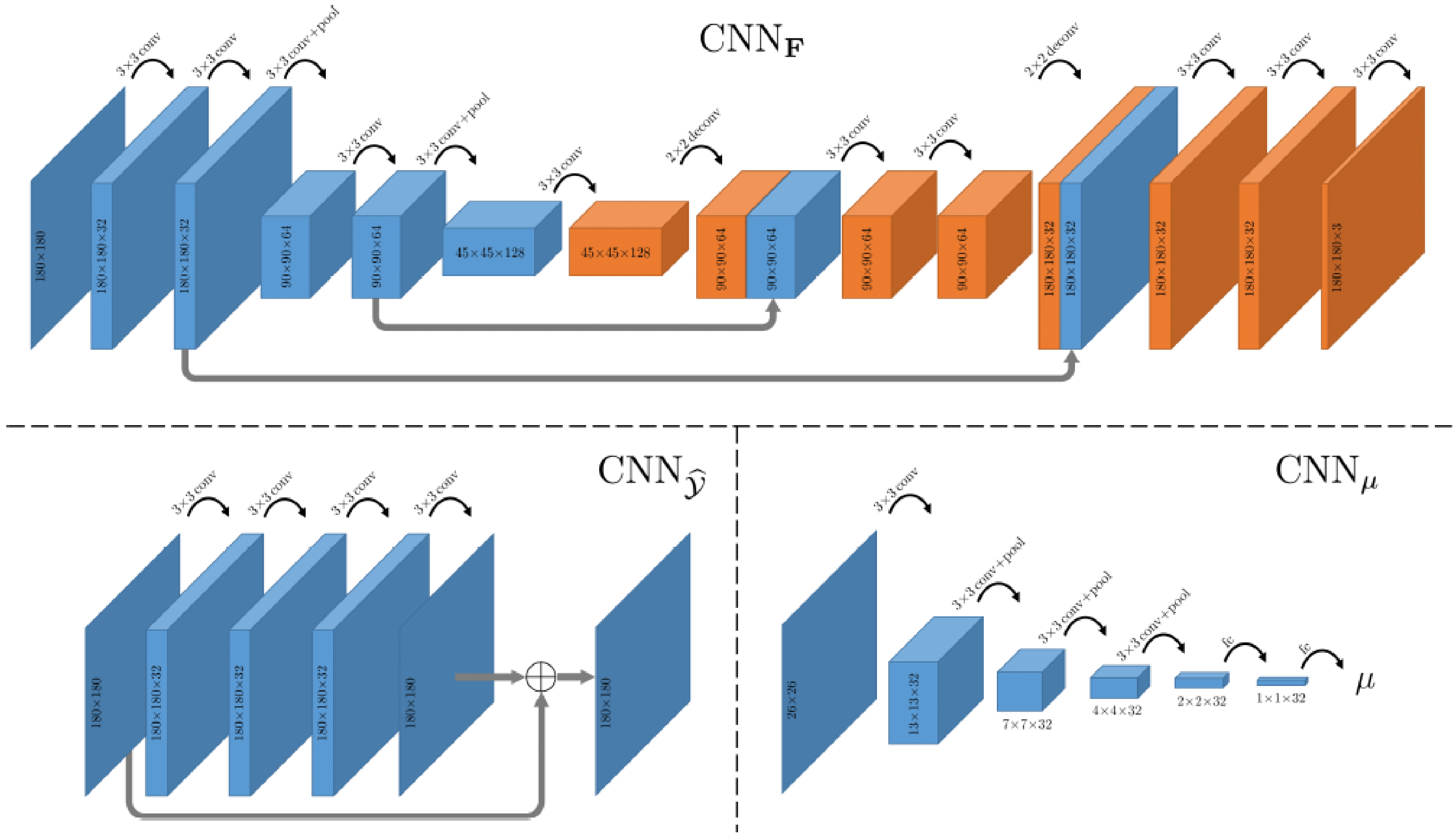


Fig. 3. Network architectures of CNN$_{\mathbf{F}}$, CNN$_{\hat{\mathcal{Y}}}$ and CNN$_{\mu}$ in the experiments. Data produced by the decoder of CNN$_{\mathbf{F}}$ is colored in orange.

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

Unrolling Graph Laplacian Regularizer

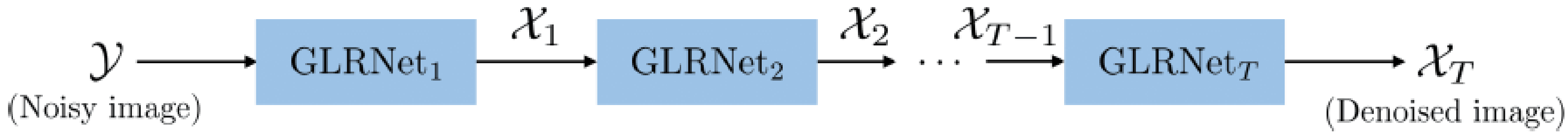


Fig. 2. Block diagram of the overall DeepGLR framework.

- **Graph 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**: By restricting search space of CNN to degree-bounded graphs, we achieve robust learning.

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

Experimental Results – 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.

Experimental Results – 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.

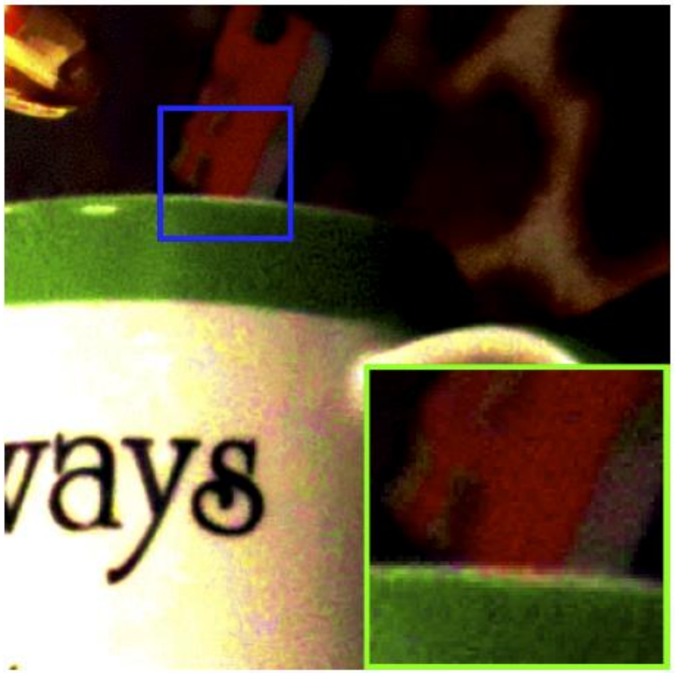
Table 4. Evaluation of cross-domain generalization for real image denoising. The best results are highlighted in boldface.

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**

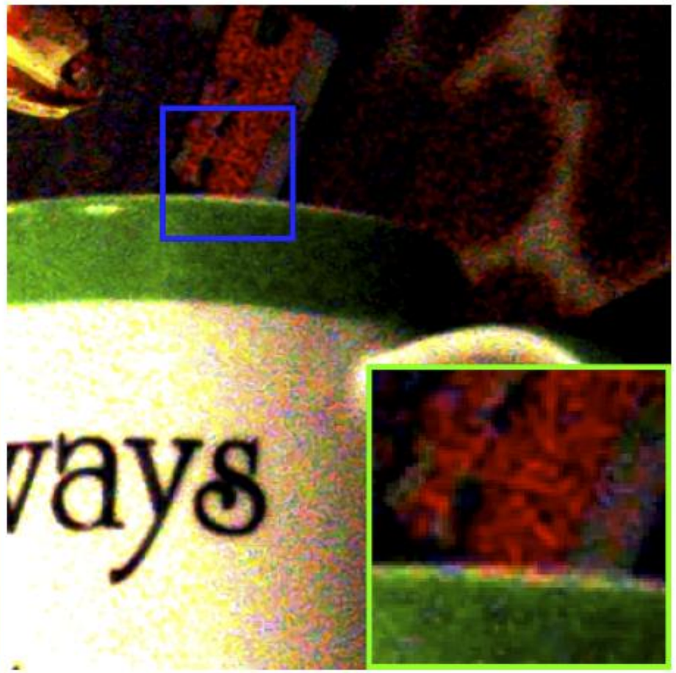
[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Experimental Results – Visual Comparison

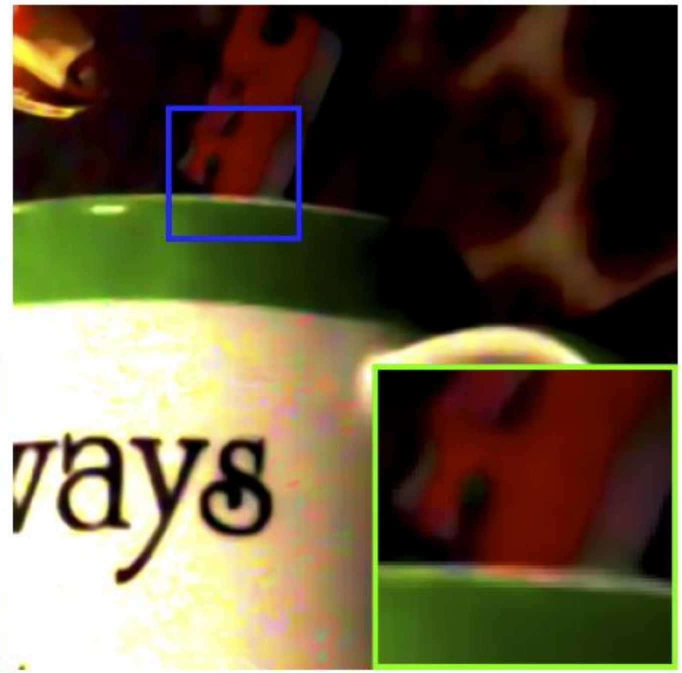
- trained on Gaussian noise, tested on low-light images in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- outperformed DnCNN by 5.74 dB, and noise clinic by 1.87 dB.



Noise Clinic



CDnCNN



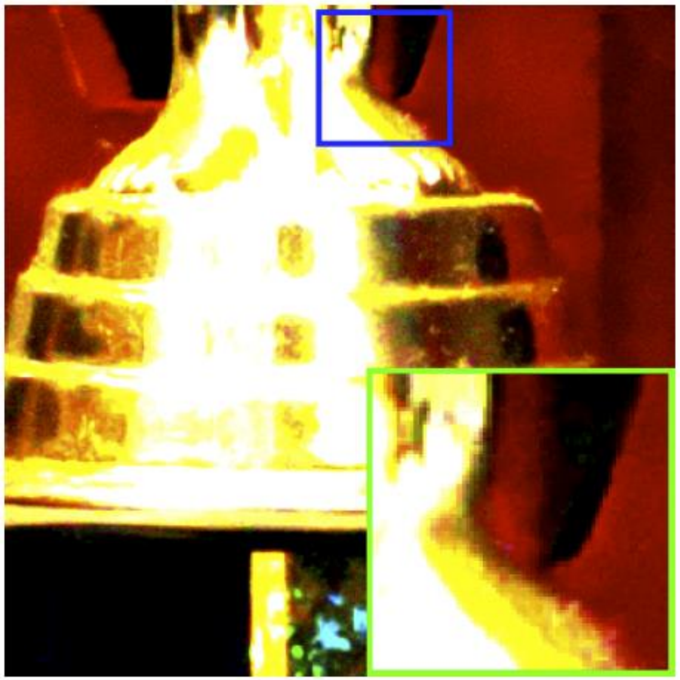
DeepGLR

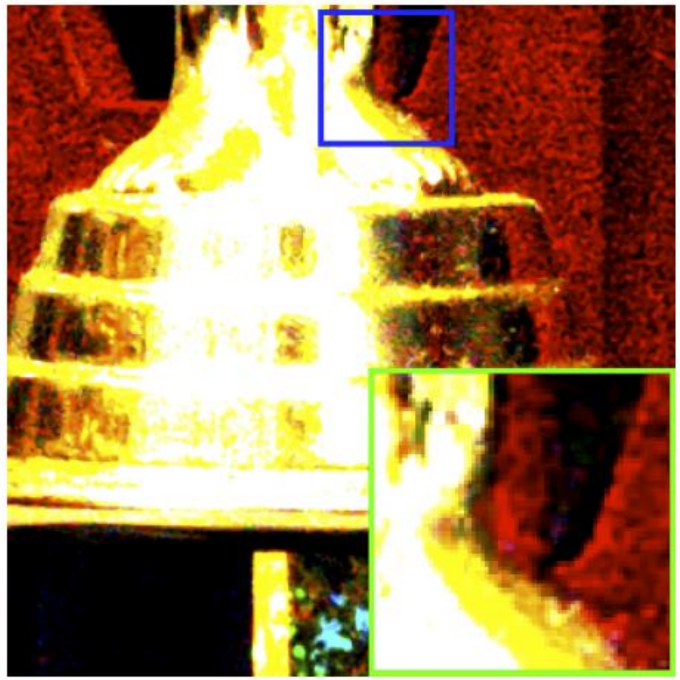
[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.

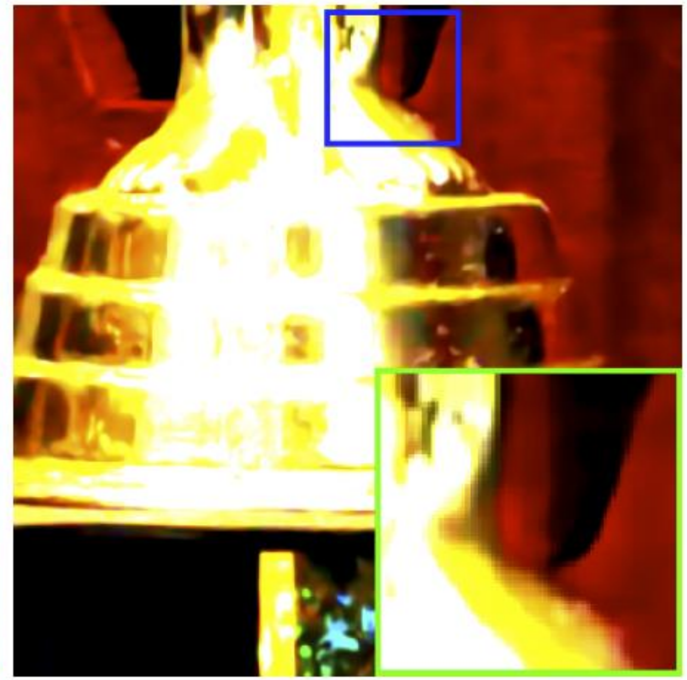
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Experimental Results – 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.

Outline

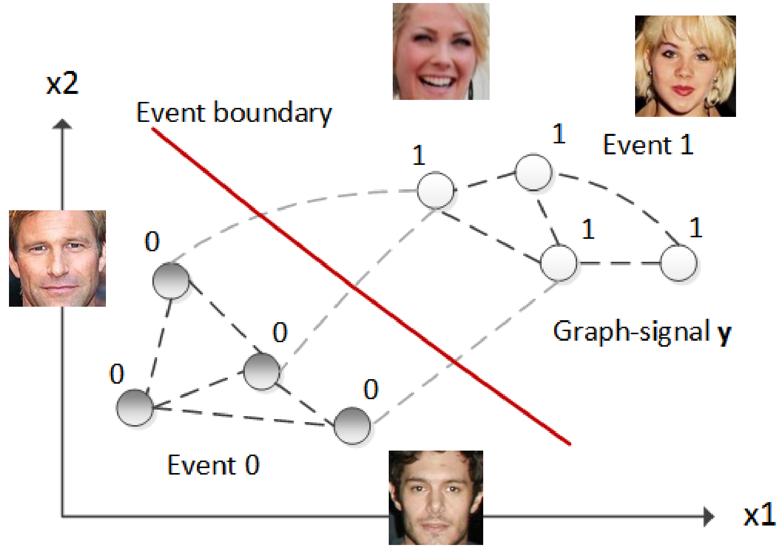
- Defining Graph frequencies
- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution
- Deep GLR
- Semi-Supervised Learning
- Graph Sampling
  - Matrix completion

Semi-Supervised Graph Classifier Learning

- **Binary Classifier**: given feature vector x_i of dimension K, compute $f(x_i) \in \{0,1\}$.

- **Classifier Learning**: given partial, noisy labels (x_i, y_i), train classifier $f(x_i)$.

- **GSP Approach** [1]:
  1. Construct ***signed similarity graph*** with +/- edges.
  2. Pose MAP graph-signal restoration problem.
  3. Perturb graph Laplacian to ensure PSD.
  4. Solve num. stable MAP as sparse lin. system.



example graph-based classifier

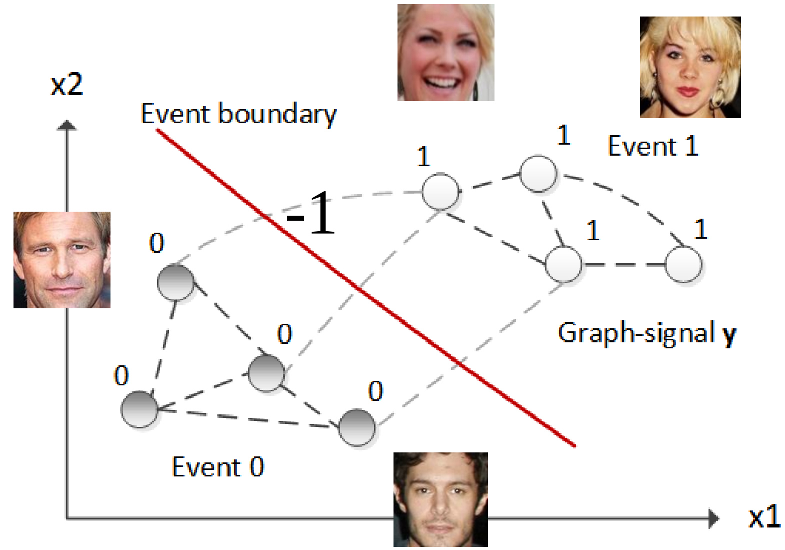
[1] Yu Mao, Gene Cheung, Chia-Wen Lin, Yusheng Ji, "**Image Classifier Learning from Noisy Labels via Generalized Graph Smoothness Priors**," *IEEE IVMSP Workshop*, Bordeaux, France, July 2016. (**Best student paper award**)

[2] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "**Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights**," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no.4, pp.712-726, December 2018.

Semi-Supervised Graph Classifier Learning

- **Binary Classifier**: given feature vector x_i of dimension K, compute $f(x_i) \in \{0,1\}$.

- **Classifier Learning**: given partial, noisy labels (x_i, y_i), train classifier $f(x_i)$.



example graph-based classifier

- **GSP Approach** [1]:
  1. Construct ***signed similarity graph*** with +/- edges.
  2. Pose MAP graph-signal restoration problem.
  3. Perturb graph Laplacian to ensure PSD.
  4. Solve num. stable MAP as sparse lin. system.

[1] Yu Mao, Gene Cheung, Chia-Wen Lin, Yusheng Ji, "**Image Classifier Learning from Noisy Labels via Generalized Graph Smoothness Priors**," *IEEE IVMSP Workshop*, Bordeaux, France, July 2016. (**Best student paper award**)

[2] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "**Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights**," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no.4, pp.712-726, December 2018.

Graph-Signal Smoothness Prior for signed graphs

w=1

- **Graph Laplacian Regularizer [1]:**

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{(i,j)\in\mathcal{E}} w_{i,j}\,(x_i - x_j)^2 = \sum_k \lambda_k\,\alpha_k^2$$

GFT coeff

eigenvalues / graph freqs

- Promote large / small inter-node differences depending on edge signs.

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = -1(x_1 - x_2)^2 + (x_2 - x_3)^2$$

Promote large difference

Promote small difference

- Sensible, but numerically unstable.

[1] J. Pang and G. Cheung, "**Graph Laplacian regularization for image denoising: Analysis inn the continuous domain**," in *IEEE Transactions on Image Processing*, vol. 26, no.4, April 2017, pp. 1770–1785.

Semi-Supervised Learning Formulation

- MAP formulation:

perturbation matrix
to ensure PSD!

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 + \mu\, \mathbf{x}^T \left(\mathbf{L} + \boldsymbol{\Delta}\right) \mathbf{x}$$

l0 fidelity term

smoothness prior for signed graph

- One sol'n is $\triangle = \lambda_{\min} \mathrm{I}$, *i.e.* shift all eigenvalues up by $\eta = \lambda_{\min}$.
- **Intuition**: signal variations + signal energies

$$\begin{aligned} \mathbf{x}^T(\mathbf{L}+\boldsymbol{\Delta})\mathbf{x} &= \mathbf{x}^T\mathbf{L}\mathbf{x} + \eta\, \mathbf{x}^T\mathbf{I}\mathbf{x} \\ &= \sum_{i,j} w_{i,j}(x_i - x_j)^2 + \eta \sum_i x_i^2 \end{aligned}$$

[1] G. Cheung, W.-T. Su, Y. Mao, C.-W. Lin, "**Robust Semi-Supervised Graph Classifier Learning with Negative Edge Weights**," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no.4, pp.712-726, December 2018.

Results: Semi-Supervised Learning

- Comparisons w/ other classifiers:

TABLE II
CLASSIFICATION ERROR RATES IN THE BANANA DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	54.71%	54.97%	54.70%	53.95%	53.42%
SVM-RBF	12.49%	13.27%	13.72%	16.23%	18.63%
RobustBoost [26]	20.42%	22.73%	24.53%	25.12%	27.52%
Graph-Pos	14.05%	15.89%	18.02%	20.76%	21.93%
Graph-MinNorm	10.23%	12.37%	14.44%	17.41%	18.69%
Graph-Bandlimited [58]	7.53%	11.77%	15.80%	19.14%	21.07%
Graph-AdjSmooth [9]	8.85%	12.08%	15.28%	18.26%	20.67%
Graph-Wavelet [6]	23.18%	24.25%	25.70%	27.15%	30.13%
Proposed-Centroid	**5.17%**	10.50%	13.79%	16.80%	19.39%
Proposed-Boundary	13.37%	15.68%	18.27%	20.51%	22.72%
Proposed-Hybrid	5.36%	**9.43%**	**12.79%**	**16.04%**	**18.43%**
Proposed-Rej	3.74% (9.59%)	6.57% (9.89%)	9.26% (9.14%)	12.19% (9.96%)	14.06% (9.95%)

Results: Semi-Supervised Learning

- Comparisons w/ other classifiers:

TABLE II
CLASSIFICATION ERROR RATES IN THE BANANA DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	54.71%	54.97%	54.70%	53.95%	53.42%
SVM-RBF	12.49%	13.27%	13.72%	16.23%	18.63%
RobustBoost [26]	20.42%	22.73%	24.53%	25.12%	27.52%
Graph-Pos	14.05%	15.89%	18.02%	20.76%	21.93%
Graph-MinNorm	10.23%	12.37%	14.44%	17.41%	18.69%
Graph-Bandlimited [58]	7.53%	11.77%	15.80%	19.14%	21.07%
Graph-AdjSmooth [9]	8.85%	12.08%	15.28%	18.26%	20.67%
Graph-Wavelet [6]	23.18%	24.25%	25.70%	27.15%	30.13%
Proposed-Centroid	**5.17%**	10.50%	13.79%	16.80%	19.39%
Proposed-Boundary	13.37%	15.68%	18.27%	20.51%	22.72%
Proposed-Hybrid	5.36%	**9.43%**	**12.79%**	**16.04%**	**18.43%**
Proposed-Rej	3.74% (9.59%)	6.57% (9.89%)	9.26% (9.14%)	12.19% (9.96%)	14.06% (9.95%)

Results: Semi-Supervised Learning

- Comparisons w/ other classifiers:

TABLE III

CLASSIFICATION ERROR RATES IN THE FACE GENDER DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	17.65%	18.22%	18.77%	19.59%	21.6%
SVM-RBF	12.14%	12.16%	12.83%	16.30%	24.01%
RobustBoost [26]	9.15%	11.09%	14.36%	17.36%	20.68%
Graph-Pos	13.15%	13.62%	14.38%	15.39%	16.54%
Graph-MinNorm	7.15%	8.26%	9.48%	10.37%	12.01%
Graph-Bandlimited [58]	5.78%	11.83%	15.30%	19.74%	23.44%
Graph-AdjSmooth [9]	**1.25%**	5.01%	7.94%	11.45%	15.39%
Graph-Wavelet [6]	20.02%	19.95%	20.12%	20.7%	21.43%
Proposed-Centroid	1.44%	**2.96%**	4.46%	5.88%	8.07%
Proposed-Boundary	10.81%	12.09%	13.17%	14.33%	15.96%
Proposed-Hybrid	1.71%	3.02%	**4.22%**	**5.75%**	**7.71%**
Proposed-Rej	0.36% (9.70%)	0.68% (9.29%)	1.08% (9.85%)	2.39% (9.08%)	4.18% (9.05%)

Results: Semi-Supervised Learning

- Comparisons w/ other classifiers:

TABLE III

CLASSIFICATION ERROR RATES IN THE FACE GENDER DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	17.65%	18.22%	18.77%	19.59%	21.6%
SVM-RBF	12.14%	12.16%	12.83%	16.30%	24.01%
RobustBoost [26]	9.15%	11.09%	14.36%	17.36%	20.68%
Graph-Pos	13.15%	13.62%	14.38%	15.39%	16.54%
Graph-MinNorm	7.15%	8.26%	9.48%	10.37%	12.01%
Graph-Bandlimited [58]	5.78%	11.83%	15.30%	19.74%	23.44%
Graph-AdjSmooth [9]	**1.25%**	5.01%	7.94%	11.45%	15.39%
Graph-Wavelet [6]	20.02%	19.95%	20.12%	20.7%	21.43%
Proposed-Centroid	1.44%	**2.96%**	4.46%	5.88%	8.07%
Proposed-Boundary	10.81%	12.09%	13.17%	14.33%	15.96%
Proposed-Hybrid	1.71%	3.02%	**4.22%**	**5.75%**	**7.71%**
Proposed-Rej	0.36% (9.70%)	0.68% (9.29%)	1.08% (9.85%)	2.39% (9.08%)	4.18% (9.05%)

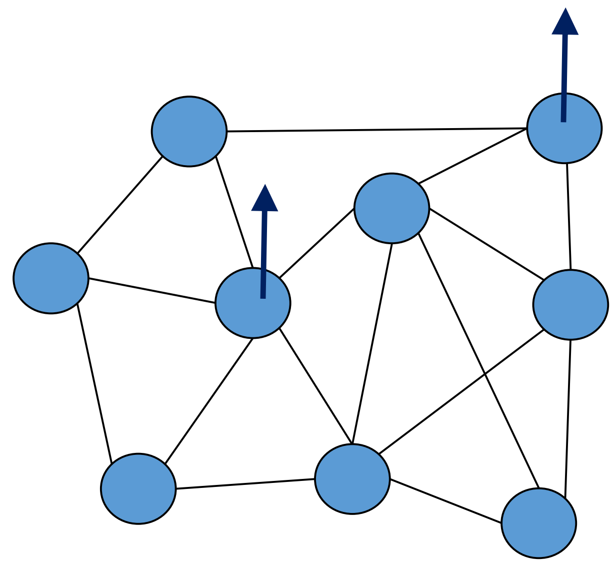
Outline

- Defining Graph frequencies
- Inverse Imaging
  - Image denoising
  - Image contrast enhancement
  - 3D point cloud denoising / super-resolution
- Deep GLR
- Semi-Supervised Learning
- Graph Sampling
  - Matrix completion

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.

Related Works

- Graph sampling
  - Aggressive sampling [1]
  - Local measurement [2]
  - Selection sampling
    - Deterministic
      - Bandlimitedness
        - A-optimal (M. Tsitsvero.TSP2016; F. Wang. SPL2018)
        - E-optimal (S. Chen. TSP2015)
        - Spectral proxies (A. Anis. TSP2016; A. Anis. TIT2018)
        - Localized coverage (A. Sakiyama. TSP2019)
      - Smoothness: GMRF model (P. Chen. ICASSP2018; Y. Bai. ICASSP 2019)
    - Random selection [3]

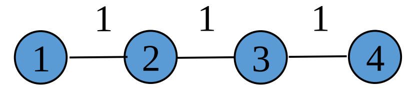
Eigen-decomposition Free

[1] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "**Sampling of graph signals with successive local aggregations**." *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1832–1843, 2016.

[2] X. Wang, J. Chen, and Y. Gu, "**Local measurement and reconstruction for noisy bandlimited graph signals**," *Signal Processing*, vol. 129, pp. 119–129, 2016.

[3] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "**Random sampling of bandlimited signals on graphs**," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, 2018.

Signal Reconstruction using GLR



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

Sample set {2, 4}

sampling matrix

observation

desired signal

- **Signal Model**: $$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

noise

- Signal prior is graph Laplacian regularizer (GLR) [1]:

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

signal contains
mostly low graph freq.

signal smooth w.r.t. graph

- **MAP Formulation**:

fidelity term

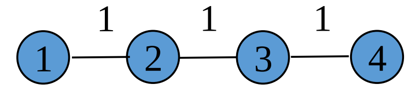
signal prior

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

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

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

Stability of Linear System



- Examine system of linear equations :

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

coefficient matrix **B**

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

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

Sample set {2, 4}

- Stability depends on the **condition number** (λ_{max}/ λ_{min}) of coeff. matrix **B**.

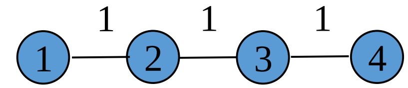
- λ_{max} is upper-bounded by $1+\mu\, 2^*d_{max}$.

- **Goal**: select samples to maximize λ_{min} (without computing eigen-pairs)!

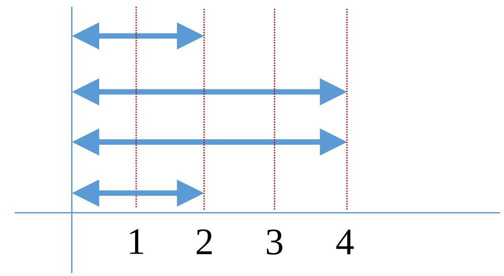
- Also minimizes worst-case MSE:

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

Gershgorin Circle Theorem



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



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

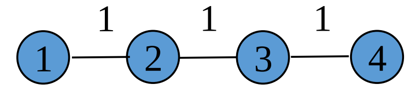
$$R_i = \sum_{j \neq i} |L_{ij}|$$

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

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

- Graph Laplacian **L** has all Gershgorin disc left-ends at 0 → **L** is psd.

Gershgorin Disc Alignment

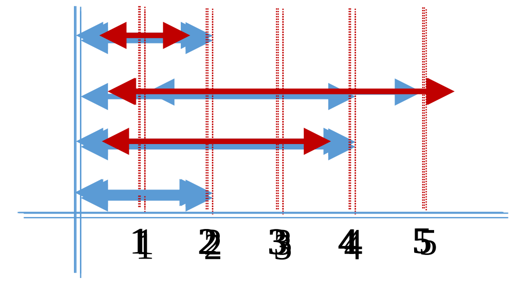


- **Main Idea**: Select samples to max smallest disc left-end of coefficient matrix **B**:

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

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

- Sample node → shift disc.



- Consider similar transform of **B**:

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

diagonal matrix w/ scale factors

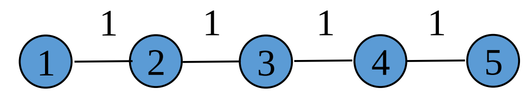
Sample set {2}

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

- Scale row → **expand** disc radius.

  → **shrink** neighbors' disc radius.

Aligning discs at threshold T

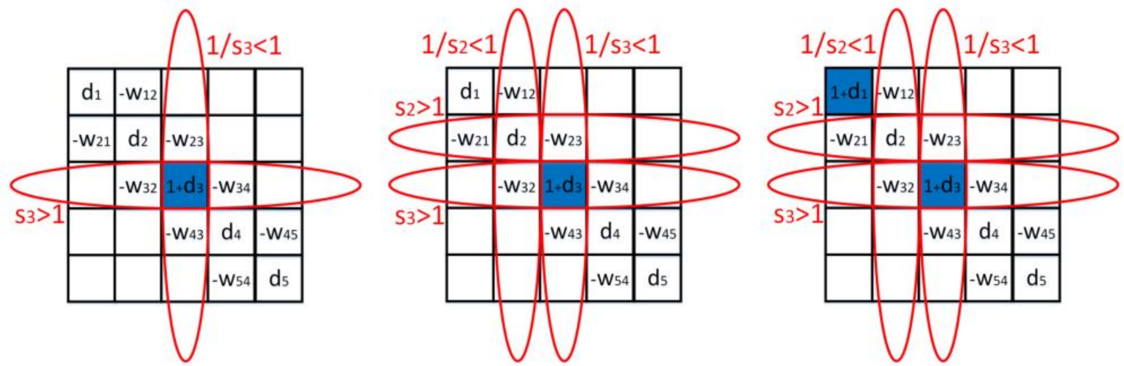


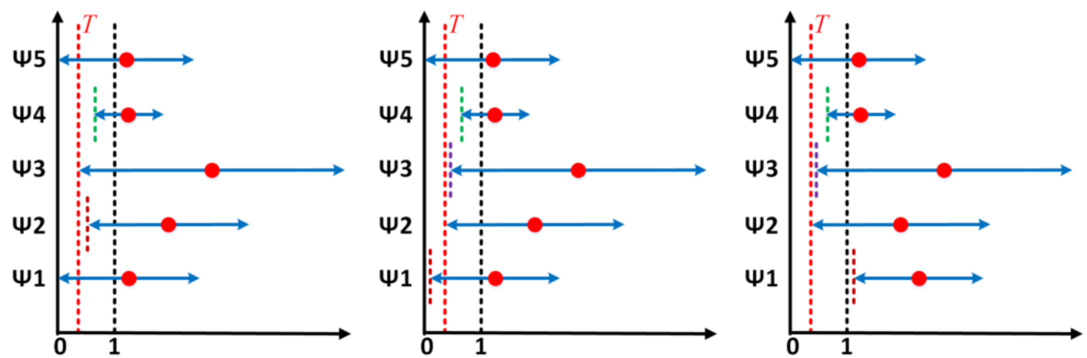
- **Breadth First Iterative Sampling (BFIS)**:
  - Given initial node set, threshold T.

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 1 if node set not empty.
5. Output sample set and count K.

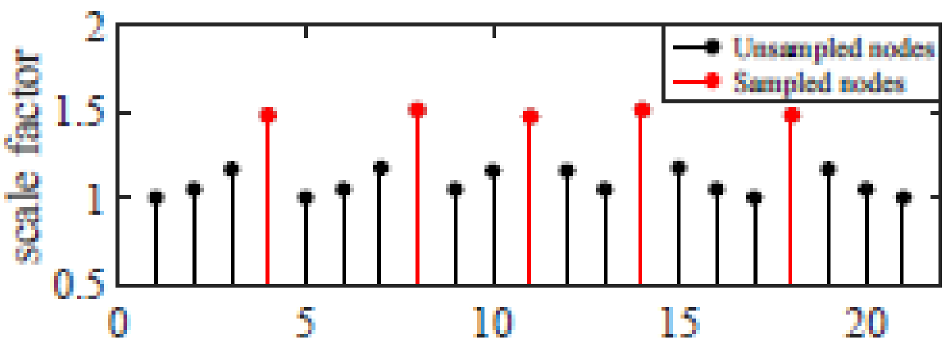




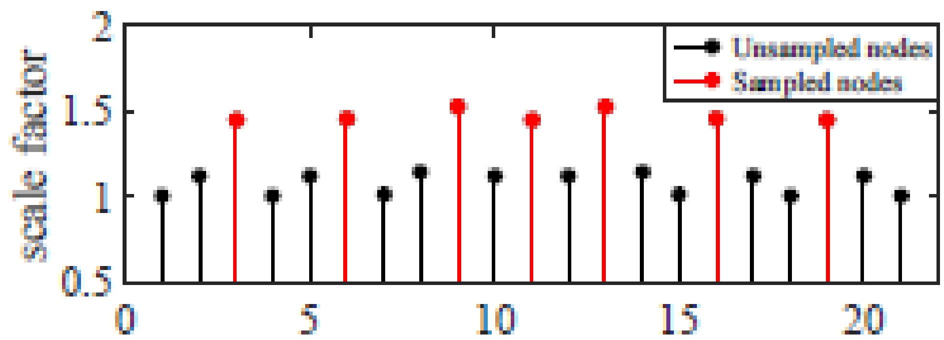
Gershgorin Disc Alignment (math)

- **Binary Search with BFIS**:
  - Sample count *K* inverse proportional to threshold *T*.
  - Binary search on *T* to drive count *K* to budget.

- **Example**: line graph with equal edge weight.
  - Uniform sampling.



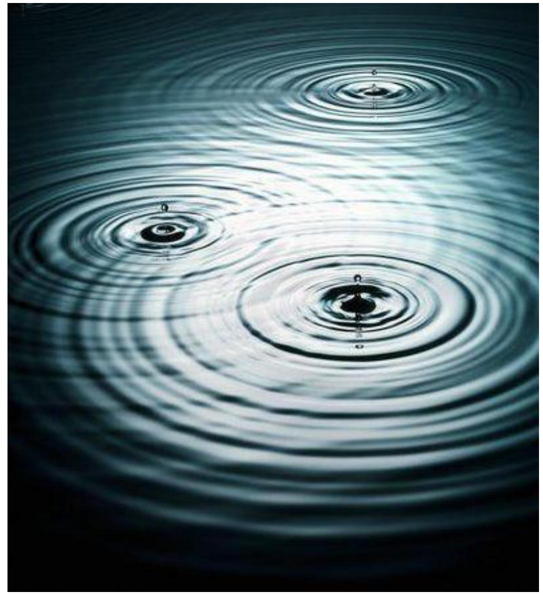
(a)

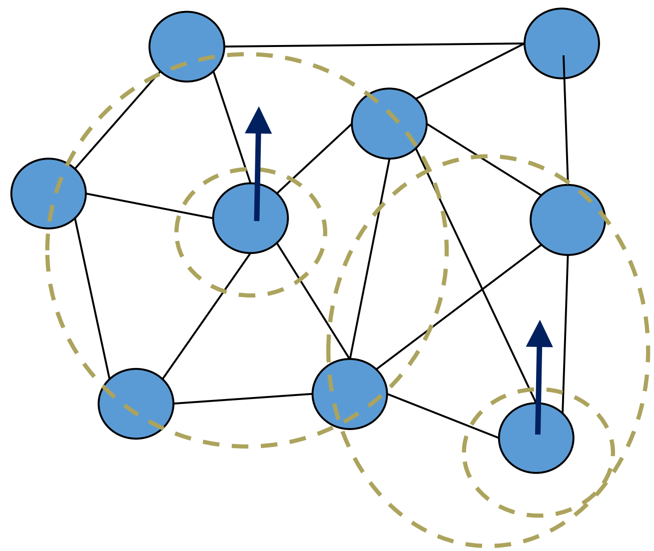


(b)

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*.





Results: Graph Sampling

- GDA is 100x to 1000x faster than state-of-art methods computing e-vectors.
- GDA is “comparable” in complexity to Random [23] and Ed-free [8].

TABLE II

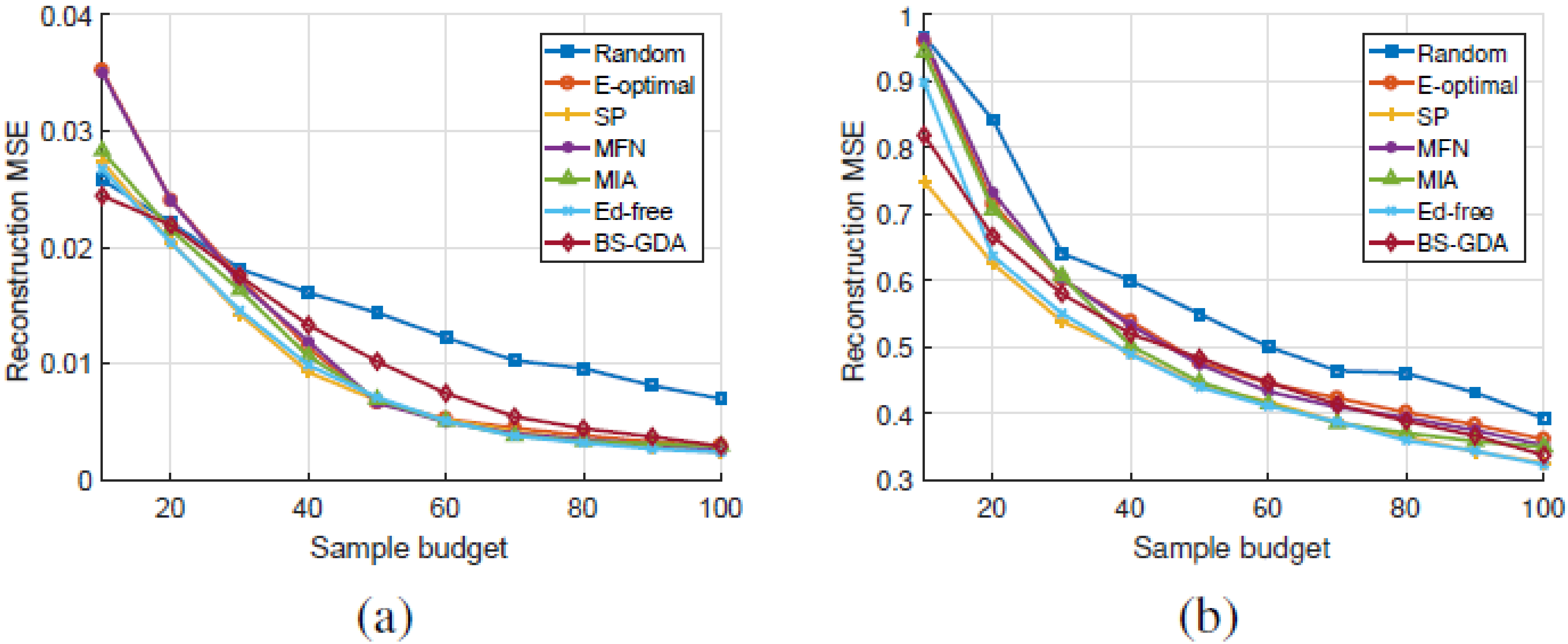
SPEEDUP FACTORS OF OUR ALGORITHM WITH RESPECT TO OTHER SAMPLING ALGORITHMS FOR $N = 3000$

Sampling Methods	Sensor	Community
Random [23]	0.22	0.21
E-optimal [20]	2812.77	1360.76
SP [12]	174.09	466.18
MFN [18]	2532.91	1184.23
MIA [16]	1896.19	964.65
Ed-free [8]	1.82	8.11

[1] Yuanchao Bai, Fen Wang, Gene Cheung, Yuji Nakatsukasa, Wen Gao, "**Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment**," submitted to *IEEE Transactions on Signal Processing*, July 2019.

Results: Graph Sampling

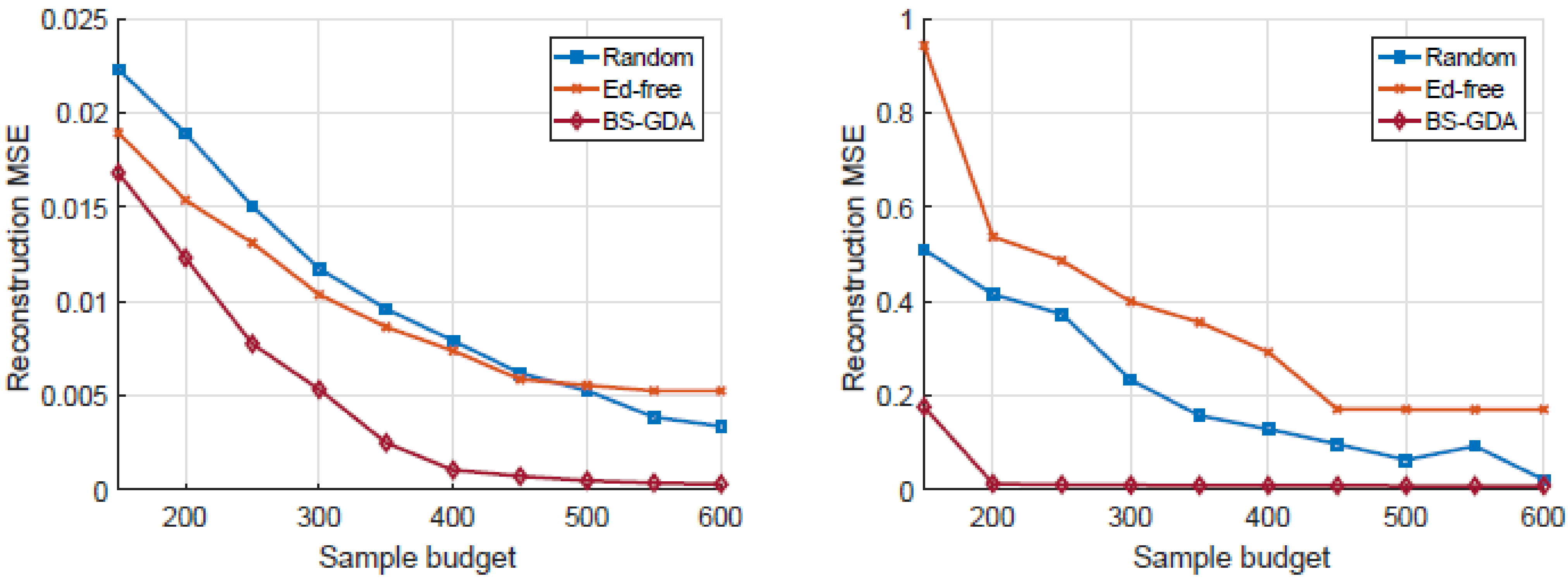
- **Small graphs**: GDA has roughly the same reconstruction MSE.
  - Random sensor graph of size 500 for two signal types.



[1] Yuanchao Bai, Fen Wang, Gene Cheung, Yuji Nakatsukasa, Wen Gao, "**Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment**," submitted to *IEEE Transactions on Signal Processing*, July 2019.

Results: Graph Sampling

- **Large graphs**: GDA has smallest reconstruction MSE.
  - Minnesota road graph of size 2642 and for two signal types.



(e) (f)

[1] Yuanchao Bai, Fen Wang, Gene Cheung, Yuji Nakatsukasa, Wen Gao, "**Fast Graph Sampling Set Selection Using Gershgorin Disc Alignment**," submitted to *IEEE Transactions on Signal Processing*, July 2019.

Matrix Completion

1	2			
	4			
		3		
				2
			3	

- Fill in missing entries in a matrix:
  (Low-rank matrix recovery problem)

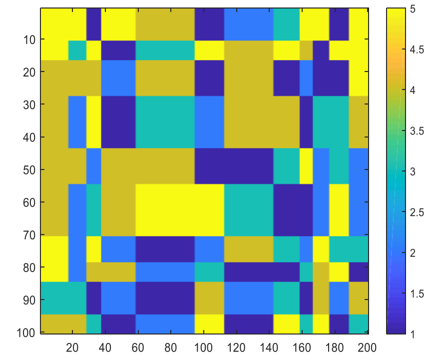
$$\min_{X \in R^{M \times N}} \operatorname{rank}(X)$$

$$s.t. \quad X_{i,j} = M_{i,j}, \quad \forall i, j \in S$$

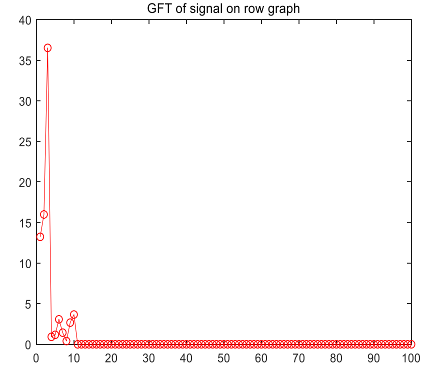
- Examples of applications:
  - **Recommendation system**—making rating prediction.
  - **Remote sensing**—infer full covariance matrix from partial correlations.
  - **Structure-from-motion** in computer vision.

[1] E. Candes and Y. Plan, "**Matrix completion with noise**," *Proceedings of the IEEE*, vol. 98, no.6, April 2010, pp. 925–936.

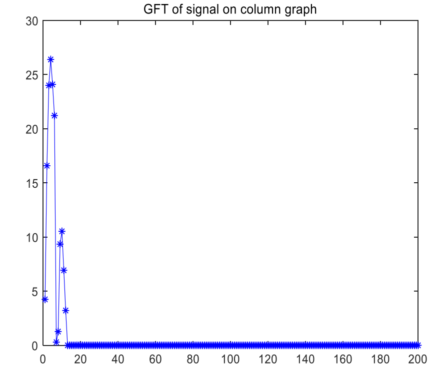
Matrix Completion



Synthetic Netflix matrix



Graph Fourier transform on $\mathbf{L}_r$



Graph Fourier transform on $\mathbf{L}_c$

- Convex relaxation to **nuclear norm**:

$$\min_{\mathrm{X}\in R^{M\times N}} \|\mathrm{X}\|_*$$

$$s.t. \quad X_{i,j} = M_{i,j}, \quad \forall i,j \in S$$

  - **Proximal Gradient**: SVD plus singular value soft-thresholding.

- Use **dual graph-signal smoothness prior** to promote low rank [1]:

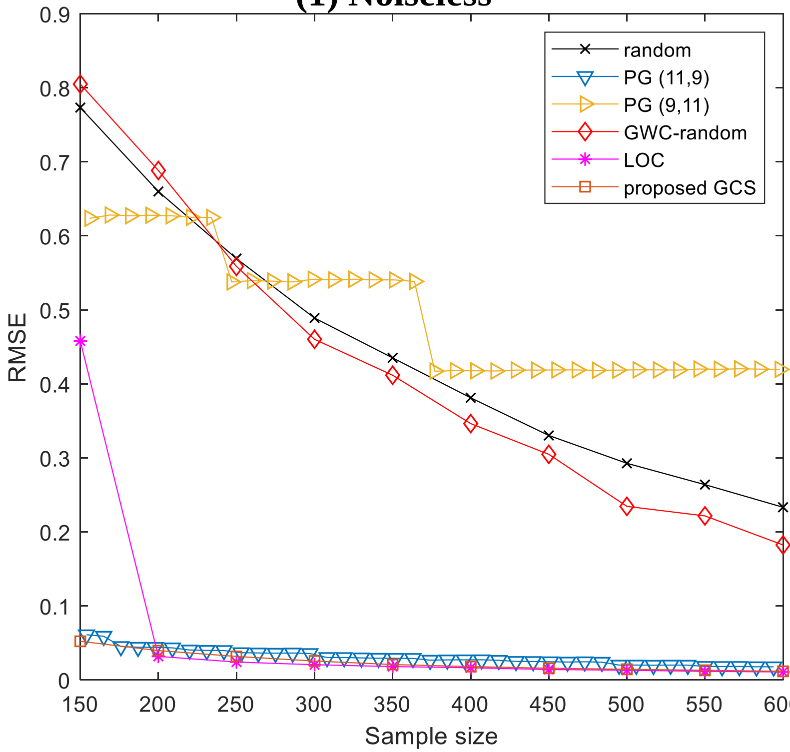
$$\min_{\mathrm{X}\in R^{M\times N}} \mathrm{tr}\left(\mathrm{X}^T \mathrm{L}_r \mathrm{X}\right) + \gamma\, \mathrm{tr}\left(\mathrm{X}\mathrm{L}_c \mathrm{X}^T\right) + \mu \left\| \mathrm{S}\circ\mathrm{M} - \mathrm{S}\circ\mathrm{X} \right\|_F^2$$

  - Unconstrained convex objective solvable via ADMM, conjugate gradient.

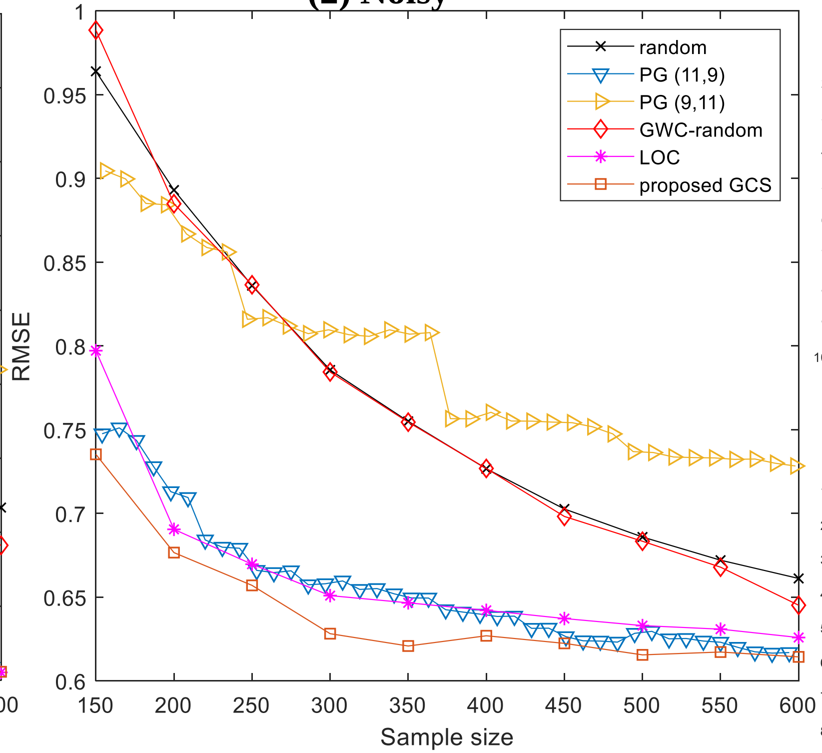
[1] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "**Matrix completion on graphs**,"arXiv preprint arXiv:1408.1717, 2014.

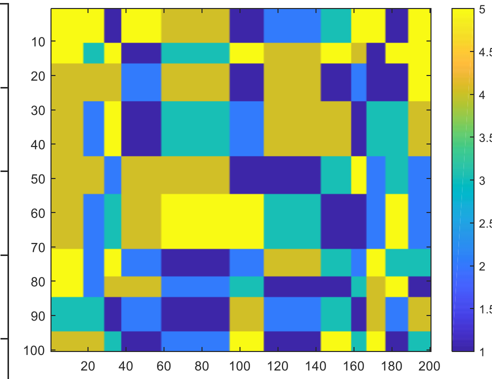
Results: Sampling for matrix completion

(1) Noiseless

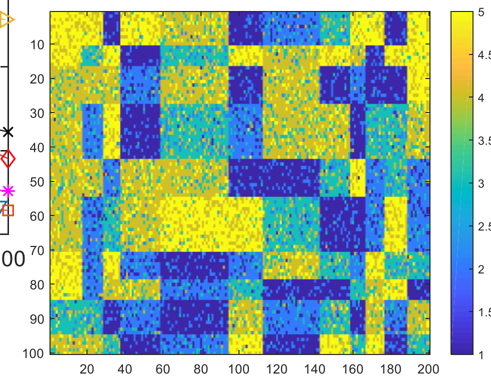


(2) Noisy





Noiseless synthetic rating matrix



Noisy synthetic rating matrix

Figure: Reconstruction MSE of different sampling methods on synthetic dataset. The reconstruction method for matrix completion is dual graph smoothness based method.

Comparison methods: PG [1]; GWC-random [2]; LOC [3]

- [1] Guillermo Ortiz-Jiménez, Mario Coutino, Sundeep Prabhakar Chepuri, and Geert Leus. "Sampling and reconstruction of signals on product graphs". arXiv preprint arXiv:1807.00145, 2018.
- [2] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," Applied and Computational Harmonic Analysis, vol. 44, no. 2, pp. 446–475, 2018.
- [3] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Eigendecomposition-free sampling set selection for graph signals,"IEEE Transactions on Signal Processing, 2019.

Results: Sampling for matrix completion

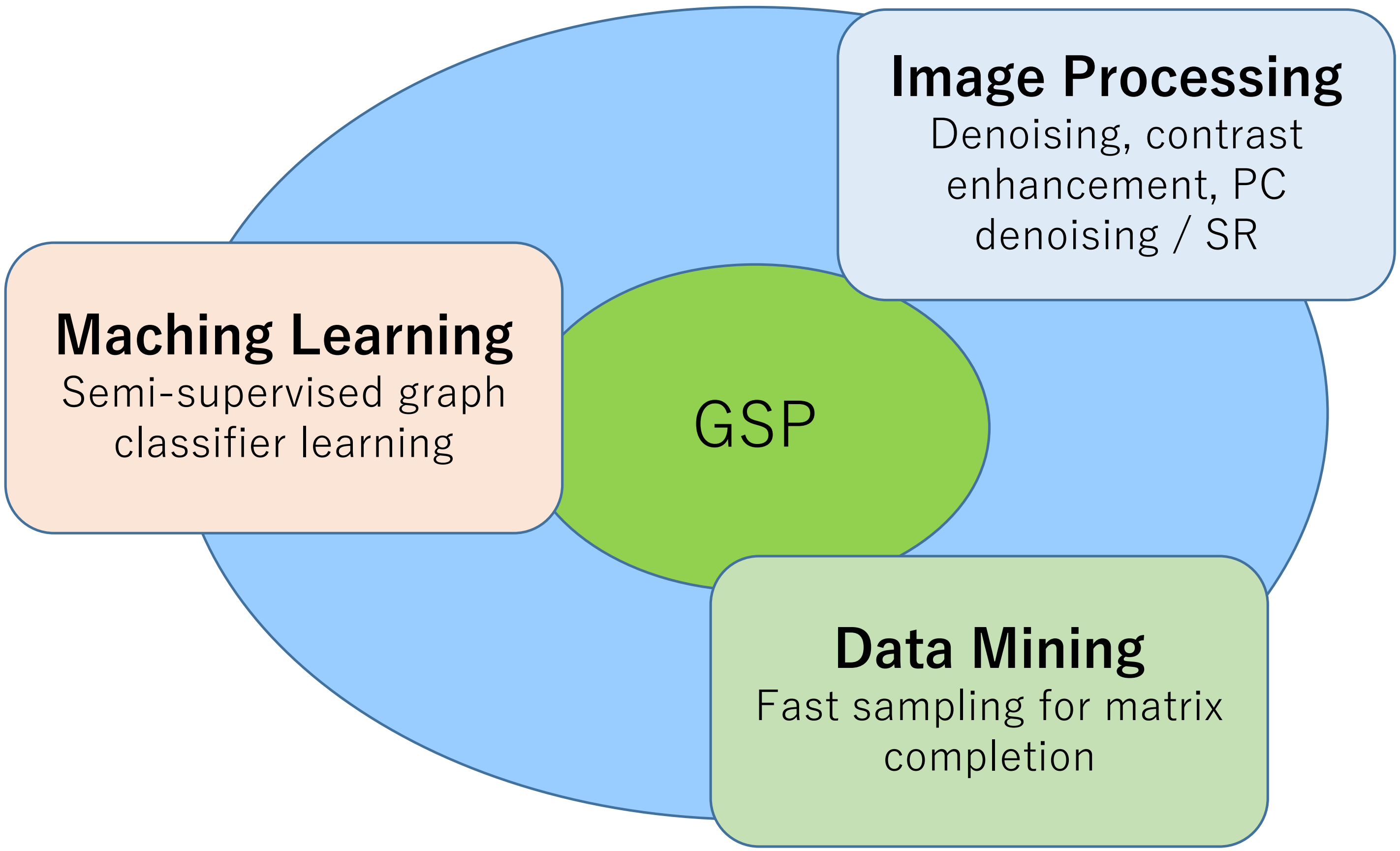
	SVT [1]	GRALS [2]	GMC [3]	NMC [4]
G1	1.021 | 1.031	0.947 | 0.931	1.036 | 1.037	0.890 | **0.888**
G2	1.021 | 0.983	0.945 | 0.893	1.118 | 1.054	0.890 | **0.858**

Table: RMSE of different matrix completion methods on **Mocielens_100k** dataset with different sampling strategies on **Feature-based graph** (G1) and **Content-based graph** (G2). In each grid, the value on left side belongs to random sampling; the right side value is of *our proposed IGCS sampling*. The best performance in each row is marked in bold and red. In our experiments, the sampling budget is 80k out of 100k available ratings; We first use random 60k samples as given, and then proceed to sample the next 20k samples base on random sampling or the proposed IGCS sampling.

- [1] J. Cai, E. J. Candes, and Z. Shen. "A singular value thresholding algorithm for matrix completion". preprint, 2008.
- [2] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon. "Collaborative filtering with graph information: Consistency and scalable methods". In Proc. NIPS, 2015.
- [3] V. Kalofolias, X. Bresson, M. M. Bronstein, and P. Vandergheynst." Matrix completion on graphs. "2014.
- [4] D. M. Nguyen, E. Tsiligianni, and N. Deligiannis, "Extendable neural matrix completion," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2018, pp. 1–5.

Summary

- Graph Spectral Analyis Tools
  - Similarity graph, graph frequencies.



Q&A

- Email: genec@yorku.ca
- Homepage: https://www.eecs.yorku.ca/~genec/index.html

Primal Sample Selection Problem

- **Optimization**: Select sample vector **a** and scalars **s**:

$$\max_{\mathbf{a},\mathbf{s}} \boxed{\min_{i\in\{1,\dots,N\}} \quad c_{ii} - \sum_{j\neq i} |c_{ij}|}$$

smallest disc left-end of **C**

$$\text{s.t.} \; \boxed{\mathbf{C} = \mathbf{S}\left(\mathbf{A} + \mu\mathbf{L}\right)\mathbf{S}^{-1}}$$

C is similar transform of coeff. matrix

$$\mathbf{A} = \mathrm{diag}(\mathbf{a}), \quad \boxed{a_i \in \{0,1\}}, \quad \boxed{\sum_{i=1}^{N} a_i \leq K,}$$

sample vector **a** is binary and within budget K

$$\mathbf{S} = \mathrm{diag}(\mathbf{s}), \quad \boxed{s_i > 0.}$$

scalars **s** are positive

- **Difficulty**: max-min objective is hard to optimize.

Dual Sample Selection Problem

- **Dual Formulation**: Select sample vector **a** and scalars **s**:

$$\min_{\mathbf{a},\mathbf{s}} \boxed{\sum_{i=1}^{N} a_i}$$

total number of samples

$$\text{s.t. } \mathbf{C} = \mathbf{S}\left(\mathbf{A} + \mu\mathbf{L}\right)\mathbf{S}^{-1}, \quad \boxed{c_{ii} - \sum_{j \neq i} |c_{ij}| \geq T, \quad \forall i}$$

$$\mathbf{A} = \text{diag}(\mathbf{a}), \qquad a_i \in \{0, 1\},$$

$$\mathbf{S} = \text{diag}(\mathbf{s}), \qquad s_i > 0.$$

all disc left-ends are at least T

- **Proposition**: If there exists threshold T s.t. optimal sol'n (**a**,**s**) to dual satisfies $\sum a_i = K$, one dual sol'n is also optimal to primal.