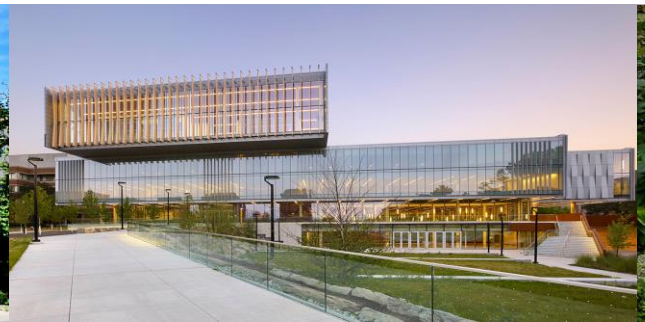
Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors

Gene Cheung
York University, Toronto, Canada

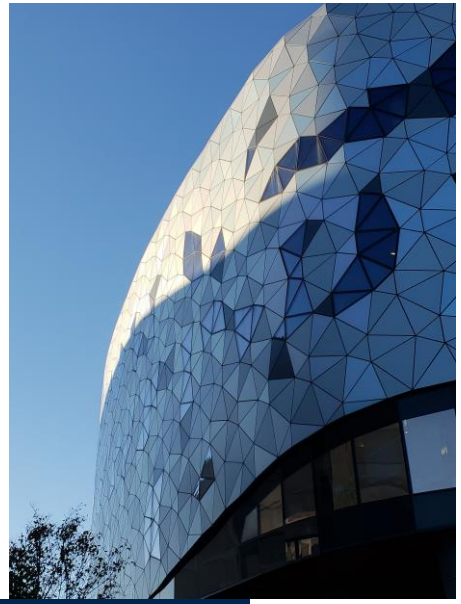
October 31, 2024

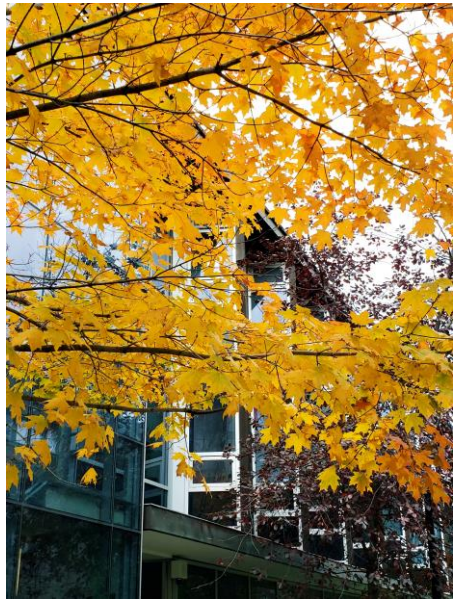


Acknowledgement

- **Graph and Image Signal Processing (GISP) Lab** (York University, Toronto, Canada)
  - Grad students: Saghar Bagheri, Tam Thuc Do, Yeganeh Gharedaghi, Niruhan Viswarupan, Parham Eftekhar, Seyed Alireza Hosseini
  - Co-supervised students: Sadid Sahami (NTHU)
  - Visiting researcher: Fei Chen (Fudan)

- **Collaborators**
  - Michael Brown, Andrew Eckford, Pirathayini Srikantha (York Univ., Canada)
  - Ivan V. Bajic (Simon Fraser Univ., Canada)
  - Antonio Ortega (Univ. of Southern California, USA)
  - Phil Chou (Google, USA)
  - Tim Eadie (Growers Edge, USA)
  - Wei Hu (Peking Univ., China), Jin Zeng (Tongji Univ., China)
  - Vicky H. Zhao (Tsinghua Univ., China)
  - Chia-Wen Lin (NTHU, Taiwan)
  - Kazuya Kodama (NII, Japan), Yuichi Tanaka (Osaka Univ. Japan)





Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

- Conclusion

Interpretable Neural Nets with fewer parameters → less training data required

Gene Cheung (genec@yorku.ca)

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

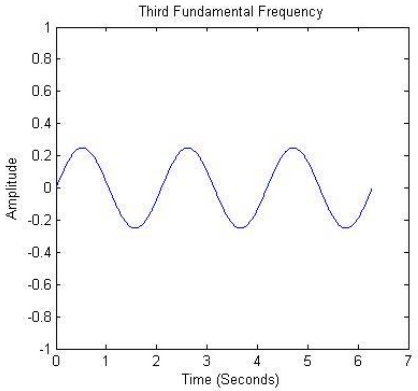
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

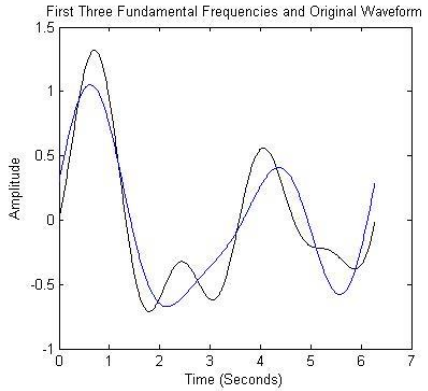
- Conclusion

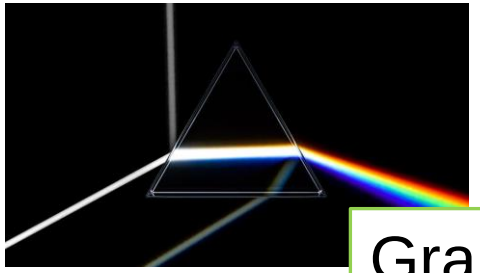
Gene Cheung (genec@yorku.ca)

Graph Signal Processing

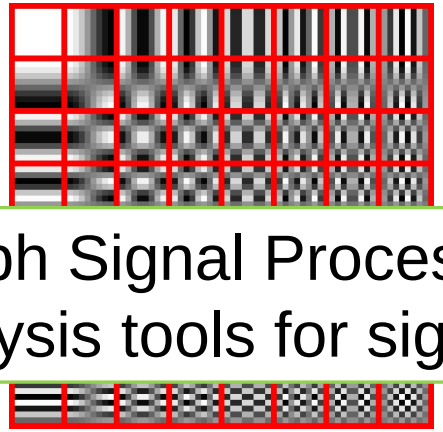
Frequency analysis + Graph kernels = Graph Signal Processing



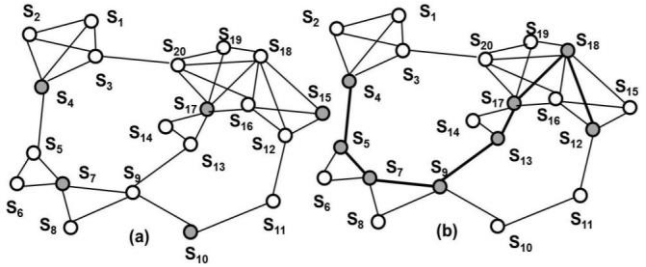


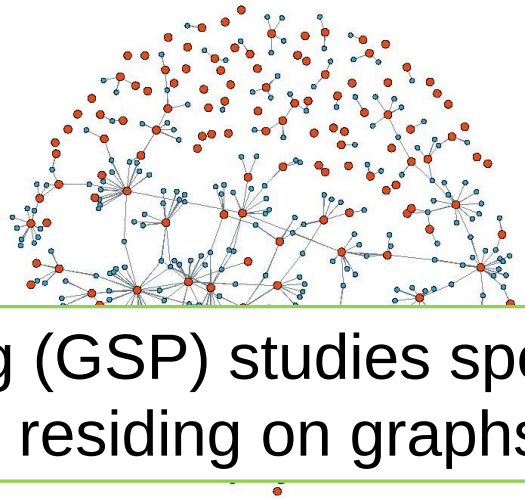


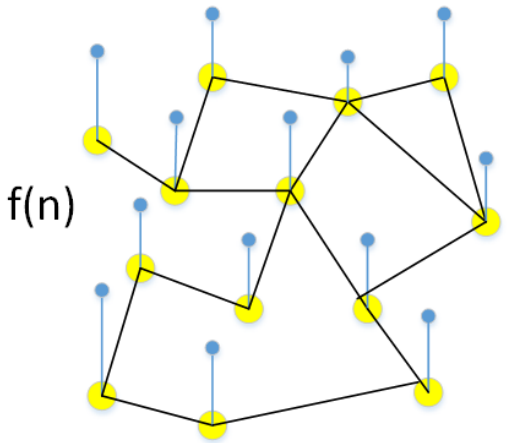
Newton decomposed white light into color components (1730).



2D-DCT basis

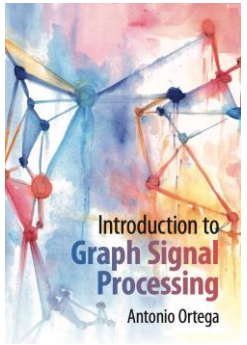


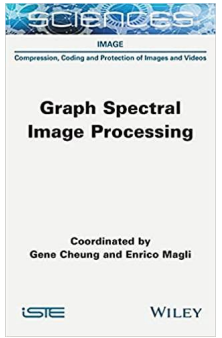




signal on graph kernel

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





[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.

Gene Cheung (genec@yorku.ca)

Graph Spectrum

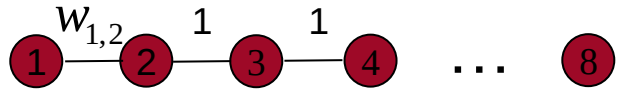
Graph Fourier modes: eigenvectors of *graph Laplacian matrix* **L = D - W**.

eigenvalues along diagonal

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

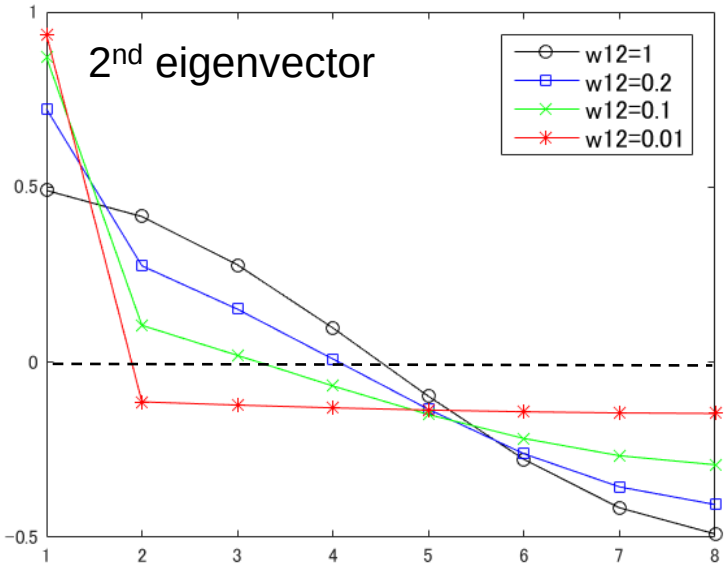
Graph Fourier Transform (GFT)

eigenvectors in columns



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

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



1. **Eigenvectors** are (***global***) aggregates of (***local***) edge weights.
   - More variations for larger eigenvalues.
2. **Eigenvalues** (≥ 0) as *graph frequencies*.

[1] G. Cheung, E. Magli, Y. Tanaka, M. Ng, "**Graph Spectral Image Processing**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907-930, May 2018. .

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

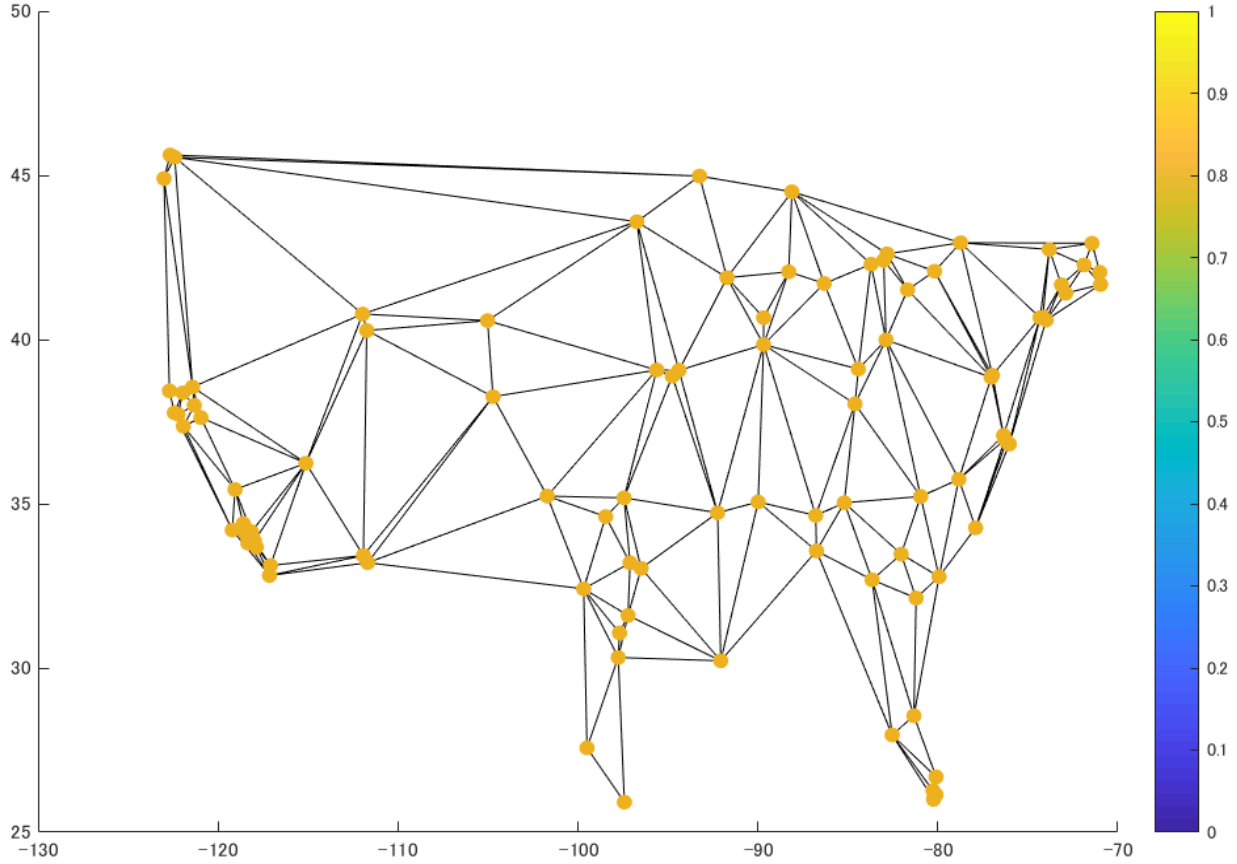
Graph connections from Delaunay Triangulation*.

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



V1: DC component

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

Gene Cheung (genec@yorku.ca)

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

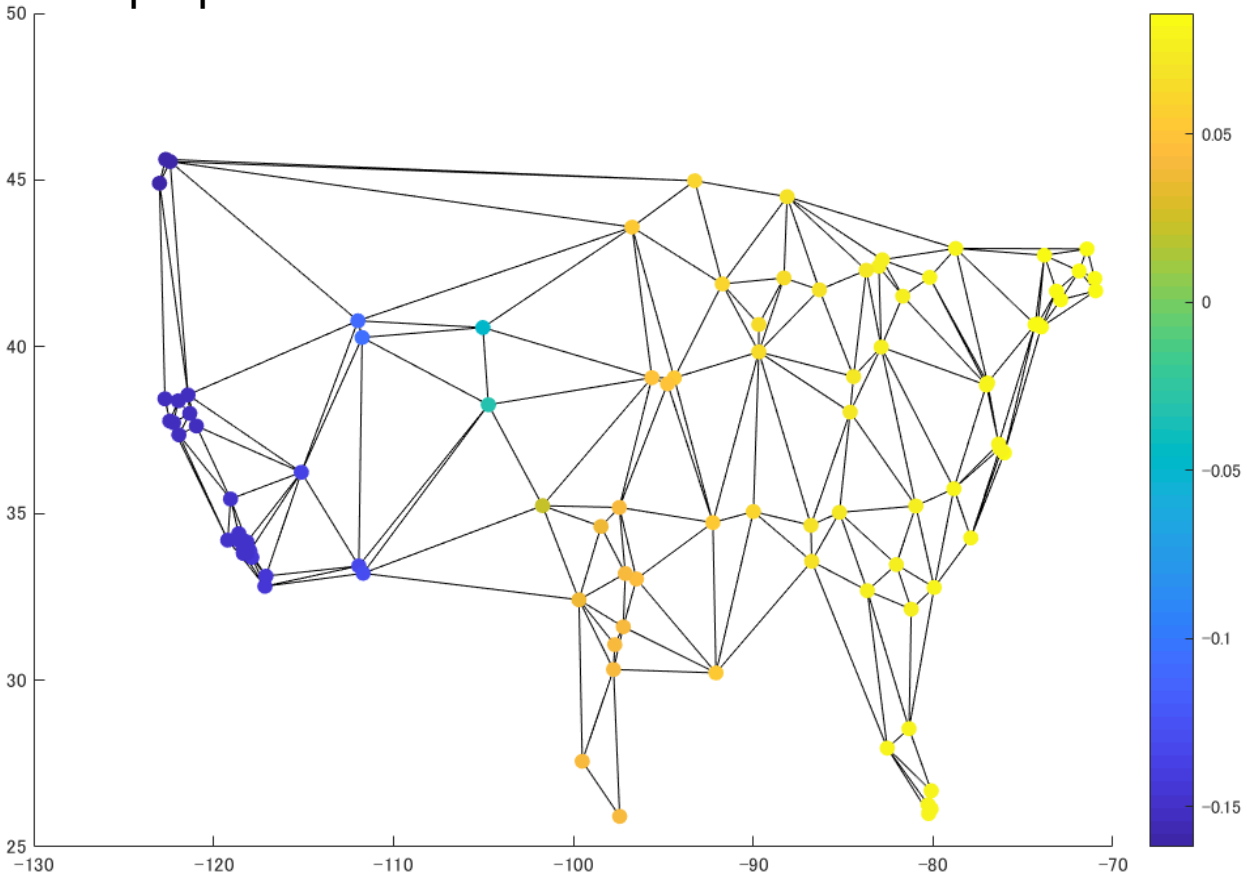
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

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

Edge weights



V2: 1st AC component

Gene Cheung (genec@yorku.ca)

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

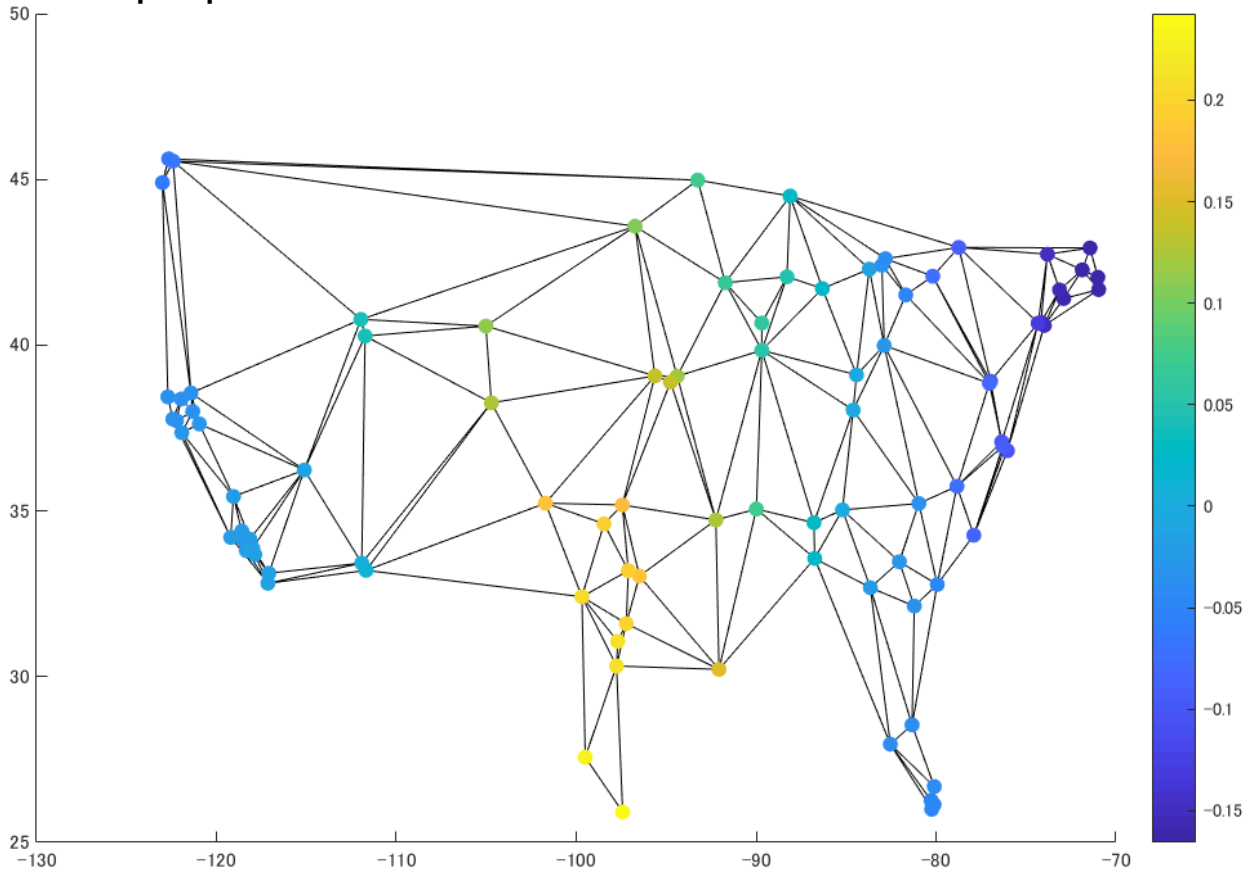
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

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

Edge weights



V3: 2nd AC component

Gene Cheung (genec@yorku.ca)

Graph Frequency Examples (US Temperature)

Weather stations from 100 most populated cities.

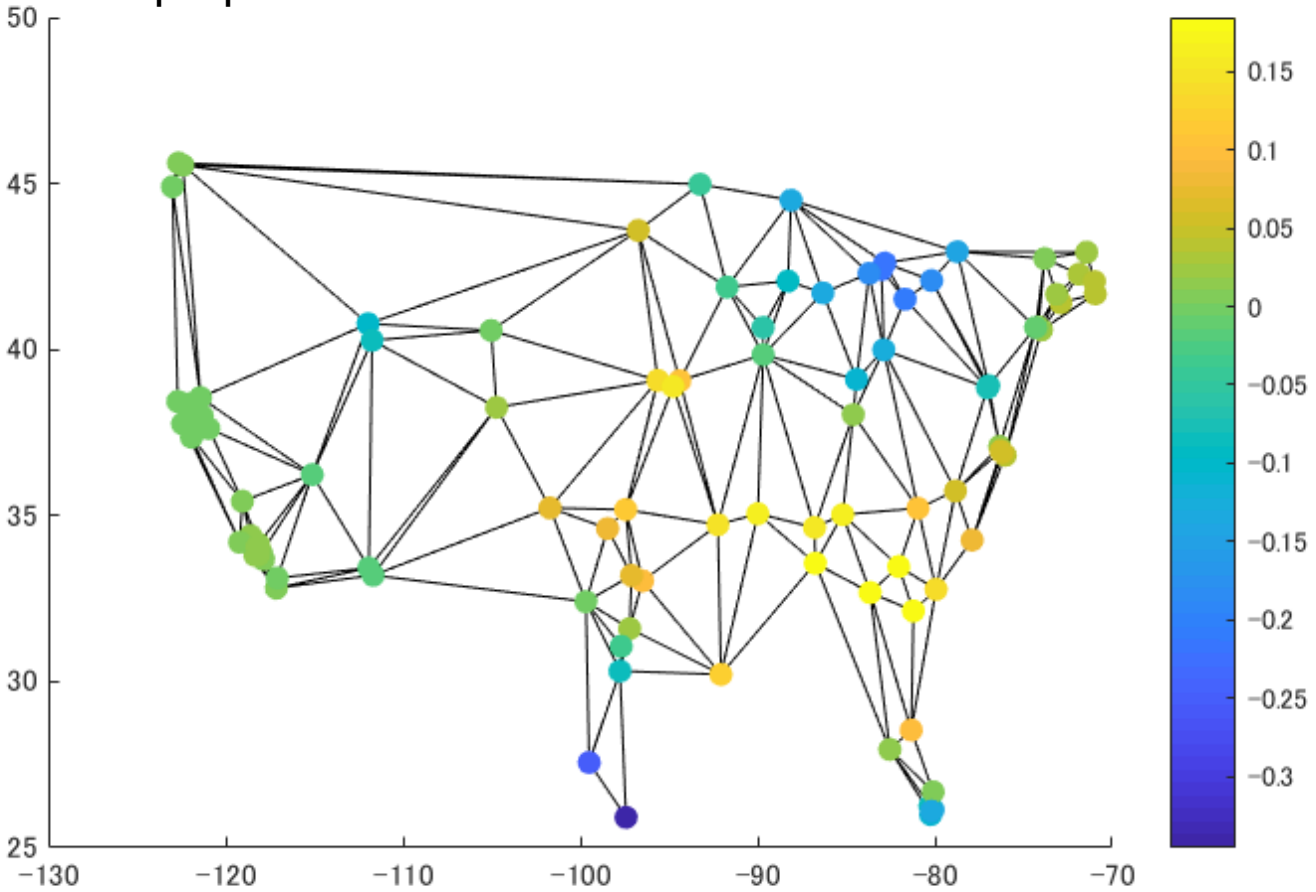
Graph connections from Delaunay Triangulation*.

Edge weights inverse proportional to distance.

location diff.

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

Edge weights



V4: 9th AC component

Gene Cheung (genec@yorku.ca)

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

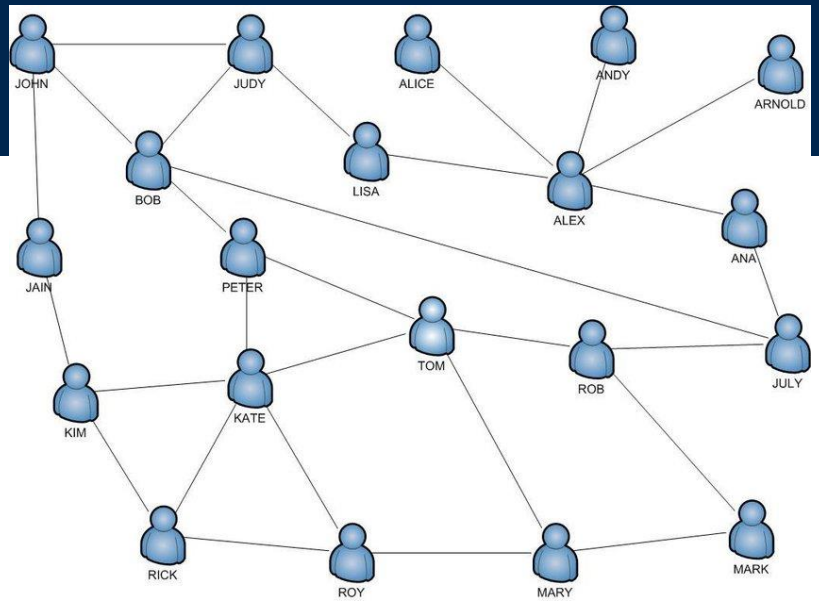
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

- Conclusion

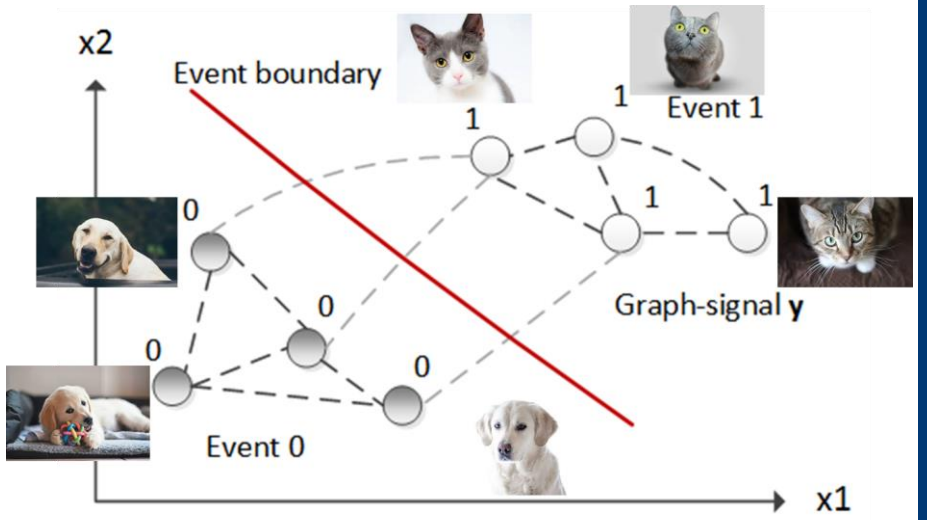
Gene Cheung (genec@yorku.ca)

Graph Construction

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



- **Graph Learning from Data**:
  1. Learn sparse **inverse covariance matrix** from observations [1].
     - Graphical Lasso, CLIME.
  2. Learn metric to determine **feature distance** [2].



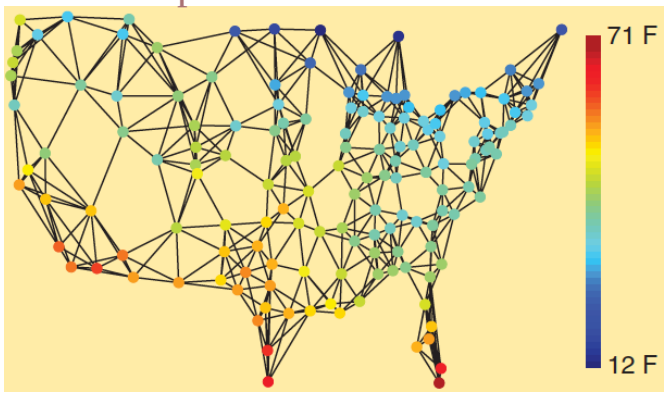
[1] S. Bagheri, G. Cheung, A. Ortega, F. Wang, "**Learning Sparse Graph Laplacian with *K* Eigenvector Prior via Iterative GLASSO and Projection**," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, June 2021 (**best student paper finalist**).

[2] C. Yang, G. Cheung, W. Hu, "**Signed Graph Metric Learning via Gershgorin Disc Alignment**," *IEEE TPAMI*, 2022.

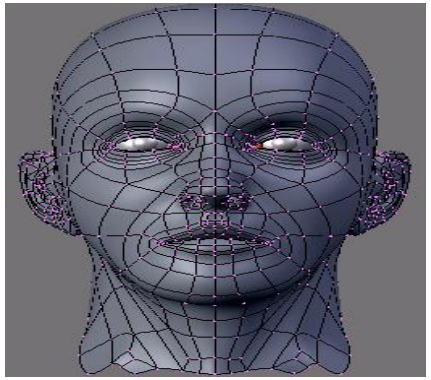
Signed Graphs

- Most GSP works assume **positive graphs**.

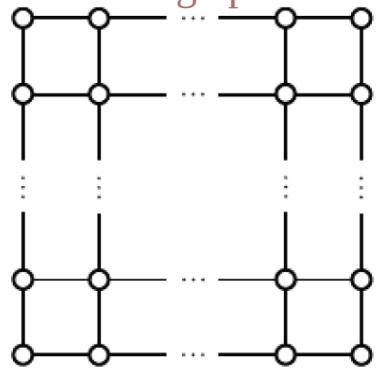
Temperature sensor network



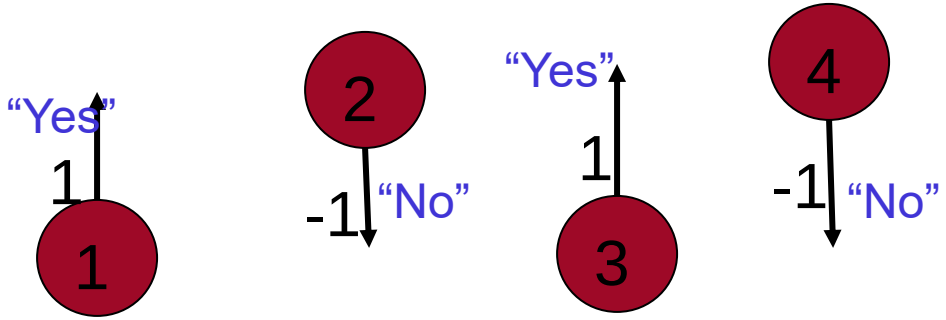
3D geometry

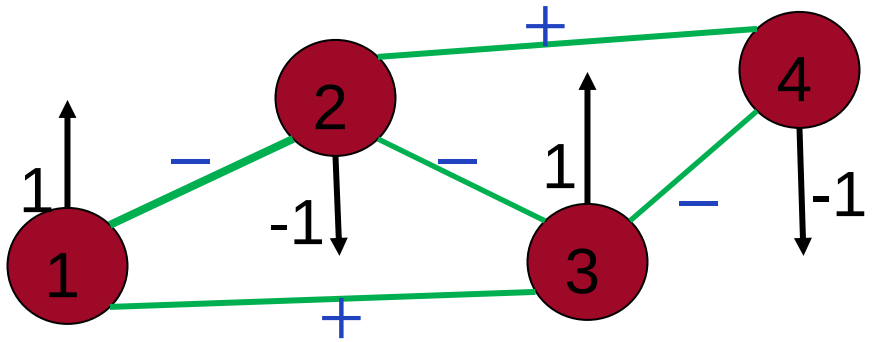


2D image pixels



- Voting records in a Parliament ⟶ ***anti-correlation*** represented as **negative edges** [1,2].





Signed graph

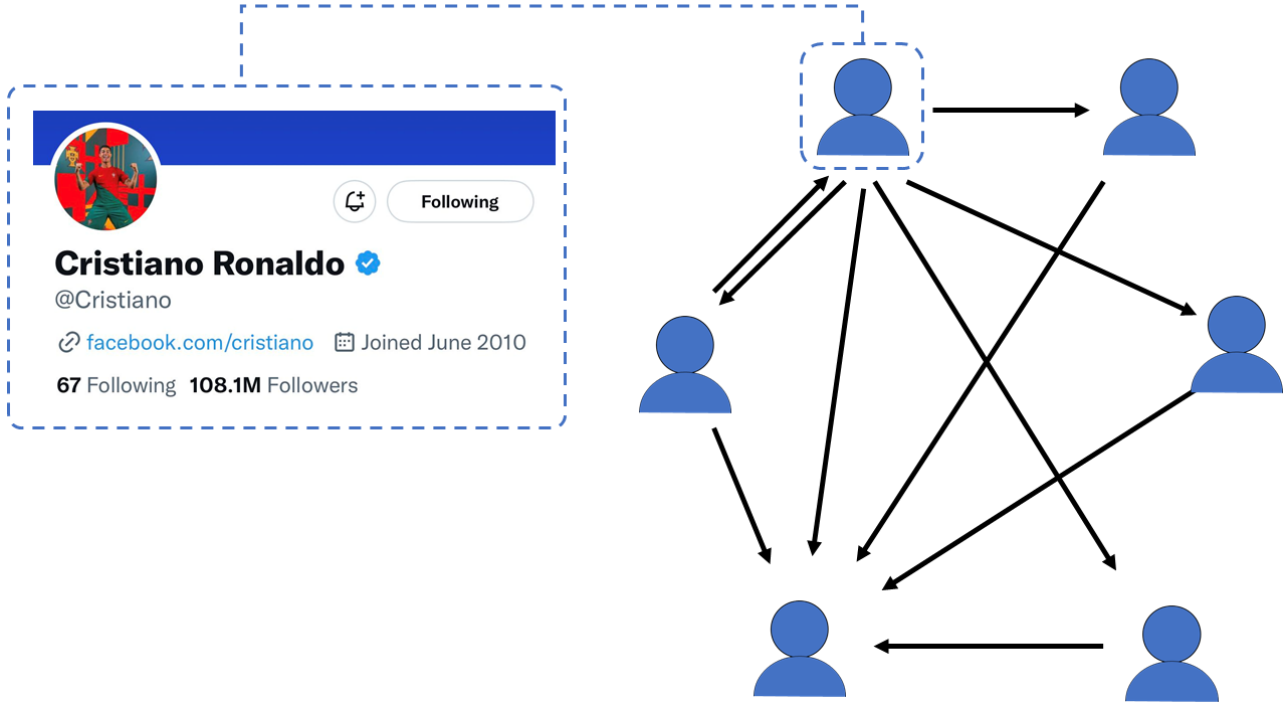
[1] C. Dinesh, G. Cheung, I. V. Bajic, "**Point Cloud Sampling via Graph Balancing and Gershgorin Disc Alignment**," *IEEE TPAMI* vol. 45, no.1, pp. 868-886, January 2023.

[2] C. Dinesh, G. Cheung, S. Bagheri, I. V. Bajic, "**Efficient Signed Graph Sampling via Balancing & Gershgorin Disc Perfect Alignment**," submitted to *IEEE TPAMI*, January 2023.

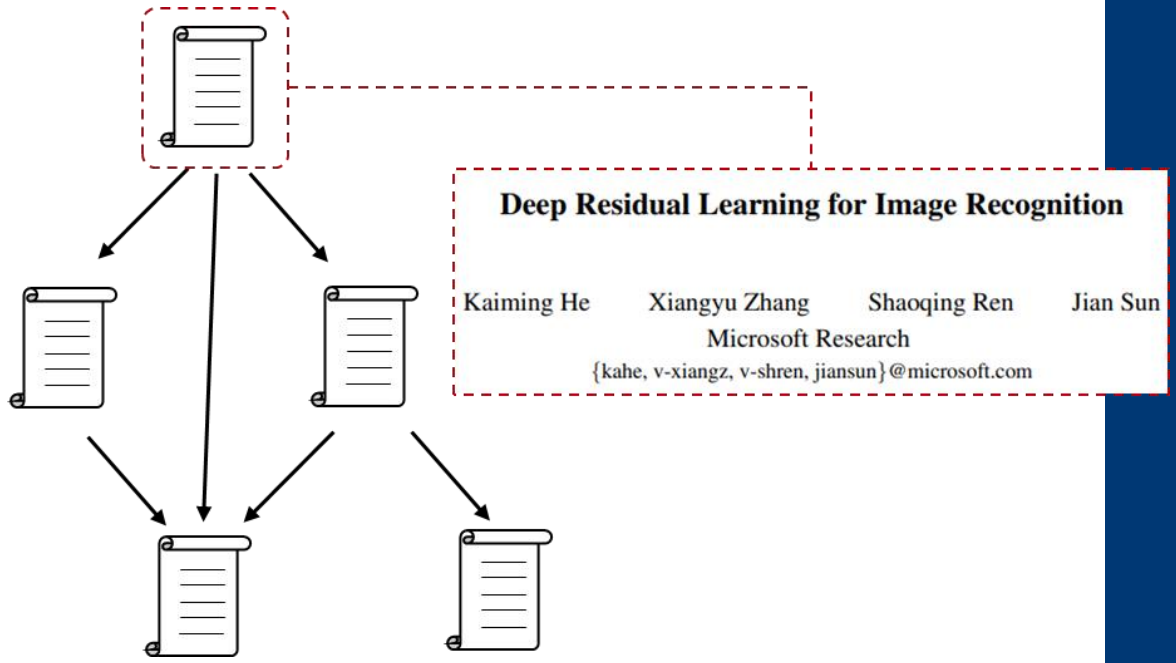
Directed Graphs

- Describes ***causal*** relationships.

(a) Following Network in Twitter/X



(b) Paper Citation Network



[1] Y. Li, H. V. Zhao, G. Cheung, "**Eigen-Decomposition-Free Directed Graph Sampling via Gershgorin Disc Alignment**," *ICASSP'23*, Rhodes, Greece, June 2023.

[1] C. Dinesh, G. Cheung, F. Chen, Y. Li, H. V. Zhao, "**Modeling Viral Information Spreading via Directed Acyclic Graph Diffusion**," *IEEE Globecom*, Malaysia, December 2023.

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

- Conclusion

Gene Cheung (genec@yorku.ca)

Spectral Graph Filter for Image Denoising

- **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

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

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

smooth signal — low-pass signal

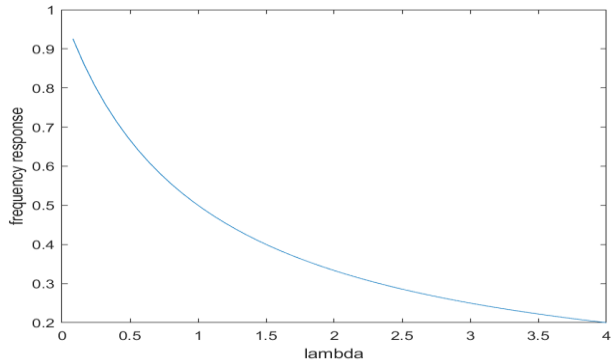
$$(\mathbf{I}+\mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

Solvable in linear time via CG

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

$$\mathbf{x}^* = \mathbf{V}\mathrm{diag}(1+\mu\lambda_1, 1+\mu\lambda_2, \dots)^{-1}\mathbf{V}^T\mathbf{y}$$

low-pass filter!



[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 ($\sigma_I = 40$)

Original

Noisy, 16.48 dB

K-SVD, 26.84 dB

BM3D, 27.99 dB

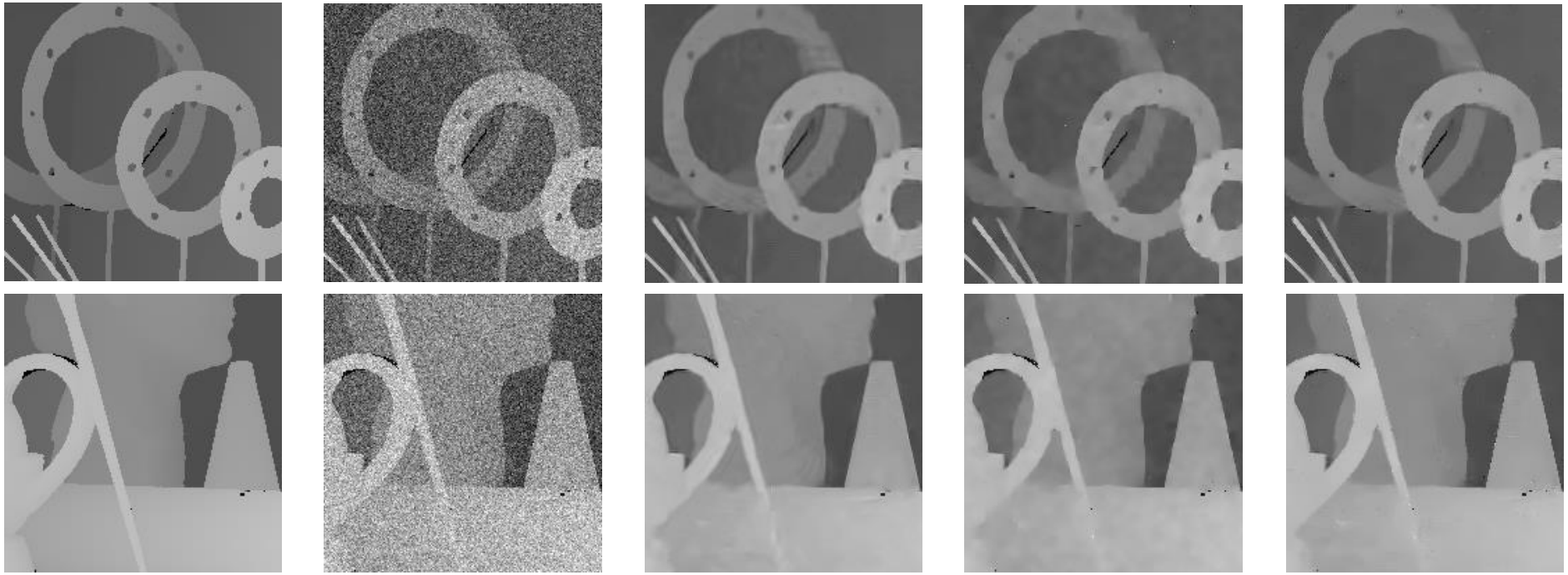
PLOW, 28.11 dB

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.

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

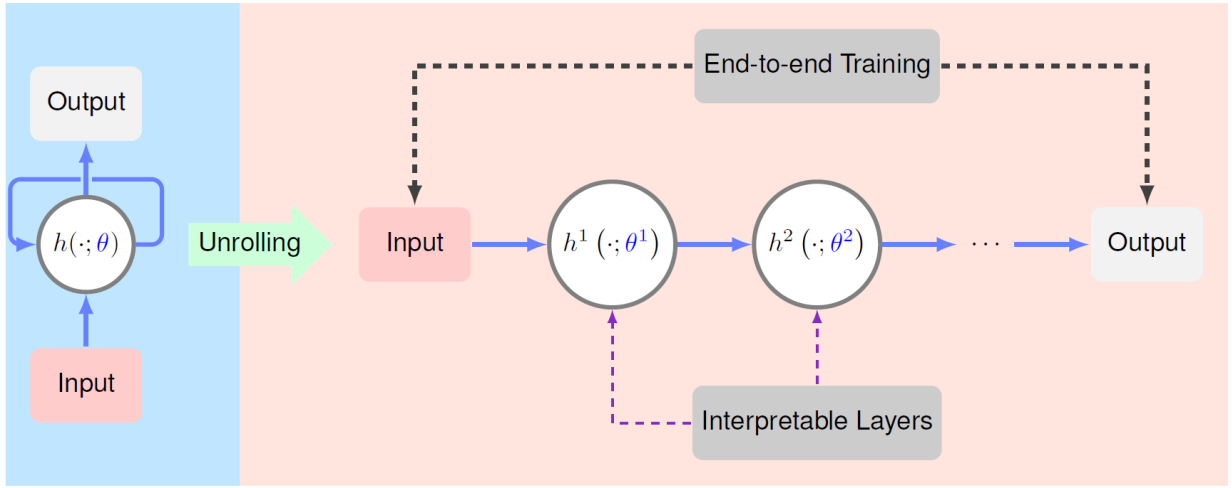
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

- Conclusion

Gene Cheung (genec@yorku.ca)

GSP with Algorithm Unrolling

- **Algorithm Unrolling:** implements each iteration of an iterative algorithm as neural layer + parameter tuning [2].
  - e.g., ISTA to LISTA [1].
  - *100% mathematically interpretable*.
  - Train *fewer* parameters.
  - Robust to **covariate shift**.



- **"White-box" transformer** by unrolling ***sparse rate reduction*** algorithm [3].

Our approach:
1. design GSP algorithms,
2. unroll + parameter tuning.

[1] J. K. Gregor and Y. LeCun, "**Learning fast approximations of sparse coding**," *ICML*, Madison, WI, USA, 2010, ICML'10, p. 399–406.
[2] V. Monga, Y. Li, and Yonina C. Eldar, "**Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing**," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
[3] Y. Yu, S. Buchanan, D. Pai, T. Chu, Z. Wu, S. Tong, B.D. Haeffele, Y. Ma, "**White-box transformers via sparse rate reduction**," ArXiv abs/2306.01129 (2023).

GLR for Image Interpolation

- Formulate interpolation problem using **GLR** as objective:

GLR → $\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{L}\mathbf{x}, \quad \text{s.t. } \mathbf{H}\mathbf{x} = \mathbf{y}$ ← sampling matrix; ← observation

- Define corresponding unconstrained Lagrangian function:

Lagrange multiplier →

$$f(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{x}^\top \mathbf{L}\mathbf{x} + \boldsymbol{\mu}^\top (\mathbf{H}\mathbf{x} - \mathbf{y})$$

- Take derivative w.r.t. x and μ, set to zero, get **linear system**:

$$\underbrace{\begin{bmatrix} 2\mathbf{L} & \mathbf{H}^\top \\ \mathbf{H} & \mathbf{0}_{K,K} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{N,N} \\ \mathbf{y} \end{bmatrix} \quad \text{or} \quad \mathbf{L}_{\bar{\mathcal{S}},\bar{\mathcal{S}}}\, \mathbf{x}_{\bar{\mathcal{S}}} = -\mathbf{L}_{\bar{\mathcal{S}},\mathcal{S}}\, \mathbf{y}$$

index set of non-samples ← ; index set of samples →

- Solve via **Conjugate Gradient** (CG) in *linear time**.
- Can interpret output $\mathbf{x}^*$ as **LP filter** of upsampled input, $\mathbf{L}^{-1}\mathbf{H}^\top \mathbf{L}_{\mathcal{S}}^{\#}\mathbf{y}$.

LP filter ↓ ; upsampling operator ↓

[1] Do, Tam Thuc, et al. "**Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors**." accepted to *NeurIPS'24*.

Gene Cheung (genec@yorku.ca)

GTV for Image Interpolation

- Formulate interpolation problem using **GTV** as objective:

GTV → $\|\mathbf{Cx}\|_1$, sampling matrix → $\mathbf{H}$, observation → $\mathbf{y}$

$$\min_{\mathbf{x}} \|\mathbf{Cx}\|_1, \quad \text{s.t. } \mathbf{Hx} = \mathbf{y}$$

- Rewrite as standard form of ***linear programming*** (LP):

$$\min_{\mathbf{z},\mathbf{x},\mathbf{q}} \mathbf{1}_M^\top \mathbf{z}, \quad \text{s.t.} \underbrace{\begin{bmatrix} \mathbf{I}_M & -\mathbf{C} & -(\mathbf{I}_M \ \ \mathbf{0}_{M,M}) \\ \mathbf{I}_M & \mathbf{C} & -(\mathbf{0}_{M,M} \ \ \mathbf{I}_M) \\ \mathbf{0}_{K,M} & \mathbf{H} & \mathbf{0}_{K,2M} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{z} \\ \mathbf{x} \\ \mathbf{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_M \\ \mathbf{0}_M \\ \mathbf{y} \end{bmatrix}}_{\mathbf{b}}, \quad \mathbf{q} \geq \mathbf{0}_{2M}$$

- Solve ***sparse LP*** (SLP) via **ADMM** in linear time [2].
- Can interpret output as **LP filter** of upsampled input:

LP filter → $(\mathbf{C}^\top\mathbf{C} + \mathbf{H}^\top\mathbf{H})\mathbf{x}^{t+1}$; upsampling operator → $\mathbf{H}^\top\mathbf{y}$

$$(\mathbf{C}^\top\mathbf{C} + \mathbf{H}^\top\mathbf{H})\mathbf{x}^{t+1} = \frac{1}{2\gamma}\mathbf{C}^\top\left(\boldsymbol{\mu}_a^t - \boldsymbol{\mu}_b^t + \boldsymbol{\mu}_d^t - \boldsymbol{\mu}_e^t\right) - \frac{1}{\gamma}\mathbf{H}^\top\boldsymbol{\mu}_c^t - \frac{1}{2}\mathbf{C}^\top(\tilde{\mathbf{q}}_1^t - \tilde{\mathbf{q}}_2^t) + \mathbf{H}^\top\mathbf{y}$$

[1] Do, Tam Thuc, et al. "**Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors**." accepted to *NeurIPS'24*.
[2] Sinong Wang and Ness Shroff, "**A new alternating direction method for linear programming**," *NeurIPS'17*.

Graph Learning from Data

- **Graph Edge Definition**: exponential of feature distance.

$$w_{i,j} = \exp\left(-d(i,j)\right), \qquad d(i,j) = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)$$

PSD metric matrix

edge weight

Mahalanobis distance

feature vector
(computed via feature function from input embedding)

- With random walk **normalization**, then

$$\bar{w}_{i,j} = \frac{\exp(-d(i,j))}{\sum_{l|(i,l)\in\mathcal{E}} \exp(-d(i,l))}.$$

edge set stemming from node *i*

[1] Do, Tam Thuc, et al. "**Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors**." accepted to *NeurIPS'24*.

Self-Attention in Transformer

- **Graph Edge Definition**: exponential of feature distance.

$$w_{i,j} = \exp\left(-d(i,j)\right), \qquad d(i,j) = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)$$

  - With normalization, then

$$\bar{w}_{i,j} = \frac{\exp(-d(i,j))}{\sum_{l|(i,l)\in\mathcal{E}} \exp(-d(i,l))}.$$

- **Self-Attention Mechanism** in Transformer: dot product of linear transformed embeddings, using ***key*** and ***query*** matrices, **K** and **Q**:

1. Graph learning with normalization from data is a self-attention mechanism!

transformed dot product

$$\sum_{l=1} \exp(c(i,l)) \qquad \mathbf{x}_j)^\top (\mathbf{K}\mathbf{x}_i).$$

[1] Do, Tam Thuc, et al. "**Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors**." accepted to *NeurIPS'24*.

Gene Cheung (genec@yorku.ca)

Output Embedding in Transformer

- **Output Embedding** in transformer: using *value* matrix **V**,

$$\mathbf{y}_i = \sum_{l=1}^{N} a_{i,l} \mathbf{x}_l \mathbf{V}$$

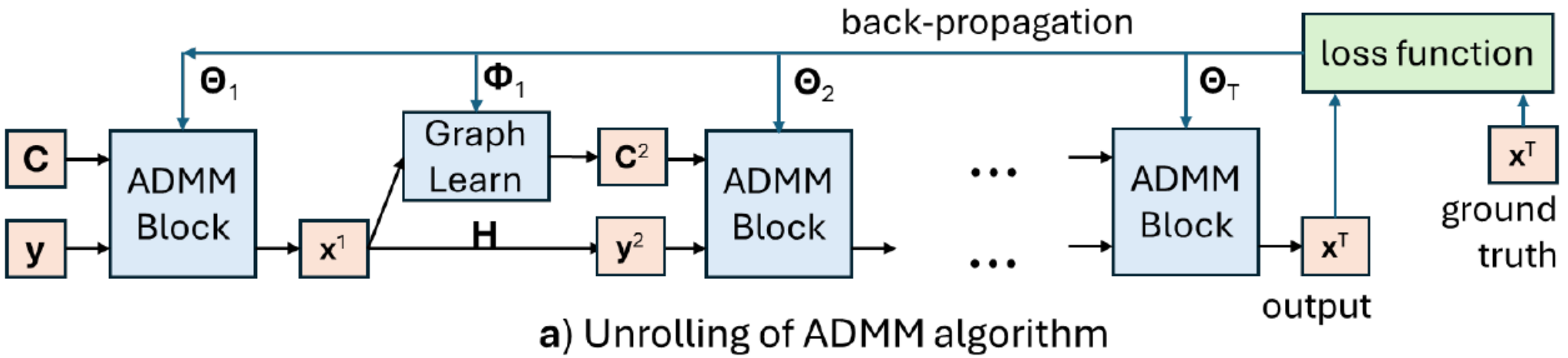
- **Interpretation**:
  - Pairwise similarities (affinities) define *directed* graph.
  - Value matrix define filter response.
- **Graph Approach**:
  - *Undirected* sparse graph via low-dimensional feature vectors $\mathbf{f}_i$.
  - Derived low-pass filter from optimization (no learning!).

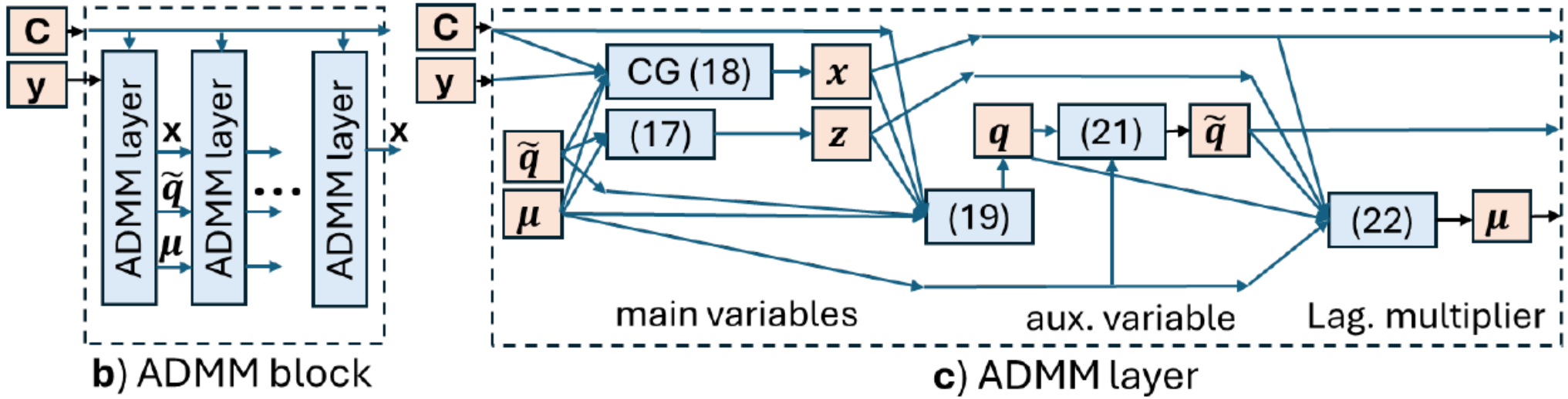
2. Unrolling of graph-based algorithm leads to **lightweight**, interpretable transformer!

[1] Do, Tam Thuc, et al. "**Interpretable Lightweight Transformer via Unrolling of Learned Graph Smoothness Priors**." accepted to *NeurIPS'24*.

Gene Cheung (genec@yorku.ca)

Unrolling GTV-based ADMM Alg. for Image Interpolation





Gene Cheung (genec@yorku.ca)

Experiments: Unrolled GLR/GTV for Demosaicking

Method	Params#	McM		Kodak		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bilinear	-	29.71	0.9304	28.22	0.8898	24.18	0.8727
RST-B [46]	931763	34.85	0.9543	38.75	0.9857	32.82	0.973
RST-S [46]	3162211	35.84	0.961	**39.81**	**0.9876**	33.87	0.9776
Menon [47]	-	32.68	0.9305	38.00	0.9819	31.87	0.966
Malvar [48]	-	32.79	0.9357	34.17	0.9684	29.00	0.9482
iGLR	-	29.39	0.8954	27.50	0.8487	23.13	0.8406
iGTV	-	30.43	0.8902	28.66	0.8422	24.91	0.8114
uGLR	323410	36.09	0.9650	37.88	0.9821	33.60	0.9772
uGTV	323435	**36.59**	**0.9665**	39.11	0.9855	**34.01**	**0.9792**

Table 2: Demosaicking performance for different models, trained on 10k sample dataset.

- **Demosaicking**: fill in missing color pixels.
- Compared with two variants of RSTCANet employing Swin Transformer [1].
- Models trained on subset of DIV2K with 10K of 64x64 patches and same number of epochs (30).
- uGTV *employed 10% parameters of **RSTCANet** w/ comparable demosaicking performance*.

[1] Wenzhu Xing and Karen Egiazarian, "**Residual swin transformer channel attention network for image demosaicing**," *10th EUVIP. IEEE*, 2022, pp. 1–6.

Gene Cheung (genec@yorku.ca)

Experiments: Unrolled GLR/GTV for Demosaicking

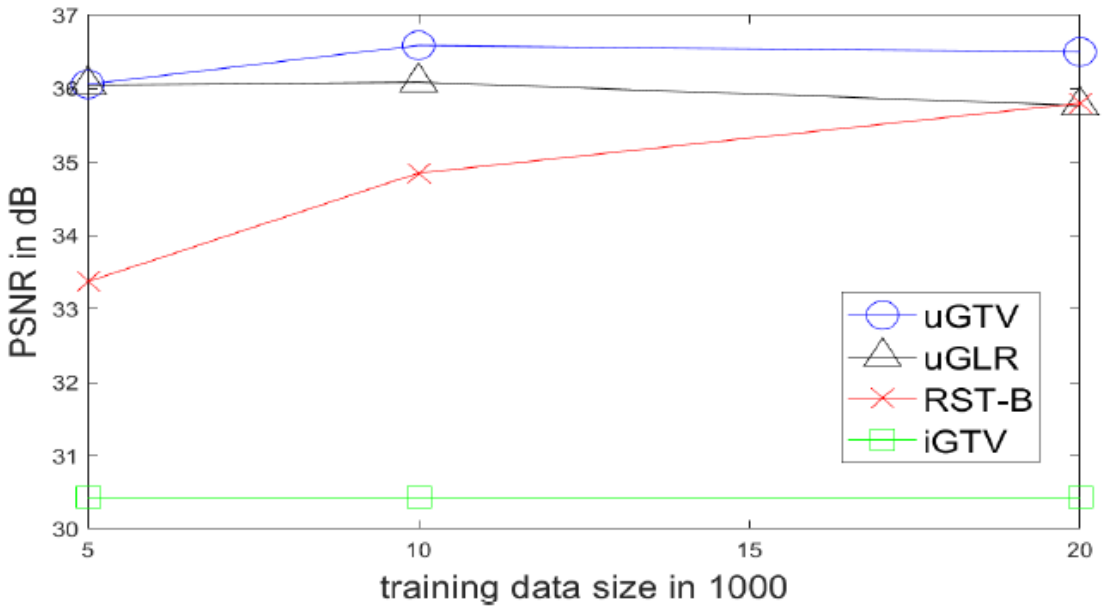


Figure 2: Demosaicking performance vs. training size for different models.

Method	McM PSNR			
	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 50$
RST-B	28.01	22.7	19.34	15.03
uGLR	28.24	22.84	19.49	15.203
uGTV	**28.31**	**22.89**	**19.56**	**15.38**

Table 1: Demosaicking performance in noisy scenario, where models are trained on noiseless dataset.

- **Demosaicking**: fill in missing color pixels.
- Requires less data to train.
- More robust to covariate shift.

[1] Wenzhu Xing and Karen Egiazarian, "**Residual swin transformer channel attention network for image demosaicing**," *10th EUVIP. IEEE*, 2022, pp. 1–6.

Gene Cheung (genec@yorku.ca)

Experiments: Unrolled GLR/GTV for Interpolation

Method	Params#	McM		Kodak		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	-	29.01	0.8922	26.75	0.8299	22.95	0.7911
MAIN [49]	10942977	32.72	0.822	28.23	0.728	25.46	0.806
SwinIR-lightweight [50]	904744	32.24	0.9354	28.62	0.8794	25.08	0.8553
iGLR	-	28.53	0.8537	26.71	0.8005	22.87	0.7549
iGTV	-	30.41	0.887	28.05	0.832	24.26	0.7855
uGLR	319090	33.31	0.9431	**29.10**	0.8870	25.94	0.8777
uGTV	319115	**33.36**	**0.9445**	29.08	**0.8888**	**26.12**	**0.8801**

Table 3: Interpolation performance for different models, trained on 10k sample dataset.

- **Interpolation**: interploated a LR image to a corresponding HR image.
- Models trained on subset of DIV2K with 10K of 64x64 patches and same number of epochs (15).
- Outperformed MAIN [1] in all three benchmark datasets by about 0.7 dB.
- uGTV *employed 3% parameters of **MAIN***.

[1] J. Ji, B. Zhong, and K.-K. Ma, "**Image interpolation using multi-scale attention-aware inception network**," *IEEE TIP*, vol. 29, pp.9413–9428, 2020.

Gene Cheung (genec@yorku.ca)

Unrolling PnP Gradient GLR for Image Restoration

- **Linear Image Formation:**

degradation matrix

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

additive noise

- Regularize ill-posed restoration problem with **Gradient GLR** [2]:

GLR on pixel row/col gradients

$$\min_{\mathbf{x}} \; \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu\, \mathbf{x}^\top \underbrace{\left(\sum_{k=1}^{N} \mathbf{H}_k^\top \mathcal{L}_{r,k} \mathbf{H}_k + \mathbf{G}_k^\top \mathcal{L}_{c,k} \mathbf{G}_k \right)}_{\mathcal{L}} \mathbf{x}$$

- Introduce ***extra*** auxiliary variables, solve using **PnP ADMM** algorithms [3]:

$$\min_{\mathbf{x},\mathbf{z}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu\, \mathbf{z}\mathcal{L}\mathbf{z} + \boldsymbol{\lambda}^\top (\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}\|_2^2$$

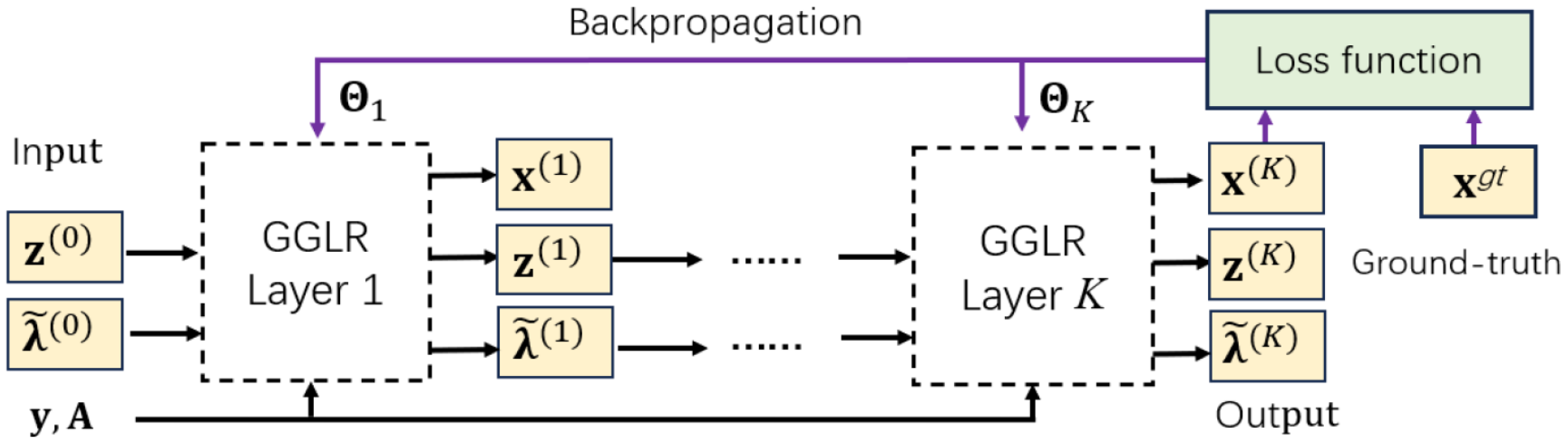
aux var.

Lagrange multiplier

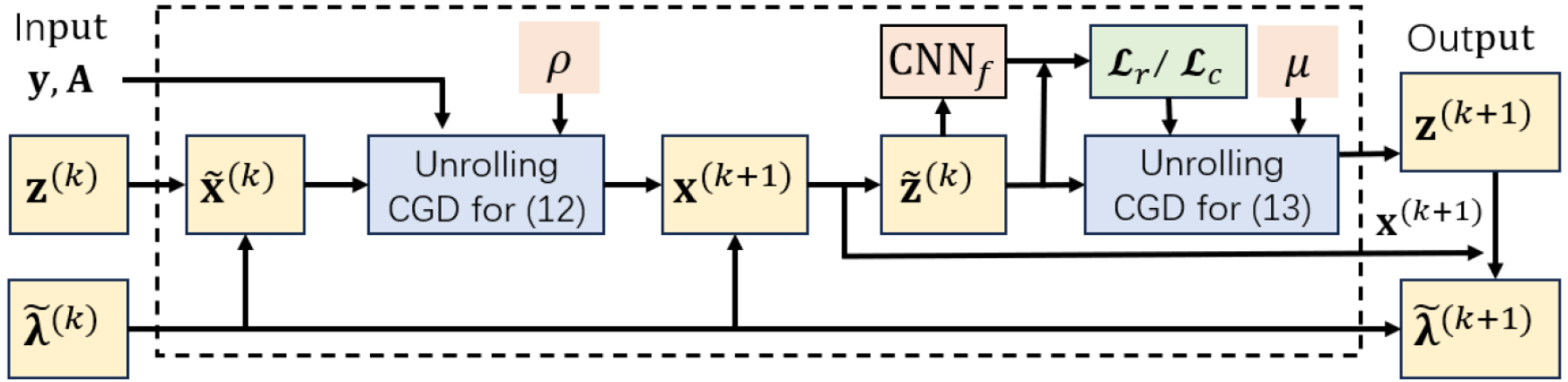
[1] J. Cai, G. Cheung, F. Chen. "**Unrolling Plug-and-Play Gradient Graph Laplacian Regularizer for Image Restoration**," submitted to *TIP*.
[2] F. Chen, G. Cheung and X. Zhang, "**Manifold Graph Signal Restoration Using Gradient Graph Laplacian Regularizer**," in *IEEE TSP*, 2024.
[3] S. H. Chan, X. Wang and O. A. Elgendy, "**Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications**," *IEEE TCI*, 2017,

Gene Cheung (genec@yorku.ca)

Unrolling PnP Gradient GLR for Image Restoration



(a) Unrolling Plug-and-Play GGLR



(b) GGLR Layer k

Gene Cheung (genec@yorku.ca)

Experiments: Unrolling PnP GGLR for Image Denoising

- Trained on Gaussian noise.
- *Employ <1% parameters of **Restormer** [1] w/ comparable denoising performance*.

TABLE I
AVERAGE PSNR(DB) RESULTS OF DIFFERENT METHODS FOR NOISE LEVELS 15, 25, AND 50 ON CBSD68 [47] DATASET.

Method	Paras(M)	CBSD68 [47]		
	-	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
CBM3D [37]	-	33.49/0.922	30.68/0.867	27.35/0.763
TWSC [38]	-	33.41/0.918	30.64/0.867	27.43/0.763
NSS [39]	-	33.33/0.918	30.76/0.868	27.61/0.769
CDnCNN [26]	0.56	33.90/0.929	31.22/0.883	27.95/0.790
IRCNN [48]	4.75	33.87/0.928	31.18/0.882	27.86/0.789
FFDNet [27]	0.67	33.85/0.929	31.22/0.883	27.98/0.792
DeepGLR [40]	0.93	33.75/0.926	31.13/0.881	27.86/0.791
DeepGTV [41]	**0.10**	31.80/0.927	31.08/0.879	27.90/0.791
DRUNet [49]	32.64	34.30/0.934	31.69/0.893	28.51/0.810
Restormer [25]	26.11	**34.39/0.935**	**31.78/0.894**	**28.59/0.813**
UPnPGGLR	0.23	34.15/0.932	31.42/0.889	28.18/0.802

[1] S. W. Zamir et al., "**Restormer: Efficient transformer for high-resolution image restoration**," *CVPR*, pp.5718–5729, 2022.

Gene Cheung (genec@yorku.ca)

Experiments: Unrolling PnP GGLR for Image Denoising

TABLE II

EVALUATION OF CROSS-DOMAIN GENERALIZATION FOR REAL IMAGE DENOISING ON RENOIR [50] AND NAM-CC15 [51] DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE.

Dataset	Method	CDnCNN [26]	IRCNN [48]	FFDNet [27]	DeepGLR [40]	DeepGTV [41]	DRUNet [49]	Restormer [25]	UPnPGGLR
RENOIR [50]	PSNR	26.79	25.77	29.19	30.25	30.14	29.81	29.55	**30.38**
	SSIM	0.638	0.651	0.762	0.809	0.808	0.784	0.780	**0.814**
Nam-CC15 [51]	PSNR	33.86	35.68	34.00	35.85	35.98	35.52	35.32	**36.26**
	SSIM	0.864	0.932	0.906	0.935	0.932	0.927	0.921	**0.939**

- Trained on Gaussian noise, tested on real noise dataset RENOIR, Nam-CC15.
- UPnPGGLR surpasses Restormer by 0.83dB on RENOIR, by 0.94dB on Nam-CC.
- Restormer is *overfitted* to Gaussian noise and fails to generalize to real noise.

[1] S. W. Zamir et al., "**Restormer: Efficient transformer for high-resolution image restoration**," *CVPR*, pp.5718–5729, 2022.

Experiments: Unrolling PnP GGLR for Image Denoising

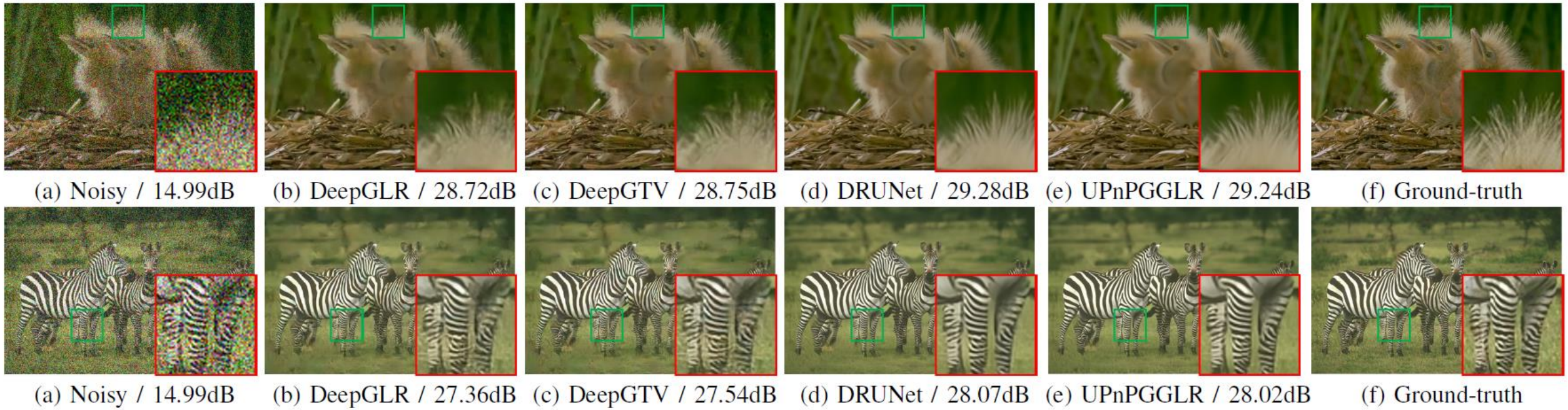


Fig. 3. Color image denoising results of different methods on image "163085" and image "253027" from CBSD68 dataset with noise level 50.

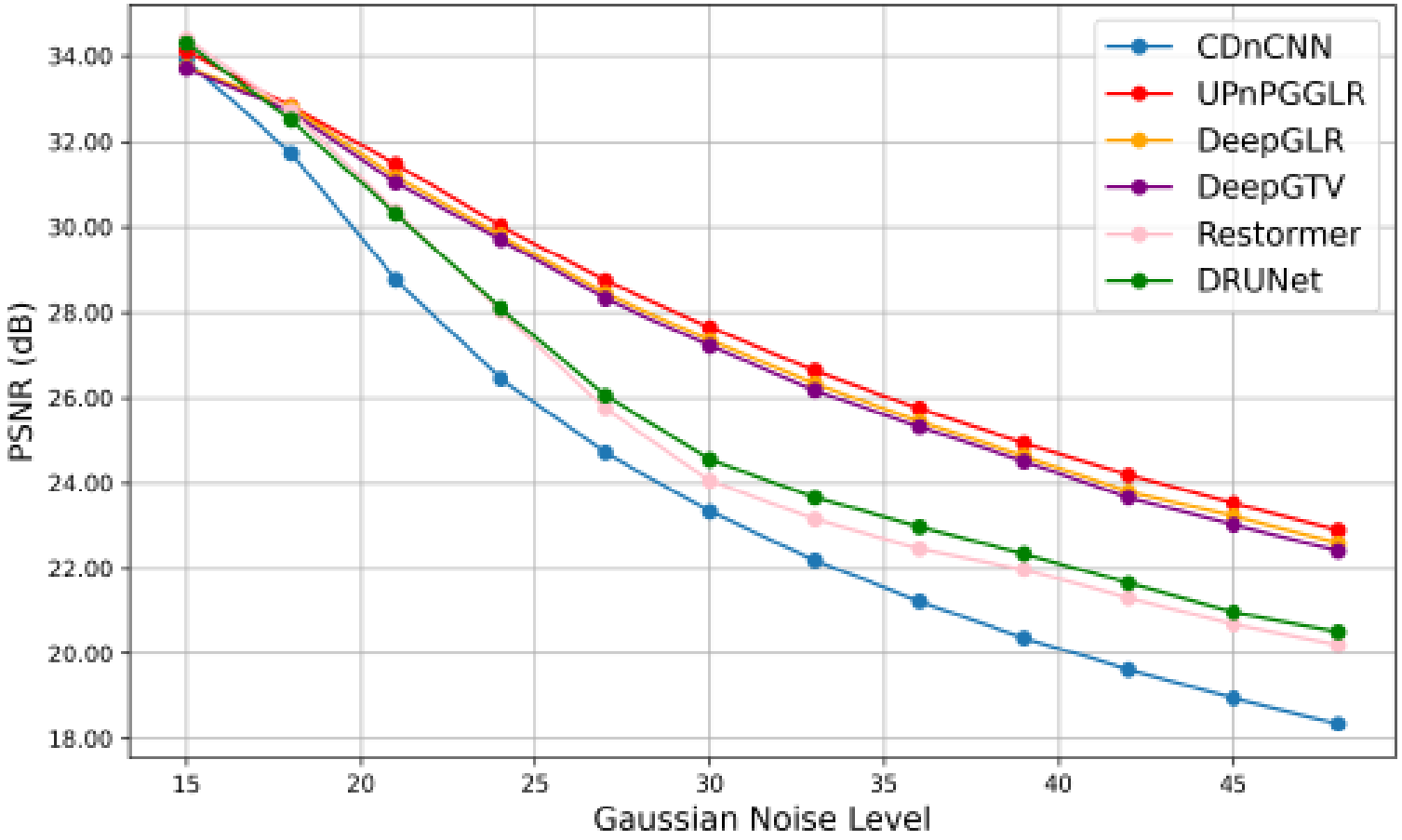
- DeepGLR and DeepGTV fail to fully remove noises in the restored images.
- UPnPGGLR performs comparably to DRUNet while using <1% of parameters.

Gene Cheung (genec@yorku.ca)

Experiments: Unrolling PnP GGLR for Image Denoising

- Trained with noise σ = 15 on BSDS500 dataset and tested at various noise levels.

- Shows improved robustness to *covariate shift*.



(a) Covariance shift

[1] S. W. Zamir et al., "**Restormer: Efficient transformer for high-resolution image restoration**," *CVPR*, pp.5718–5729, 2022.

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Interpolation
- Unrolling GGTV for Image Denoising

- Conclusion

Gene Cheung (genec@yorku.ca)

Conclusion

- Build parameter-efficient neural nets via unrolling of graph-based algorithms.
  - Fewer model parameters → less training data required.

- **Main idea**:
  1. Graph learning from data with normalization is **self-attention**.
  2. Unrolling of graph algorithms leads to ***lightweight*** **transformers**.

- **Future work**:
  - Analytically study unrolling of different graph smoothness priors.
  - Different graphs (directed, signed) as attention mechanism.

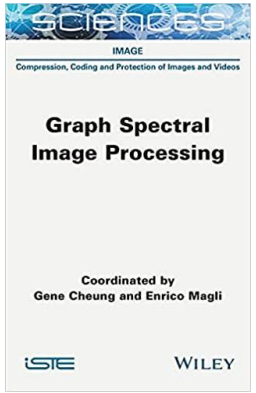
Contact Info

- **Homepage**:

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

- **E-mail**:

genec@yorku.ca

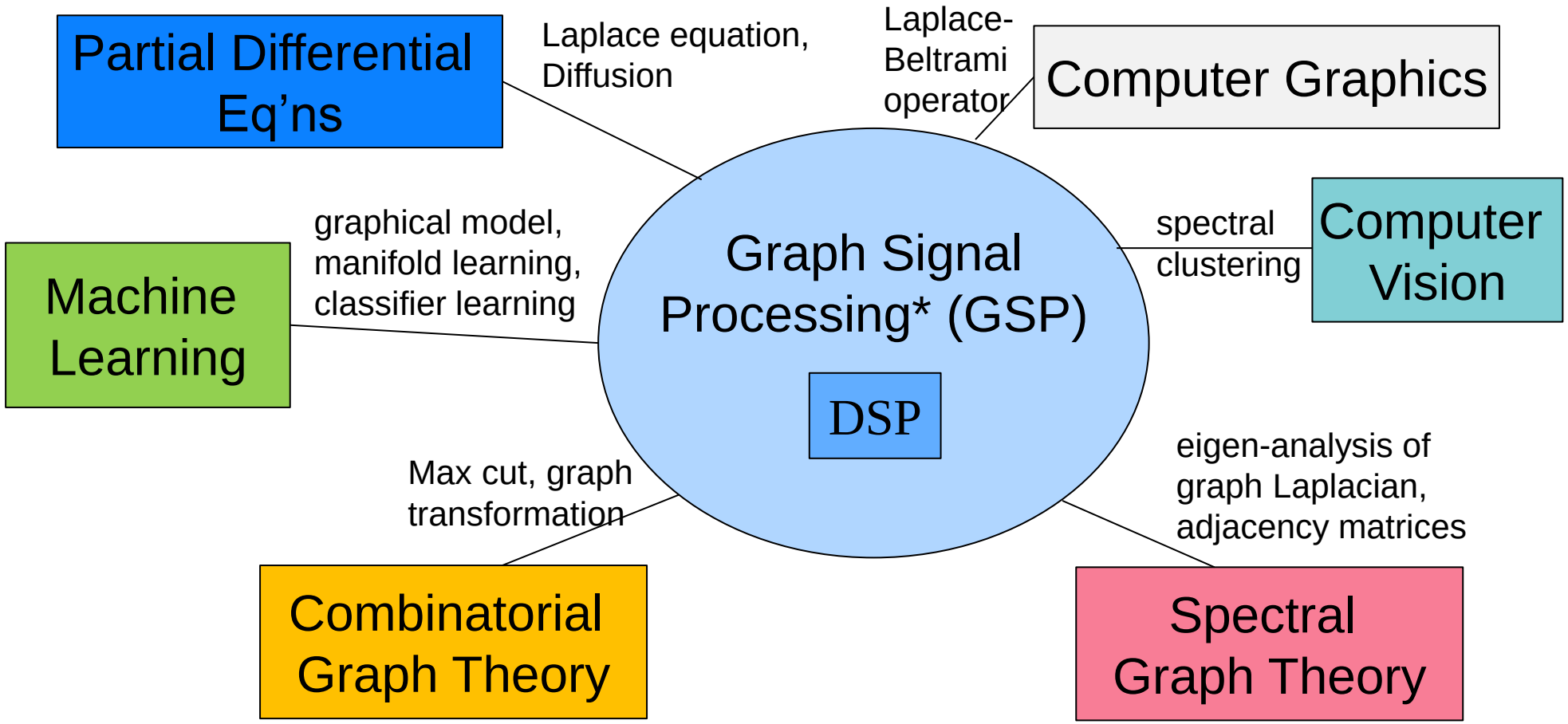


- **New book**:

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

GSP and Graph-related Research

GSP: SP framework that unifies concepts from multiple fields.



Gene Cheung (genec@yorku.ca)