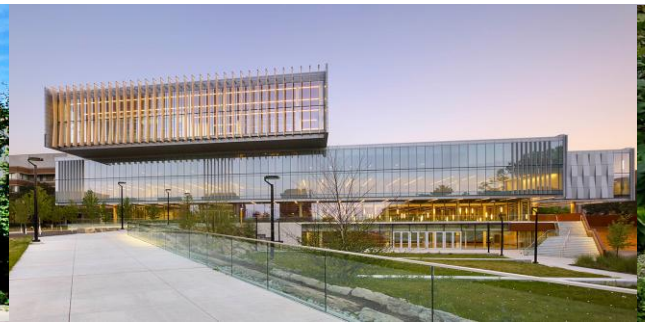
YORK UNIVERSITÉ UNIVERSITY

Self-Attention as Graph Filters: Lightweight Transformer via Unrolling of Graph Smoothness Algorithms

Gene Cheung
York University, Toronto, Canada

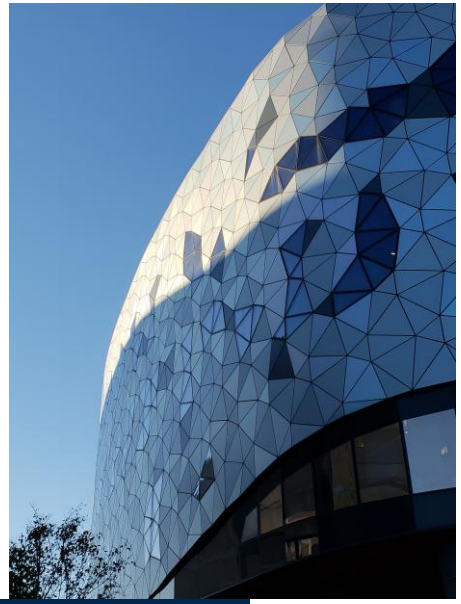
December 16, 2025

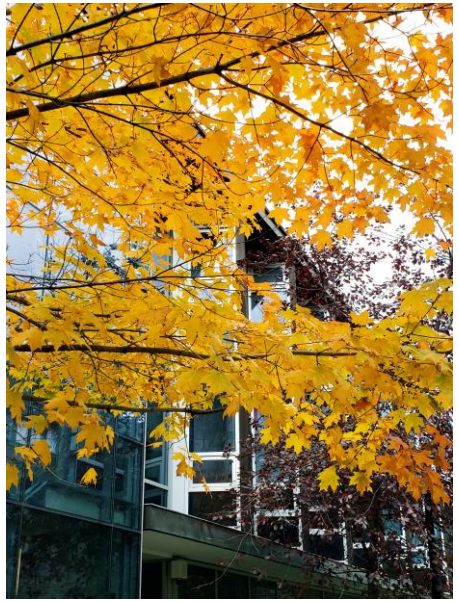


Acknowledgement

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

- **Collaborators**
  - Michael Brown, Andrew Eckford, Pirathayini Srikantha (York Univ., Canada)
  - Antonio Ortega (Univ. of Southern California, USA)
  - Yao Wang, Ivan Selesnick (New York Univ., USA)
  - Phil Chou (Google, 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 Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- Conclusion

We miniaturize transformers!

Gene Cheung (genec@yorku.ca)

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

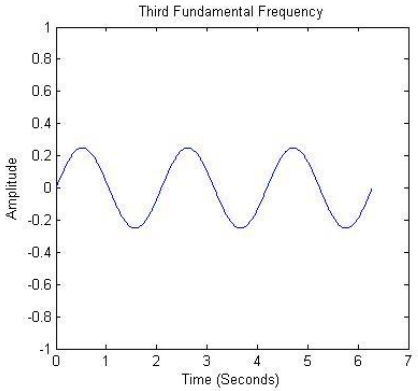
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

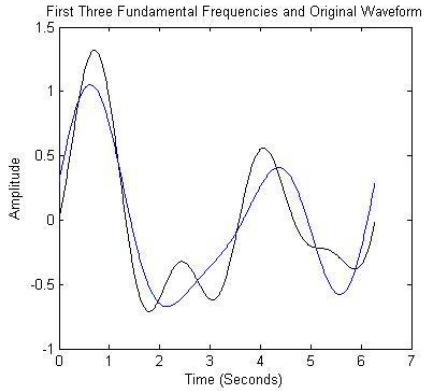
- Conclusion

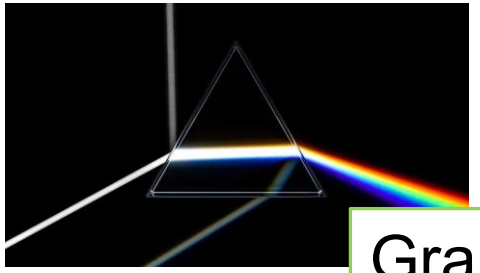
Gene Cheung (genec@yorku.ca)

Graph Signal Processing

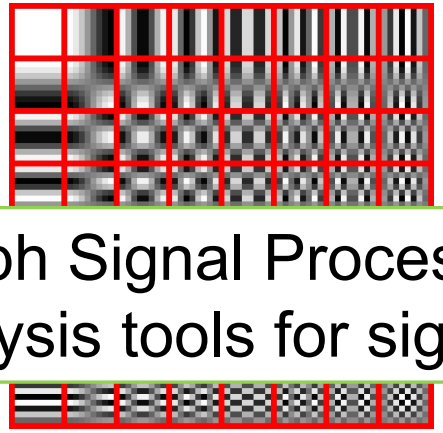
Frequency analysis + Graph kernels = Graph Signal Processing



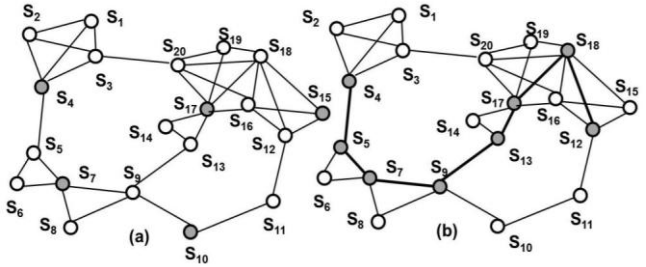


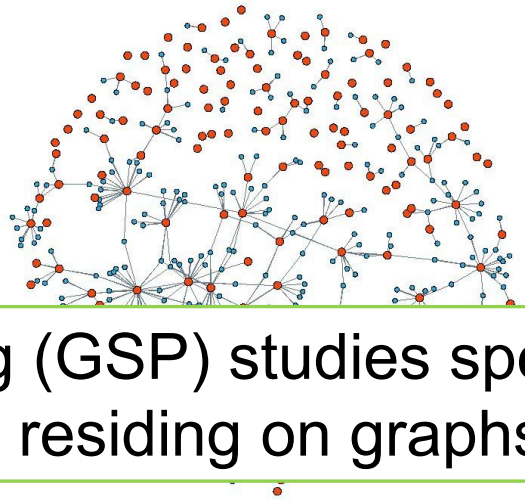


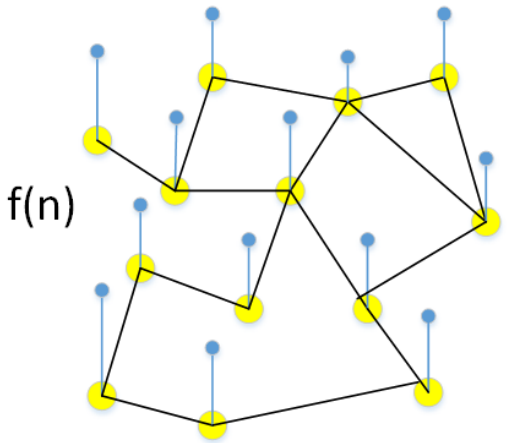
Newton decomposed white light into color components (1730).



2D-DCT basis

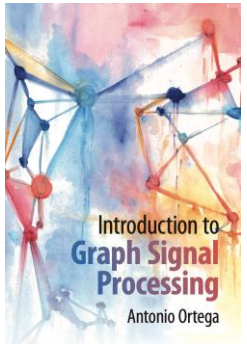


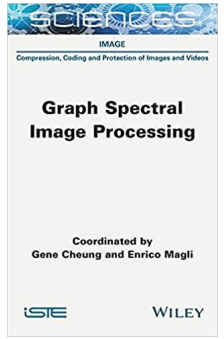




signal on graph kernel

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

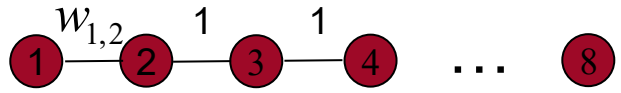
Graph Fourier modes: eigenvectors of *graph Laplacian matrix* **L = D - W**.

eigenvalues along diagonal

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

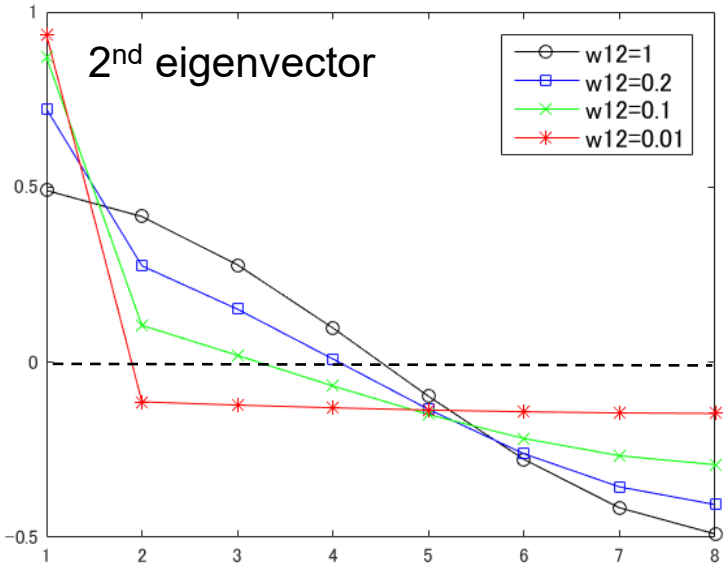
Graph Fourier Transform (GFT)

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.

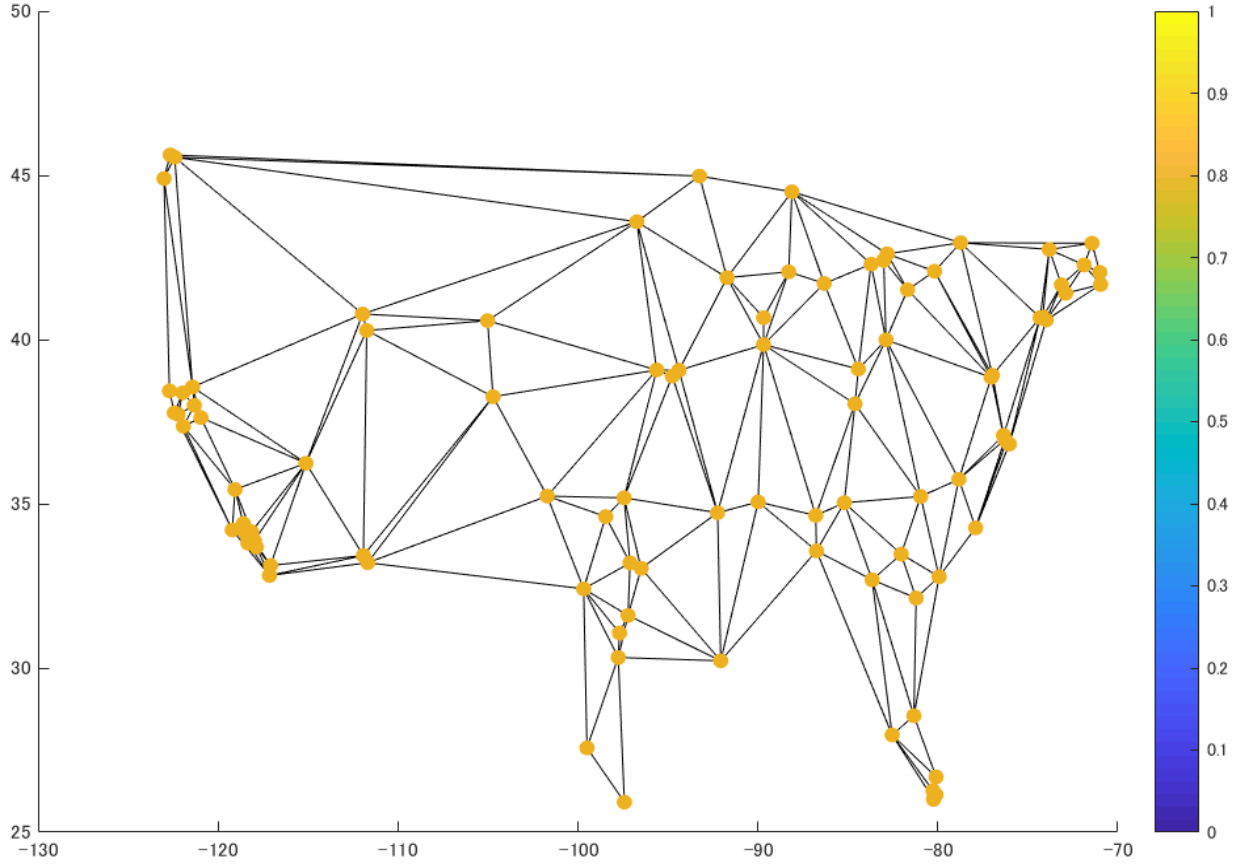
Graph connections from Delaunay Triangulation*.

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.

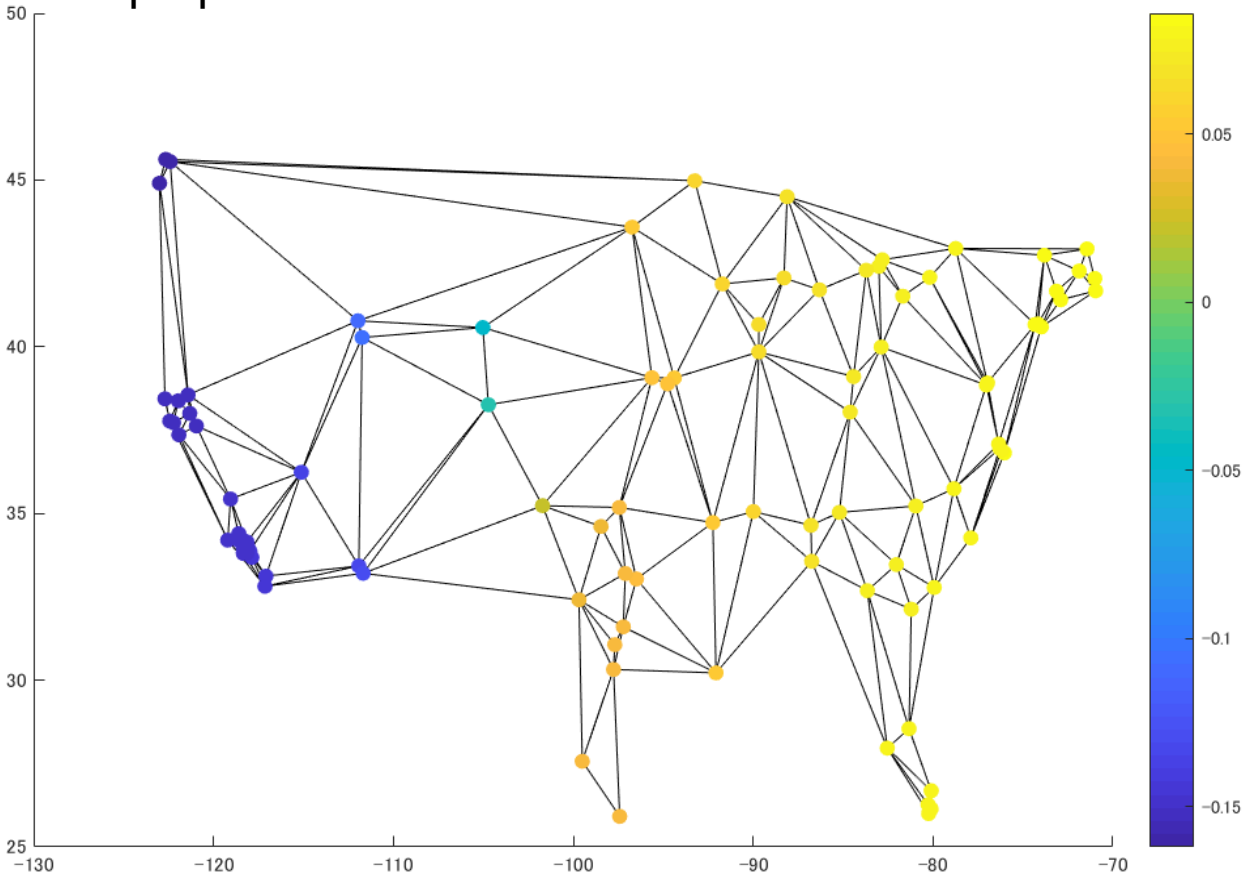
Graph connections from Delaunay Triangulation*.

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.

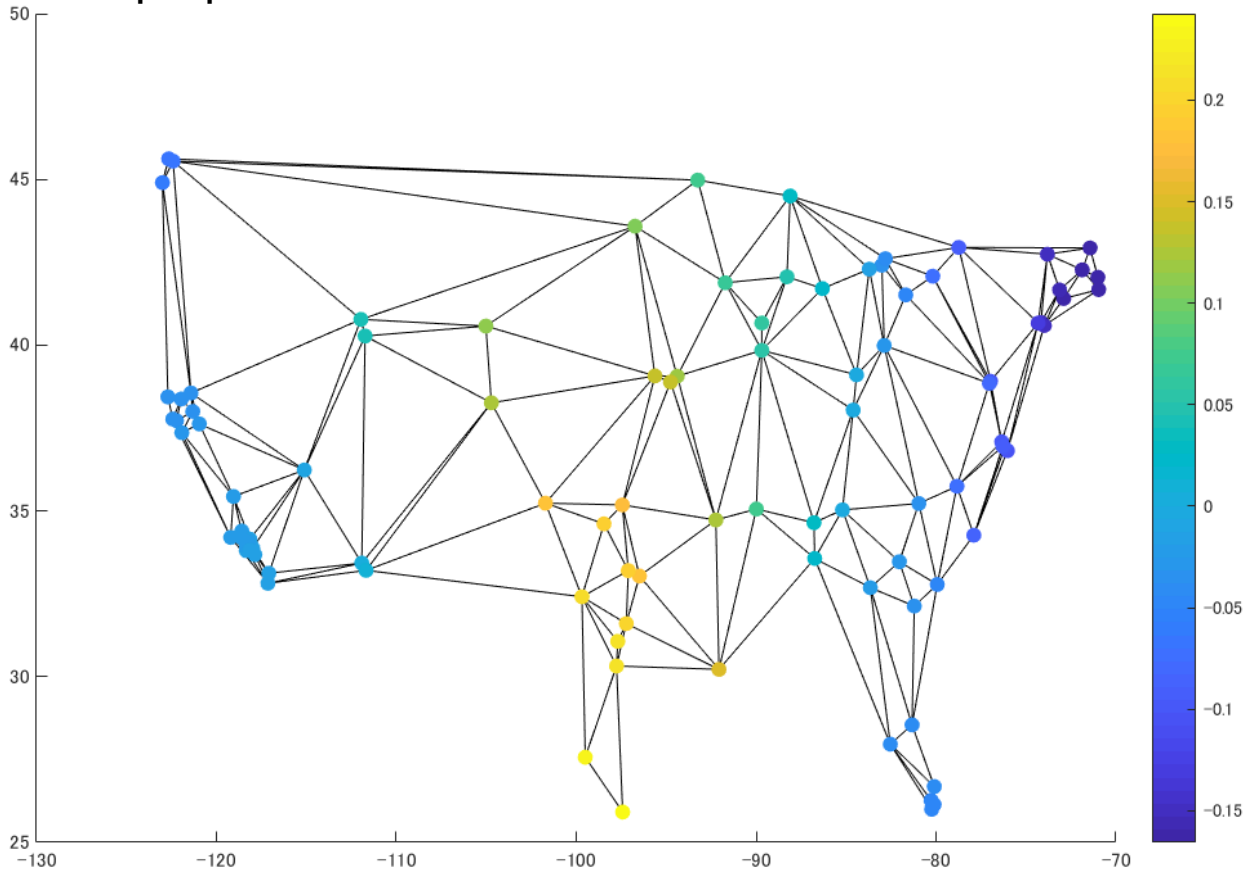
Graph connections from Delaunay Triangulation*.

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.

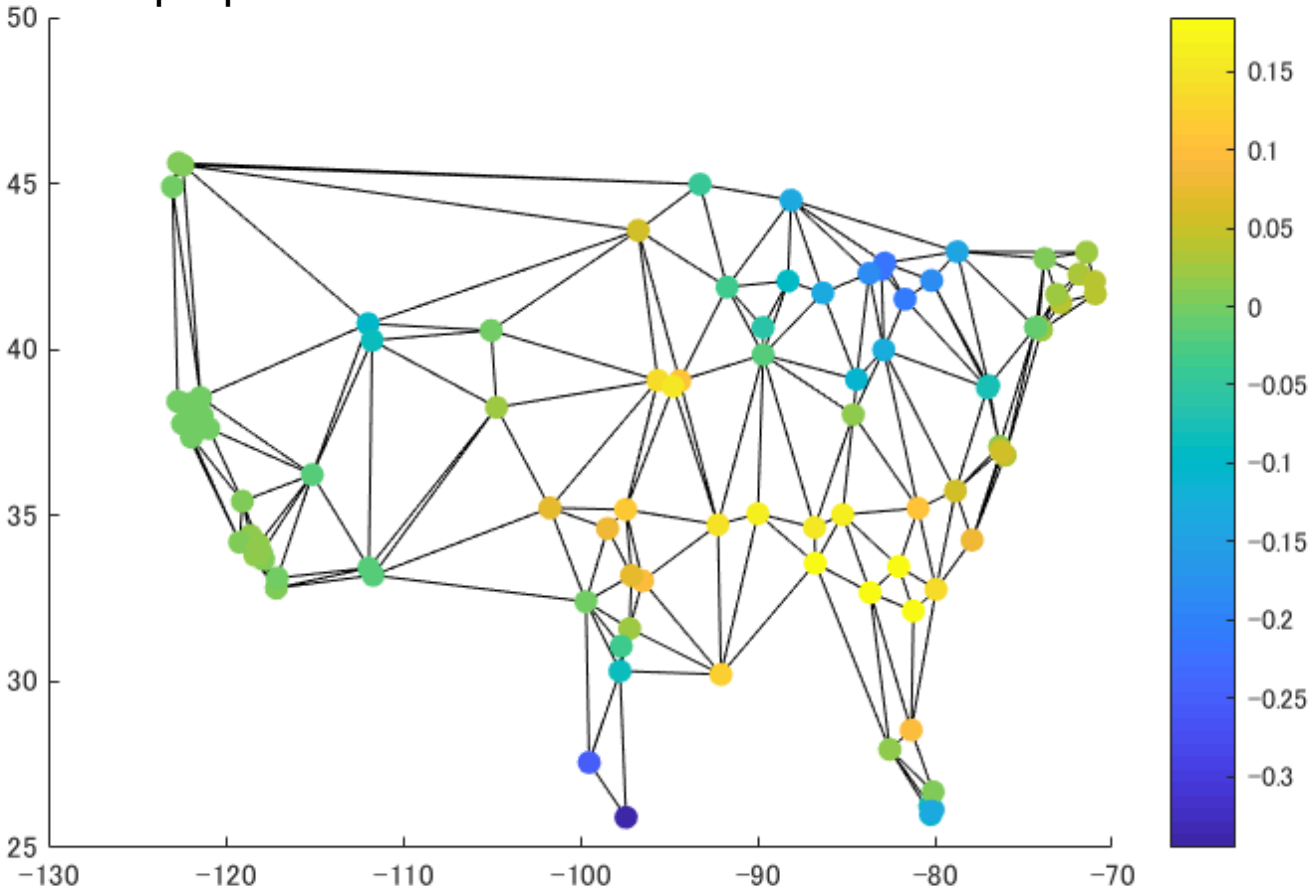
Graph connections from Delaunay Triangulation*.

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 Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- Conclusion

Gene Cheung (genec@yorku.ca)

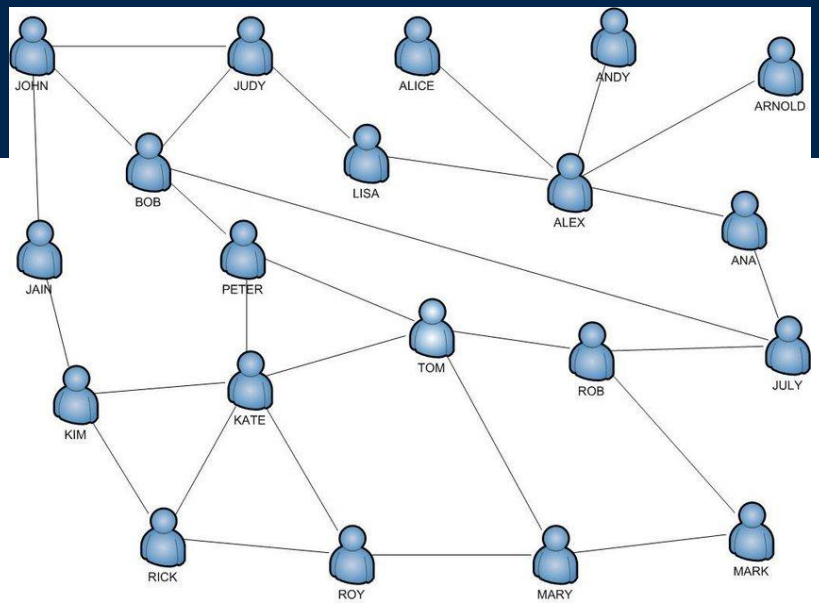
Graph Construction

- Graph captures ***pairwise similarities***.

  Connectivity: Domain knowledge.

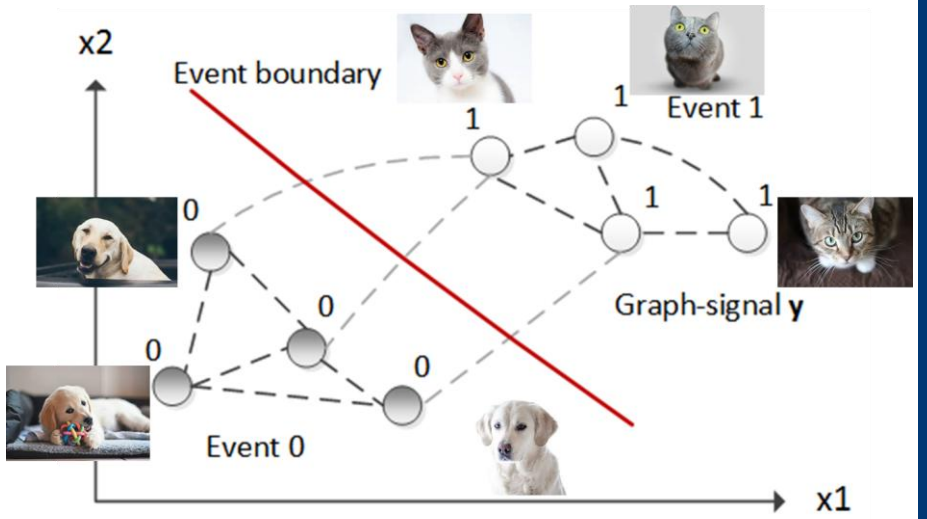
  Edge weights:

  1. Correlations (statistical).
  2. Feature distance (deterministic).



- **Graph Learning from Data**:

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



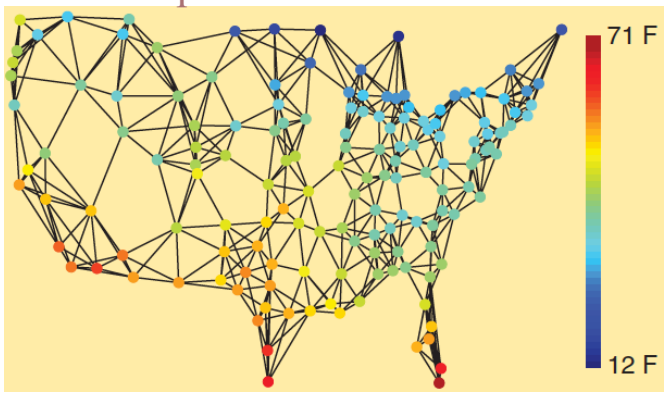
[1] S. Bagheri, G. Cheung, A. Ortega, F. Wang, "**Learning Sparse Graph Laplacian with *K* Eigenvector Prior via Iterative GLASSO and Projection**," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, June 2021 (**best student paper finalist**).

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

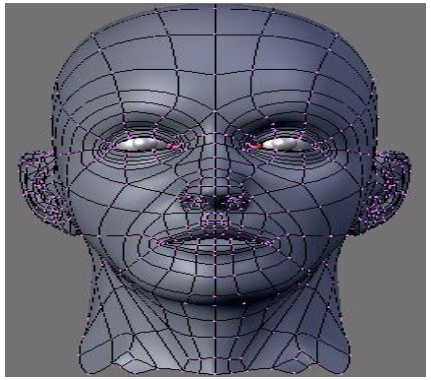
Signed Graphs

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

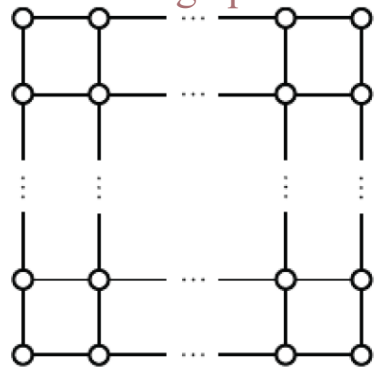
Temperature sensor network



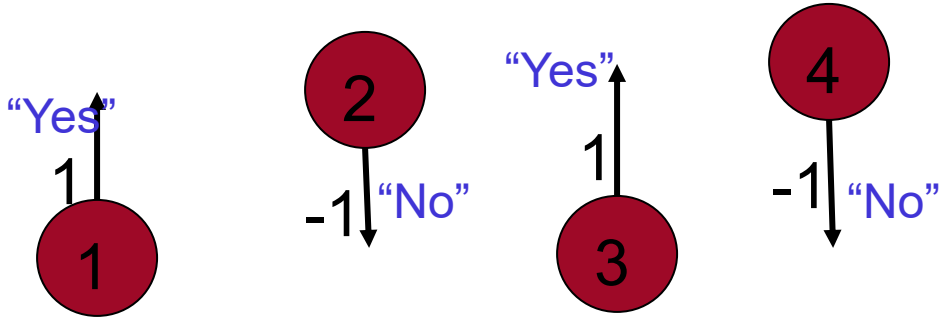
3D geometry

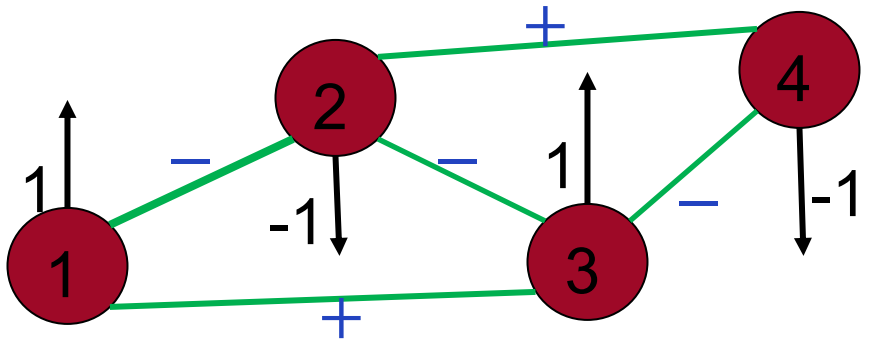


2D image pixels



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





Signed graph

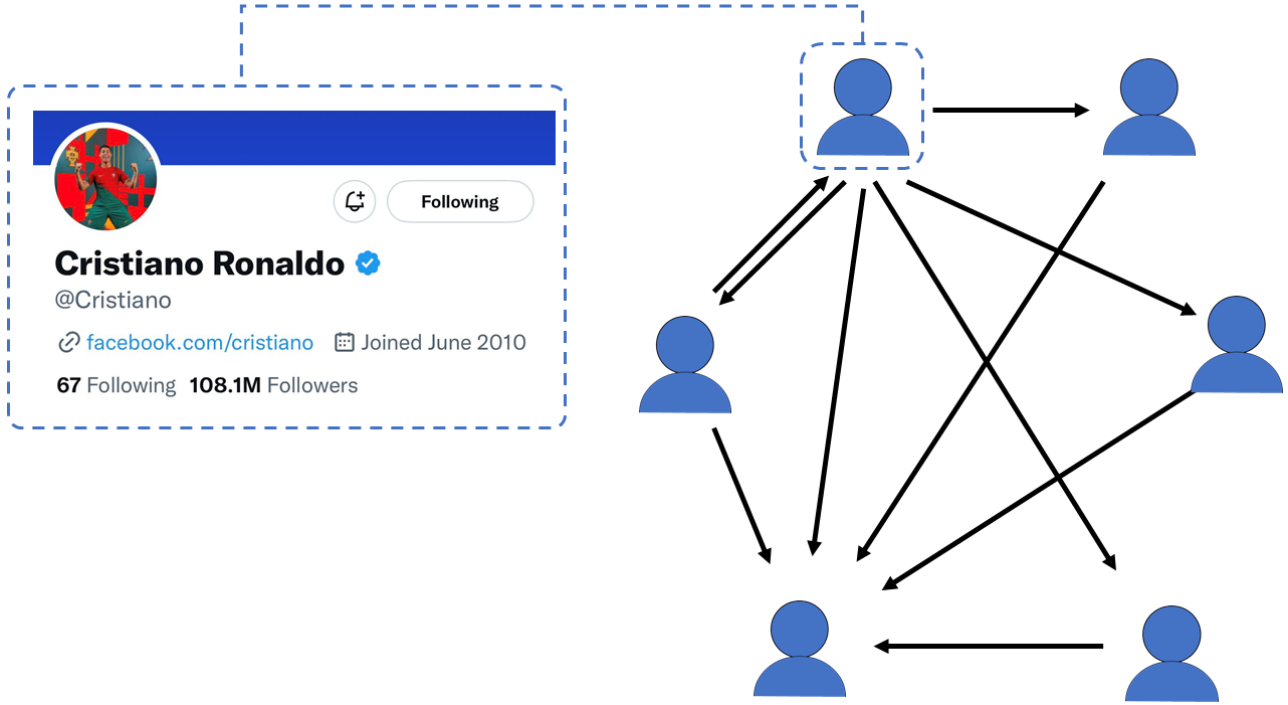
[1] C. Dinesh, G. Cheung, I. V. Bajic, "**Point Cloud Sampling via Graph Balancing and Gershgorin Disc Alignment**," *IEEE TPAMI* vol. 45, no.1, pp. 868-886, January 2023.

[2] C. Dinesh, G. Cheung, S. Bagheri, I. V. Bajic, "**Efficient Signed Graph Sampling via Balancing & Gershgorin Disc Perfect Alignment**," *IEEE TPAMI*, April 2025.

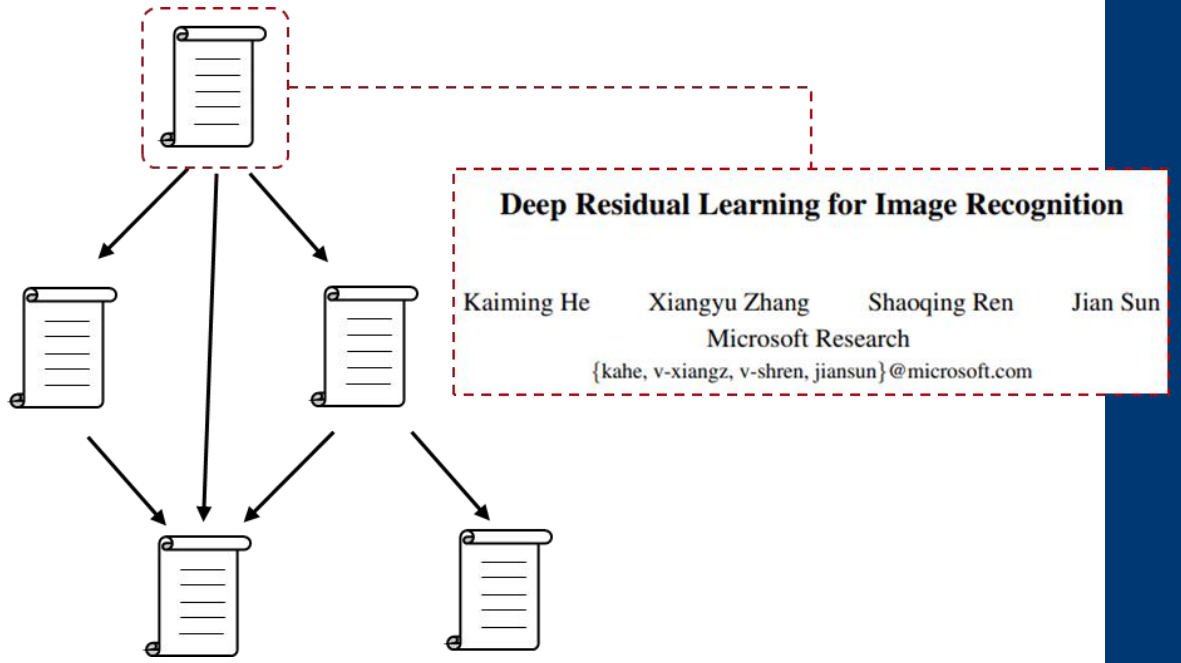
Directed Graphs

- 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 Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- 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

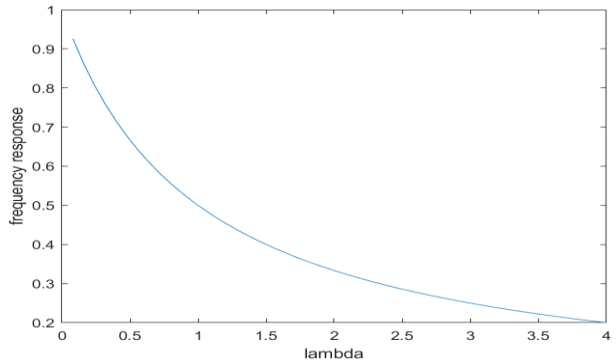
$$(\mathbf{I}+\mu\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

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

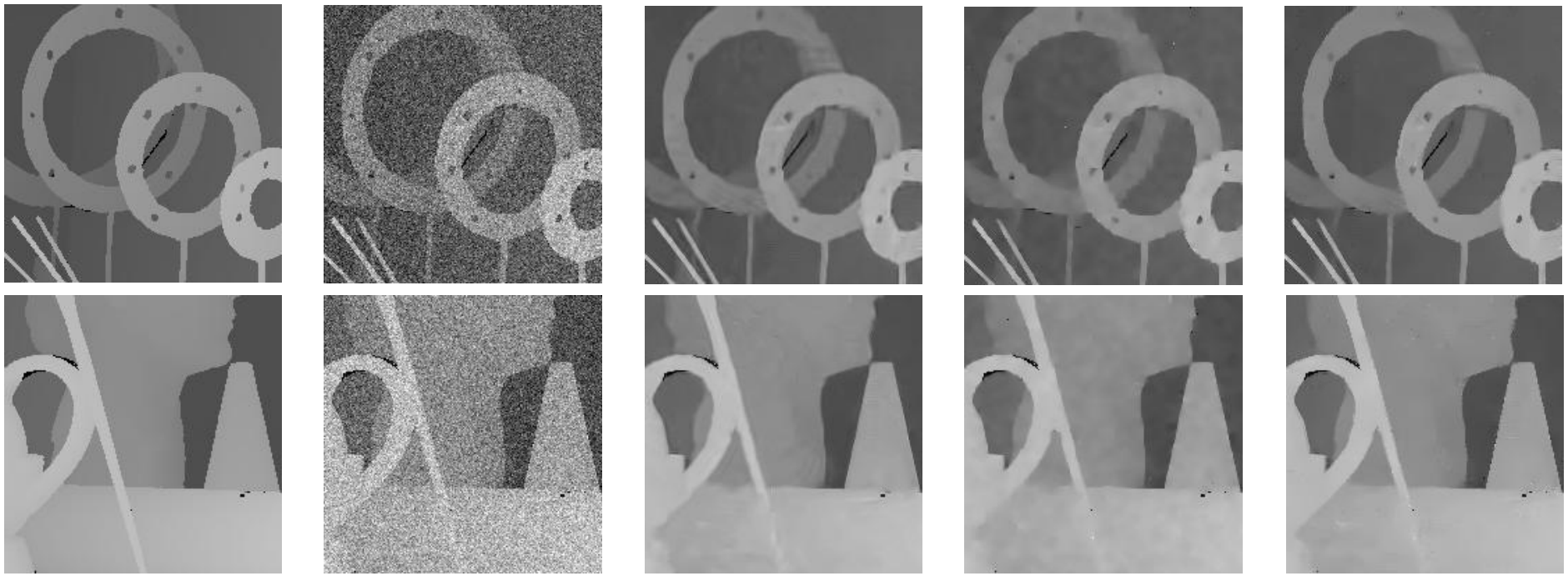
PLOW, 28.11 dB

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

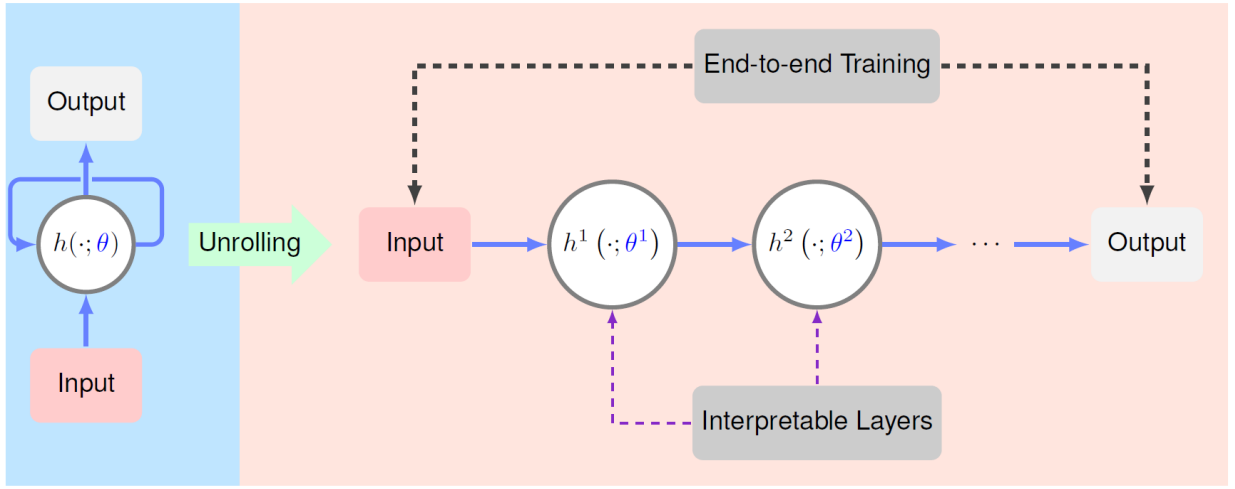
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- 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].

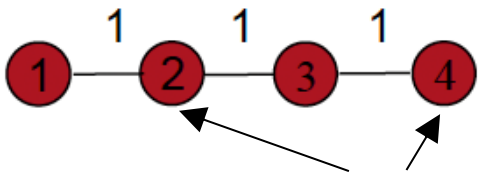
Our approach:
1. design GSP algorithms,
2. unroll + parameter tuning.

[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**," *NeurIPS'23*.

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



Sample set $\{2, 4\}$

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

- 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**." *NeurIPS'24*.

Gene Cheung (genec@yorku.ca)

GTV for Image Interpolation

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

GTV → $\|\mathbf{Cx}\|_1$ ← incidence matrix, $\mathbf{C}^T\mathbf{C} = \mathbf{L}$

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 $\mathbf{H}^\top\mathbf{y}$

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

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

Self-Attention in Transformer

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

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

Feature distance *d(i,j)* as Mahalanobis distance

  - 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**:

$$a_{i,j} = \frac{\exp(e(i,j))}{\sum_{l=1}^{N} \exp(e(i,l))}, \qquad e(i,j) = (\mathbf{Q}\mathbf{x}_j)^\top (\mathbf{K}\mathbf{x}_i)$$

Affinity *e(i,j)* as transformed dot product

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

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

Gene Cheung (genec@yorku.ca)

Output Embedding in Transformer

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

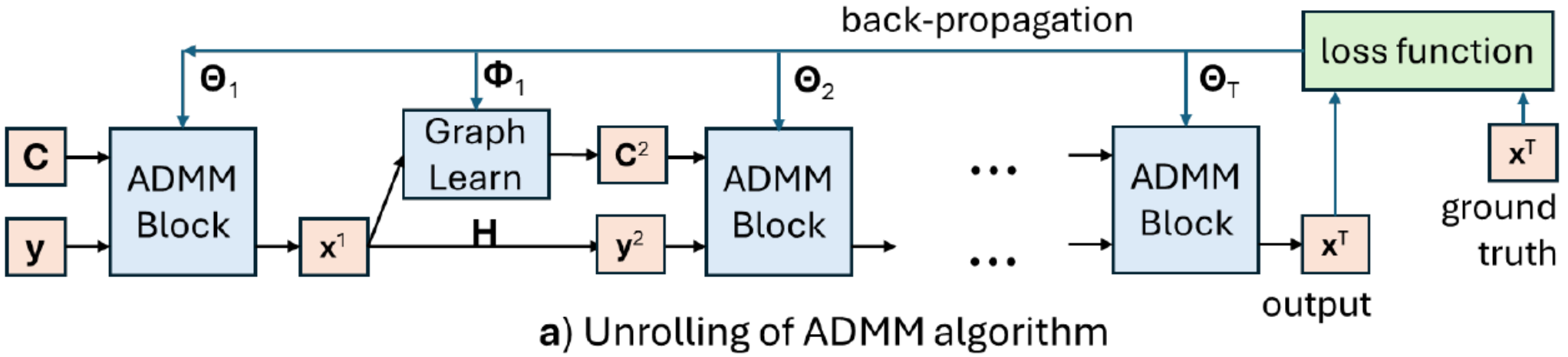
$$\mathbf{y}_i = \sum_{l=1}^{N} a_{i,l} \mathbf{x}_l \mathbf{V}$$

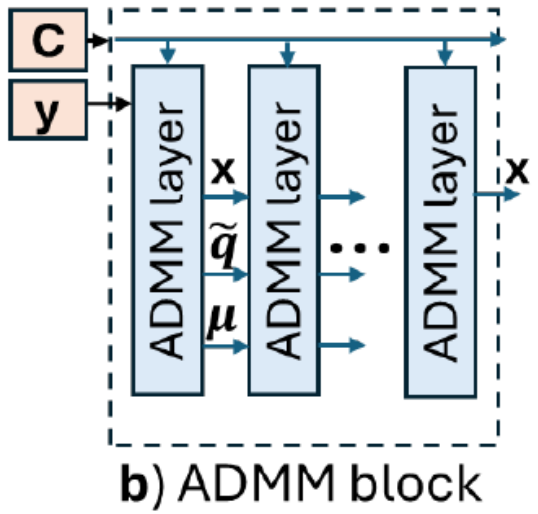
- **Interpretation**:
  - Pairwise similarities (affinities) define *directed* graph.
  - Value matrix defines filter response.
- **Graph Approach**:
  - *Undirected* sparse graph via low-dimensional feature vectors $\mathbf{f}_i$.
  - Derived **low-pass filter** from optimization (no learning!).

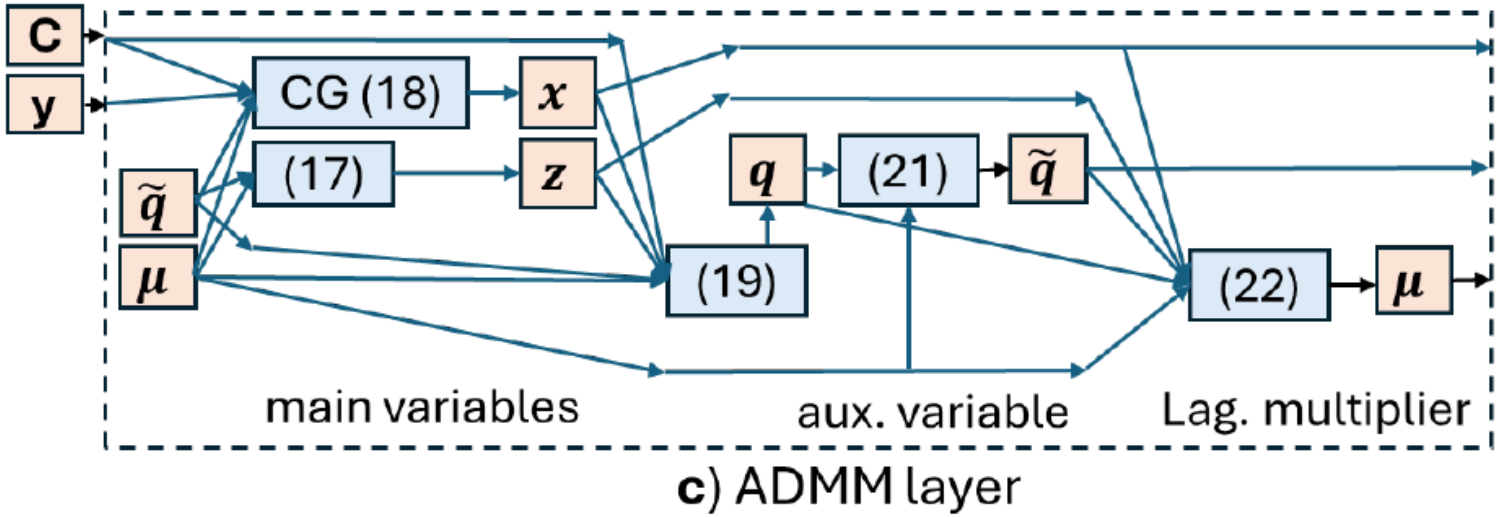
2. Unrolling of graph-based algorithm leads to **lightweight**, interpretable transformer!

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

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

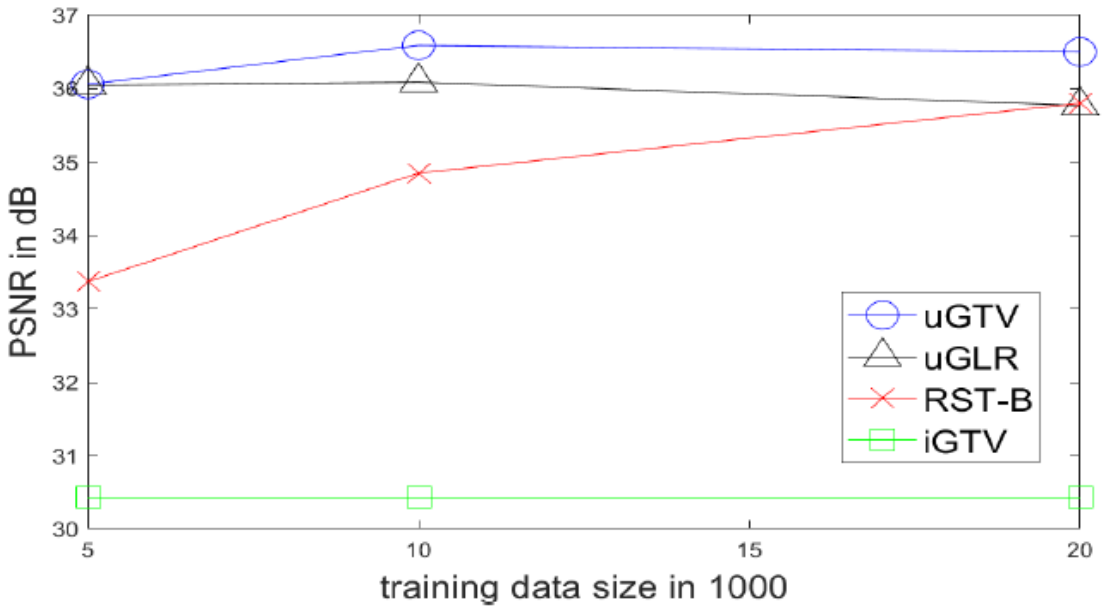


Figure 2: Demosaicking performance vs. training size for different models.

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.

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**," *TIP*, 2025.
[2] F. Chen, G. Cheung and X. Zhang, "**Manifold Graph Signal Restoration Using Gradient Graph Laplacian Regularizer**," *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,

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

(a) Noisy / 14.99dB (b) DeepGLR / 28.72dB (c) DeepGTV / 28.75dB (d) DRUNet / 29.28dB (e) UPnPGGLR / 29.24dB (f) Ground-truth

(a) Noisy / 14.99dB (b) DeepGLR / 27.36dB (c) DeepGTV / 27.54dB (d) DRUNet / 28.07dB (e) UPnPGGLR / 28.02dB (f) Ground-truth

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)

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 Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

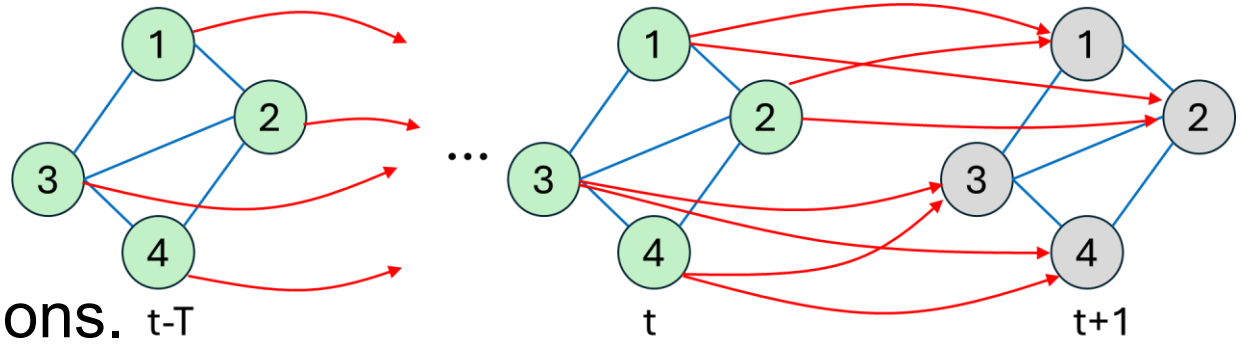
- Conclusion

Gene Cheung (genec@yorku.ca)

Unrolling of Mixed Graph Algorithms for Traffic Forecast

- **Spatial-temporal data:**
  - Time-varying graph signals (*e.g.*, traffic, weather).
  - Spatial correlations + temporal causal relationships.

Product Graph



- **Mixed Graph Model** [1]:
  - ***Undirected*** graph for spatial correlations.
  - ***Directed*** graph for temporal relationships.
- ***Directed GLR / Directed GTV*** to quantify & promote signal smoothness (**low frequencies**) on ***directed*** graph.

DGLR DGTV

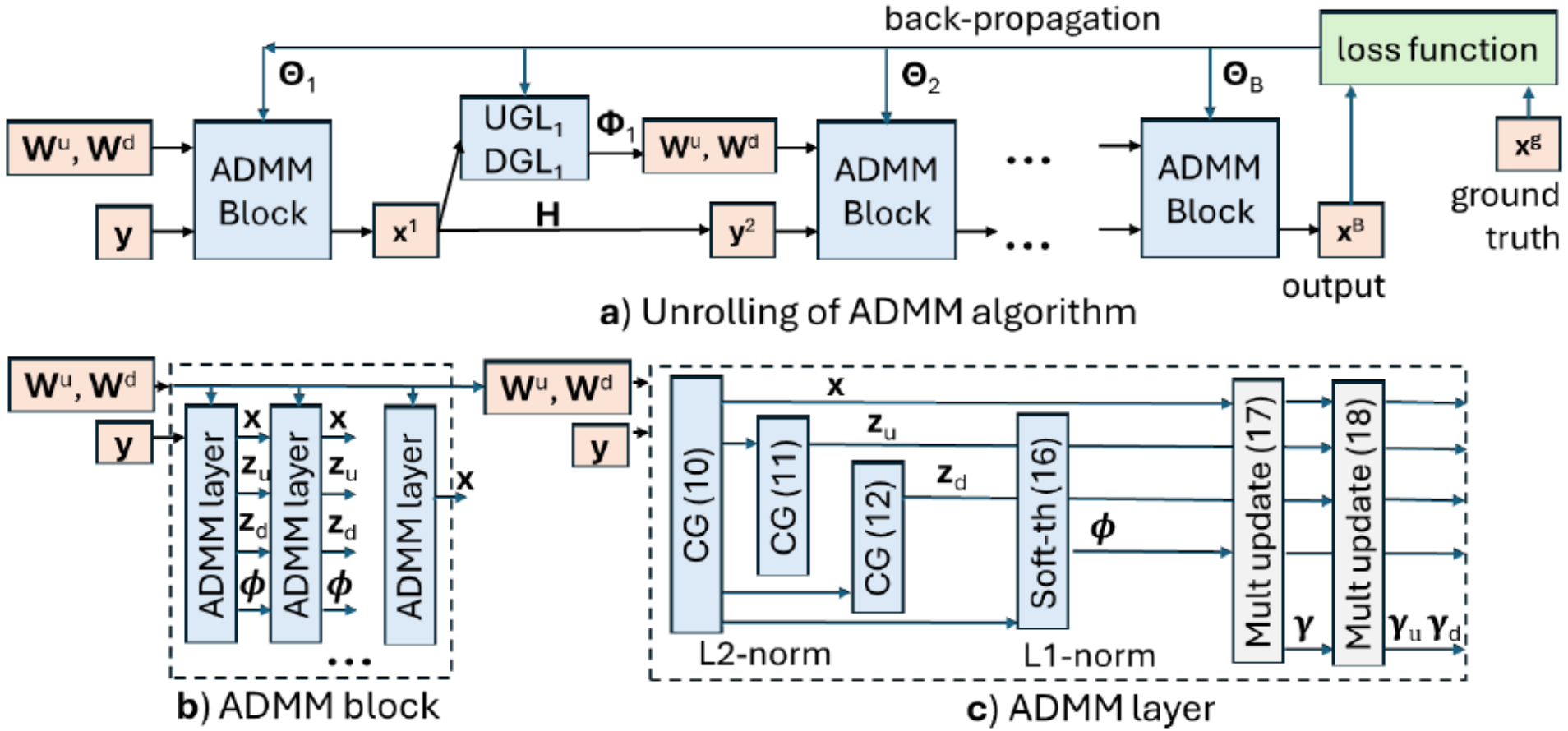
$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \mu_u \mathbf{x}^\top \mathbf{L}^u \mathbf{x} + \mu_{d,2} \mathbf{x}^\top \mathcal{L}_r^d \mathbf{x} + \mu_{d,1} \|\mathbf{L}_r^d \mathbf{x}\|_1$$

[1] J. Qi et al., "**Lightweight and Interpretable Transformer via Mixed Graph Algorithm Unrolling for Traffic Forecast**," submitted to *ICLR'26.*

Gene Cheung (genec@yorku.ca)

Unrolling of Mixed Graph Algorithms for Traffic Forecast

- **Convex Optimization:**
  - **ADMM**: i) L2-norm (conjugate gradient), ii) L1-norm (soft-thresholding).



[1] J. Qi et al., "**Lightweight and Interpretable Transformer via Mixed Graph Algorithm Unrolling for Traffic Forecast**," submitted to *ICLR'26.*

Gene Cheung (genec@yorku.ca)

Unrolling of Mixed Graph Algorithms for Traffic Forecast

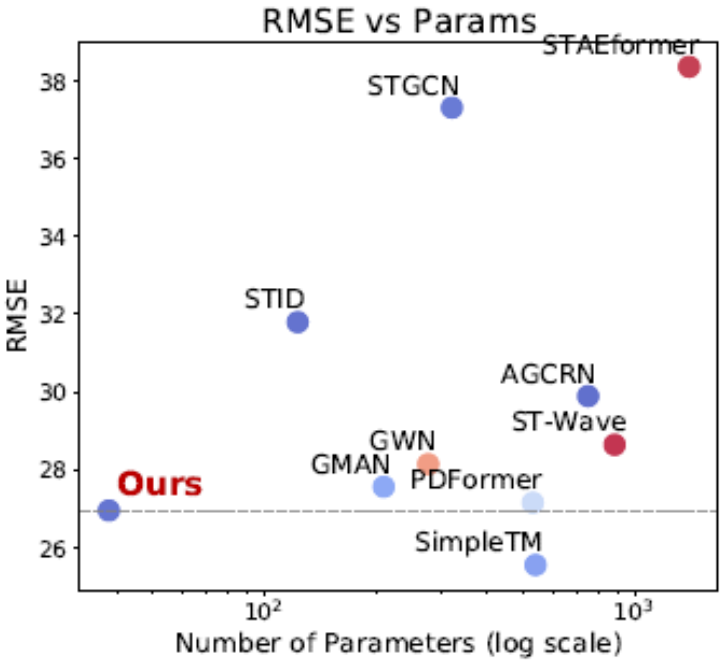
- **Experiments**:
  - 7.2%of transformer-based PDFormer
  - boldface for smallest, blue for 2nd and 3rd smallest, underline for next 2 smallest.

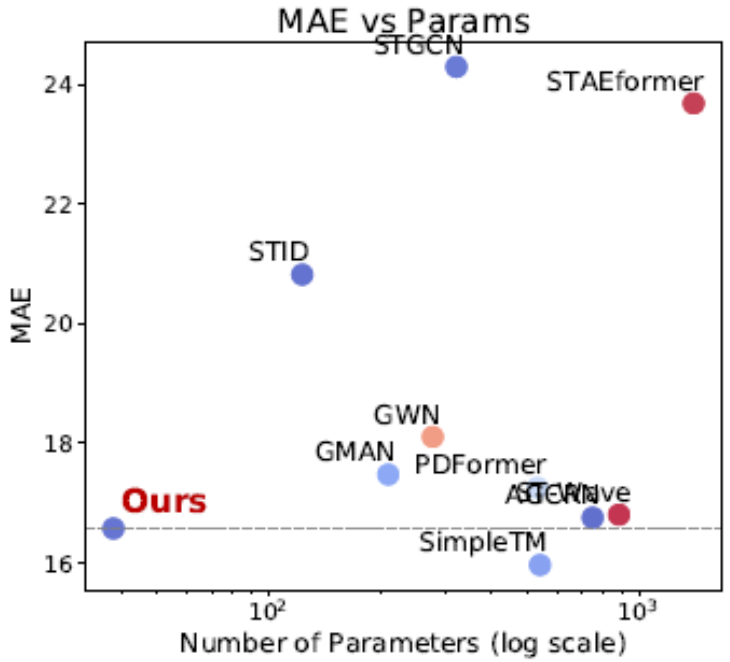
Dataset & Horizon	PEMS03 (358 nodes, 547 edges)			METR-LA (207 nodes, 1,315 edges)		
	30 minutes	60 minutes	120 minutes	30 minutes	60 minutes	120 minutes
VAR	28.07 / 16.53 / 17.49	30.54 / 18.31 / 19.61	36.65 / 22.40 / 24.64	10.72 / 5.55 / 11.29	12.59 / 6.99 / 13.46	**14.83** / 8.90 / 16.54
STGCN	28.06 / 18.24 / 17.76	37.31 / 24.29 / 23.00	54.34 / 34.83 / 30.52	11.26 / 5.08 / 10.93	13.91 / 6.35 / 13.67	16.92 / 8.11 / 17.69
STSGCN	26.41 / 16.75 / 16.86	30.62 / 19.35 / 19.15	38.04 / 23.86 / 23.62	10.25 / 4.05 / 9.18	12.65 / 5.18 / 11.48	15.74 / **6.84** / 15.33
GMAN	25.79 / 16.23 / 20.04	27.57 / 17.48 / 24.33	**30.69** / 19.20 / 26.69	11.97 / 5.34 / 10.66	14.49 / 6.93 / 13.12	15.53 / 7.59 / 14.70
ST-Wave	25.57 / **15.11** / **15.04**	28.65 / 16.81 / 19.24	**29.88** / **17.11** / **17.71**	10.81 / 4.11 / 9.16	13.24 / 5.33 / 11.32	23.18 / 11.22 / **12.71**
PDFormer	**23.71** / **15.05** / 18.16	**27.16** / 17.26 / 21.21	35.77 / 22.25 / 25.01	10.21 / **3.89** / 8.50	**12.17** / **4.81** / **10.92**	17.27 / 9.55 / 19.10
STAEformer	30.22 / 18.85 / 26.62	38.36 / 23.68 / 29.21	48.72 / 31.96 / 44.71	10.16 / **3.73** / **8.25**	12.58 / **4.79** / **10.08**	**14.63** / **5.82** / **11.87**
GWN	25.76 / 15.22 / 16.83	28.15 / 18.11 / **17.56**	34.60 / 21.10 / 22.45	11.42 / 5.18 / **8.21**	12.46 / 5.17 / 11.97	23.13 / 11.32 / 13.53
AGCRN	27.40 / 15.19 / **14.35**	29.90 / **16.76** / **15.32**	32.79 / **18.72** / **17.03**	**10.11** / **3.82** / 8.61	12.56 / **5.00** / 11.25	**14.77** / **6.33** / 14.05
STID	26.50 / 17.27 / 18.59	31.88 / 20.82 / 24.89	42.98 / 28.03 / 42.41	10.21 / 4.05 / 9.47	12.61 / 5.21 / 12.22	15.48 / 6.98 / 16.78
SimpleTM	**23.75** / **15.13** / **15.59**	**25.56** / **15.97** / **15.49**	**30.58** / **18.73** / **18.18**	**8.76** / 4.12 / **7.63**	**11.96** / 5.49 / **9.65**	15.44 / 7.64 / **12.27**
Ours	**25.05** / 15.85 / 16.49	**26.96** / **16.58** / 18.07	34.06 / 20.23 / 23.86	**10.06** / 4.05 / 9.23	**12.17** / 5.27 / 11.78	15.19 / 7.14 / 16.44

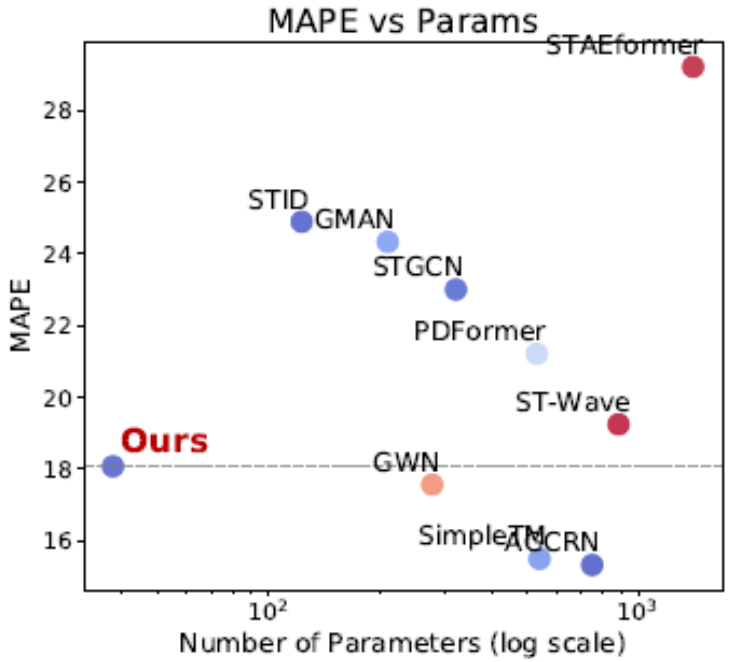
Gene Cheung (genec@yorku.ca)

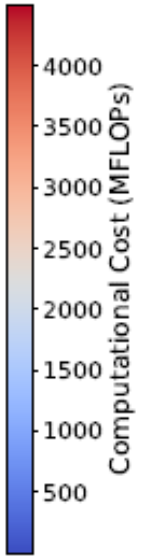
Unrolling of Mixed Graph Algorithms for Traffic Forecast

- **Experiments**:
  - Tradeoff between parameter count and performance for 60-minute forecast on PEMS03.









Gene Cheung (genec@yorku.ca)

Outline

- GSP Overview
  - Graph frequencies from eigen-pairs

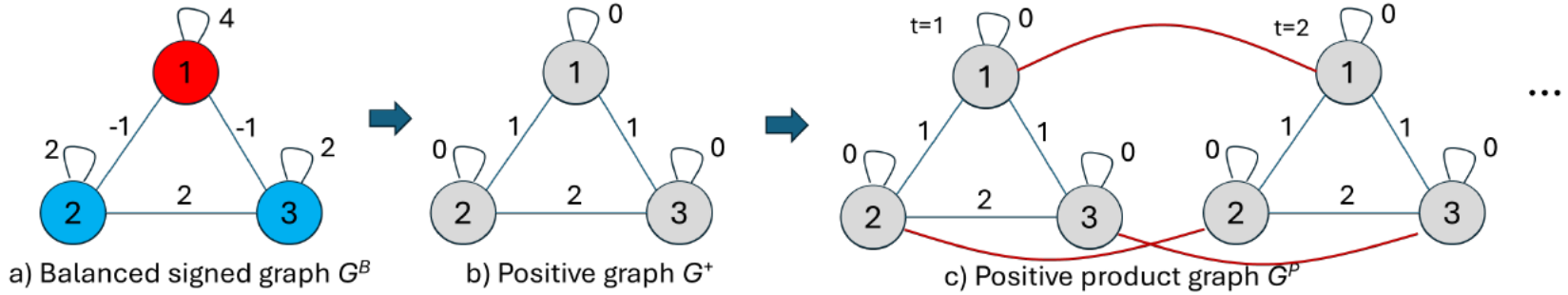
- Graph Construction
- Low-pass Graph Filter for Image Denoising
- GSP + Algorithm Unrolling
- Unrolling GLR/GTV for Image Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- Conclusion

Gene Cheung (genec@yorku.ca)

EEG Classification via Balanced Signed Graph Algo. Unrolling

- **Balanced Signed Graph:**
  - No cycle of odd number of negative edges.
  - 1-to-1 mapping to positive graph via similarity transform of Laplacians [1].



- **Train 2 Denoisers**:
  - Learn balanced signed graph from EEG data.
  - Approximate ideal LP-filter with Lanczos approx.
  - Trained denoiser learns signal class posterior.

Pretext task

- **Binary Classification**:
  - Inject input signal to two trained denoisers.
  - Examine residual size for classification.

Generative modeling + Discriminative classification

[1] C. Dinesh et al., "**Efficient Signed Graph Sampling via Balancing & Gershgorin Disc Perfect Alignment**," *IEEE TPAMI*, April 2025.
[2] J. Yao et al., "**Lightweight Transformer for EEG Classification via Balanced Signed Graph Algorithm Unrolling**," submitted to *ICLR'26.*

EEG Classification via Balanced Signed Graph Algo. Unrolling

- **Datasets:**
  - Epilepsy patients vs. normal subjects.
  - Training / testing in 8:1:1 split, "leave-one-subject-out" (LOSO) settings.
- **Experiments**:
  - 1% of Transformer.
  - 0.1% of STFT+CNN.

Method	Params #	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
Non-graph-based Methods in Default Task Setting						
MDTW + KNN (Tasci et al., 2023)	-	87.78	89.39	81.32	92.68	85.16
TF + ROD (Shen et al., 2024)	18,400	88.08	89.22	82.28	92.46	85.61
mAtt (Pan et al., 2022)	46,542	92.00	95.27	85.68	96.78	90.22
EEG2Rep (Mohammadi Foumani et al., 2024)	96,886	91.41	94.79	86.09	91.48	93.11
CWT + DCNN (Dicsli et al., 2025)	143,297	95.91	94.55	**96.98**	96.06	95.30
Ours	**14,787**	**97.57**	**98.58**	95.98	**97.45**	**98.01**
Large model in Default Task Setting						
Transformer (Lih et al., 2023)	1,849,771	85.12	82.00	82.00	87.32	82.00
STFT + CNN (Bhandage et al., 2024)	11,533,928	99.20	99.14	99.46	98.98	99.30
Graph-based Methods in LOSO Task Setting						
DGCNN (Song et al., 2018)	149,466	76.74	69.56	62.74	84.60	65.97
GIN (Zhang & Yao, 2021)	25,794	68.82	58.78	44.36	82.55	50.56
EEGNet (Lawhern et al., 2018)	**9,170**	78.78	81.26	53.25	93.11	64.34
FBCSPNet (Schirrmeister et al., 2017)	98,242	81.76	92.80	53.40	**97.67**	67.79
Deep4Net (Schirrmeister et al., 2017)	321,227	78.62	73.06	64.20	86.72	68.34
Ours(LOSO)	14,787	**90.06**	**93.48**	**86.10**	91.70	**92.59**

Gene Cheung (genec@yorku.ca)

EEG Classification via Balanced Signed Graph Algo. Unrolling

Experiments:

- Compare performance of three graph types on the LOSO task.
- Balanced signed graph + Lanczos approx. filter is best.

Setting	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
Positive Graph	84.30	88.49	76.88	86.00	87.23
Unbalanced Signed Graph	78.87	86.74	68.22	78.69	82.52
Balanced Signed Graph	**93.68**	**96.32**	**89.45**	**93.61**	**94.94**

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 Restoration
- Unrolling DGLR/DGTV for Traffic Forecast
- Unrolling Ideal LP Filters for EEG Classification

- 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**.

     - Low-pass filter + graph learning → compact model

Conclusion

- **Future work**:
  - Denoiser-based diffusion model for **image generation**.
  - Unrolled denoisers / interpolators with **guaranteed initialization**.
  - Optimizing connectivity *and* edge weights of graphs (e.g., for LLMs).

Unet

LGF

Table 1: Interpolation performance for different models.

Method	Params	Kodak[22] PSNR | SSIM	Urban100[23] PSNR | SSIM	McM[24] PSNR | SSIM
Bicubic	–	32.59 | 0.9332	28.38 | 0.9176	34.88 | 0.9578
SRCNN [25]	57,281	35.25 | 0.9599	31.00 | 0.9478	39.02 | 0.9786
SwinIR [2]	910,152	36.06 | 0.9619	31.69 | 0.9525	39.93 | 0.9809
Restormer [3]	26,124,052	36.95 | 0.9656	**33.97** | **0.9657**	**40.92** | 0.9825
uGTV [13]	319,115	36.81 | 0.9653	32.92 | 0.9610	40.43 | 0.9818
Ours (Bicubic)	**283,819**	36.56 | 0.9651	32.80 | 0.9605	40.29 | 0.9814
Ours (Bicubic+)	**288,781**	**36.95** | **0.9660**	**33.48** | **0.9636**	**40.82** | **0.9825**

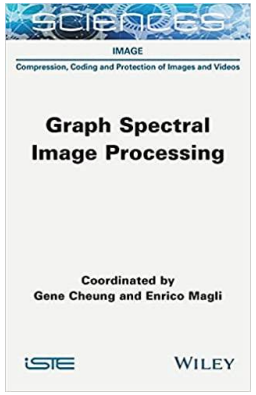
Contact Info

- **Homepage**:

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

- **E-mail**:

genec@yorku.ca



- **GSP book**:

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