Bridging graph signal processing and graph neural networks

Siheng Chen

Mitsubishi Electric Research Laboratories

Graph-structured data

Data with explicit graph

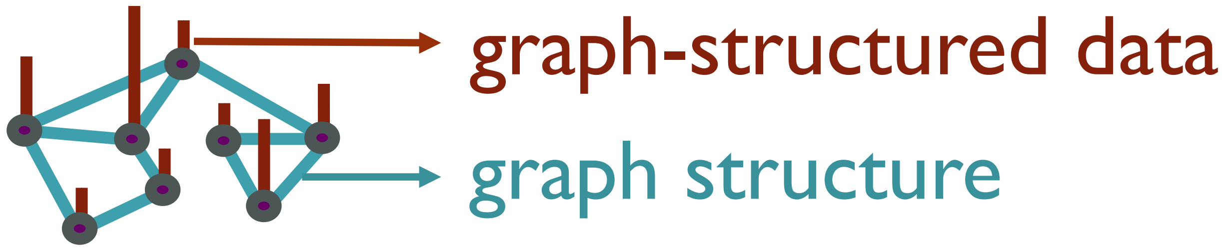
Social networks

Internet of things

Traffic networks

Human brain networks

Graph-structured data



Data with explicit graph

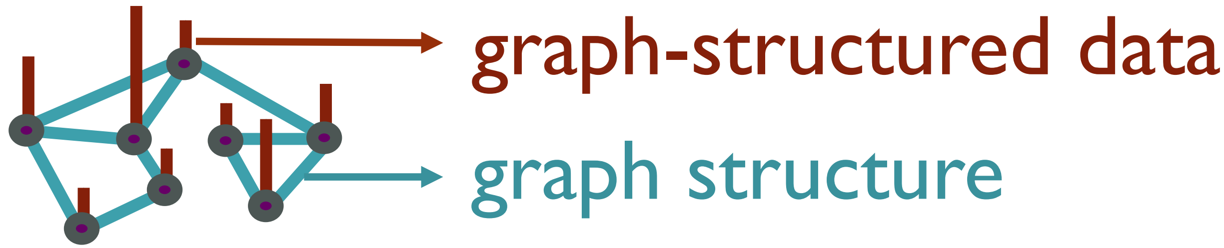
Social networks

Internet of things

Traffic networks

Human brain networks

Graph-structured data



Data with explicit graph

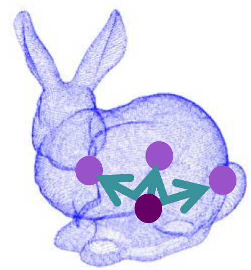
Social networks

Internet of things

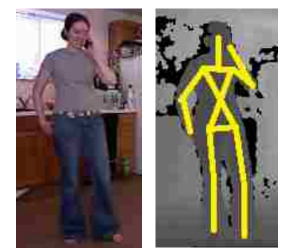
Traffic networks

Human brain networks

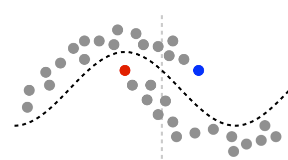
Data with implicit graph



3D point cloud



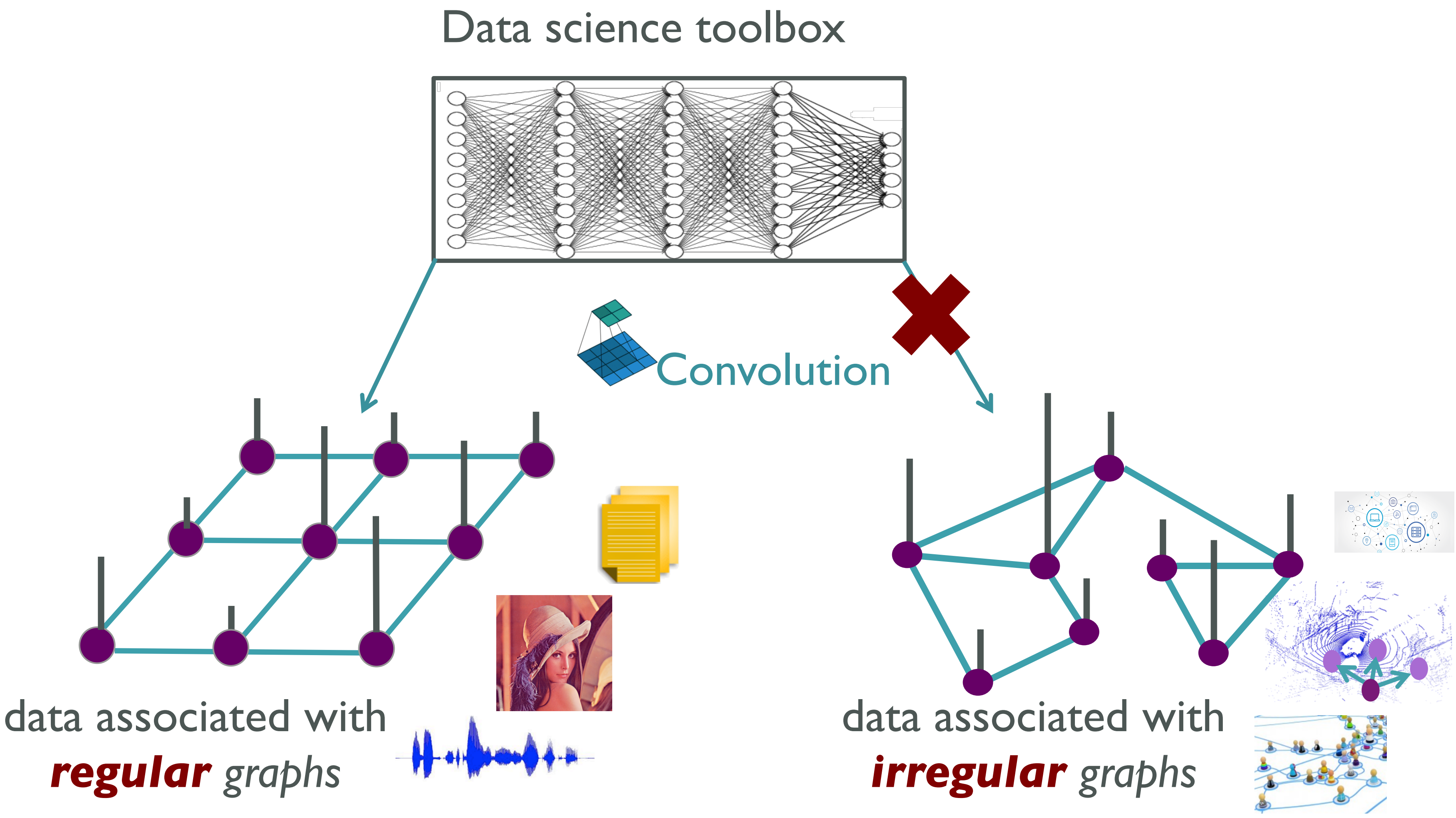
Action recognition



Semi-supervised learning

Recommender systems

Graph-structured data



Graph-structured data science

A data-science toolbox to process and learn from data associated with large-scale, complex, irregular graphs

Graph-structured data science

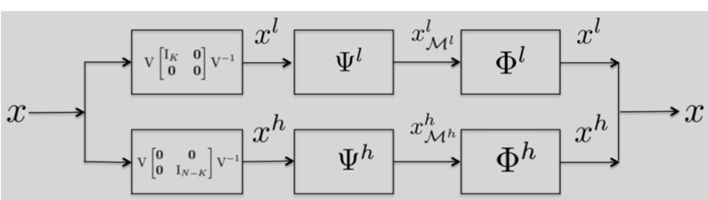
A data-science toolbox to process and learn from data associated with large-scale, complex, irregular graphs

Graph signal processing (GSP)

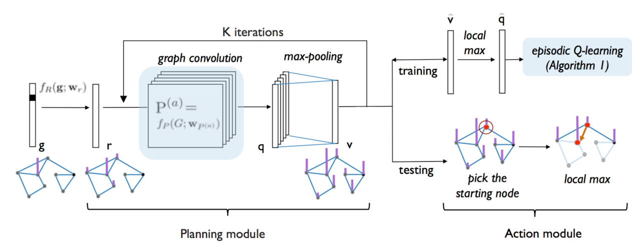
- Extend classical signal processing
- Analytical model
- Theoretical guarantee

Graph neural network (GNN)

- Extend deep learning techniques
- Data-driven model
- Empirical performance



Graph filter bank



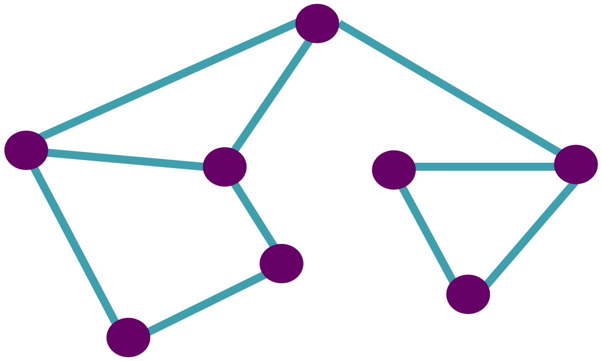
Graph neural network

Sampling & recovery of graph-structured data

GSP: Sampling & recovery

Task: Approximate the original graph-structured data by exploiting information from its subset

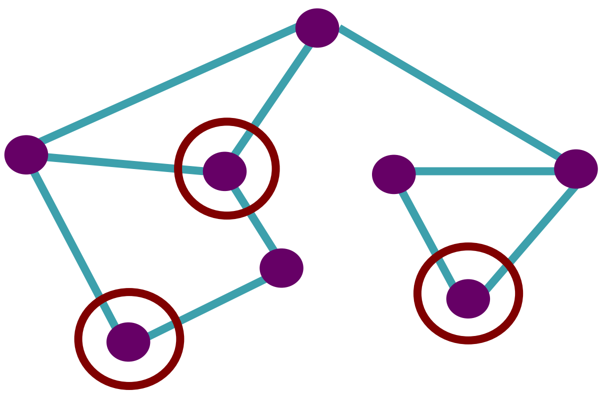
Look for data at each node



GSP: Sampling & recovery

Task: Approximate the original graph-structured data by exploiting information from its subset

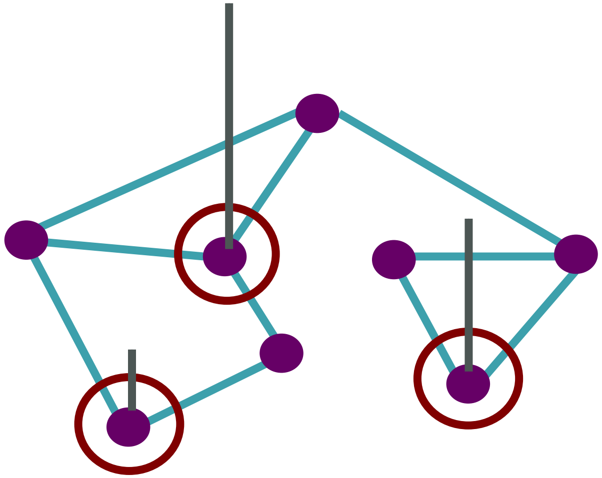
Select a few representative nodes



GSP: Sampling & recovery

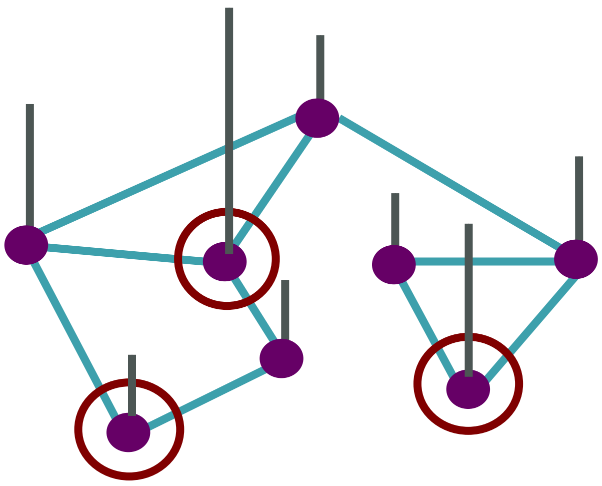
Task: Approximate the original graph-structured data by exploiting information from its subset

Query values at selected nodes



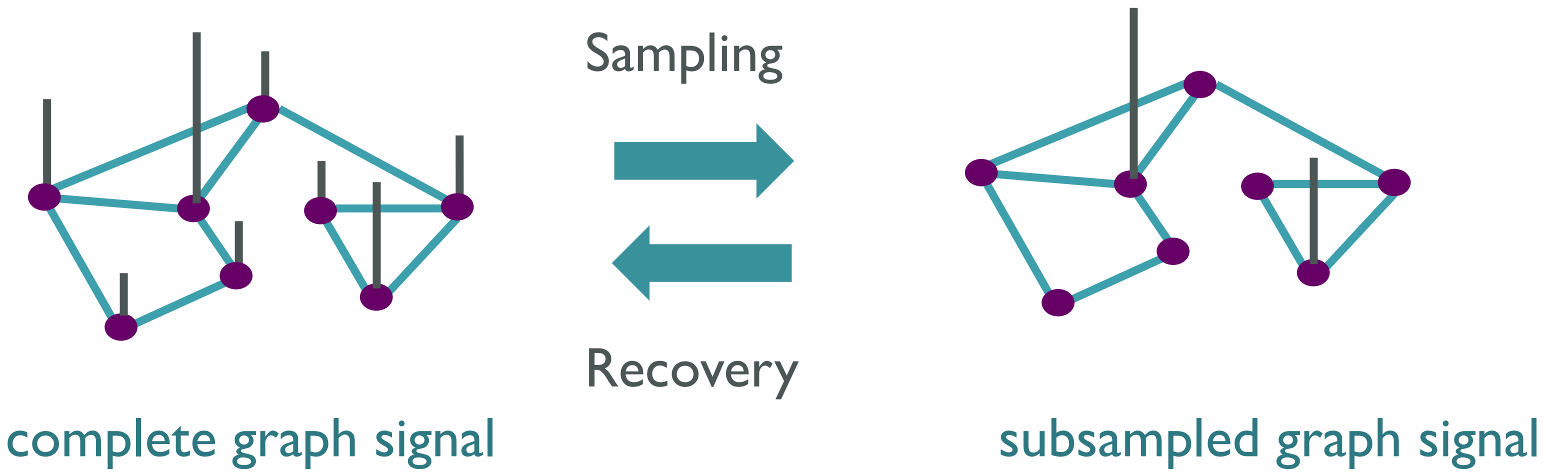
GSP: Sampling & recovery

Task: Approximate the original graph-structured data by exploiting information from its subset



GSP: Sampling & recovery

Task: Approximate the original graph-structured data by exploiting information from its subset

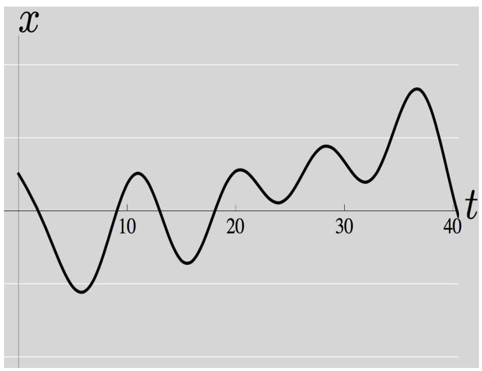


$$\text{sampling}: \qquad \mathbf{x}' = \Psi \mathbf{x}$$

$$\text{recovery}: \qquad \mathbf{x} = \Phi \mathbf{x}'$$

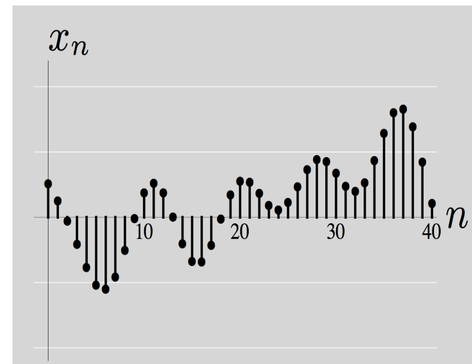
GSP: Sampling & recovery

Task: Approximate the original graph-structured data by exploiting information from its subset



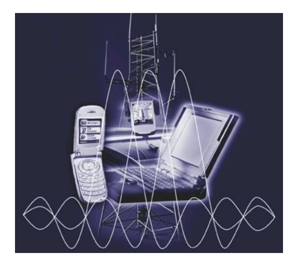
Sampling

Recovery



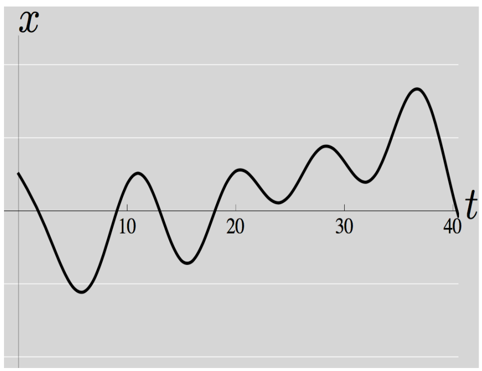
complete continuous signal

subsampled sequence



GSP: Sampling & recovery

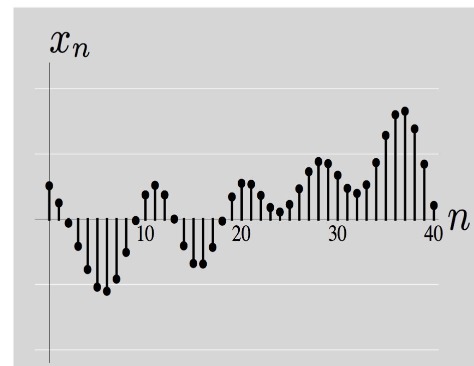
Task: Approximate the original graph-structured data by exploiting information from its subset



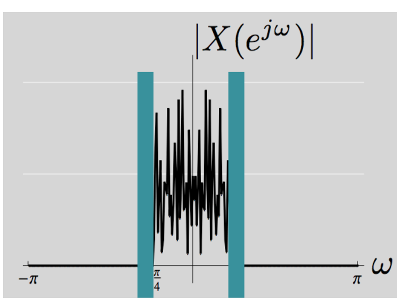
complete continuous signal

Sampling

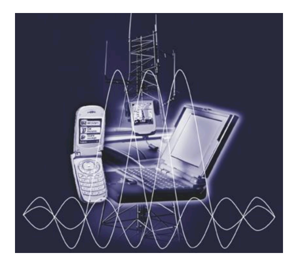
Recovery



subsampled sequence

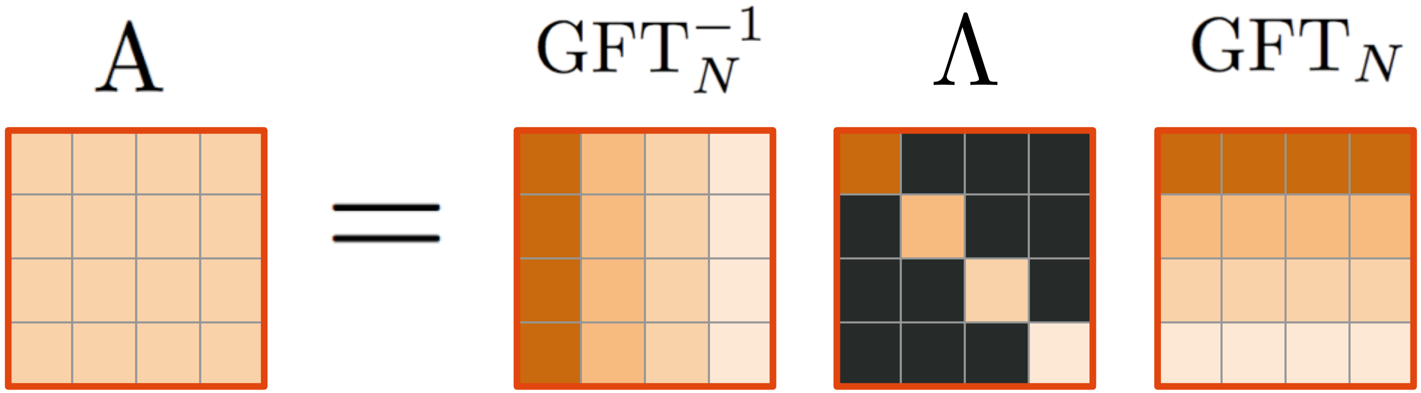


bandlimited in the Fourier domain



GSP: Sampling & recovery

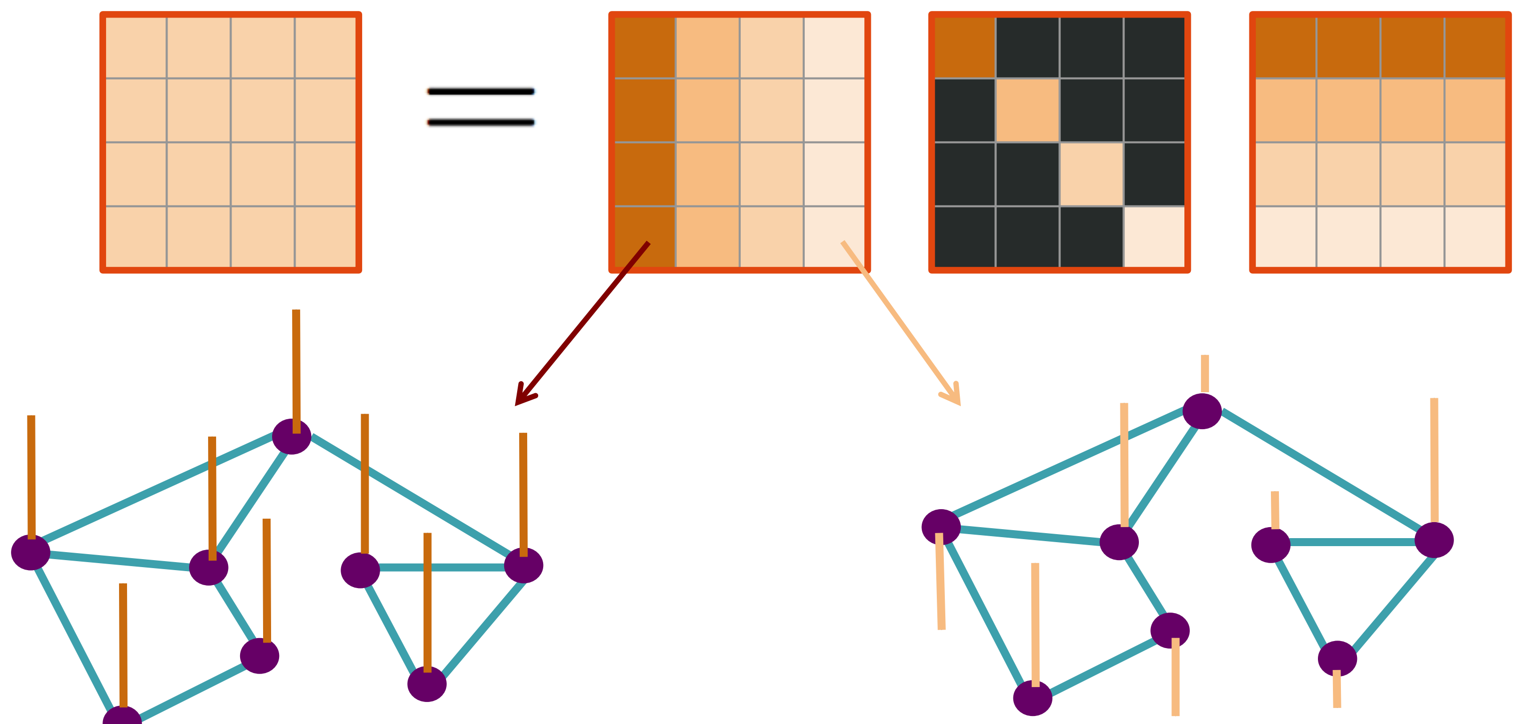
Graph Fourier transform



GSP: Sampling & recovery

Graph Fourier transform

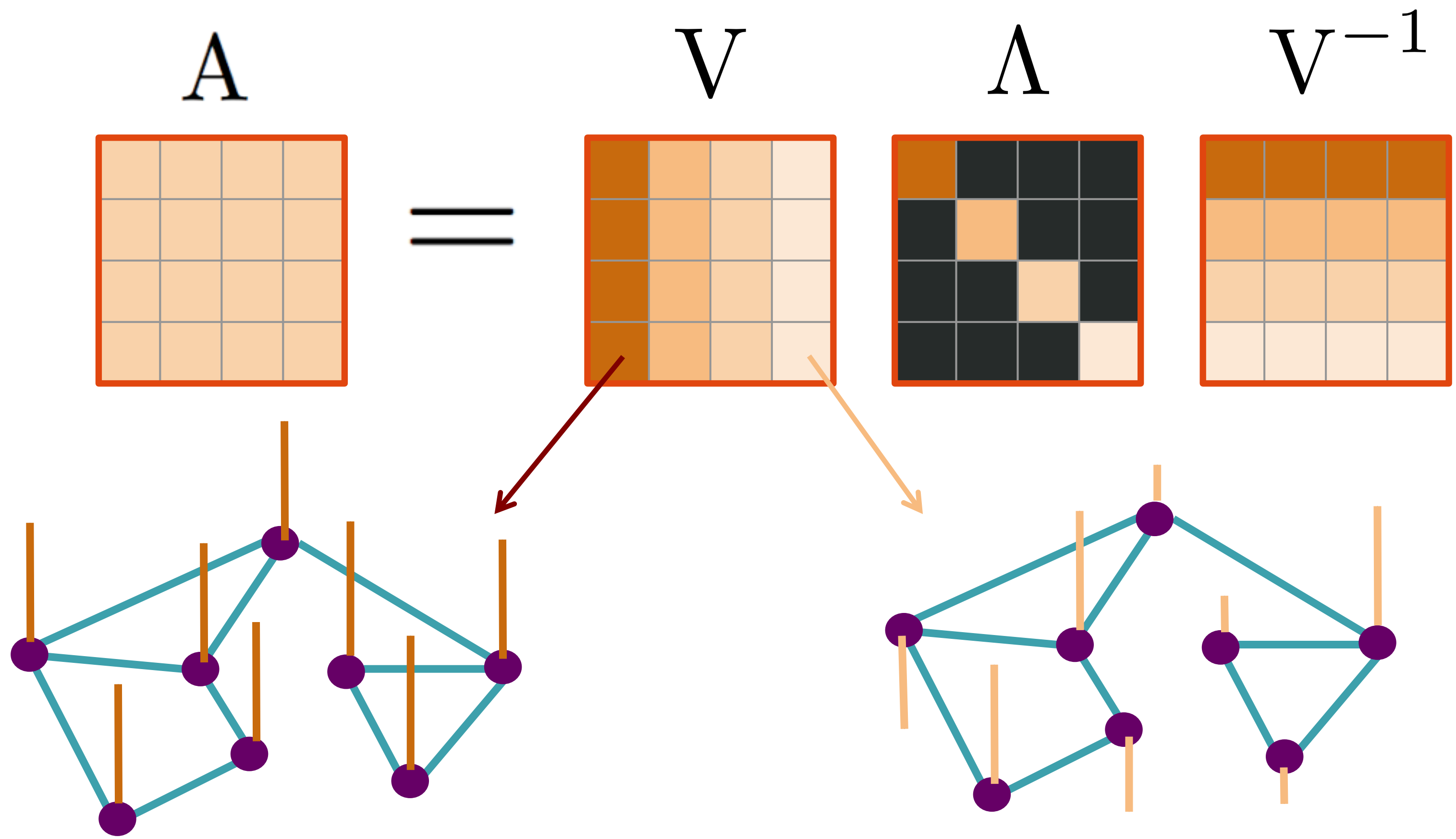
$$\mathbf{A} = \mathrm{GFT}_N^{-1} \; \Lambda \; \mathrm{GFT}_N$$



low-pass (smooth) Fourier vector high-pass (nonsmooth) Fourier vector

GSP: Sampling & recovery

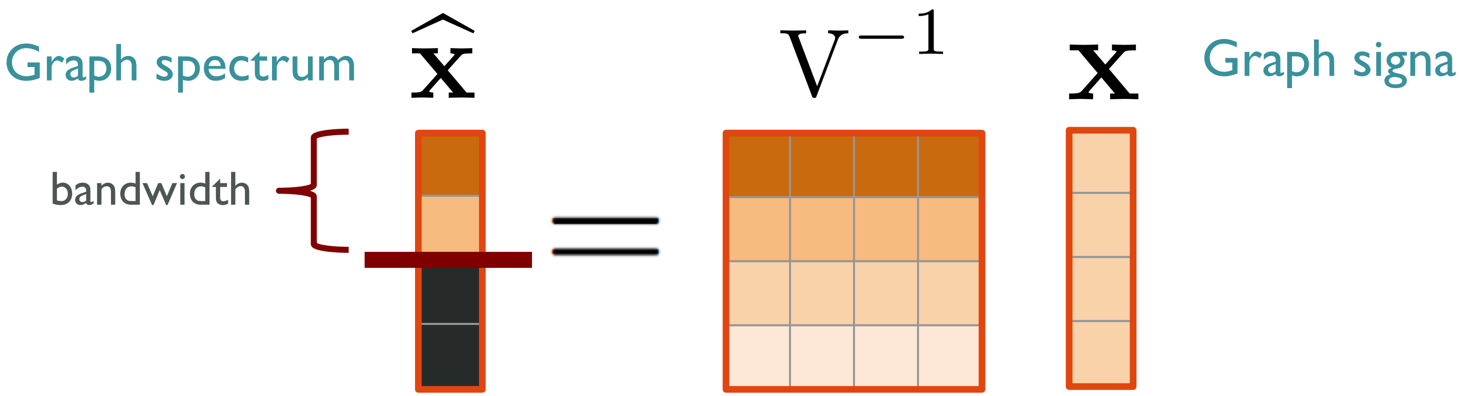
Graph Fourier transform



low-pass (smooth) Fourier vector   high-pass (nonsmooth) Fourier vector

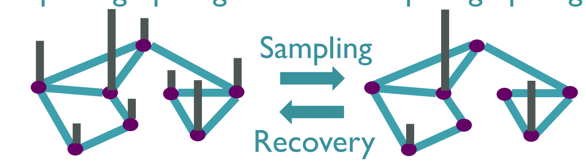
GSP: Sampling & recovery

Bandlimited graph signals



low-pass, smooth

GSP: Sampling & recovery



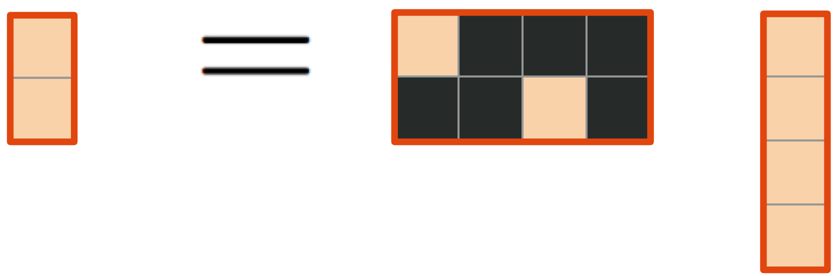
Problem formulation

- Sampling process

sampling operator: information bottleneck

graph signal: bandlimited class

$$\mathbf{x}' = \Psi \, \mathbf{x}$$



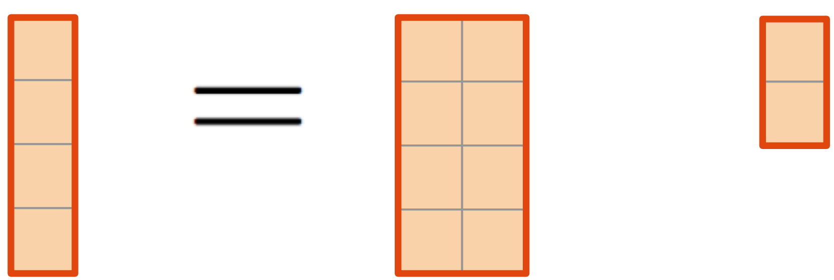
- Recovery process

recovery operator

reconstruction

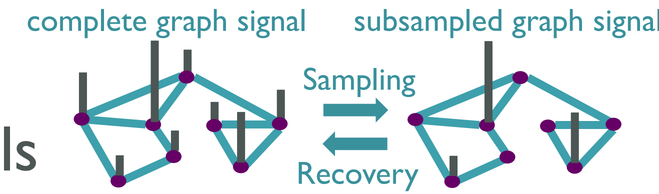
subsamples

$$\mathbf{x} = \Phi \, \mathbf{x}'$$



GSP: Sampling & recovery

Perfect recovery of bandlimited graph signals



Theorem 1. Let Ψ satisfy

qualified sampling operator

$$\boxed{\operatorname{rank}(\Psi\,\mathrm{V}_{(K)}) = K,}$$

where $\mathrm{V}_{(K)} \in \mathbb{R}^{N\times K}$ denotes the first K columns of V. For all $x \in \mathrm{BL}_K(\mathrm{V}^{-1})$, perfect recovery, $x = \Phi\Psi x$, is achieved by choosing

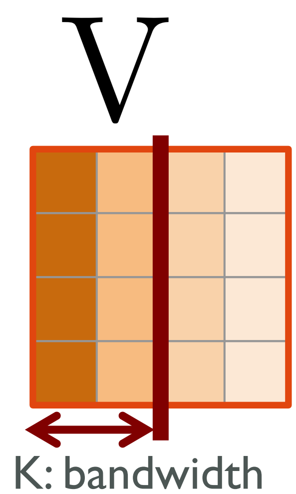
$$\Phi = \mathrm{V}_{(K)}\,\mathrm{U},$$

with $\mathrm{U}\,\Psi\,\mathrm{V}_{(K)}$ a $K \times K$ identity matrix.

S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, "Discrete signal processing on graphs: Sampling Theory", IEEE Transactions on Signal Processing, 2015.

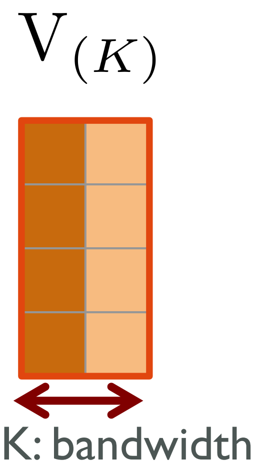
GSP: Sampling & recovery

Sufficient condition for a qualified sampling operator



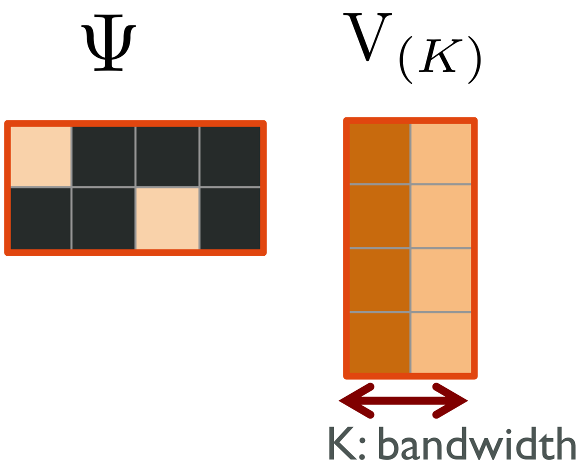
GSP: Sampling & recovery

Sufficient condition for a qualified sampling operator



GSP: Sampling & recovery

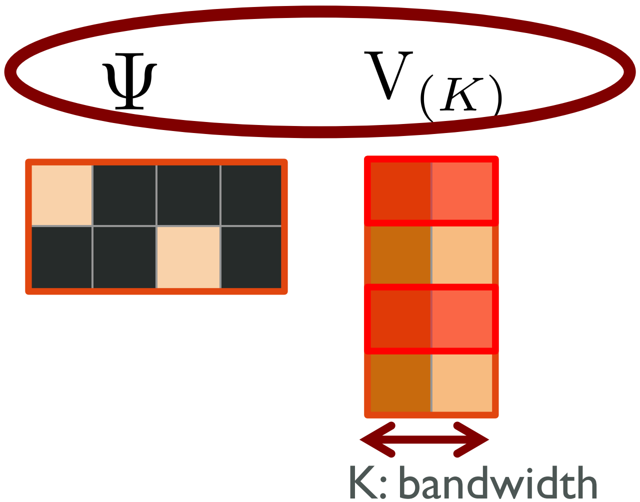
Sufficient condition for a qualified sampling operator



GSP: Sampling & recovery

Sufficient condition for a qualified sampling operator

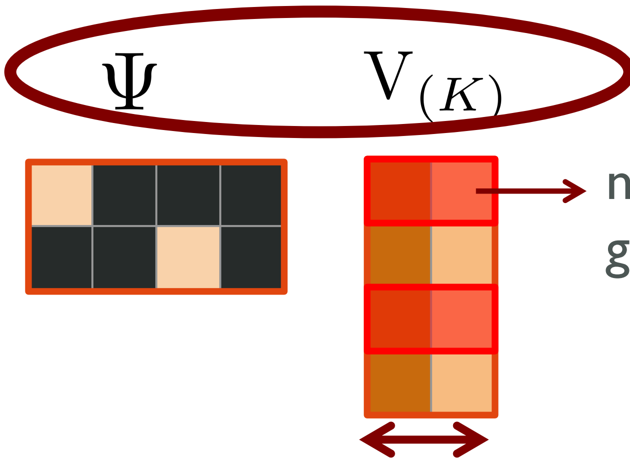
full rank: linearly independent



GSP: Sampling & recovery

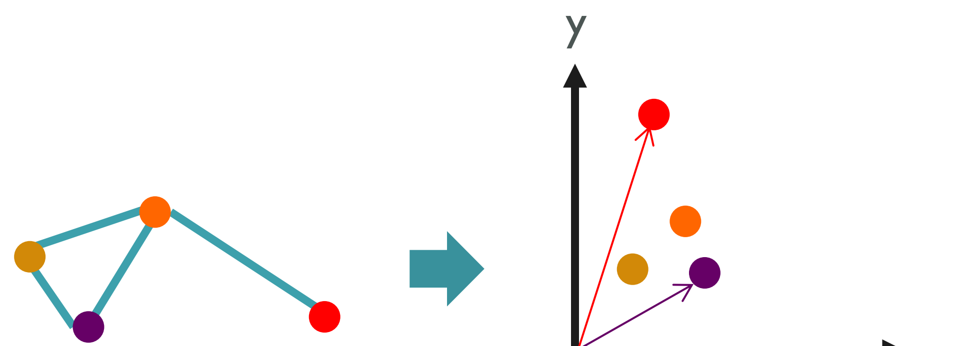
Sufficient condition for a qualified sampling operator

full rank: linearly independent



node features/coordinates in the graph spectral domain

K: bandwidth



Graph vertex domain

Graph spectral domain

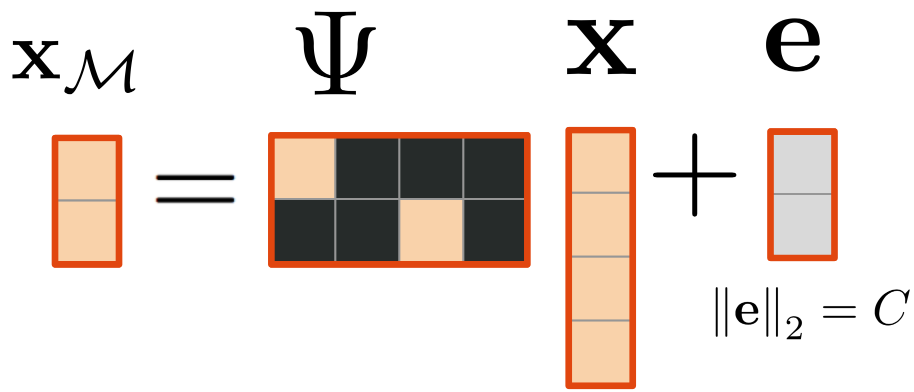
GSP: Sampling & recovery

Sufficient condition for a qualified sampling operator

- Full rank $\operatorname{rank}(\Psi\, \mathrm{V}_{(K)}) = K$

Optimal sampling operator

- Robust to noise
- Greedy algorithm



$$
\begin{aligned}
\Psi^{\mathrm{opt}} &= \arg\min_{\Psi} \max_{\mathbf{x}} \|\Phi(\Psi\, \mathbf{x} + \mathbf{e}) - \mathbf{x}\|_2 \\
&= \arg\min_{\Psi} \left\| \left(\Psi\, \mathrm{V}_{(K)}\right)^{\dagger} \right\|_2
\end{aligned}
$$

Algorithm 1 Optimal Sampling Operator via Greedy Algorithm

Input $\mathrm{V}_{(K)}$ the first K columns of V
M the number of samples
Output $\mathcal{M}$ sampling set
Function
while $|\mathcal{M}| < M$
$m = \arg\max_i \sigma_{\min}\left((\mathrm{V}_{(K)})_{\mathcal{M}+\{i\}}\right)$
$\mathcal{M} \leftarrow \mathcal{M} + \{m\}$
end
return $\mathcal{M}$

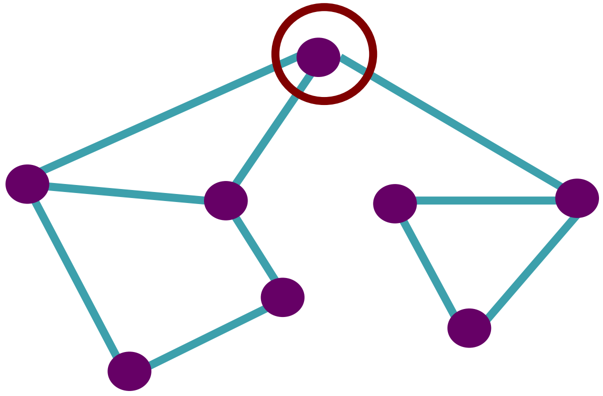
S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, "Discrete signal processing on graphs: Sampling Theory", IEEE Transactions on Signal Processing, 2015.

GSP: Sampling & recovery

Sampling process

- Deterministic: Either choose a node or discard; Accurate
- Randomized: Sample nodes according to a distribution; fast, scalable
  - Fundamental limits of sampling ability

sampling probability π_i



GSP: Sampling & recovery

Fundamental limits of sampling strategies $|\mathcal{M}| = m \to \infty$

- Minmax recovery error

Reconstruction error

$$\inf_{(\mathbf{x}^*, \boxed{\mathcal{M}) \in \Theta_{\exp}}} \sup_{\mathbf{x} \in \mathrm{ABL}_{\mathrm{A}}(K, \beta, \mu)} \boxed{\mathbb{E}_{\mathcal{M}} \left(\frac{\|\mathbf{x}^* - \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \right)}$$

Sampling strategy

S. Chen, R. Varma, A. Singh, J. Kovačević, "Signal Recovery on Graphs: Fundamental Limits of Sampling Strategies ", IEEE Transactions on Signal and Information Processing over Networks, 2016.

GSP: Sampling & recovery

Fundamental limits of sampling strategies $|\mathcal{M}| = m \to \infty$

- Minmax recovery error

lower bound

upper bound

$$cm^{-\gamma_1} \leq \inf_{(\mathbf{x}^*, \mathcal{M}) \in \Theta_{\exp}} \sup_{\mathbf{x} \in \mathrm{ABL}_{\mathrm{A}}(K, \beta, \mu)} \mathbb{E}_{\mathcal{M}} \left(\frac{\|\mathbf{x}^* - \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \right) \leq Cm^{-\gamma_2}$$

$$\boxed{\gamma_1 = \gamma_2}$$

Tight bound

- Lower bound: fundamental limits
- Upper bound: optimal algorithm

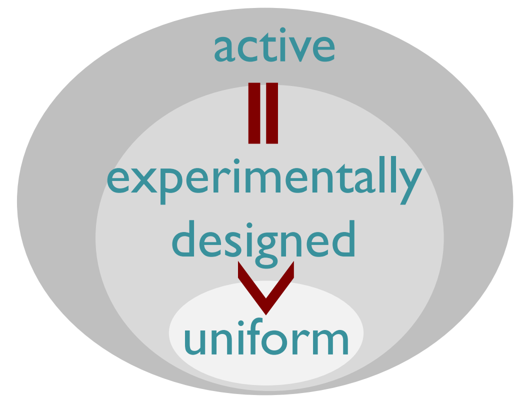
higher rate => faster decay => better sampling

S. Chen, R. Varma, A. Singh, J. Kovačević, "Signal Recovery on Graphs: Fundamental Limits of Sampling Strategies ", IEEE Transactions on Signal and Information Processing over Networks, 2016.

GSP: Sampling & recovery

Sampling process

- Fundamental limits of sampling strategies
  - Minmax recovery error
  - Lower bound
  - Sampling algorithm
  - Upper bound



Corollary 6. Let $A \in \mathbb{R}^{N \times N}$ be a type-2 graph with parameter K, for the class $ABL_A(K, \beta, \mu)$.

Uniform sampling

- Under uniform sampling,

$$\gamma_1 = \gamma_2 = \frac{2\beta}{2\beta+\gamma}, \ \gamma > 1$$

where constant $C > c > 0$, and the rate is achieved when κ is in the order of $m^{1/(2\beta+\gamma)}$ and $\gamma = \log(N)/\log(\kappa) \geq 1$;

Experimentally designed sampling & feedback-based sampling

$$\gamma_1 = \gamma_2 = \frac{2\beta}{2\beta+1}$$

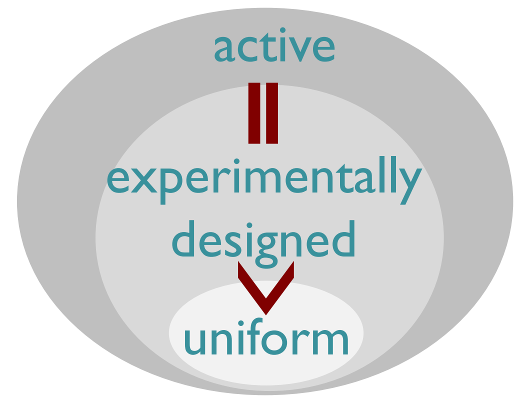
where constant $C > 0$, the rate is achieved when κ is in the order of $m^{1/(2\beta+1)}$ and upper bounded by N.

S. Chen, R. Varma, A. Singh, J. Kovačević, “Signal Recovery on Graphs: Fundamental Limits of Sampling Strategies ”, IEEE Transactions on Signal and Information Processing over Networks, 2016.

GSP: Sampling & recovery

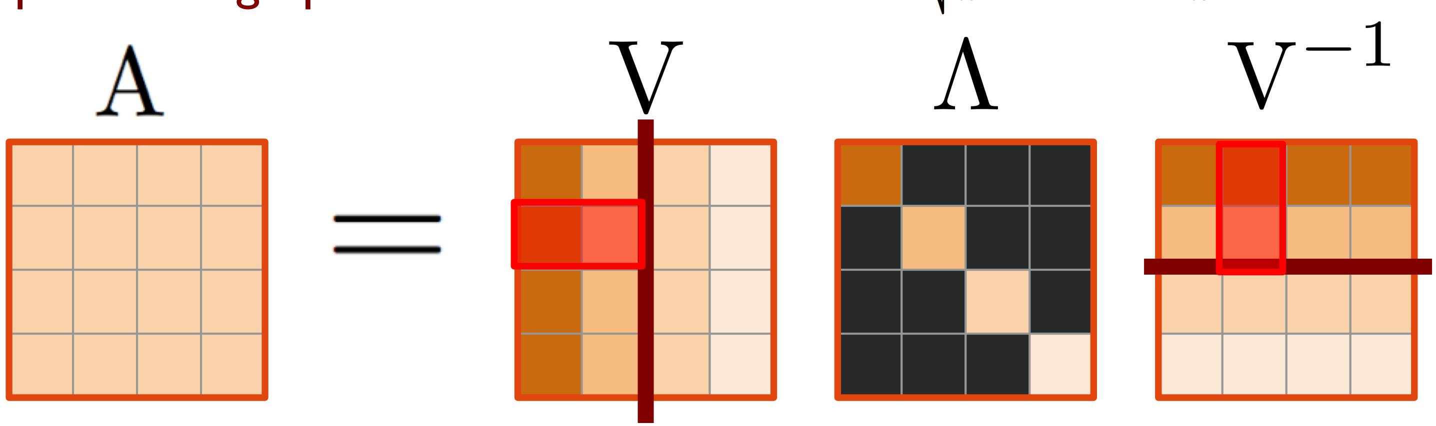
Sampling process

- Fundamental limits of sampling strategies
  - Minmax recovery error
  - Lower bound
  - Sampling algorithm
  - Upper bound



Optimal sampling distribution depends on properties of graph Fourier

$$\pi_i \propto \sqrt{\sum_{k=1}^{\kappa} \left(V^{-1}_{k,i}\right)^2 \sum_{k=1}^{\kappa} (V_{i,k})^2}$$

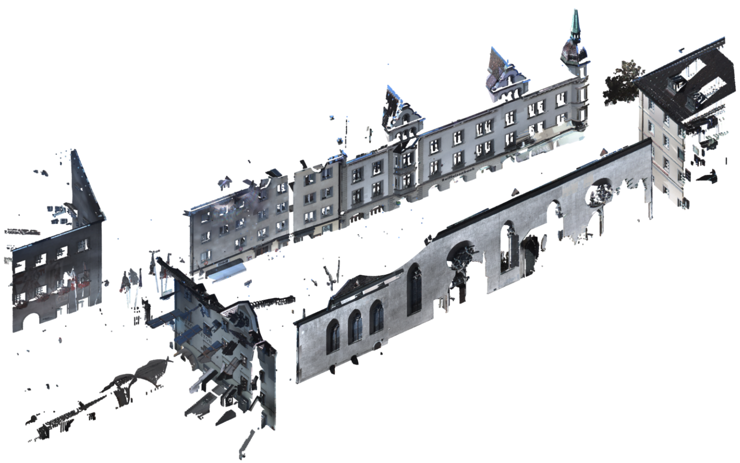


S. Chen, R. Varma, A. Singh, J. Kovačević, "Signal Recovery on Graphs: Fundamental Limits of Sampling Strategies ", IEEE Transactions on Signal and Information Processing over Networks, 2016.

GSP: Sampling & recovery

Applications

- Resampling of 3D point clouds



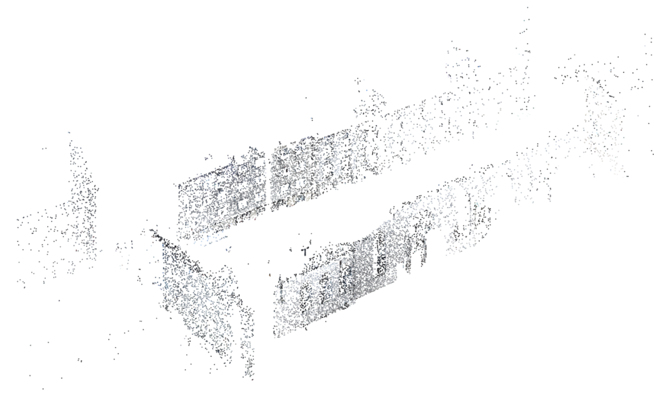
Over 30 million 3D points

S Chen, D Tian, C Feng, A Vetro, J. Kovačević, “Fast Resampling of Three-Dimensional Point Clouds via Graphs”, IEEE Transactions on Signal Processing, 2018

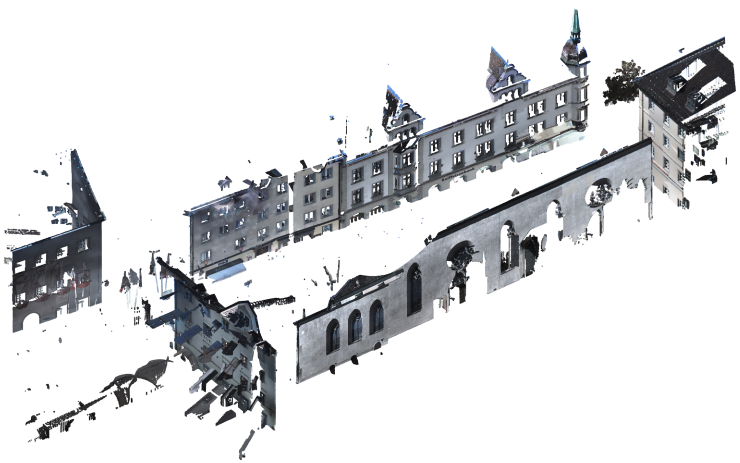
GSP: Sampling & recovery

Applications

- Resampling of 3D point clouds



uniform sampling (1000)



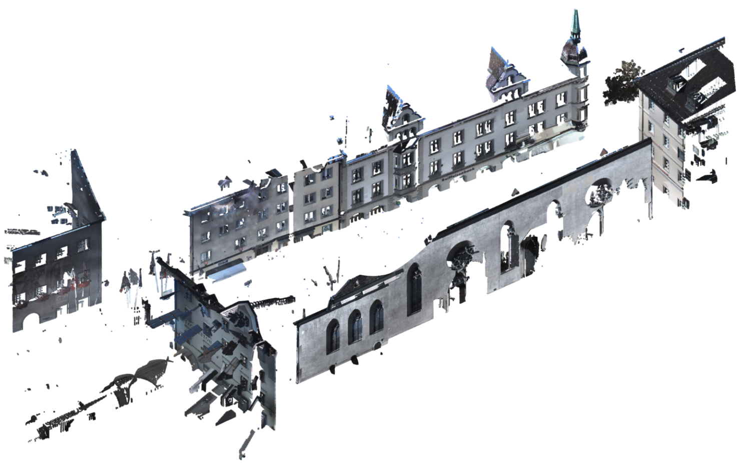
Over 30 million 3D points

S Chen, D Tian, C Feng, A Vetro, J. Kovačević, "Fast Resampling of Three-Dimensional Point Clouds via Graphs", IEEE Transactions on Signal Processing, 2018

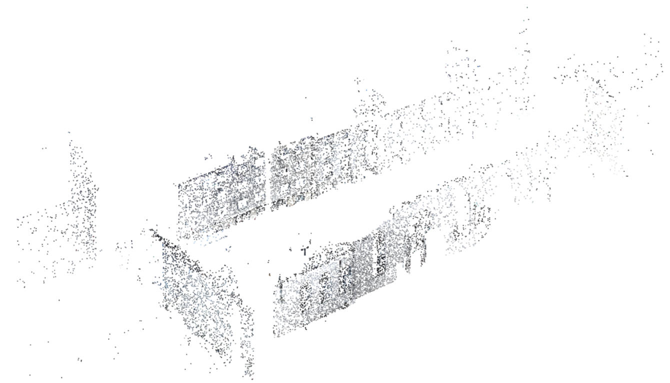
GSP: Sampling & recovery

Applications

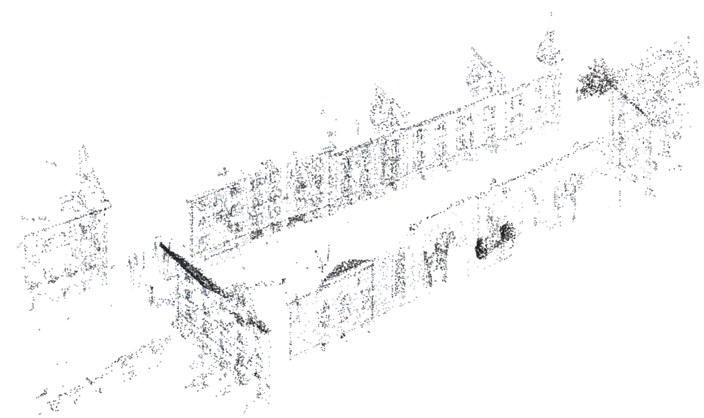
- Resampling of 3D point clouds



Over 30 million 3D points



uniform sampling (1000)

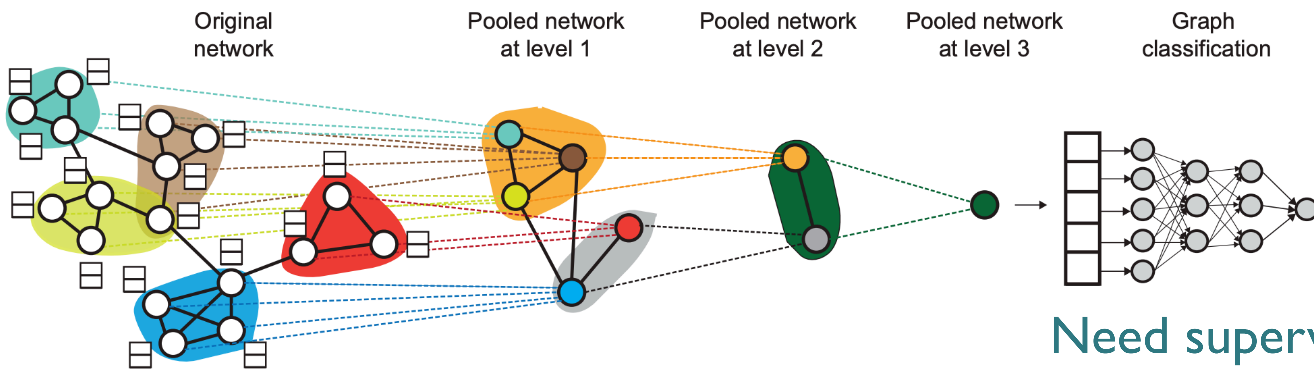


designed sampling (1000)

S Chen, D Tian, C Feng, A Vetro, J. Kovačević, “Fast Resampling of Three-Dimensional Point Clouds via Graphs”, IEEE Transactions on Signal Processing, 2018

GNN: Sampling & recovery

Diffpool: Graph pooling in multiscale graph neural network



Need supervision

Trainable sampling operator $\Psi = \mathrm{softmax}\left(\mathrm{GNN}\left(\mathrm{A}, \mathbf{x}\right)\right)$

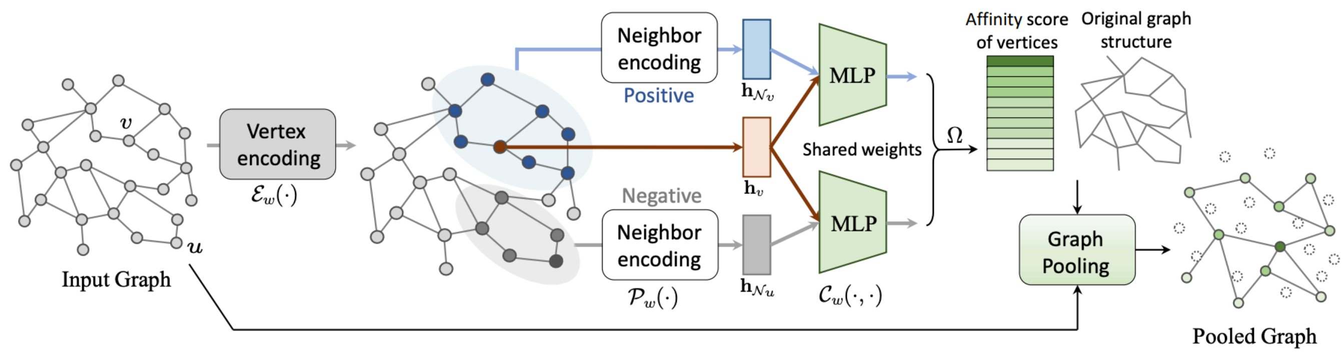
Data pooling $\mathbf{x}' = \Psi \mathbf{x}$

Structure pooling $\mathrm{A}' = \Psi \, \mathrm{A} \, \Psi^T$

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, Jure Leskovec, Hierarchical Graph Representation Learning with Differentiable Pooling, NeurIPS 18

GNN: Sampling & recovery

Proposed graph pooling in multiscale graph neural network



Self-supervision / supervision

$$\max_{\mathcal{M} \subset \mathcal{V}} \quad C(\mathcal{M}), \quad \text{subject to} \quad |\mathcal{M}| = M.$$

Vertex-selection criterion

$$C(\mathcal{M}) = I^{(\mathcal{M})}(\mathbf{v}, \mathbf{n}) = D_{\mathrm{KL}}(P_{\mathbf{v},\mathbf{n}} \| P_{\mathbf{v}} \otimes P_{\mathbf{n}})$$
$$\geq \sup_{T \in \mathcal{T}} \left\{ \mathbb{E}_{\mathbf{s}_v, \mathbf{q}_{\mathcal{N}_v} \sim P_{\mathbf{v},\mathbf{n}}} \left[T(\mathbf{s}_v, \mathbf{q}_{\mathcal{N}_v}) \right] - \mathbb{E}_{\mathbf{s}_v \sim P_{\mathbf{v}}, \mathbf{q}_{\mathcal{N}_u} \sim P_{\mathbf{n}}} \left[e^{T(\mathbf{s}_v, \mathbf{q}_{\mathcal{N}_u}) - 1} \right] \right\}$$

GNN

node feature

neighborhood feature

S. Chen, M. Li, "Sampling and Recovery of Graph Signals: A Neural Network's Perspective", 2020

GNN: Sampling & recovery

Experimental results

- Graph classification

Dataset # Graphs (Classes) Avg. # Vertices	IMDB-B 1000 (2) 19.77	IMDB-M 1500 (3) 13.00	COLLAB 5000 (3) 74.49	D&D 1178 (2) 284.32	PROTEINS 1113 (2) 39.06	ENZYMES 600 (6) 32.63
PatchySAN [34]	76.27 ± 2.6	69.70 ± 2.2	43.33 ± 2.8	72.60 ± 2.2	75.00 ± 2.8	-
ECC [40]	-	-	67.79	72.54	72.65	53.50
Set2Set [20]	-	-	71.75	78.12	74.29	60.15
DGCNN [49]	70.00 ± 0.9	47.83 ± 0.9	73.76 ± 0.5	79.37 ± 0.9	73.68 ± 0.9	-
DiffPool [46]	70.40	47.83	75.84	80.64	76.25	62.53
Graph U-Net [19]	72.10	48.33	77.56	82.43	77.68	58.57
SAGPool [28]	72.80	49.43	78.52	82.84	78.28	60.23
AttPool [24]	73.60	50.67	77.04	79.20	76.50	59.76
StructPool [47]	74.70	52.47	74.22	84.19	80.36	**63.83**
GXN	$\mathbf{77.30 \pm 0.8}$	$\mathbf{54.57 \pm 0.9}$	$\mathbf{80.62 \pm 0.8}$	$\mathbf{84.26 \pm 1.3}$	$\mathbf{80.38 \pm 1.2}$	60.43 ± 1.0

S. Chen, M. Li, “Sampling and Recovery of Graph Signals: A Neural Network’s Perspective”, 2020

GNN: Sampling & recovery

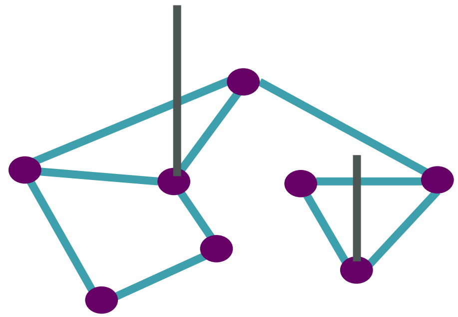
Experimental results

- Node classification

Dataset # Vertices (Classes) Supervision	Cora 2708 (7) full-sup.	 semi-sup.	Citeseer 3327 (6) full-sup.	 semi-sup.	Pubmed 19717 (3) full-sup.	 semi-sup.
DeepWalk [36]	78.4 ± 1.7	67.2 ± 2.0	68.5 ± 1.8	43.2 ± 1.6	79.8 ± 1.1	65.3 ± 1.1
ChebNet [10]	86.4 ± 0.5	81.2 ± 0.5	78.9 ± 0.4	69.8 ± 0.5	88.7 ± 0.3	74.4 ± 0.4
GCN [27]	86.6 ± 0.4	81.5 ± 0.5	79.3 ± 0.5	70.3 ± 0.5	90.2 ± 0.3	79.0 ± 0.3
GAT [43]	87.8 ± 0.7	83.0 ± 0.7	80.2 ± 0.6	73.5 ± 0.7	90.6 ± 0.4	79.0 ± 0.3
FastGCN [7]	85.0 ± 0.8	80.8 ± 1.0	77.6 ± 0.8	69.4 ± 0.8	88.0 ± 0.6	78.5 ± 0.7
ASGCN [25]	87.4 ± 0.3	-	79.6 ± 0.2	-	90.6 ± 0.3	-
Graph U-Net [19]	-	84.4	-	73.2	-	79.6
GXN	$\mathbf{88.9 \pm 0.4}$	$\mathbf{85.1 \pm 0.6}$	$\mathbf{80.9 \pm 0.4}$	$\mathbf{74.8 \pm 0.4}$	$\mathbf{91.8 \pm 0.3}$	$\mathbf{80.2 \pm 0.3}$

S. Chen, M. Li, "Sampling and Recovery of Graph Signals: A Neural Network's Perspective", 2020

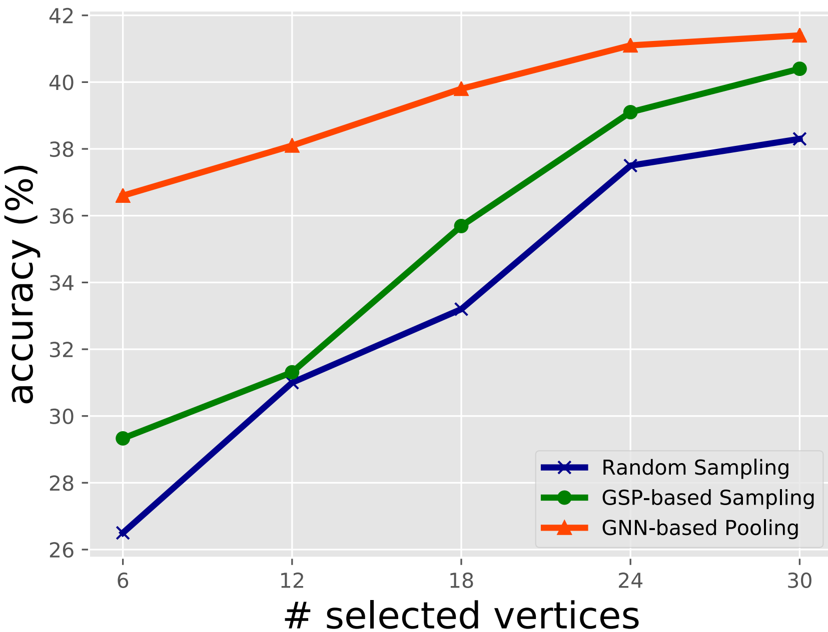
GNN: Sampling & recovery



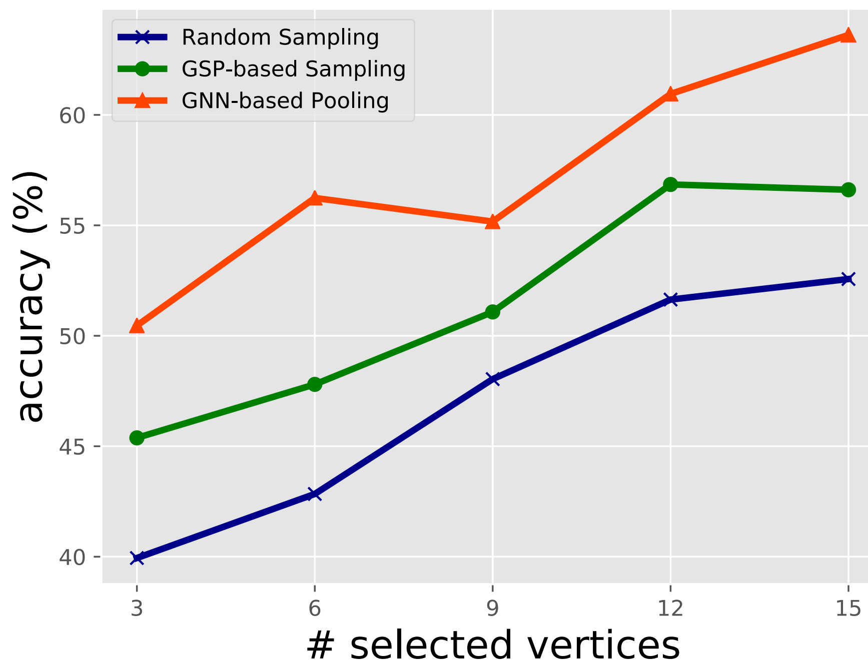
Experimental results

- Active-sampling-based semi-supervised learning

Citeseer: 3327 nodes

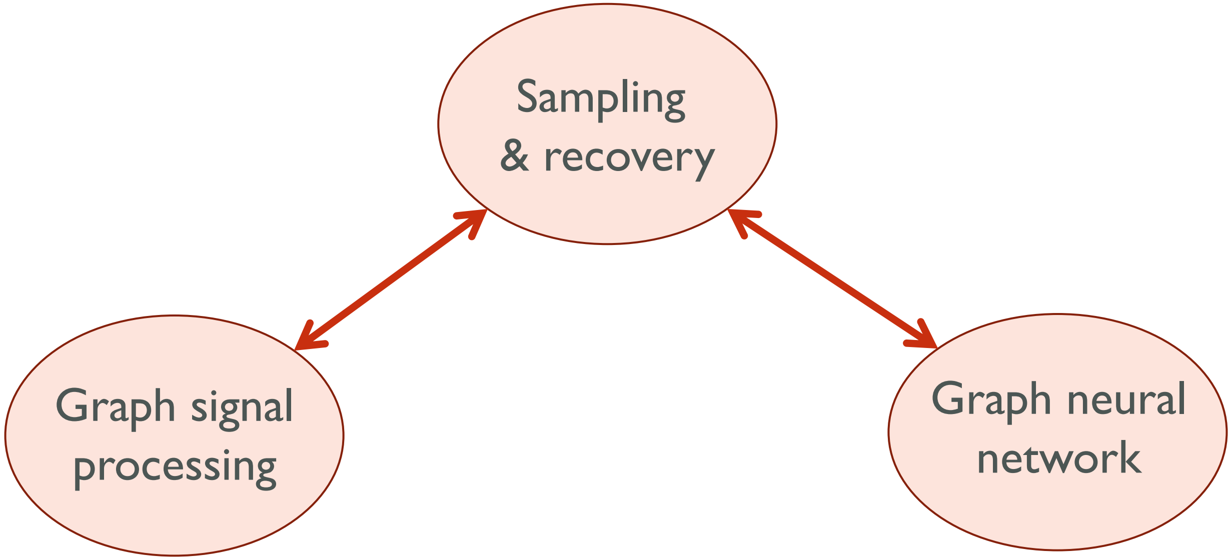


Pubmed: 19717 nodes



S. Chen, M. Li, “Sampling and Recovery of Graph Signals: A Neural Network’s Perspective”, 2020

Bridging GSP and GNN



Bridging GSP and GNN

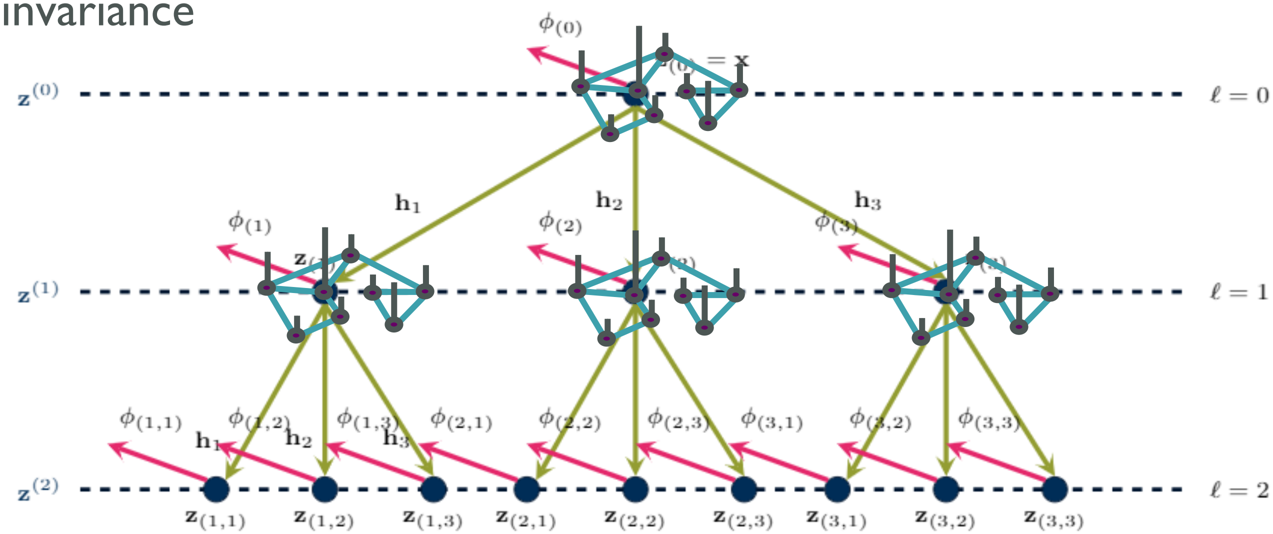
Bridging GSP and GNN

- Scattering
- Algorithm unrolling

Graph scattering transforms

Graph scattering transforms are nontrainable GCNs

- Parameters of the graph convolutions are mathematically designed
- Theoretical property
  - Energy preservation
  - Permutation invariance
  - Stability

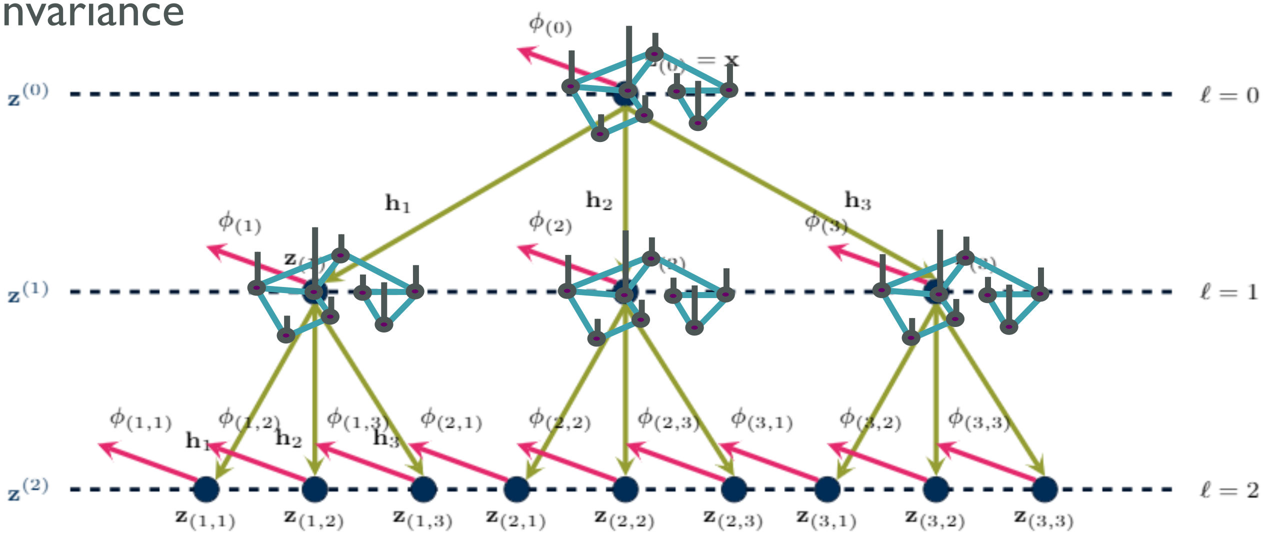


- Dongmian Zou, Gilad Lerman, "Graph Convolutional Neural Networks via Scattering", Applied and Computational Harmonic Analysis, 2019
- Fernando Gama, Alejandro Ribeiro, Joan Bruna, "Diffusion Scattering Transforms on Graphs" ICLR, 2019

Graph scattering transforms

Graph scattering transforms are nontrainable GCNs

- Parameters of the graph convolutions are mathematically designed
- Theoretical property
  - Energy preservation
  - Permutation invariance
  - Stability



Scattering => exponential growth!

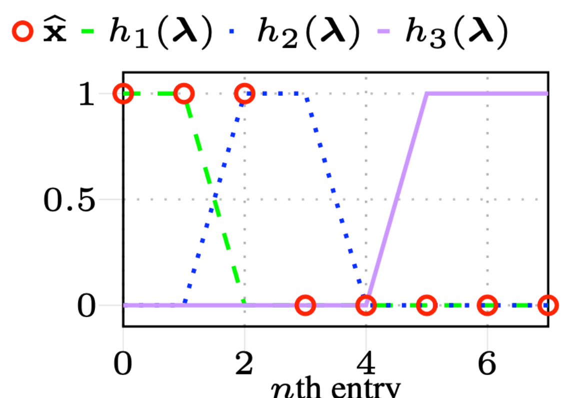
- Dongmian Zou, Gilad Lerman, "Graph Convolutional Neural Networks via Scattering", Applied and Computational Harmonic Analysis, 2019
- Fernando Gama, Alejandro Ribeiro, Joan Bruna, "Diffusion Scattering Transforms on Graphs" ICLR, 2019

Graph scattering transforms

Main idea: a pruning framework to select informative features of the graph scattering transforms

- Optimal pruning

$$\max_{\{f_{(p,j)}\}_{j=1}^{J}} \quad \sum_{j=1}^{J} \left(\sum_{n=1}^{N} \left(\widehat{h}_j(\lambda_n)^2 - \tau \right) [\widehat{\mathbf{z}}_{(p)}]_n^2 \right) f_{(p,j)}$$
$$\text{s. t.} \quad f_{(p,j)} \in \{0,1\}, \;\; j = 1, \ldots, J$$



GFTs of $\mathbf{x}$, $h_1(\mathbf{S})$, $h_2(\mathbf{S})$, $h_3(\mathbf{S})$

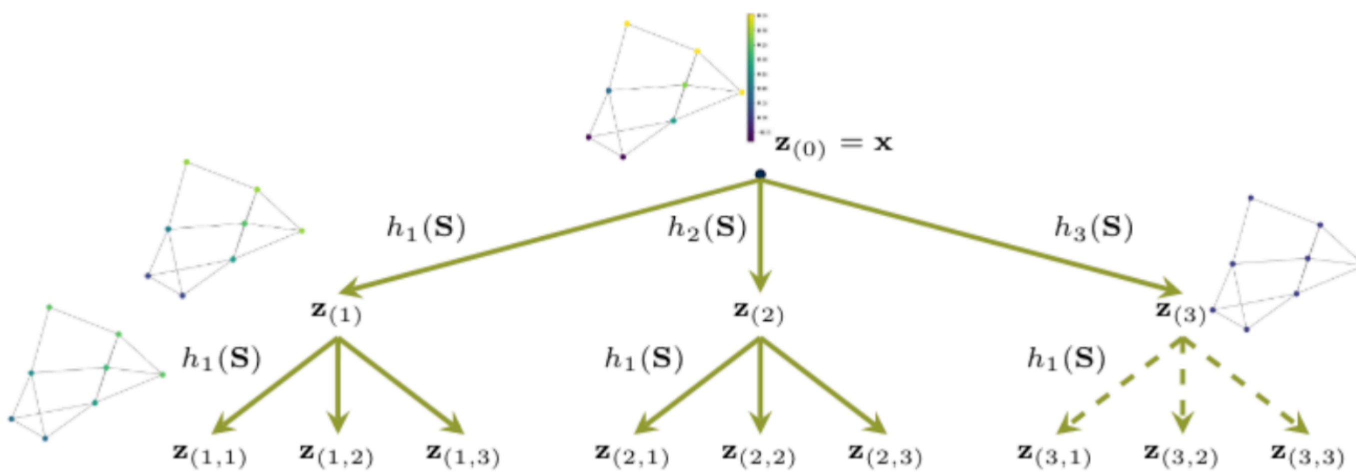
V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph scattering transforms

Main idea: a pruning framework to select informative features of the graph scattering transforms

- Optimal pruning

$$f^*_{(p,j)} = \begin{cases} 1 & \textit{if } \dfrac{\|\mathbf{z}_{(p,j)}\|^2}{\|\mathbf{z}_{(p)}\|^2} > \tau, \\ 0 & \textit{if } \dfrac{\|\mathbf{z}_{(p,j)}\|^2}{\|\mathbf{z}_{(p)}\|^2} < \tau. \end{cases}, \; j = 1, \ldots, J$$

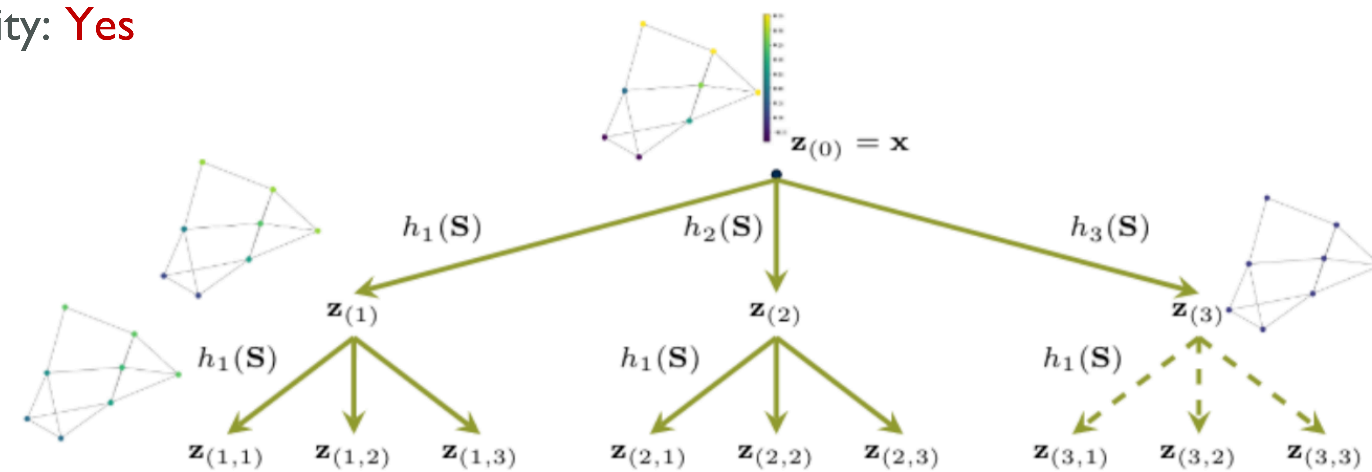


V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph scattering transforms

Main idea: a pruning framework to select informative features of the graph scattering transforms

- Optimal pruning
- Theoretical property
  - Energy preservation: No
  - Permutation invariance: ?
  - Stability: Yes



V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph scattering transforms

Main idea: a pruning framework to select informative features of the graph scattering transforms

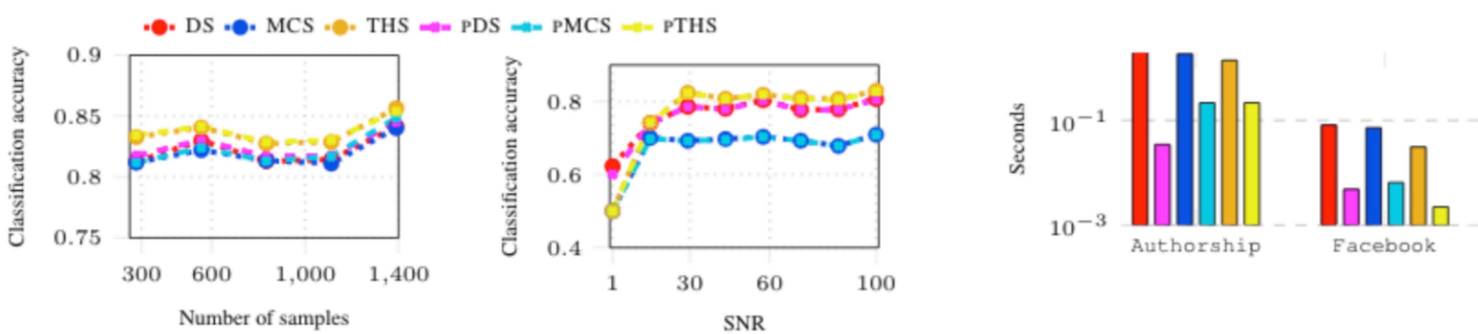
	Method	Data Set			
		ENZYMES	D&D	COLLAB	PROTEINS
Kernel	SHORTEST-PATH	42.32	78.86	59.10	76.43
	WL-OA	60.13	79.04	80.74	75.26
GNNs	PATCHYSAN	–	76.27	72.60	75.00
	GRAPHSAGE	54.25	75.42	68.25	70.48
	ECC	53.50	74.10	67.79	72.65
	SET2SET	60.15	78.12	71.75	74.29
	SORTPOOL	57.12	79.37	73.76	75.54
	DIFFPOOL-DET	58.33	75.47	**82.13**	75.62
	DIFFPOOL-NOLP	62.67	79.98	75.63	77.42
	DIFFPOOL	**64.23**	81.15	75.50	78.10
Scattering	GSC	53.88	76.57	76.88	74.03
	GST	59.84	79.28	77.32	76.23
	PGST (Ours)	60.25	**81.27**	78.40	**78.57**

Mathematically designed framework is comparable with trained framework

V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph scattering transforms

Main idea: a pruning framework to select informative features of the graph scattering transforms



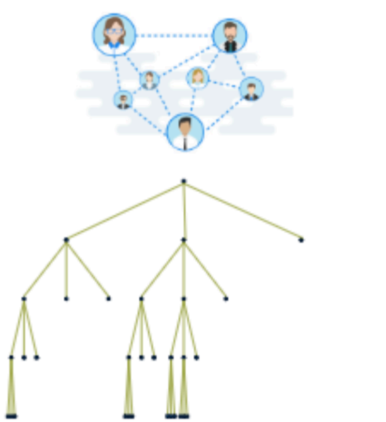
(a) Authorship (b) Facebook (c) Runtime

Similar performance, but 10 times faster!

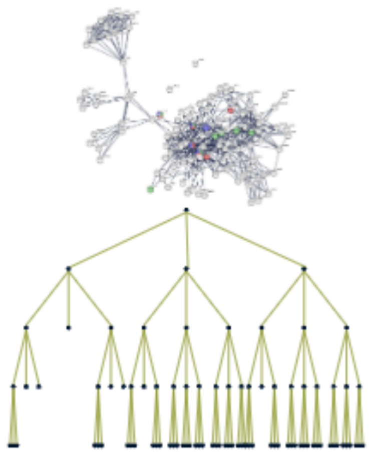
V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph scattering transforms

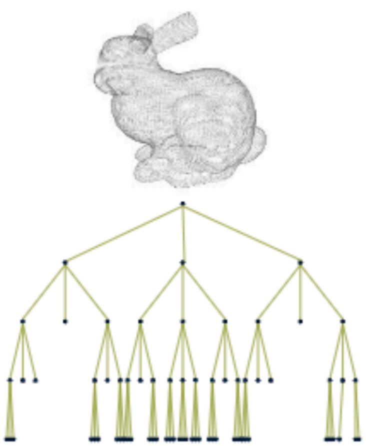
Main idea: a pruning framework to select informative features of the graph scattering transforms



(a) Academic collaboration



(b) Protein-protein network



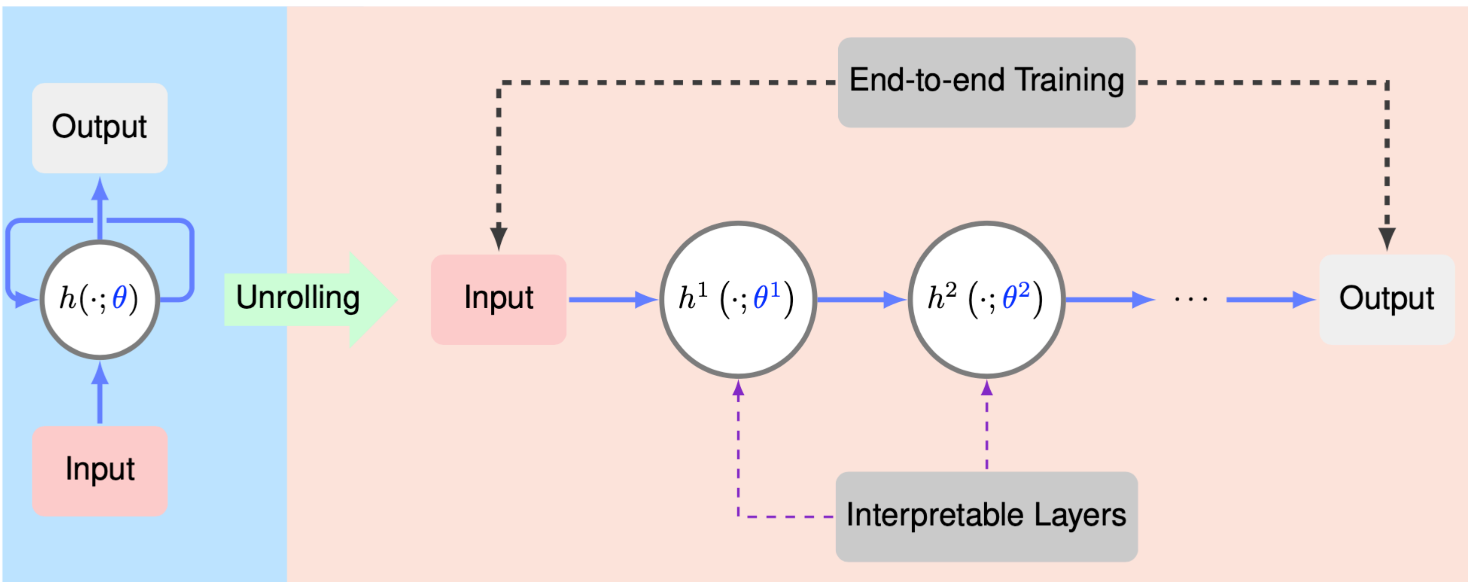
(c) 3D point cloud

Pruning pattern reflects the complexity of dataset

V. N. Ioannidis, S. Chen, G. B. Giannakis, Pruned graph scattering transforms, ICLR 2020

Graph unrolling networks

Algorithm unrolling framework: Transform an iterative algorithm to a multilayer neural network



Signal processing

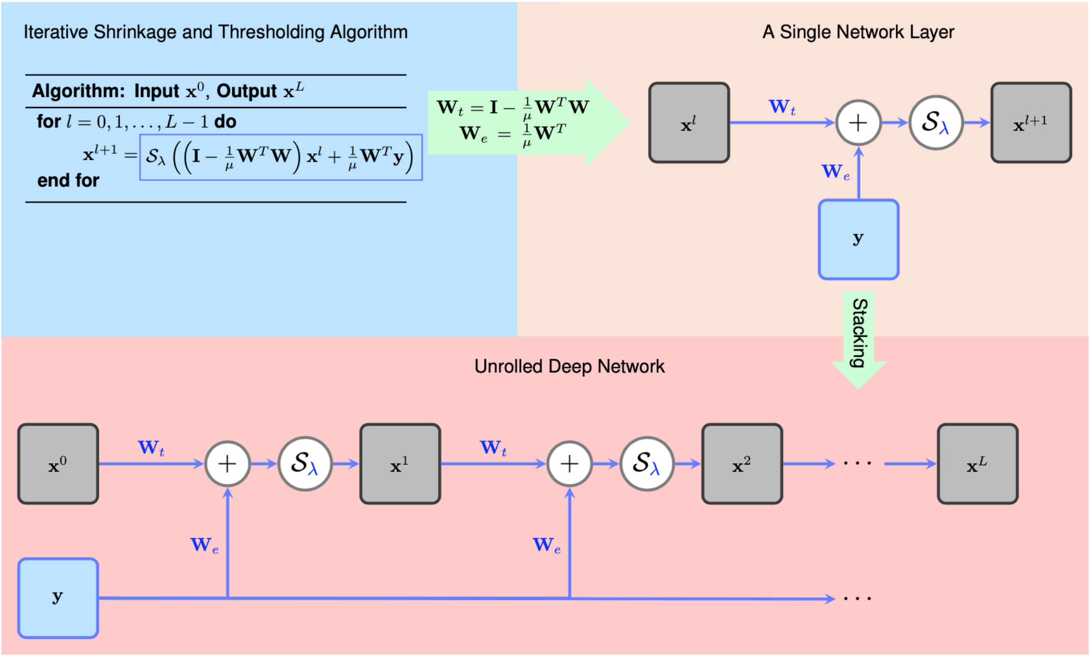
Neural network

Vishal Monga, Yuelong Li, Yonina C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing". IEEE Signal Processing Magazine, 2020

Graph unrolling networks

Algorithm unrolling framework: Transform an iterative algorithm to a multilayer neural network

Sparse coding $$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathrm{W}\,\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1$$



K. Gregor and Y. LeCun, "Learning Fast Approximations of Sparse Coding," ICML 2010.

Graph unrolling networks

Algorithm unrolling framework: Transform an iterative algorithm to a multilayer neural network

$$\min_{\mathbf{s} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{t} - \mathbf{x}\|_2^2 + u(\mathbf{y}) + r(\mathbf{z}),$$
$$\text{subject to} \quad \mathbf{x} = \mathbf{h} *_v \mathbf{s}, \quad \mathbf{y} = \mathrm{P}\,\mathbf{x}, \quad \mathbf{z} = \mathrm{Q}\,\mathbf{s}$$

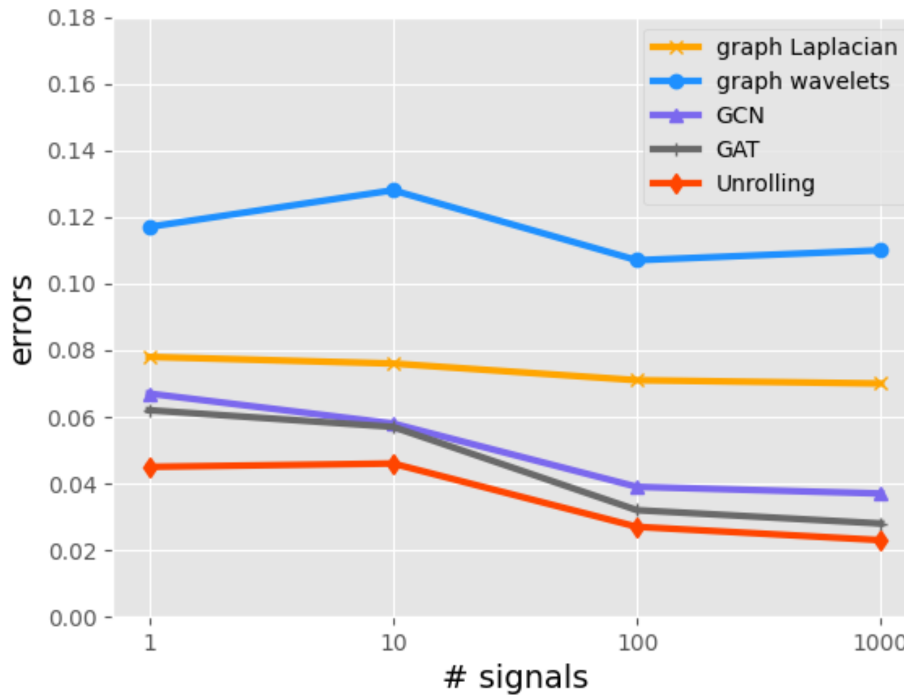
Analytical iterative solution		One network layer
$\mathbf{x} \leftarrow \widetilde{\mathrm{P}}\left(\mu_1 \mathbf{h} *_v \mathbf{s} + \mathbf{t} + \mu_2 \mathrm{P}^T \mathbf{y}\right),$	$\longleftrightarrow$	$\mathbf{x} \leftarrow \mathbb{A} *_a \mathbf{s} + \mathbb{B} *_a \mathbf{t} + \mathbb{C} *_a \left(\mathrm{P}^T \mathbf{y}\right)$
$\mathbf{s} \leftarrow \widetilde{\mathrm{Q}}\left(\mu_1 \mathbf{h} *_v^T \mathbf{x} + \mu_3 \mathrm{Q}^T \mathbf{z}\right),$	$\longleftrightarrow$	$\mathbf{s} \leftarrow \mathbb{D} *_a \mathbf{x} + \mathbb{E} *_a \left(\mathrm{Q}^T \mathbf{z}\right)$
$\mathbf{y} \leftarrow \arg\min_{\mathbf{y}} \frac{\mu_2}{2} \|\mathbf{y} - \mathrm{P}\,\mathbf{x}\|_2^2 + u(\mathbf{y}),$	$\longleftrightarrow$	$\mathbf{y} \leftarrow \mathrm{NN}_u\left(\mathrm{P}\,\mathbf{x}\right)$
$\mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \frac{\mu_3}{2} \|\mathbf{z} - \mathrm{Q}\,\mathbf{s}\|_2^2 + r(\mathbf{z}),$	$\longleftrightarrow$	$\mathbf{z} \leftarrow \mathrm{NN}_r\left(\mathrm{Q}\,\mathbf{s}\right)$

Siheng Chen, Yonina C. Eldar, Lingxiao Zhao, "Graph Unrolling Networks: Interpretable Neural Networks for Graph Signal Denoising", arXiv:2006.01301

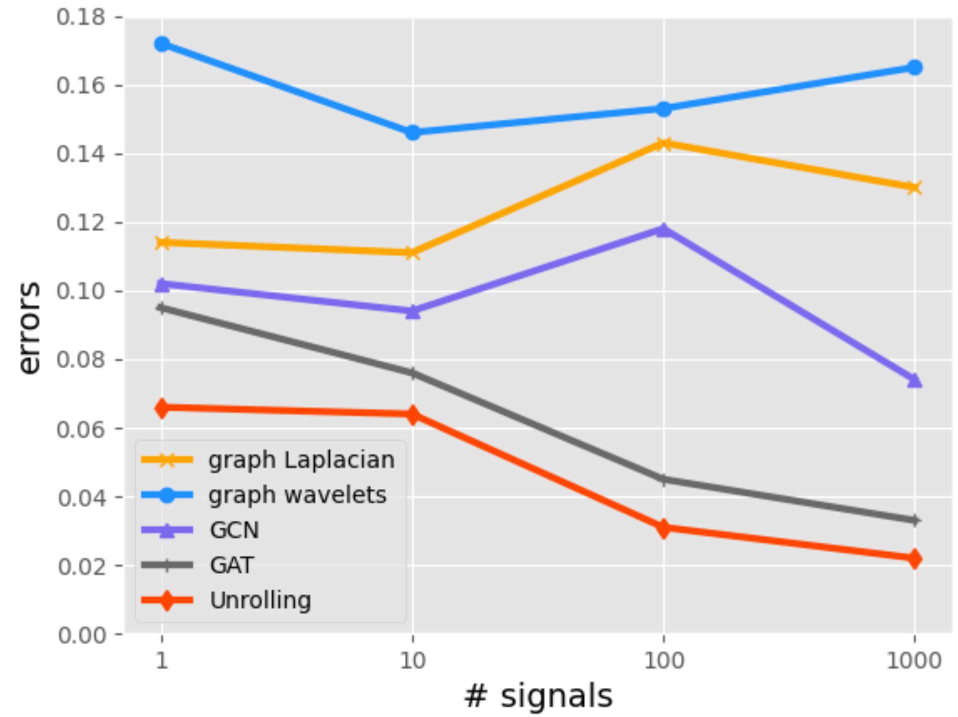
Graph unrolling networks

Denoising of graph-structured data

Smooth graph signals



Piecewise-smooth graph signals



Siheng Chen, Yonina C. Eldar, Lingxiao Zhao, "Graph Unrolling Networks: Interpretable Neural Networks for Graph Signal Denoising", arXiv:2006.01301

Conclusions

Graph-structured data science

- Graph signal processing (GSP)
- Graph neural network (GNN)

Sampling and recovery of graph-structured data

- GSP: Mathematically designed graph sampling operator
- GNN: Trainable graph pooling operator

Bridging GSP and GNN

- Graph scattering transform
- Graph unrolling network

Thank you!