The Scattering Transform on Graphs and Manifolds

Michael Perlmutter

Department of Mathematics
University of California, Los Angeles

Outline

- The Euclidean Scattering Transform
- Graph and Manifold Scattering
- Incorporating Learning
- Application to drug discovery

The (Euclidean) Scattering Transform - S. Mallat (2012)

Overview:

- Model of Convolutional Neural Networks.
- Predefined (wavelet) filters.

Advantages:

- Provable stability and invariance properties.
- Very good numerical results in certain situations.
- Needs less training data.

Example Task: Image Classification

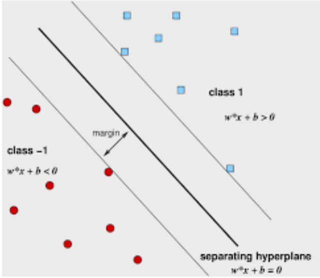
- CNNs are commonly used for image classification
- You have 5000 photos of cats and 5000 photos of dogs.
- Given a new image, how do you decide if its a cat or a dog?

Scattering is an Embedding

- Deep Neural Networks consist of an embedding an a classifier
- An **embedding** (front end) creates a hidden representation of each input in some high-dimensional vector space

$$\mathbf{x} \mapsto h(x) = (h_i(\mathbf{x}))_{i=1}^{H}$$

- The **classifier** (back end) then makes the final prediction



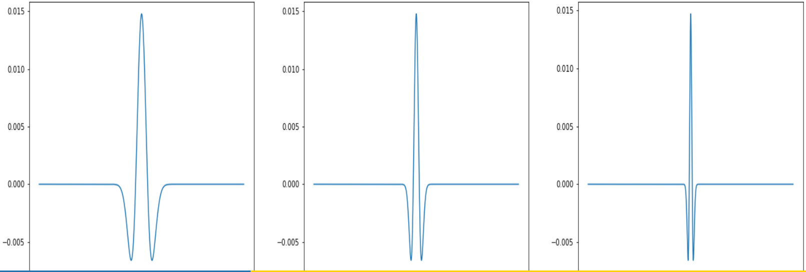
The Wavelet Transform

Definition:

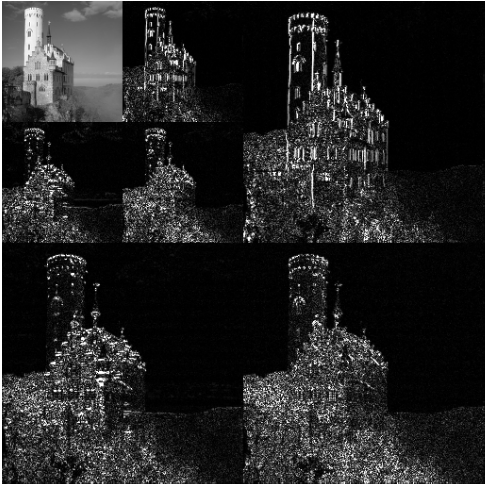
- $W_j f(x) = (\psi_j \star f)(x)$,
- $\psi_j(x) = \frac{1}{2^j}\psi\left(\frac{x}{2^j}\right)$ for some mean zero “mother wavelet” ψ.

Properties

- Collects information at different scales of resolution or frequency bands
- Heuristic: $\text{supp}(\hat{\psi}_j) \approx [2^{-j}a, 2^{-j}b]$



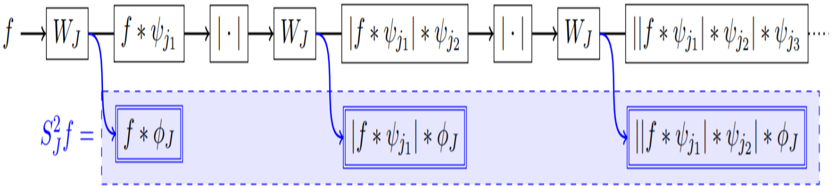
Wavelets Sparsify Natural Images



The Scattering Transform

The Scattering Transform:

- Multilayered cascade of nonlinear measurements.
- Each "layer" uses a wavelet transform W_J and a nonlinearity,
- $U_j f(x) = \sigma((\psi_j \star f)(x)),\ j \leq J, \quad \sigma(x) = M(x) = |x|.$
- $U_{j_1,j_2} f(x) = U_{j_2} U_{j_1} f(x)$
- $U_{j_1,\ldots,j_m} f(x) = U_{j_m} \ldots U_{j_1} f(x)$
- $S_{j_1,\ldots j_m} f(x) = \phi_J \star U_{j_1,\ldots,j_m} f(x), \quad \phi_J(x) = \frac{1}{2^J}\phi\left(\frac{x}{2^J}\right),$ or,
- $\bar{S}_{j_1,\ldots j_m} f = \|U_{j_1,\ldots,j_m} f\|_1.$



Why a Nonlinear Structure?

A good representation should be:

- Stable on $\mathbf{L}^2$
- Invariant to translations (or rotations etc.)
- Sufficiently descriptive

The limits of linearity:

A linear network can be invariant or descriptive, but not both.

- $\widehat{f}(0) = \int_{\mathbb{R}^d} f(x)dx$ is invariant, but throws away all high-frequency information.
- Filters which focus in on high-frequency information are unstable to translations.

The wavelet transform captures high-frequency information, and the modulus pushes this information down to lower frequencies.

Theorem (Mallat 2012)

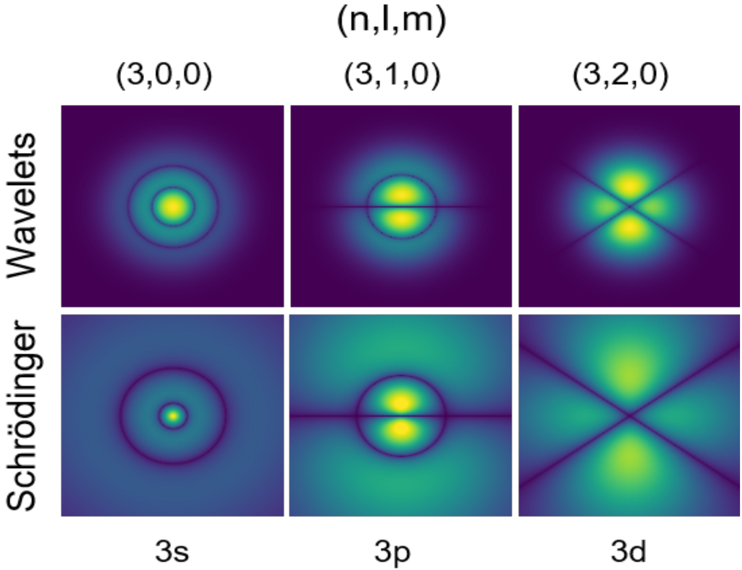
Scattering is stable on $\mathbf{L}^2$ and invariant to translations.

Limited Data Environment - Scattering for Stylometry

Which one is a Van Gogh?

- *Scattering Transform and Sparse Linear Classifiers for Art Authentication* (Leonarduzzi, Liu, and Wang)
- Dataset of 64 real Van Gogh's and 15 fakes.
- Scattering achieves state-of-the-art (96%) accuracy.

Scattering for Quantum Chemistry



Same Power Spectrum, Different Scattering

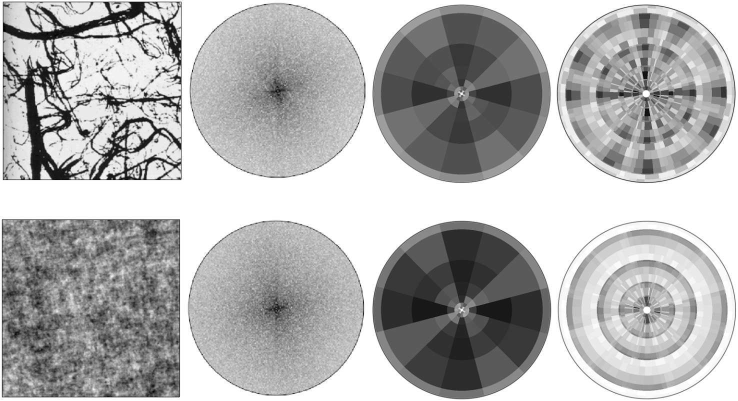
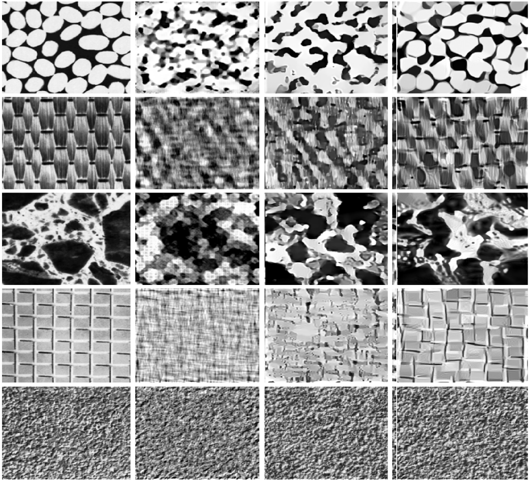


Figure 9: Two different textures having the same Fourier power spectrum. (a) Textures $X(u)$. Top: Brodatz texture. Bottom: Gaussian process. (b) Same estimated power spectrum $\hat{R}X(\omega)$. (c) Nearly same scattering coefficients $S_J[p]X$ for $m = 1$ and 2^J equal to the image width. (d) Different scattering coefficients $S_J[p]X$ for $m = 2$.

Synthesis of random textures



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

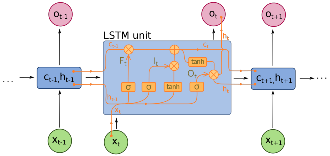
(a): Original texture. (b): texture synthesized with wavelet $\mathbf{l}^2$ norms. (c): synthesized with wavelet $\mathbf{l}^1$ norms. (d): synthesized with scattering coefficients.

Geometric deep learning

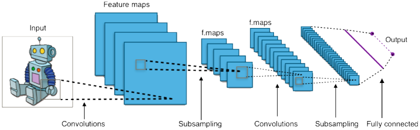
Popular Network Architectures Leverage the Structure of the Data

Examples

Recurrent Neural Nets:



Convolutional Nerual Nets:



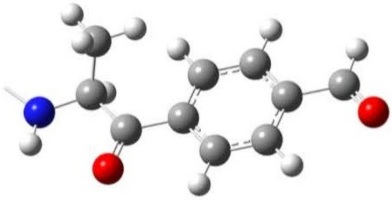
Both Sequences and Images have a Euclidean grid-like structure

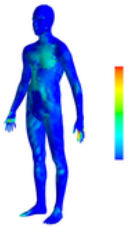
Question: Can we extend these insights to data with a non-Euclidean structure such as graphs and manifolds?

Geometric Scattering

Geometric Wavelets

- Probabilistic Methods: Heat semi-group on a manifold or random walk on a graph.
- Spectral Methods: Eigenfunctions / eigenvectors of an appropriate Laplacian.





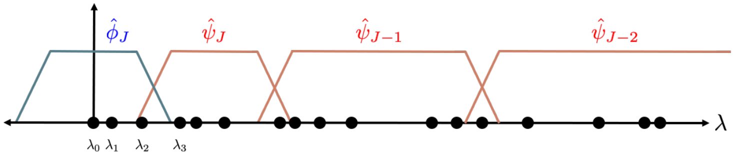
Geometric Wavelets vs GCN style filters

GCN Style Filters

- Take averages over local neighborhoods - promote smoothness
- Low-pass filter

Wavelets

- Detects changes at different scales
  - How is my four-step neighborhood different than my two-step neighborhood?
- Band-pass filter
- Capture long range interactions



Spatial Geometric Wavelets

Definition

Let $\mathcal{X}$ be a graph or a manifold and let $\{P_t\}_{t\geq 0}$ be the heat-semigroup or random walk diffusion. For $0 \leq j \leq J$, let

$$\Psi_j^{(2)} = P_{2^{j+1}} - P_{2^j}, \quad \Phi_J^{(2)} = P_{2^{J+1}},$$

Theorem: P., Gao, Wolf, Hirn

$\mathcal{W}_J^{(2)}$ is a non-expansive frame on a suitable weighted space, i.e.,

$$c\|f\|^2 \leq \sum_j \|\Psi_j^{(2)} f\|^2 + \|\Phi_J^{(2)} f\|^2 \leq \|f\|^2.$$

Remark

Subsequent work with Tong et. al showed that dyadic scales are unnecessary and the same result holds with any sequence of increasing scales. Moreover, one may learn the scales through data.

Spectral Convolution

Generalized Fourier Multiplication

Let L be the Laplace-Beltrami operator or graph Laplacian with eigenbasis $\{\varphi_k\}$, $L\varphi_k = \lambda_k\varphi_k$. A spectral convolution operator has the form

$$Tf = \sum_{k=0}^{\infty} h_k \langle f, \varphi_k \rangle \varphi_k.$$

This notion of convolution is used in many popular Graph Neural Networks such as ChebNet (Defferrard et al. 2016)

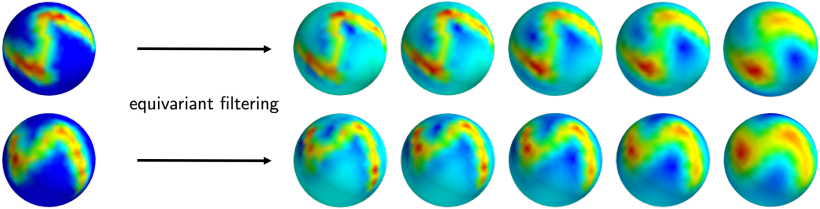
Spectral filters

T is called a spectral filter if $h_k = h(\lambda_k)$

Spectral Representation of the Heat Semigroup

$$P_t f(x) = \sum_{k=0}^{\infty} g(\lambda_k)^t \langle f, \varphi_k \rangle \varphi_k, \quad g(\lambda) = e^{-\lambda}$$

Equivariant Filters



Theorem: (P., Gao, W., Hirn)

Spectral filters commute with isometries.

Spectral Wavelets

Definition

$$\mathcal{W}_J^{(1)} f(x) = \{\Psi_j^{(1)} f(x), \Phi_J^{(1)} f(x)\}_{0 \leq j \leq J},$$

where $\Phi_J^{(1)} = P_{2^J}$, $g(\lambda) = e^{-\lambda}$ and

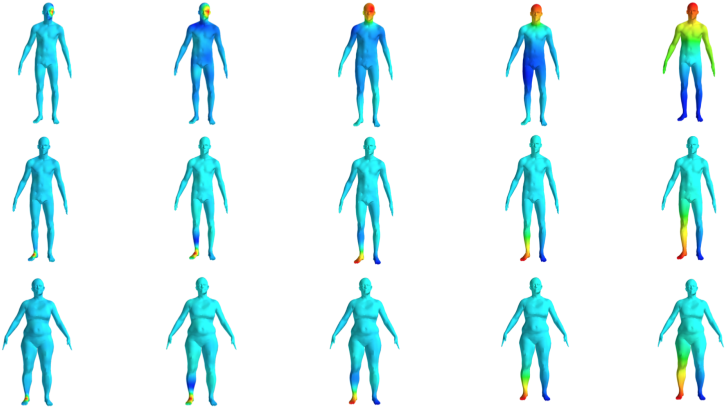
$$\Psi_j^{(1)} f = (P_{2^{j+1}} - P_{2^j})^{1/2} f = \sum_{k=0}^{\infty} [g(\lambda_k)^{2^{j+1}} - g(\lambda_k)^{2^j}]^{1/2} \langle f, \varphi_k \rangle \varphi_k.$$

Theorem: P., Gao, Wolf, Hirn

$\mathcal{W}_J^{(1)}$ is an isometry, i.e.,

$$\sum_j \|\Psi_j^{(1)} f\|^2 + \|\Phi_J^{(1)} f\|^2 = \|f\|^2.$$

Wavelets on the Faust Dataset



Theoretical Guarantees Manifold Scattering

Theorem (P. Gao, Wolf, Hirn)

$$\|Sf_1 - Sf_2\| \leq \|f_1 - f_2\|, \quad \forall f_1, f_2 \in \mathbf{L}^2(\mathcal{M}).$$

Theorem (P. Gao, Wolf, Hirn)

Let ζ be an isometry, $V_\zeta f(x) = f(\zeta^{-1}(x))$.

$$\|Sf - SV_\zeta f\| = \mathcal{O}\left(2^{-dJ}\right) \quad \forall f \in \mathbf{L}^2(\mathcal{M}) \ .$$

Theorem (P. Gao, Wolf, Hirn)

Let ζ be an diffeomorphism, and assume f is bandlimited (finitely many non-zero Fourier coefficients). Then

$$\|Sf - SV_\zeta f\| = \mathcal{O}\left(2^{-dJ}\right) + \mathcal{O}\left(\lambda_{\max}^d d(\zeta, Isom)\right).$$

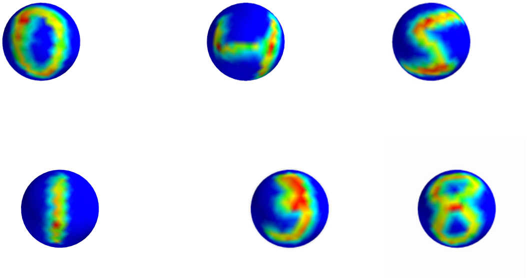
Theorem (P., Gao, Wolf, Hirn)

Similar results hold for graph scattering.

Manifold Scattering Results

Example (Spherical MNIST)

MNIST digits projected on the sphere:

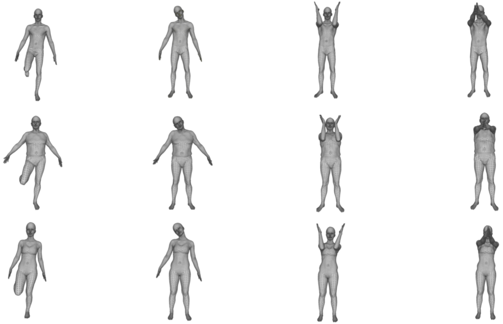


- Single manifold, multiple signals
- 95% classification accuracy from scattering features

Manifold Scattering Results

Example (FAUST dataset)

Ten people in ten different poses:

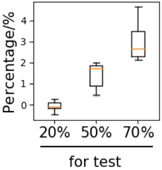


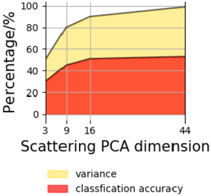
- Mesh grids & Shot features (Tombari et al., 2010; Prakya et al.,2015)
- Accuracy: 81% person recognition, 95% pose classification

Graph Scattering Results

	COLLAB	IMDB-B	IMDB-M	REDDIT-B	REDDIT-5K	REDDIT-12K	
WL	77.82 ± 1.45	71.60 ± 5.16	N/A	78.52 ± 2.01	50.77 ± 2.02	34.57 ± 1.32	Graph kernel
Graphlet	73.42 ± 2.43	65.40 ± 5.95	N/A	77.26 ± 2.34	39.75 ± 1.36	25.98 ± 1.29	
WL-OA	80.70 ± 0.10	N/A	N/A	89.30 ± 0.30	N/A	N/A	
DGK	73.00 ± 0.20	66.90 ± 0.50	44.50 ± 0.50	78.00 ± 0.30	41.20 ± 0.10	32.20 ± 0.10	
DGCNN	73.76 ± 0.49	70.03 ± 0.86	47.83 ± 0.85	N/A	48.70 ± 4.54	N/A	Deep learning
2D CNN	71.33 ± 1.96	70.40 ± 3.85	N/A	89.12 ± 1.70	52.21 ± 2.44	48.13 ± 1.47	
PSCN ($k = 10$)	72.60 ± 2.15	71.00 ± 2.29	45.23 ± 2.84	86.30 ± 1.58	49.10 ± 0.70	41.32 ± 0.42	
GCAPS-CNN	77.71 ± 2.51	71.69 ± 3.40	48.50 ± 4.10	87.61 ± 2.51	50.10 ± 1.72	N/A	
S2S-P2P-NN	81.75 ± 0.80	73.80 ± 0.70	51.19 ± 0.50	86.50 ± 0.80	52.28 ± 0.50	42.47 ± 0.10	
GIN-0 (MLP-SUM)	80.20 ± 1.90	75.10 ± 5.10	52.30 ± 2.80	92.40 ± 2.50	57.50 ± 1.50	N/A	
GS-SVM	79.94 ± 1.61	71.20 ± 3.25	48.73 ± 2.32	89.65 ± 1.94	53.33 ± 1.37	45.23 ± 1.25	

Impact of training size & feature-space dimensionality [1]:





[1]Demonstrated on ENZYMES dataset (Borgwardt et al., Bioinformatics 2005)

Semi-Supervised Node Classification

Setup

- Entire Graph Structure is known (all Vertices and Edges)
- Node feature matrix $X = X^0 = (\mathbf{x}_1, \ldots, \mathbf{x}_F)$ is known for all nodes
- Labels are known for some nodes ($\leq$5%)
- Goal: Predict the labels of the remaining nodes.

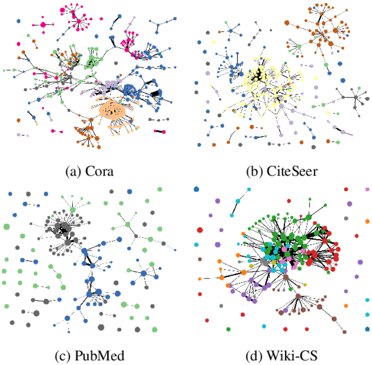


Figure: Visualizations of Common Data sets

Graph Convolutional Network (Kipf and Welling)

Layer-Wise Update Rule

- Sequentially transform node features via layerwise updates

$$X^{t+1} = \sigma(\widehat{A} X^t \Theta)$$

- $\Theta \in \mathbb{R}^{F_t \times F_{t+1}}$ is a trainable weight matrix.
- $\widehat{A}$ is a local averaging operator.
- Promotes smoothness, i.e. similarity amongst neighbors
- Θ is learned but $\widehat{A}$ is designed (as a low-degree polynomial of the graph Laplacian).

Low-pass filter

- Multiplying by $\widehat{A}$ leaves bottom eigenvector unchanged.
- All other frequencies are depressed.
- Repeated applications increasingly depress high-frequencies.
- "Deep" Graph Neural Nets typically use 2 layers.

Discriminative Power

When can a network tell two nodes apart?

- Necessary condition: The network learns different representations of the two nodes
- Lots of work on the analogous problem for graph classification
  - GCN $\lesssim$ Weisfeiler-Lehman Kernel
- Little work for node classification
- Do GCNs rely on informative features? Or can they learn from the geometry of the graph?

Theorem (Wenkel, Min, Hirn, P., and Wolf (2022))

- *There are situations where GCN provably not discriminate two nodes if their local neighborhoods have the same structure*
- *Graph Scattering can discriminate some of those nodes*
- *Thus GCN-Scattering Hybrid networks have more discriminative power than pure GCN networks.*

Limitations of GCN

Intrinsic Node Features

A node feature $\mathbf{x}$ is called intrinsic is called K-intrinsic if $\mathbf{x}(v) = \mathbf{x}(v')$ whenever the K-step neighborhood of v is isomorphic to the K-step neighborhood of v'.

Examples:

- $\mathbf{d}(v) = \text{degree}(v)$ is 1-intrinsic
- $\mathbf{t}^{(K)}(v)$ =Number of triangles in K-steb neighborhood of v is K-intrinsic

Theorem (Wenkel, Min, Hirn, P., and Wolf (2022))

If the $K + L$-step neighborhoods of v and v' are isomorphic and all node features are K-intrinsic, then an L-layer GCN can't discriminate v and v'.

Scattering Can Help

Structural differences

- Suppose the $K+L$-step neighborhood of v is isomorphic to the $K+L$-step neighborhood of v' under a mapping v'
- let X be a K-intrinsic feature matrix and let u be in the $K+L$ step-neighborhood of v.
- We say a structural difference manifests at u if $X[u] \neq X[\phi(u)]$

Theorem (Wenkel, Min, Hirn, P., and Wolf (2022))

If there is a structural difference, in the $K+L$ neighborhood of v, then (except in certain pathological cases) scattering can discriminate v and v'.

INCORPORATING LEARNING

- Scattering helps us understand GNNs and a theoretical level
- Let's use this understanding to build (trained) GNNs incorporating the principals of scattering

Scattering GCN Hybrid

Scattering Channels

Layer-wise update rule:

$$X_{sct}^{\ell} := \sigma\left((P^{2^{J+1}} - P^{2^{J}})X^{\ell-1}\Theta\right).$$

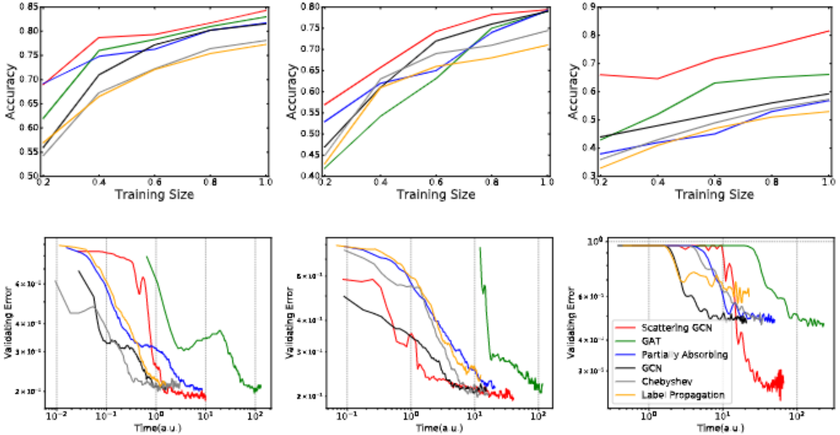
Hybrid Network

- Wenkel, Min, Hirn, P., and Wolf (2022) use both GCN channels and Scattering channels of each layer.
- GCN channels focus on low-frequency information.
- Scattering Channels retain high-frequency information.
- Can use an attention mechanism to balance channel ratios.

Hybrid Network Results

Model	Citeseer	Cora	Pubmed	DBLP
Sc-GCN (ours)	**71.7**	**84.2**	**79.4**	**81.5**
GAT [10]	**72.5**	**83.0**	79.0	66.1
Partially absorbing [9]	71.2	81.7	**79.2**	56.9
GCN [5]	70.3	81.5	79.0	59.3
Chebyshev [28]	69.8	78.1	74.4	57.3
Label Propagation [38]	58.2	77.3	71.0	53.0
Graph scattering [14]	67.5	81.9	69.8	**69.4**
Node features (SVM)	61.1	58.0	49.9	48.2

Hybrid Network Results



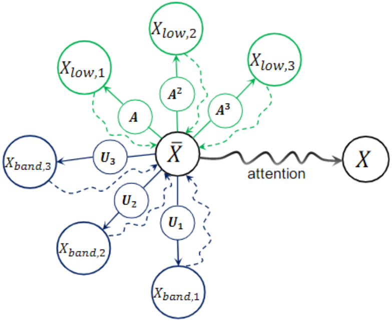
(b) Cora (c) Pubmed (d) DBLP

Scattering Attention Network

Attention Mechanism

$$\mathbf{X}^\ell = C^{-1}\tilde{\sigma}\left(\sum_{j=1}^{C_{\text{low}}} \alpha^\ell_{\text{low},j} \odot \bar{\mathbf{X}}^\ell_{\text{low},j} + \sum_{j=1}^{C_{\text{band}}} \alpha^\ell_{\text{band},j} \odot \bar{\mathbf{X}}^\ell_{\text{band},j}\right)$$

$$C = C_{\text{low}} + C_{\text{high}}, \quad \alpha \odot \mathbf{X} = \text{diag}(\alpha)\mathbf{X}$$



Attention Network Results

Dataset	Classes	Nodes	Edges	Homophily	GCN	GAT	Sc-GCN	GSAN
Texas	5	183	295	0.11	59.5	58.4	60.3	60.5
Chameleon	5	2,277	31,421	0.23	28.2	42.9	51.2	61.2
CoraFull	70	19,793	63,421	0.57	62.2	51.9	62.5	64.5
Wiki-CS	10	11,701	216,123	0.65	77.2	77.7	78.1	78.6
Citeseer	6	3,327	4,676	0.74	70.3	72.5	71.7	71.3
Pubmed	3	19,717	44,327	0.80	79.0	79.0	79.4	79.8
Cora	7	2,708	5,276	0.81	81.5	83.0	84.2	84.0
DBLP	4	17,716	52,867	0.83	59.3	66.1	81.5	84.3

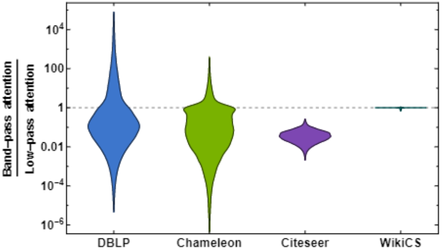
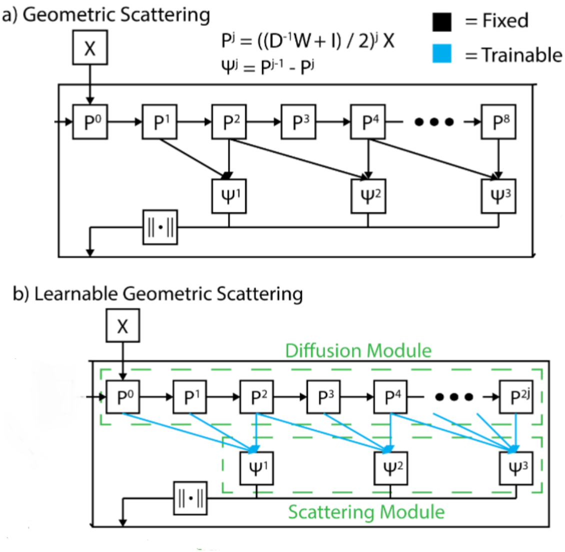


Fig. 6. Distribution of attention ratios per node between band-pass (scattering) and low-pass (GCN) channels across all heads for DBLP, Chameleon, Citeseer, and WikiCS.

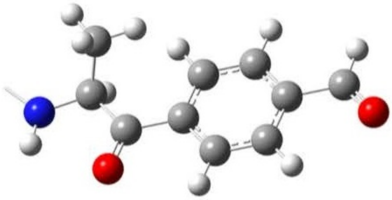
LEGS - Learning the Scales



Graph Generation

Problem:

- Given a dataset of graphs, can you generate a new graph that looks like it was a member of the original dataset
- Motivating Application - Drug Development



Encoding robust representation for graph generation (Zou and Lerman 2019)

- Encoder E = Graph Scattering Transform
- Decoder D = Fully Connected Network
- $D \circ E = Id$
- Generate new graphs by adding noise in latent space

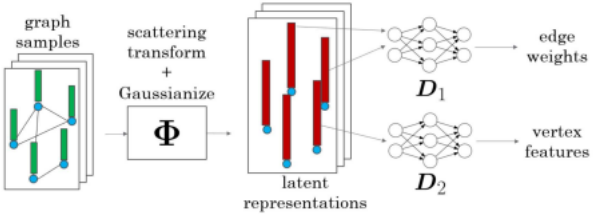


Figure: Scattering Encoder-Decoder Network

Molecular Graph Generation via Geometric Scattering (GRASSY) - Bhaskar, Grady, P., Krishnaswamy

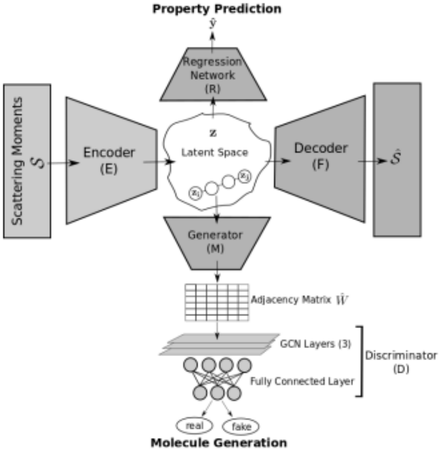
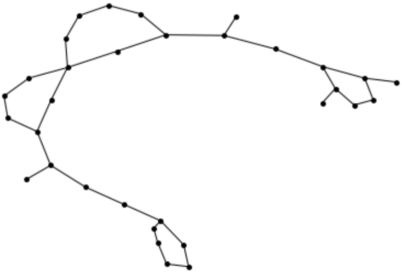


Figure: GRaph Scattering SYnthesis network

Results

Table 2. Molecule generation performance on ZINC dataset

Measure	ZINC Tranche	Models				
		GRASSY	GraphAF	MolGAN ($\lambda = 0$)	MolGAN ($\lambda = 1$)	MegaMolBART*
Validity	BBAB	**1.0**	**1.0**	0.93	0.86	0.88
	FBAB	**1.0**	**1.0**	0.90	0.71	0.96
	JBCD	**1.0**	**1.0**	0.84	0.63	0.99
Uniqueness	BBAB	0.86	**0.98**	0.07	0.11	0.43
	FBAB	0.91	**1.0**	0.04	0.03	0.41
	JBCD	0.87	**1.0**	0.05	0.04	0.37
Novelty	BBAB	**1.0**	**1.0**	**1.0**	**1.0**	0.22
	FBAB	**1.0**	**1.0**	**1.0**	**1.0**	0.15
	JBCD	**1.0**	**1.0**	**1.0**	**1.0**	0.19



Conclusion

- The Euclidean scattering transform is a model of CNNs.
  - Provable Stability / Invariance Guarantees
  - Designed filters - useful for low-data environments
  - Can be used to synthesize textures
- Geometric Versions for Graphs and Manifolds
  - Similar theoretical guarantees to the Euclidean scattering transform
  - Wavelets can be constructed either spatially or spectrally
  - Can be incorporated in hybrid Scattering - GCN networks
- The graph scattering transform can be used to synthesize molecules as part of the GRASSY framework

THANK YOU!