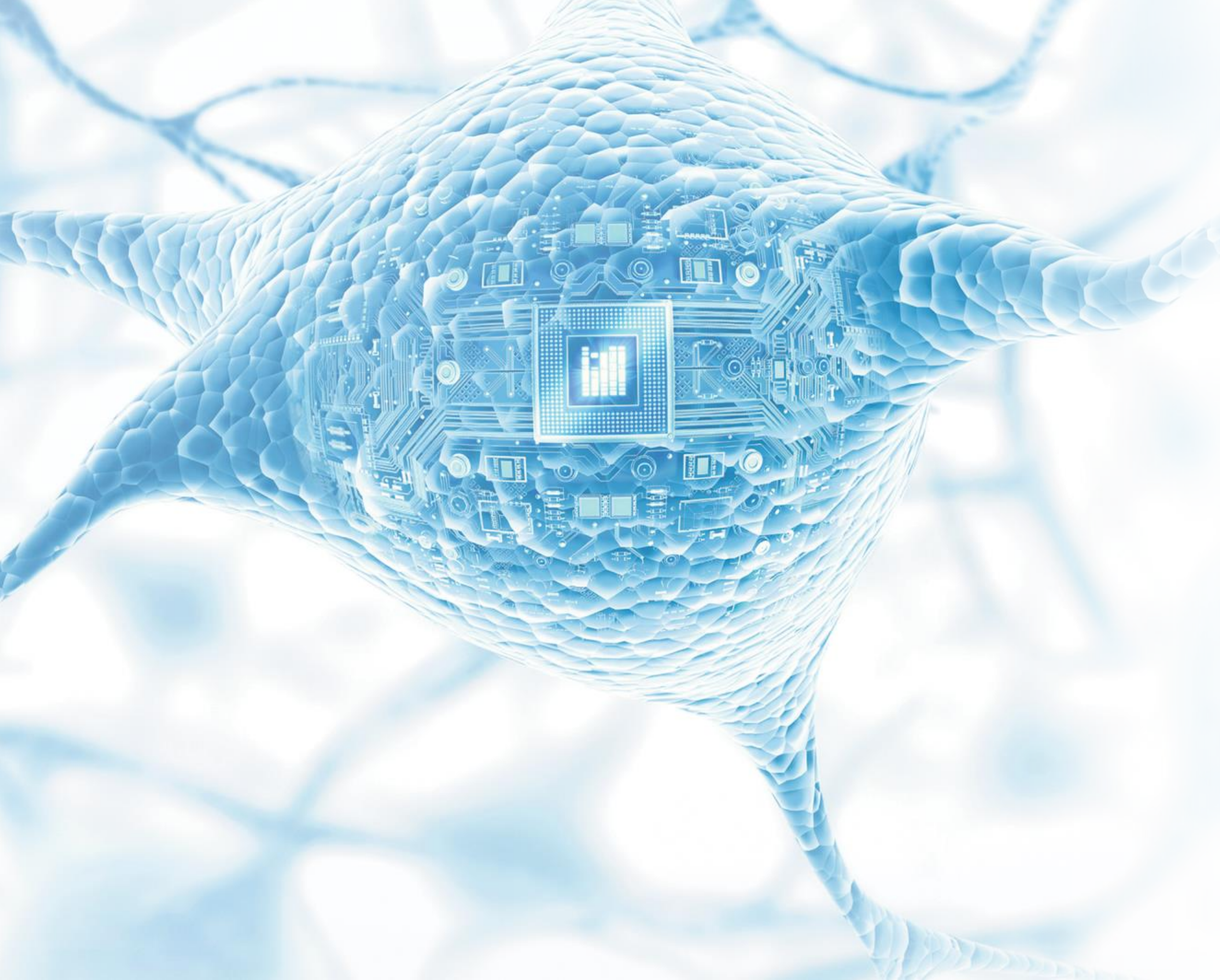
LanczosNet: Multi-Scale Deep Graph Convolutional Networks

Renjie Liao, Zhizhen Zhao, Raquel Urtasun, Richard S. Zemel

Published as a conference paper at ICLR 2019

Presented By Juan Carrillo

March 30, 2021



UNIVERSITY OF TORONTO

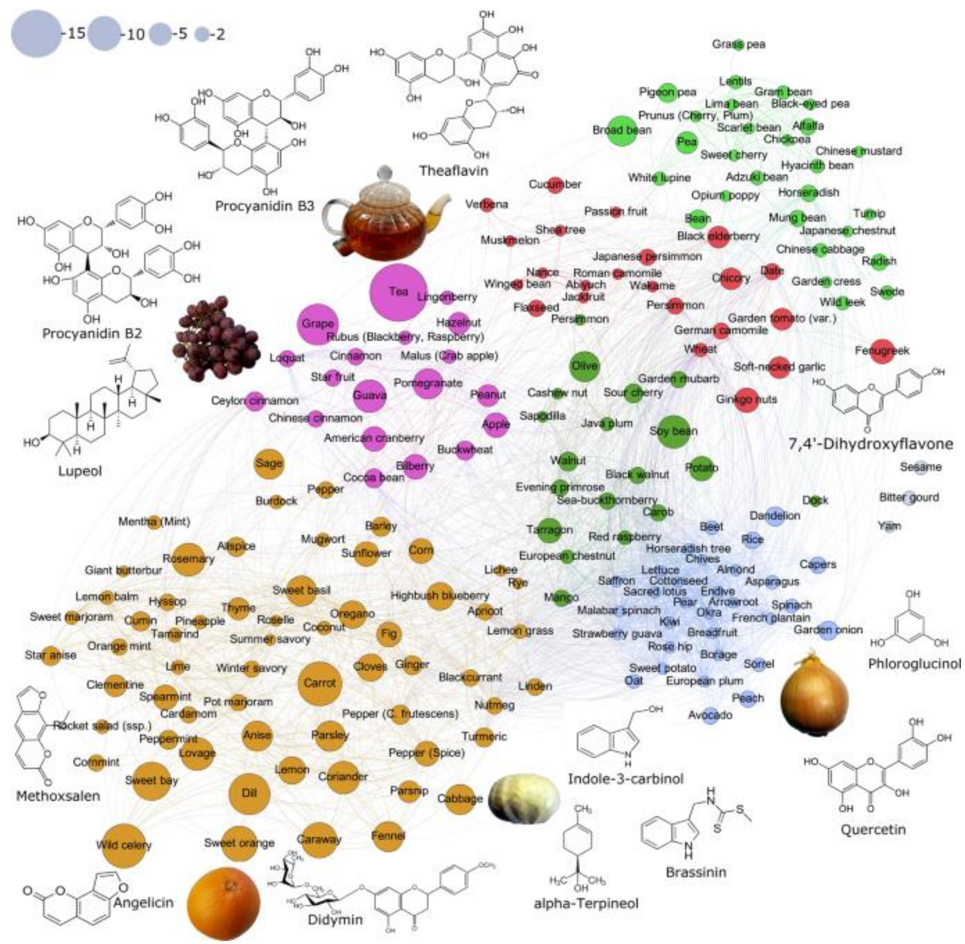
Index

- Motivation
- Problem setting
- Contributions
- Previous approaches
- Background
- Methods
- Experiments
- Conclusions
- Limitations

Motivation

- Learning representations in Graph data
  - Graph level
  - Node level
  - Multi-scale
  - Others…

- Graph are rich data structures
  - Bioinformatics
  - Transportation networks
  - Social networks
  - Point clouds
  - 3D Meshes
  - Knowledge graphs
  - Recommendation engines
  - Particle physics

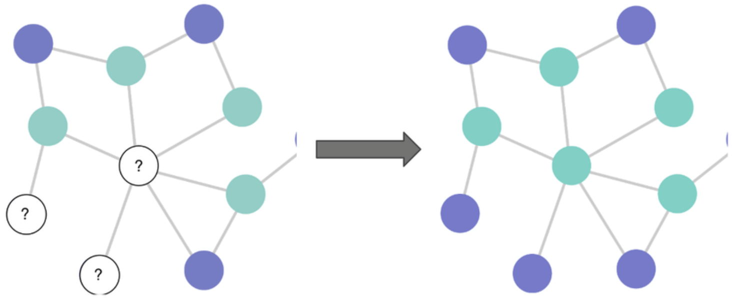


Veselkov et al. (2019)

Problem setting

Node classification

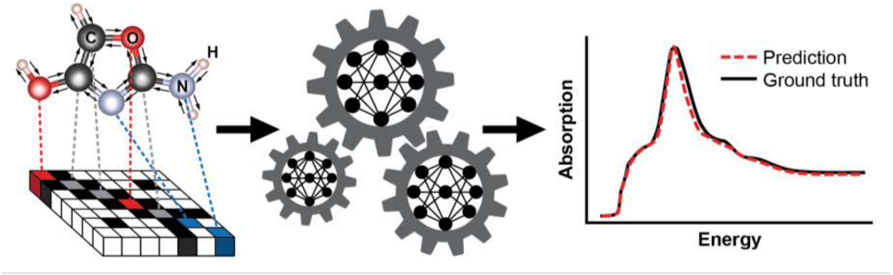
Given a graph, predict the category of unlabeled nodes



Mishra et al. (2020)

Graph regression

Given a graph, predict a quantitative attribute of it



Carbone et al. (2020)

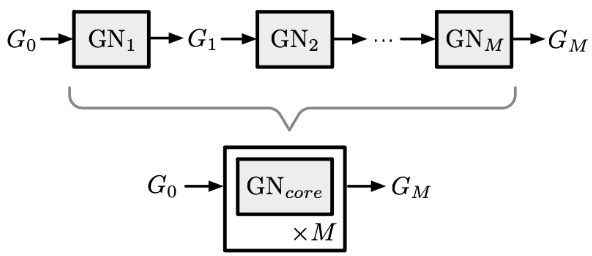
UNIVERSITY OF TORONTO

Contributions

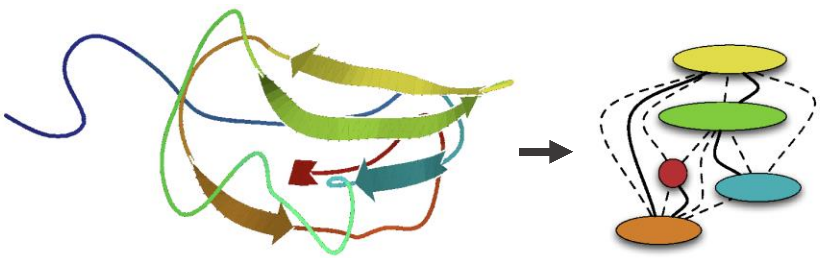
- LanczosNet uses the Lanczos algorithm to efficiently extract useful features from graphs
- The architecture allows multi-scale analysis in large graphs
- Achieves SOTA performance in two challenging benchmarks

Previous approaches

Supervised/semi-supervised learning

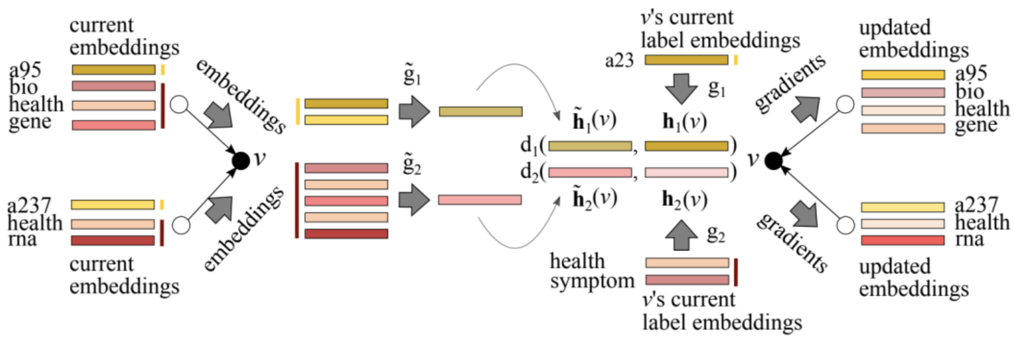


Battaglia et al. (2018)



Vishwanathan et al. (2020)

Unsupervised learning



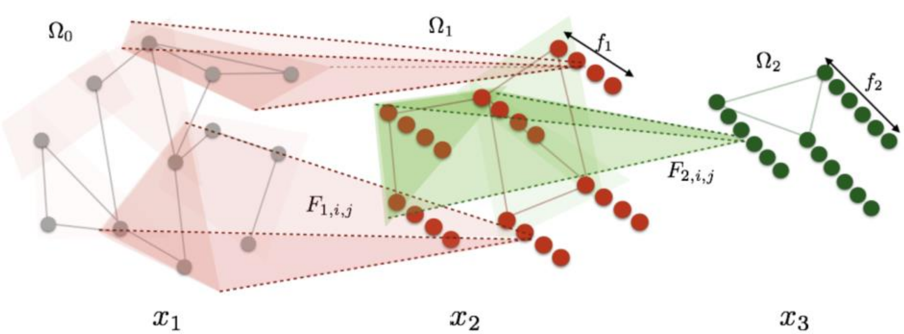
García-Durán et al. (2017)

Previous approaches

Graph Convolution Based Models

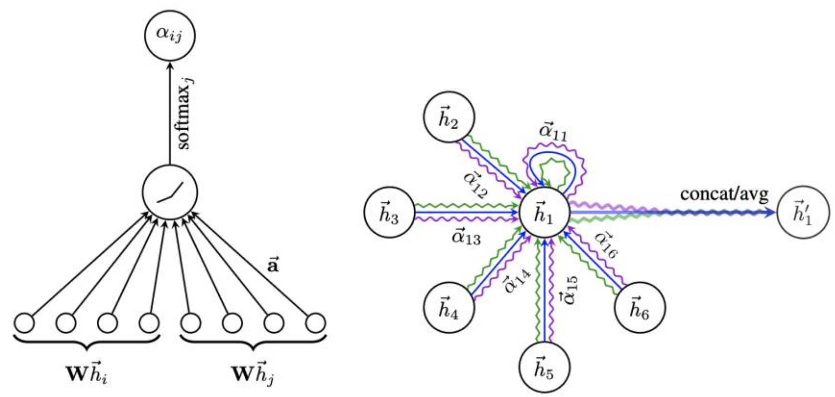
- Origins in graph signal processing (GSP)
- Supported by spectral graph theory

Spectral Networks



Bruna et al. (2014)

Graph Attention Networks



Velickovic et al. (2018)

UNIVERSITY OF TORONTO

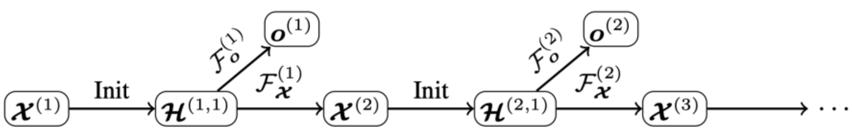
Previous approaches

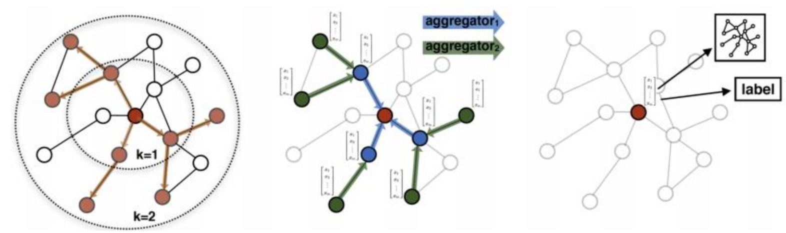
Recurrent Neural Networks based Models

- Origins in recurrent neural networks (RNNs)
- Graph neural networks (GNNs)

GraphSAGE

Gated Graph Sequence Neural Networks





Li et al. (2017)

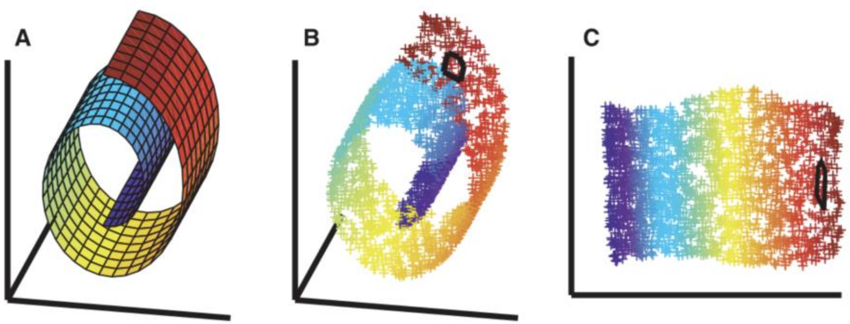
Hamilton et al. (2017)

Previous approaches

Graph based manifold learning

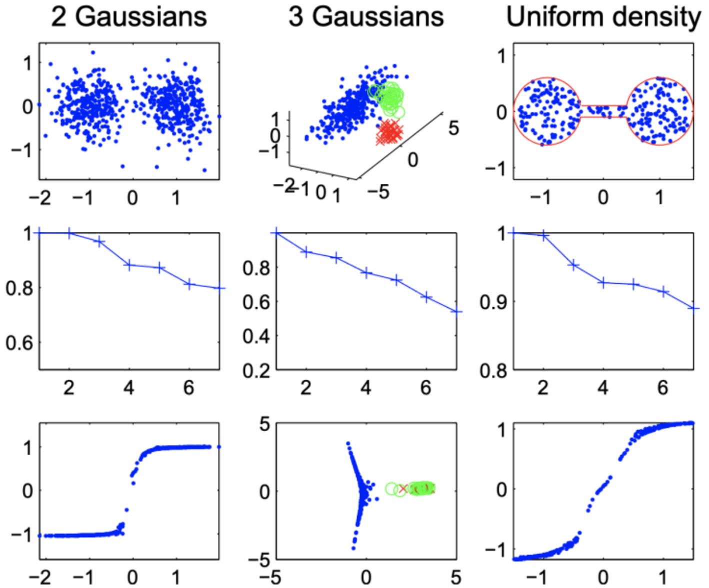
- High to low dimensional representations
- Reduces graph complexity

Locally Linear Embedding (LLE)



Roweis & Saul (2000)

Diffusion maps

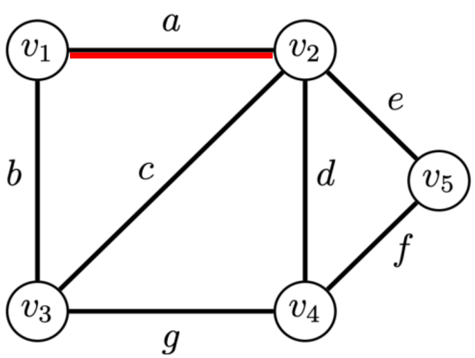


Nadler et al. (2006)

Background

Graph notation and definitions

- Undirected graph



$$G = (V, E)$$

$$V = \{v_1, \dots, v_n\}$$

$$E = \{\{v_i, v_j\}, \dots, \{v_m, v_n\}\}$$

$$with\ v_i, v_j \in V\ and\ v_i \neq v_j$$

- Adjacency matrix

$$a_{i\,j} = \begin{cases} 1 & \text{if there is some edge } \{v_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

$$A = \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ \boxed{1} & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Gallier (2020)

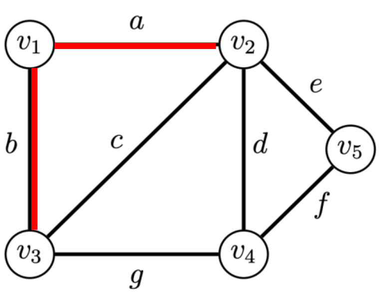
Background

Graph notation and definitions

- Degree matrix

$$d(v) = |\{u \in V \mid (v,u) \in E \text{ or } (u,v) \in E\}|$$

$$D(G) = \mathrm{diag}(d_1, \ldots, d_m)$$



$$\begin{pmatrix} \boxed{2} & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

- Laplacian matrix

$$L = D - A$$

$$\begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$L(G) = D(G) - W,$$

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = \boxed{I - D^{-1/2} W D^{-1/2}}$$

Gallier (2020)

Background

Additional background…

- Graph Fourier Transform

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$S = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$S = U \Lambda U^\top$$

$$\Lambda_{i,i} = \lambda_i \text{ and } 1 \geq \lambda_1 \geq \cdots \geq \lambda_N \geq -1$$

$$\boxed{Y = U^\top X}$$

$$X \in \mathbb{R}^{N \times F}$$

- Localized Polynomial Filter

$$g_w(\Lambda) = \sum_{t=0}^{\tau-1} w_t \Lambda^t$$

$$\boldsymbol{w} = [w_0, w_1, \ldots, w_{\tau-1}] \in \mathbb{R}^{\tau \times 1}$$

$$\boxed{Y = \sum_{t=0}^{\tau-1} g_t(S, \ldots, S^t, X) W_t}$$

$$Y \in \mathbb{R}^{N \times O} \qquad W_t \in \mathbb{R}^{F \times O}$$

Liao et al. (2019)

Methods

Lanczos Algorithm

Algorithm 1 : Lanczos Algorithm

Input: S, x, K, ϵ
Initialization: $\beta_0 = 0$, $q_0 = 0$, and $q_1 = x/\|x\|$
For $j = 1, 2, \ldots, K$:
    $z = Sq_j$
    $\gamma_j = q_j^\top z$
    $z = z - \gamma_j q_j - \beta_{j-1} q_{j-1}$
    $\beta_j = \|z\|_2$
    If $\beta_j < \epsilon$, quit
    $q_{j+1} = z/\beta_j$

$Q = [q_1, q_2, \cdots, q_K]$
Construct T following Eq. (2)
Eigen decomposition $T = BRB^\top$
Return $V = QB$ and R.

Goal: Obtain an approximation of

- Orthogonal matrix Q
- Symmetric tridiagonal matrix T
- Such that $Q^\top SQ = T$

$$T = \begin{bmatrix} \gamma_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{N-1} \\ & & \beta_{N-1} & \gamma_N \end{bmatrix}$$

Liao et al. (2019)

UNIVERSITY OF TORONTO

Methods

LanczosNet

- Localized Polynomial Filter

$$X_{:,i} \in \mathbb{R}^{N \times 1}$$

$$\tilde{Q} \text{ of } \mathcal{K}_K(S, X_{:,i}) \text{ and } \tilde{T}$$

$$Y_j = \tilde{Q}\boldsymbol{w}_{i,j}$$

$$\boldsymbol{w}_{i,j} \in \mathbb{R}^{K \times 1} \qquad \tilde{Q} \in \mathbb{R}^{N \times K}$$

- Spectral Filter

$$S \approx QTQ^\top \qquad Q \in \mathbb{R}^{N \times K}$$

$$T = BRB^\top \qquad B \in \mathbb{R}^{K \times K}$$

$$S \approx VRV^\top \qquad V = QB$$

$$S^t \approx VR^tV^\top$$

$$\begin{aligned} Y_j &= [X_i, SX_i, \ldots, S^{K-1}X_i]\boldsymbol{w}_{i,j} \\ &\approx [X_i, VRV^\top X_i, \ldots, VR^{K-1}V^\top X_i]\boldsymbol{w}_{i,j} \end{aligned}$$

Liao et al. (2019)

Methods

LanczosNet

- Learning the Spectral Filter

$$Y_j \approx [X_i, VRV^\top X_i, \ldots, VR^{K-1}V^\top X_i]\boldsymbol{w}_{i,j}$$

$$\{(r_i, v_i) | i = 1, \ldots, K\}$$

$$\hat{L}_i = \sum_{k=1}^{K} f_i(r_k^1, r_k^2, \cdots, r_k^{K-1}) v_k v_k^\top$$

$$Y_j = [X_i, \hat{L}_1 X_i, \ldots, \hat{L}_{K-1} X_i]\boldsymbol{w}_{i,j}$$

- Multi-scale Graph Convolution

$$Y = \left[L^{\mathcal{S}_1} X, \ldots, L^{\mathcal{S}_M} X, \hat{L}_1(\mathcal{I}) X, \ldots, \hat{L}_N(\mathcal{I}) X \right] W$$

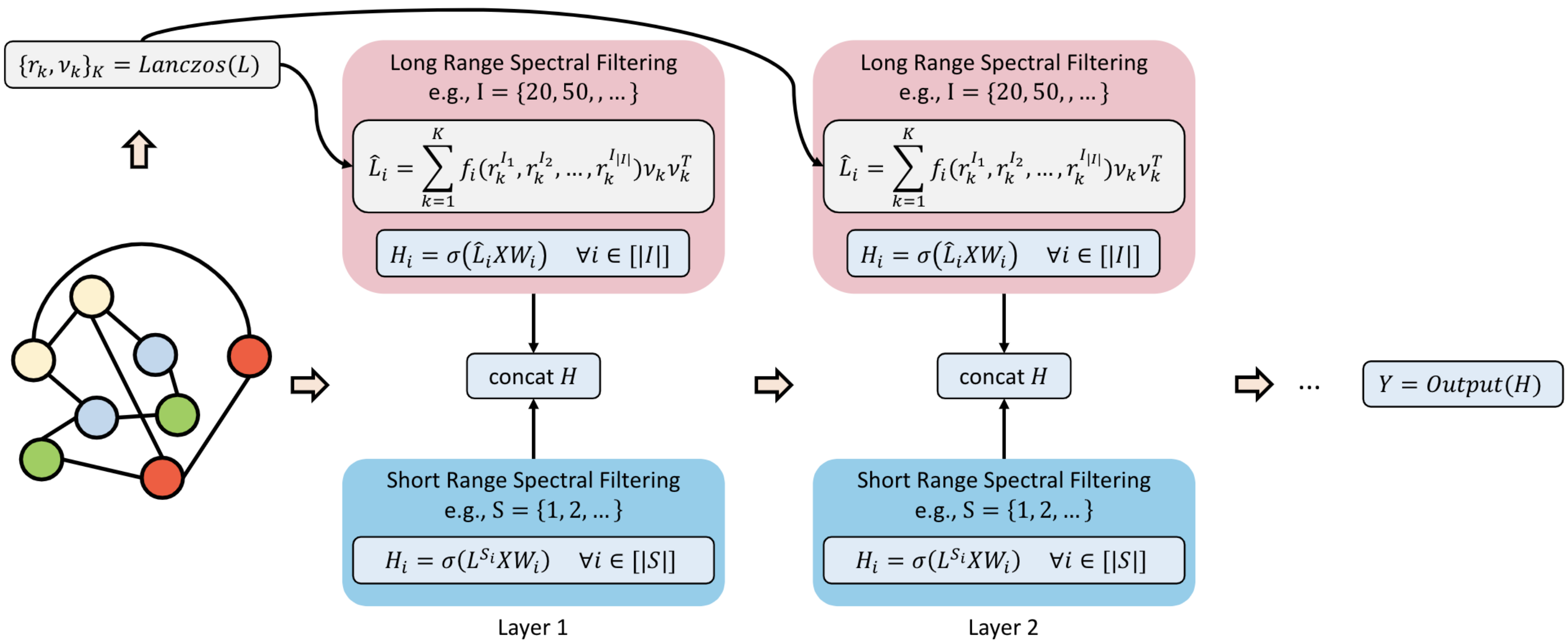
$$W \in \mathbb{R}^{(M+E)D \times O} \qquad \mathcal{S} = \{0, 1, \ldots, 5\} \qquad \mathcal{I} = \{10, 20, \ldots, 50\}$$

$$\hat{L}_i(\mathcal{I}) = \sum_{k=1}^{K} f_i(r_k^{\mathcal{I}_1}, r_k^{\mathcal{I}_2}, \cdots, r_k^{\mathcal{I}_{|\mathcal{I}|}}) v_k v_k^\top$$

Liao et al. (2019)

Methods

LanczosNet



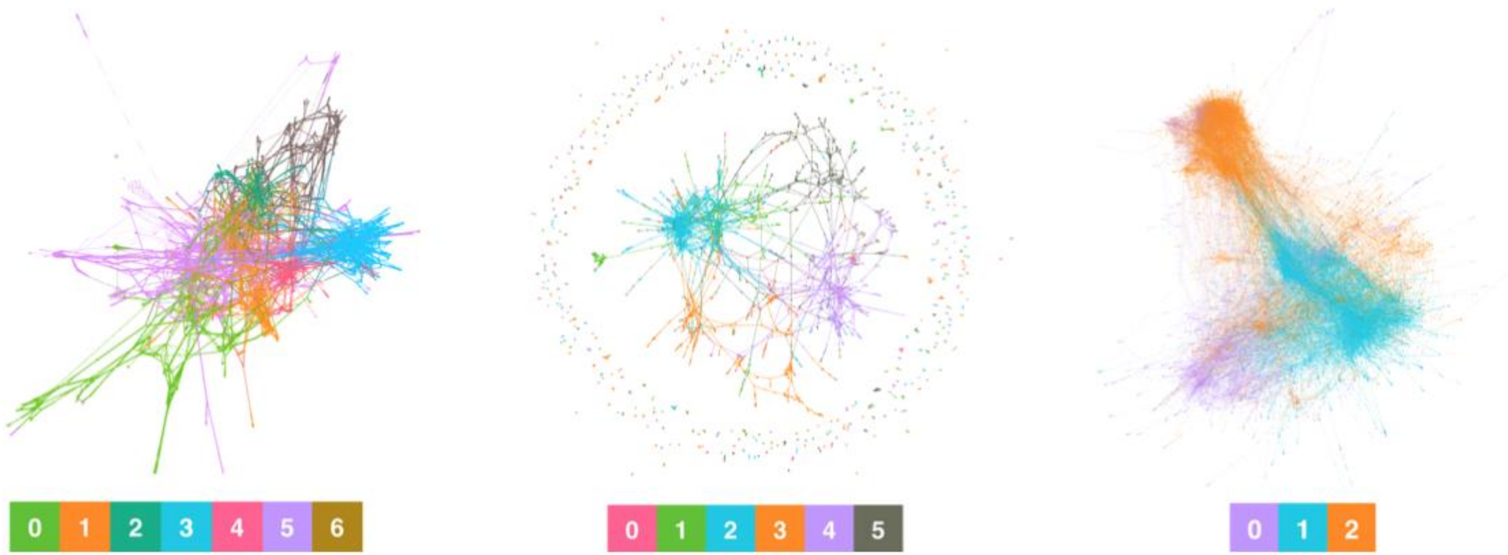
Liao et al. (2019)

UNIVERSITY OF TORONTO

Experiments

Citation networks

Goal: Predict class of unlabeled nodes (documents) in citation networks



(a) CORA-ML (b) CiteSeer (c) PubMed

Cheung et al. (2020)

Experiments

Citation networks

Goal: Predict class of unlabeled nodes (documents) in citation networks

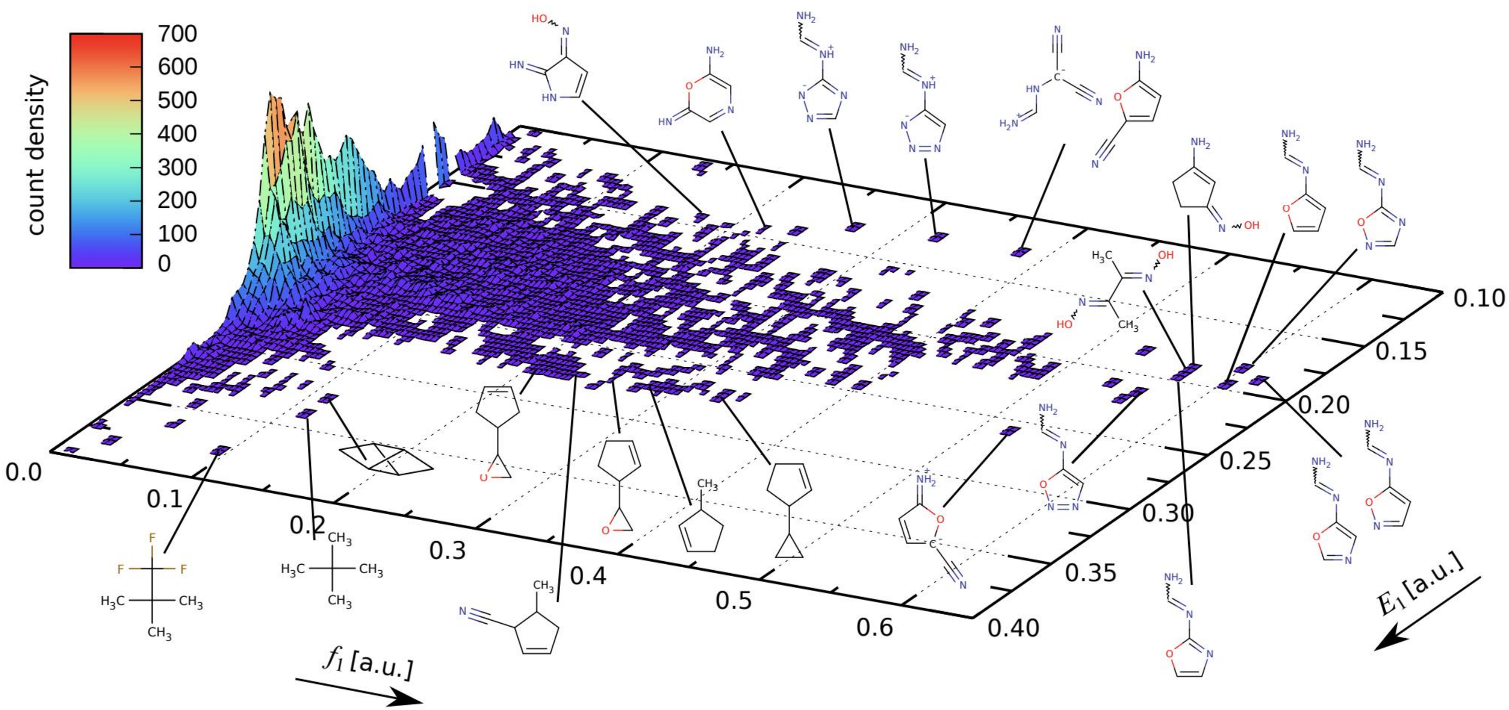
Cora	GCN-FP	GGNN	DCNN	ChebyNet	GCN	MPNN	GraphSAGE	GAT	LNet	AdaLNet
Public	74.6 ± 0.7	77.6 ± 1.7	79.7 ± 0.8	78.0 ± 1.2	80.5 ± 0.8	78.0 ± 1.1	74.5 ± 0.8	**82.6 ± 0.7**	79.5 ± 1.8	80.4 ± 1.1
3%	71.7 ± 2.4	73.1 ± 2.3	76.7 ± 2.5	62.1 ± 6.7	74.0 ± 2.8	72.0 ± 4.6	64.2 ± 4.0	56.8 ± 7.9	76.3 ± 2.3	**77.7 ± 2.4**
1%	59.6 ± 6.5	60.5 ± 7.1	66.4 ± 8.2	44.2 ± 5.6	61.0 ± 7.2	56.7 ± 5.9	49.0 ± 5.8	48.6 ± 8.0	66.1 ± 8.2	**67.5 ± 8.7**
0.5%	50.5 ± 6.0	48.2 ± 5.7	59.0 ± 10.7	33.9 ± 5.0	52.9 ± 7.4	46.5 ± 7.5	37.5 ± 5.4	41.4 ± 6.9	58.1 ± 8.2	**60.8 ± 9.0**
Citeseer	**GCN-FP**	**GGNN**	**DCNN**	**ChebyNet**	**GCN**	**MPNN**	**GraphSAGE**	**GAT**	**LNet**	**AdaLNet**
Public	61.5 ± 0.9	64.6 ± 1.3	69.4 ± 1.3	70.1 ± 0.8	68.1 ± 1.3	64.0 ± 1.9	67.2 ± 1.0	**72.2 ± 0.9**	66.2 ± 1.9	68.7 ± 1.0
1%	54.3 ± 4.4	56.0 ± 3.4	62.2 ± 2.5	59.4 ± 5.4	58.3 ± 4.0	54.3 ± 3.5	51.0 ± 5.7	46.5 ± 9.3	61.3 ± 3.9	**63.3 ± 1.8**
0.5%	43.9 ± 4.2	44.3 ± 3.8	53.1 ± 4.4	45.3 ± 6.6	47.7 ± 4.4	41.8 ± 5.0	33.8 ± 7.0	38.2 ± 7.1	53.2 ± 4.0	**53.8 ± 4.7**
0.3%	38.4 ± 5.8	36.5 ± 5.1	44.3 ± 5.1	39.3 ± 4.9	39.2 ± 6.3	36.0 ± 6.1	25.7 ± 6.1	30.9 ± 6.9	44.4 ± 4.5	**46.7 ± 5.6**
Pubmed	**GCN-FP**	**GGNN**	**DCNN**	**ChebyNet**	**GCN**	**MPNN**	**GraphSAGE**	**GAT**	**LNet**	**AdaLNet**
Public	76.0 ± 0.7	75.8 ± 0.9	76.8 ± 0.8	69.8 ± 1.1	77.8 ± 0.7	75.6 ± 1.0	76.8 ± 0.6	76.7 +- 0.5	**78.3 ± 0.3**	78.1 ± 0.4
0.1%	70.3 ± 4.7	70.4 ± 4.5	73.1 ± 4.7	55.2 ± 6.8	73.0 ± 5.5	67.3 ± 4.7	65.4 ± 6.2	59.6 +- 9.5	**73.4 ± 5.1**	72.8 ± 4.6
0.05%	63.2 ± 4.7	63.3 ± 4.0	66.7 ± 5.3	48.2 ± 7.4	64.6 ± 7.5	59.6 ± 4.0	53.0 ± 8.0	50.4 +- 9.7	**68.8 ± 5.6**	66.0 ± 4.5
0.03%	56.2 ± 7.7	55.8 ± 7.7	60.9 ± 8.2	45.3 ± 4.5	57.9 ± 8.1	53.9 ± 6.9	45.4 ± 5.5	50.9 +- 8.8	60.4 ± 8.6	**61.0 ± 8.7**

Liao et al. (2019)

Experiments

Quantum Chemistry

Goal: Predict 16 quantities per molecule in QM8 dataset



Ramakrishnan et al. (2015)

Experiments

Quantum Chemistry

Goal: Predict 16 quantities per molecule in QM8 dataset

Methods	Validation MAE ($\times 1.0e^{-3}$)	Test MAE ($\times 1.0e^{-3}$)
GCN-FP [29]	15.06 ± 0.04	14.80 ± 0.09
GGNN [37]	12.94 ± 0.05	12.67 ± 0.22
DCNN [8]	10.14 ± 0.05	9.97 ± 0.09
ChebyNet [7]	10.24 ± 0.06	10.07 ± 0.09
GCN [11]	11.68 ± 0.09	11.41 ± 0.10
MPNN [62]	11.16 ± 0.13	11.08 ± 0.11
GraphSAGE [39]	13.19 ± 0.04	12.95 ± 0.11
GPNN [40]	12.81 ± 0.80	12.39 ± 0.77
GAT [33]	11.39 ± 0.09	11.02 ± 0.06
LanczosNet	$\mathbf{9.65} \pm \mathbf{0.19}$	$\mathbf{9.58} \pm \mathbf{0.14}$
AdaLanczosNet	10.10 ± 0.22	9.97 ± 0.20

Liao et al. (2019)

Experiments

Ablation studies in QM8

	Model	Graph Kernel	Node Embedding	Spectral Filter	Short Scales	Long Scales	Lanczos Step	Validation MAE ($\times 1.0e^{-3}$)
Multi-Scale Graph Convolution	LanczosNet		one-hot		{1, 2, 3}			10.71
	LanczosNet		one-hot		{3, 5, 7}			10.60
	LanczosNet		one-hot			{10, 20, 30}	20	10.54
	LanczosNet		one-hot		{3, 5 ,7}	{10, 20, 30}	20	**10.41**
Lanczos Step	LanczosNet		one-hot			{10, 20, 30}	5	10.49
	LanczosNet		one-hot			{10, 20, 30}	10	**10.44**
	LanczosNet		one-hot			{10, 20, 30}	20	10.54
	LanczosNet		one-hot			{10, 20, 30}	40	10.49
Learning Spectral Filter	LanczosNet		one-hot	3-MLP	{3, 5 ,7}	{10, 20, 30}	20	10.44
	LanczosNet		one-hot	5-MLP	{3, 5 ,7}	{10, 20, 30}	20	10.54
	LanczosNet		✓	3-MLP	{3, 5 ,7}	{10, 20, 30}	20	10.26
	LanczosNet		✓	3-MLP		{1, 2, 3, 5, 7, 10, 20, 30}	20	**9.56**
Graph Kernel/Node Embedding	AdaLanczosNet	✓	one-hot	3-MLP	{3, 5, 7}	{10, 20, 30}	20	10.99
	AdaLanczosNet		✓	3-MLP	{3, 5, 7}	{10, 20, 30}	20	10.20
	AdaLanczosNet		✓	3-MLP	{1, 2, 3}	{5, 7, 10, 20, 30}	20	**9.96**

Liao et al. (2019)

Conclusions

- LanczosNet uses the Lanczos algorithm to extract useful features from graphs
- The method enables analysis of multi-scale patterns in graphs
- Allows efficient learning of spectral filters
- Achieves SOTA performance in two challenging benchmarks

UNIVERSITY OF TORONTO

Limitations

- The Lanczos algorithm could be time consuming, less desirable for real-time applications
- What is the applicability in directed graphs?
- What are the implications for use in graphs with significantly larger size?

Questions?

- Please reach out in Piazza