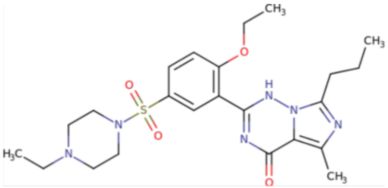
Hierarchical Graph Representation Learning via Differentiable Pooling

Rex Ying, Jiaxuan You, Christopher Morris, William L. Hamilton, Xiang Ren, Jure Leskovec

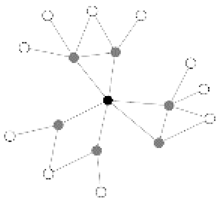
Stanford University
TU Dortmund University
University of Southern California

Motivation: ML for Graphs

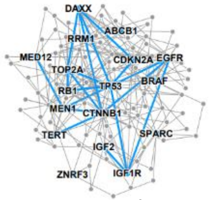
- Graph classification tasks:
  - Molecule prediction
    - Classify molecule properties (toxicity, drug-likeness etc.)



  - Social networks
    - Predict social group properties



  - Biological applications
    - Model disease pathways in PPI networks
  - Physical systems
    - Evolving dynamical systems



Graph Pooling

Graph Neural Networks (GNNs) have revolutionized machine learning with graphs

But GNNs learn individual node representations and then simply globally aggregate them:

- Mean/max/sum of all node embeddings (e.g. structure2vec)
- Pool by sorting (e.g. DGCNN, PatchySan)

Problem: How to aggregate information in a hierarchical way to capture the entire graph

Pooling for GNNs

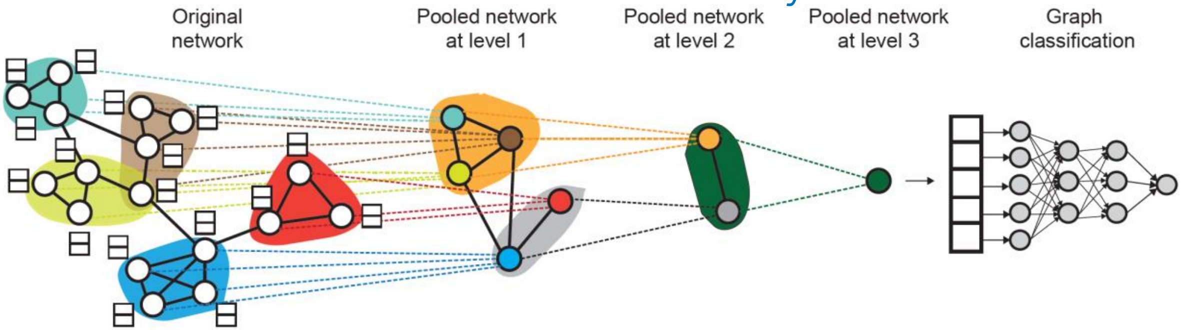
Problem: Learn a hierarchical pooling strategy that respects graph structure

Our solution: DiffPool

- Learns hierarchical pooling analogous to CNNs
- Sets of nodes are pooled hierarchically
- Soft assignment of nodes to next-level nodes

DiffPool Architecture

A different GNN is learned at every level of abstraction



Our approach: Use two sets of GNNs

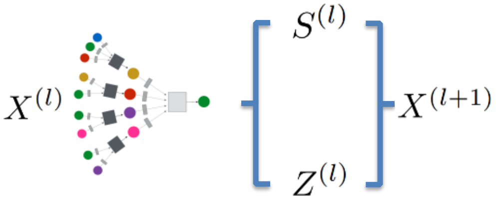
- GNN1 to learn how to pool the network
  - Learn cluster assignment matrix
- GNN2 to learn the node embeddings

DiffPool Architecture

Assuming general GNN model:

$$H^{(k)} = M(A, H^{(k-1)}; \theta^{(k)})$$

Concretely: $\mathrm{ReLU}(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(k-1)}W^{(k-1)})$



Two-tower architecture

$$Z^{(l)} = \mathrm{GNN}_{l,\mathrm{embed}}(A^{(l)}, X^{(l)})$$ Embedding

$$S^{(l)} = \mathrm{softmax}\left(\mathrm{GNN}_{l,\mathrm{pool}}(A^{(l)}, X^{(l)})\right)$$ Assignment

Aggregate embedding via assignment to generate next-level representations and adjacency

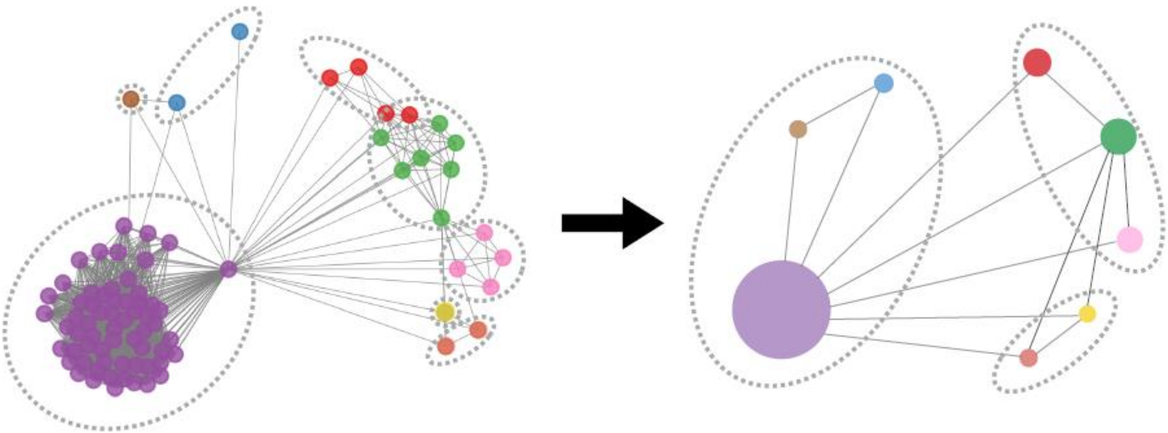
Experimental Results

An average of 6.27% improvement in accuracy for graph classification tasks on biological and social networks

	Method	Data Set					
		ENZYMES	D&D	REDDIT-MULTI-12K	COLLAB	PROTEINS	Gain
GNN	PATCHYSAN	–	76.27	41.32	72.60	75.00	4.17
	GRAPHSAGE	54.25	75.42	42.24	68.25	70.48	–
	ECC	53.50	74.10	41.73	67.79	72.65	0.11
	SET2SET	60.15	78.12	43.49	71.75	74.29	3.32
	SORTPOOL	57.12	79.37	41.82	73.76	75.54	3.39
	DIFFPOOL-DET	58.33	75.47	46.18	**82.13**	75.62	5.42
	DIFFPOOL-NOLP	61.95	79.98	46.65	75.58	76.22	5.95
	DIFFPOOL	**62.53**	**80.64**	**47.08**	75.48	**76.25**	**6.27**

Experimental Results

DIFFPOOL learns reasonable pooling architectures



Pooling at Layer 1 **Pooling at Layer 2**

Thank you!

Poster: AB #14

Code: https://github.com/RexYing/diffpool