Hierarchical Graph Representation Learning with Differentiable Pooling

Challenge: When applying GNNs to graph classification, the standard approach is to generate embeddings for all the nodes in the graph and then to globally pool all these node embeddings together. This global pooling approach ignores any hierarchical structure that might be present in the graph, and it prevents researchers from building effective GNN models for predictive tasks over entire graphs

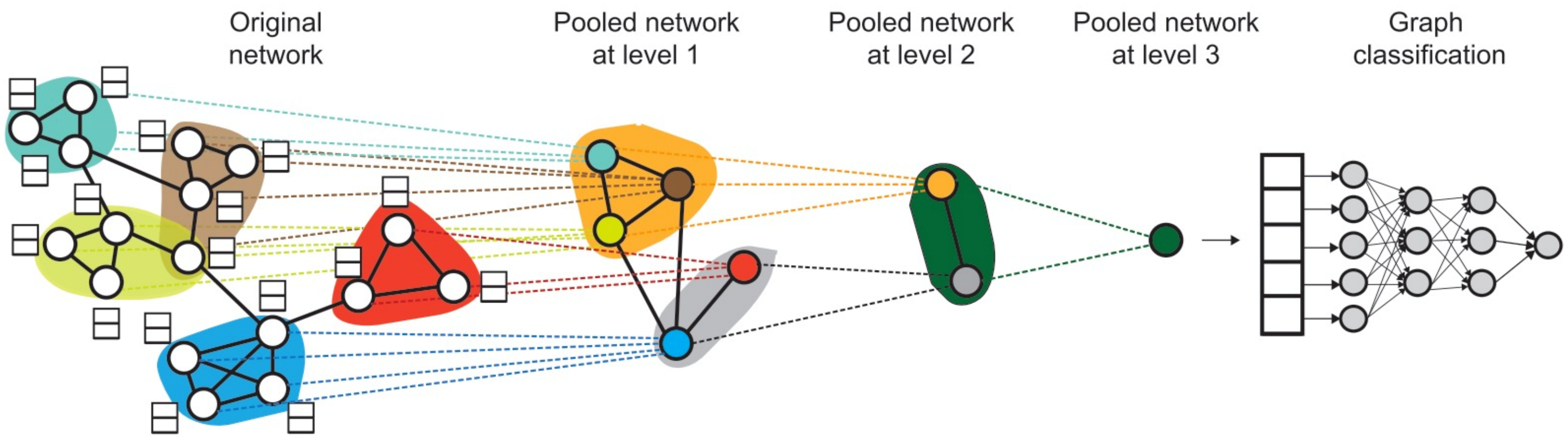


Figure 1: High-level illustration of our proposed method DIFFPOOL. At each hierarchical layer, we run a GNN model to obtain embeddings of nodes. We then use these learned embeddings to cluster nodes together and run another GNN layer on this coarsened graph. This whole process is repeated for L layers and we use the final output representation to classify the graph.

GNN $Z = \mathrm{GNN}(A, X)$

$$H^{(k)} = M(A, H^{(k-1)}; \theta^{(k)}), \quad Z = H^{(K)} \in \mathbb{R}^{n \times d},$$

Pooling $(A^{(l+1)}, X^{(l+1)}) = \textsc{DiffPool}(A^{(l)}, Z^{(l)})$

Cluster assignment matrix at layer l as $S^{(l)} \in \mathbb{R}^{n_l \times n_{l+1}}$

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

$$X^{(l+1)} = {S^{(l)}}^T Z^{(l)} \in \mathbb{R}^{n_{l+1} \times d},$$

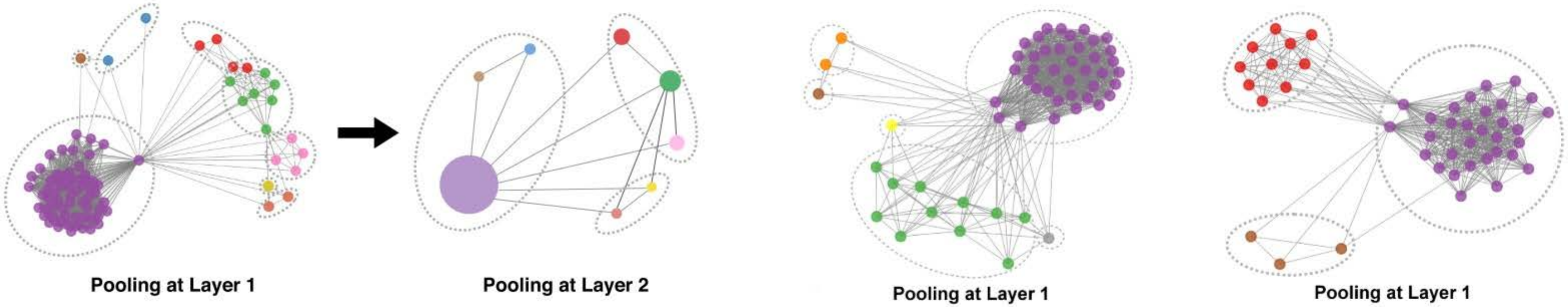
$$A^{(l+1)} = {S^{(l)}}^T A^{(l)} S^{(l)} \in \mathbb{R}^{n_{l+1} \times n_{l+1}}.$$

Auxiliary Link Prediction Objective and Entropy Regularization

Nearby nodes should be pooled together $L_{\mathrm{LP}} = ||A^{(l)}, S^{(l)} {S^{(l)}}^T||_F$

Output cluster assignment for each node should generally be close to a one-hot vector $L_{\mathrm{E}} = \frac{1}{n} \sum_{i=1}^{n} H(S_i),$

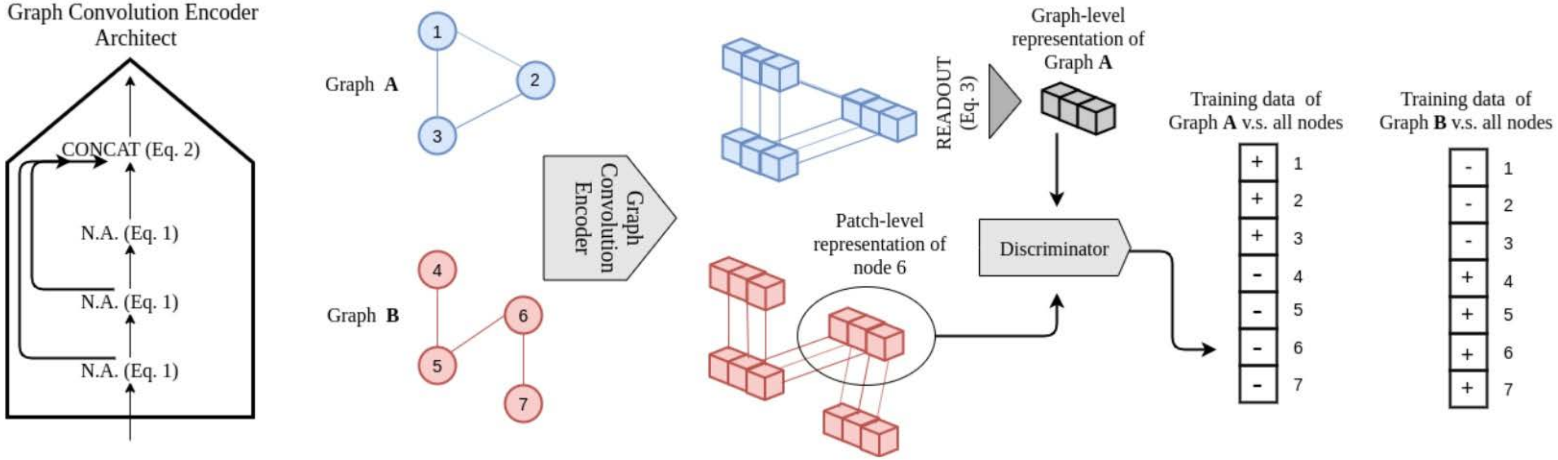
	Method	Data Set					
		ENZYMES	D&D	REDDIT-MULTI-12K	COLLAB	PROTEINS	Gain
Kernel	GRAPHLET	41.03	74.85	21.73	64.66	72.91	
	SHORTEST-PATH	42.32	78.86	36.93	59.10	76.43	
	1-WL	53.43	74.02	39.03	78.61	73.76	
	WL-OA	60.13	79.04	44.38	80.74	75.26	
GNN	PATCHYSAN	–	76.27	41.32	72.60	75.00	4.17
	GRAPHSAGE	54.25	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**



(a) (b) (c)

InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization

Motivation:
InfoGraph (unsupervised): Maximize the mutual information between the representations of entire graphs and the representations of substructures of different granularity
InfoGraph* (semi-supervised): Maximize the mutual information between intermediate representations of the two models so that the student model (supervised) learns from the teacher model (unsupervised).



$$\hat{\phi}, \hat{\psi} = \arg\max_{\phi,\psi} \sum_{G \in \mathbf{G}} \frac{1}{|G|} \sum_{u \in G} I_{\phi,\psi}(\vec{h}_\phi^u; H_\phi(G)).$$

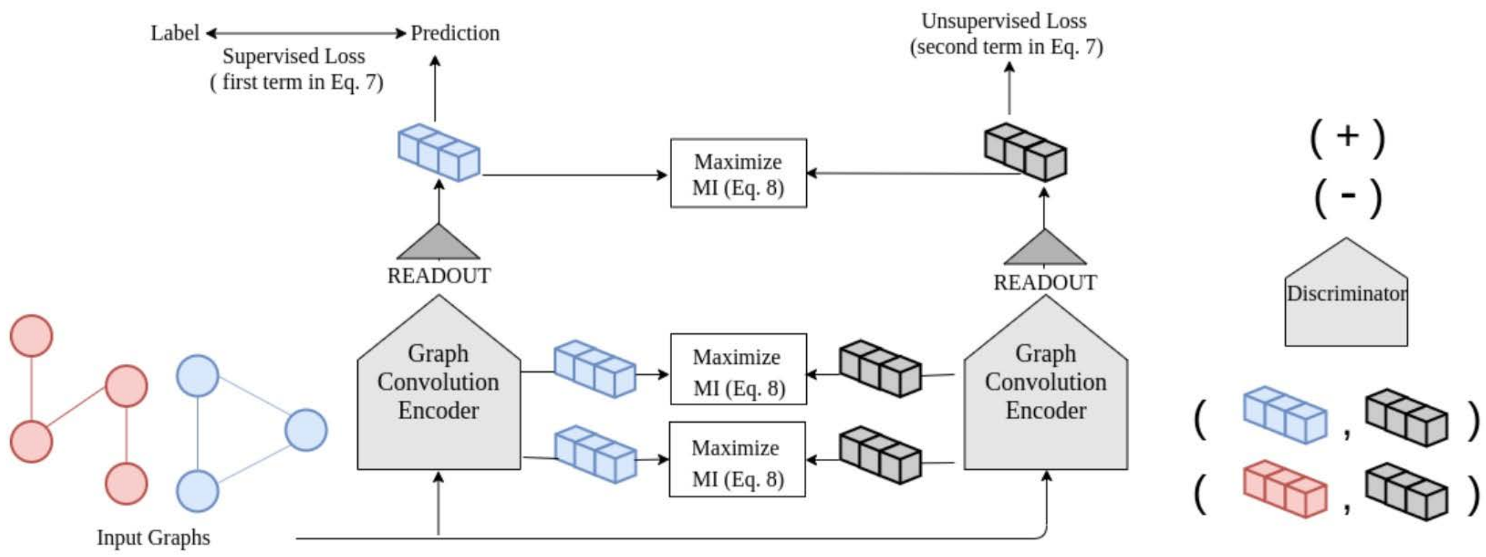
$$I_{\phi,\psi}(h_\phi^i(G); H_\phi(G)) :=$$

$$\mathbb{E}_{\mathbb{P}}[-\text{sp}(-T_{\phi,\psi}(\vec{h}_\phi^i(x), H_\phi(x)))] - \mathbb{E}_{\mathbb{P}\times\tilde{\mathbb{P}}}[\text{sp}(T_{\phi,\psi}(\vec{h}_\phi^i(x'), H_\phi(x)))]$$

* Graph neural network designs should be considered carefully so that graph representations can be discriminative towards

$$L_{\text{total}} = \sum_{i=1}^{|\mathbb{G}^L|} L_{\text{supervised}}(y_\phi(G_i), o_i) + \lambda \sum_{j=1}^{|\mathbb{G}^L|+|\mathbb{G}^U|} L_{\text{unsupervised}}(h_\phi(G_j); H_\phi(G_j))$$

Negative transfer



$$L_{\text{total}} = \sum_{i=1}^{|\mathbb{G}^L|} L_{\text{supervised}}(y_\phi(G_i), o_i) + \sum_{j=1}^{|\mathbb{G}^L|+|\mathbb{G}^U|} L_{\text{unsupervised}}(h_\varphi(G_j); H_\varphi(G_j))$$
$$- \lambda \sum_{j=1}^{|\mathbb{G}^L|+|\mathbb{G}^U|} \frac{1}{|G_j|} \sum_{k=1}^{K} I(H_\phi^k(G_j); H_\varphi^k(G_j)).$$

Dataset	MUTAG	PTC-MR	RDT-B	RDT-M5K	IMDB-B	IMDB-M
(No. Graphs)	188	344	2000	4999	1000	1500
(No. classes)	2	2	2	5	2	3
(Avg. Graph Size)	17.93	14.29	429.63	508.52	19.77	13.00
Graph Kernels						
RW	83.72 ± 1.50	57.85 ± 1.30	OMR	OMR	50.68 ± 0.26	34.65 ± 0.19
SP	85.22 ± 2.43	58.24 ± 2.44	64.11 ± 0.14	39.55 ± 0.22	55.60 ± 0.22	37.99 ± 0.30
GK	81.66 ± 2.11	57.26 ± 1.41	77.34 ± 0.18	41.01 ± 0.17	65.87 ± 0.98	43.89 ± 0.38
WL	80.72 ± 3.00	57.97 ± 0.49	68.82 ± 0.41	46.06 ± 0.21	72.30 ± 3.44	46.95 ± 0.46
DGK	87.44 ± 2.72	60.08 ± 2.55	78.04 ± 0.39	41.27 ± 0.18	66.96 ± 0.56	44.55 ± 0.52
MLG	87.94 ± 1.61	$\mathbf{63.26 \pm 1.48}$	> 1 Day	> 1 Day	66.55 ± 0.25	41.17 ± 0.03
Other Unsupervised Methods						
node2vec	72.63 ± 10.20	58.58 ± 8.00	-	-	-	-
sub2vec	61.05 ± 15.80	59.99 ± 6.38	71.48 ± 0.41	36.68 ± 0.42	55.26 ± 1.54	36.67 ± 0.83
graph2vec	83.15 ± 9.25	60.17 ± 6.86	75.78 ± 1.03	47.86 ± 0.26	71.1 ± 0.54	$\mathbf{50.44 \pm 0.87}$
InfoGraph	$\mathbf{89.01 \pm 1.13}$	61.65 ± 1.43	$\mathbf{82.50 \pm 1.42}$	$\mathbf{53.46 \pm 1.03}$	$\mathbf{73.03 \pm 0.87}$	49.69 ± 0.53

Target	Mu (0)	Alpha (1)	HOMO (2)	LUMO (3)	Gap (4)	R2 (5)	ZPVE(6)	U0 (7)	U (8)	H (9)	G(10)	Cv (11)
MAE	0.3201	0.5792	0.0060	0.0062	0.0091	10.0469	0.0007	0.3204	0.2934	0.2722	0.2948	0.2368

Semi-Supervised	Error Ratio											
Mean-Teachers	1.09	1.00	**0.99**	1.00	**0.97**	0.52	0.77	1.16	0.93	0.79	0.86	0.86
InfoGraph	1.02	0.97	1.02	**0.99**	1.01	0.71	0.96	0.85	0.93	0.93	0.99	1.00
InfoGraph*	**0.99**	**0.94**	**0.99**	**0.99**	0.98	**0.49**	**0.52**	**0.44**	**0.58**	**0.57**	**0.54**	**0.83**

Application in Urban Computing

- Wang, Pengyang, et al. "Adversarial substructured representation learning for mobile user profiling." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019
- Wang, Pengyang, et al. "You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.