HL-Pretrained Filterbank GNNs

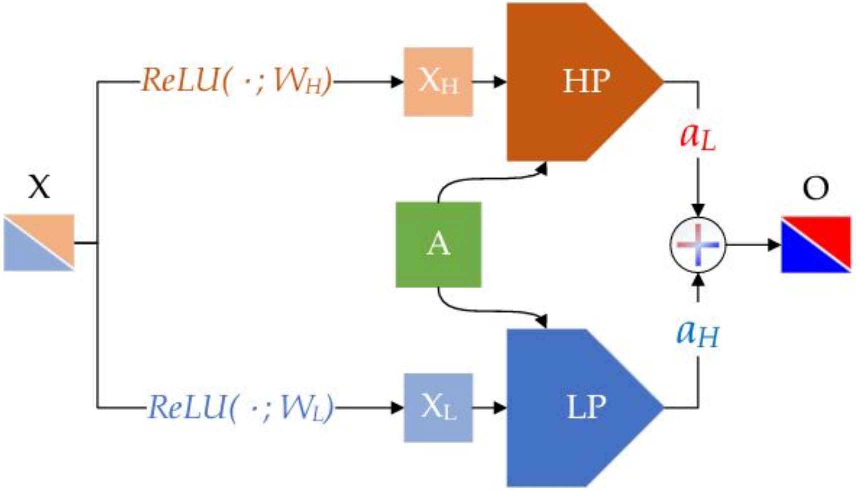
Original Idea: Complete the missing half

- GNNs message passing assumption: smoothness
  - Node aggregation: node-level low-pass filter (LP).
  - Capturing low-frequency signals (smooth features) across the graph and make nodes become similar to their neighbors.
- Problem:
  - Homophily assumption does not always hold
  - On networks like dating network and online purchasing network, non-smooth pattern turn out to be important.
  - Signal defined on graph is generally a mixture of smooth and nonsmoothgraphsignalsandeachpartplaysanindispensable role.

Original Idea: Complete the missing half

- Proposed method:
  - Normalized affinity matrix as the low-pass filter
    - $A_{sym} = D^{-1/2} A D^{-1/2}$
  - Normalized laplacian matrix as the high-pass filter
    - $L_{sym} = D^{-1/2} L D^{-1/2}$
  - low-pass filter: aim at aggregating node information
  - high-pass filter: aim at diversify node information
  - $A_{sym} + L_{sym} = I$

Original Idea: Complete the missing half

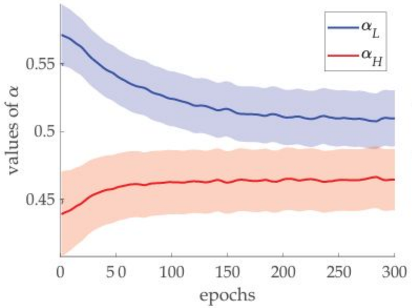


$$\boldsymbol{H}_L^l = L_{\mathrm{LP}} f(\boldsymbol{H}^{l-1} W_L^{l-1}), \quad \boldsymbol{H}_H^l = L_{\mathrm{HP}} f(\boldsymbol{H}^{l-1} W_H^{l-1})$$

$$\boldsymbol{H}^l = \alpha_L^l \cdot \boldsymbol{H}_L^l + \alpha_H^l \cdot \boldsymbol{H}_H^l, \quad l = 1, \ldots, n$$

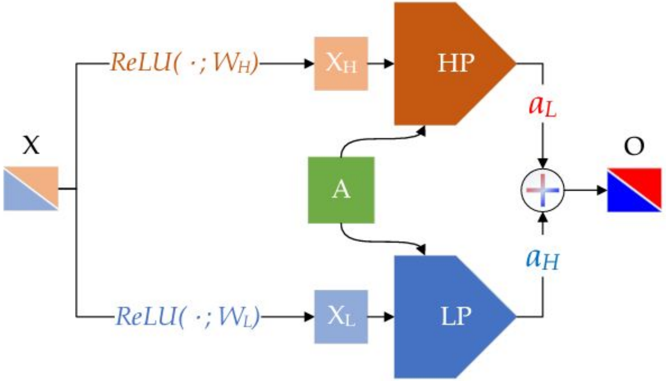
Problem

- Only applicable when there are a large amount of labeled data
  - Original paper data split: 60%/20%/20%
  - When the labels are scarce: weighting parameter becomes unstabilized and converge to different values based on different initialization.
  - Performance suffers and the average-10-run classification accuracies becomes lower than non-FB assisted GCN.
  - Filterbank cannot balance the smoothness & nonsmoothness proportion



Proposed method: HL-Pretrained Filterbank

- Utilize contrastive learning in pretraining
  - Use the high-frequency-signal extractor and low-frequency-signal extractor as augmentation methods.
  - By maximize the mutual information between two output representation, seek to balance out the smooth & nonsmooth proportion without utilizing additional labels



$$l(u_i, v_i) = log\frac{e^{\theta(u_i,v_i)/\tau}}{e^{\theta(u_i,v_i)/\tau} + \sum_{k\neq i} e^{\theta(u_i,v_k)/\tau} + \sum_{k\neq i} e^{\theta(u_i,u_k)/\tau}}$$

Datasets

- Cora (Homo)
  - 7 classes, 2708 nodes, 5429 edges
- Citeseer (Homo)
  - 6 classes, 3327 nodes, 4732 edges
- Chameleon (Hetero)
  - 5 classes, 2277 nodes, 36101 edges
- Texas (Hetero)
  - 5 classes, 183 nodes, 309 edges
- Squirrel (Hetero)
  - 5 classes, 5201 nodes, 217073 edges

Result (data split: 5%/35%/60%)

Accuracy	Chameleon	Citeseer	Cora	Texas	Squirrel
Original Idea (W/O pretraining)	0.6386	0.7867	0.8587	0.6812	0.4963
Proposed method	0.5957	0.7705	0.8275	0.5215	0.4579