Self Attention Graph Pooling

ICML 2019

Junhyun Lee, Inyeop Lee, Jaewoo Kang

Department of Computer Science and Engineering, Korea University, Seoul, Korea

Introduced by Qing Lyu

06/19/2019

Contribution

- Propose a graph pooling method based on self-attention
- The proposed pooling method considers both node features and graph topology
- The proposed method achieves superior graph classification performance on the benchmark datasets using a reasonable number of parameters

Graph convolution

- Spectral approach
  - redefine the convolution operation in the Fourier domain
    - graph Laplacian with the Chebyshev expansion (introduced by Fatir on April 17)
- Non-spectral approach
  - the central node aggregates features from adjacent nodes when its features are passed to the next layer
    - GraphSAGE: learns node embeddings through sampling and aggregation
    - Graph Attention Network (GAT): based on attention mechanisms

Graph pooling

- Topology based pooling
  - eigendecomposition
  - Graclus
- Global pooling
  - consider graph features
  - use summation or neural networks to pool all the representations of nodes in each layer
    - Set2Set
    - SortPool
- Hierarchical pooling
  - learn feature- or topology-based node assignments in each layer
    - Diffpool
    - gPool

Motivation

- Current graph pooling methods are with high storage complexity
  - DiffPool: $\mathcal{O}(k|V|^2)$
  - gPool: $\mathcal{O}(|V| + |E|)$
- The number of parameters depends on the number of nodes

To further improve graph pooling, they propose SAGPool which can use features and topology to yield hierarchical representations with a reasonable complexity of time and space.

Method

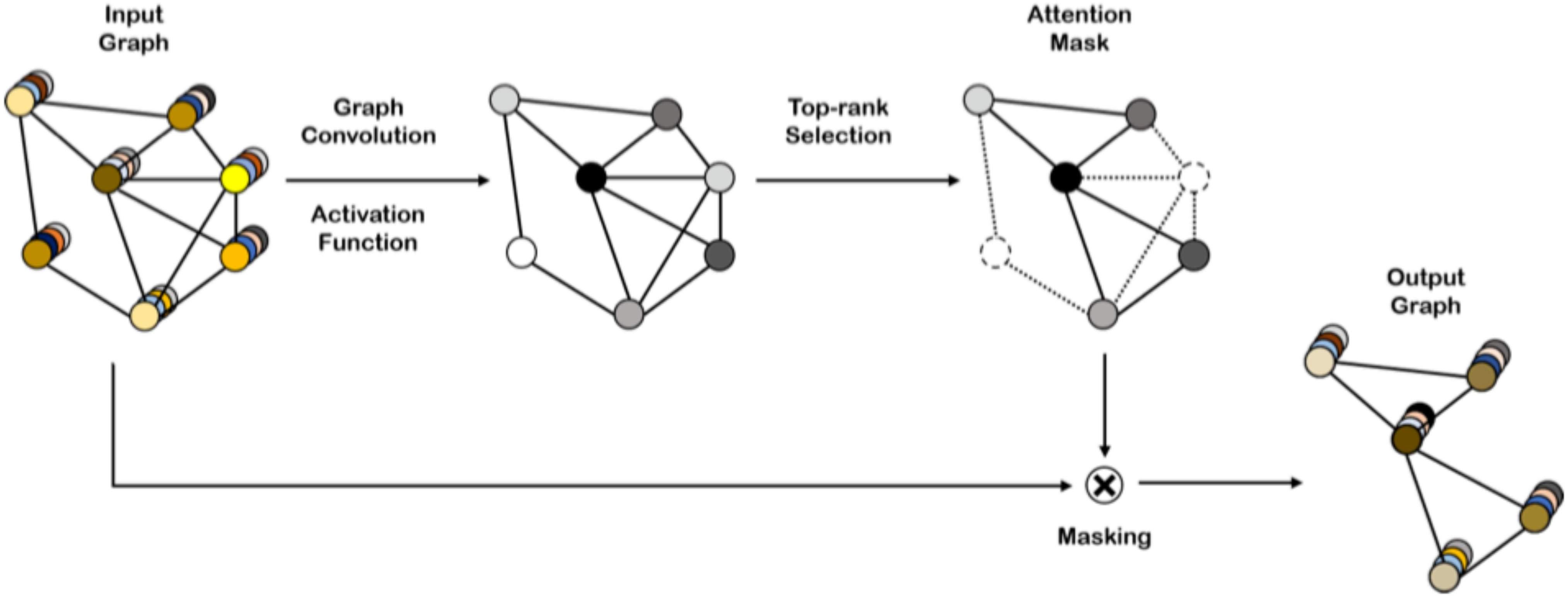


Figure 1. An illustration of the SAGPool layer.

Method

- Self-attention mask
  - Calculate self-attention score *Z*

$$Z = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta_{att})$$

  - Keep top *kN* nodes

$$\mathrm{idx} = \text{top-rank}(Z, \lceil kN \rceil), \quad Z_{mask} = Z_{\mathrm{idx}}$$

Method

- SAGPool
  - Generalized equation for attention score *Z*

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

  - Two-hop connections: $\mathrm{SAGPool}_{\mathrm{augmentation}}$

$$Z = \sigma(\mathrm{GNN}(X, A + A^2))$$

  - Stacking GNN layers: $\mathrm{SAGPool}_{\mathrm{serial}}$

$$Z = \sigma(\mathrm{GNN}_2(\sigma(\mathrm{GNN}_1(X, A)), A))$$

  - Averaging multiple attention scores: $\mathrm{SAGPool}_{\mathrm{parallel}}$

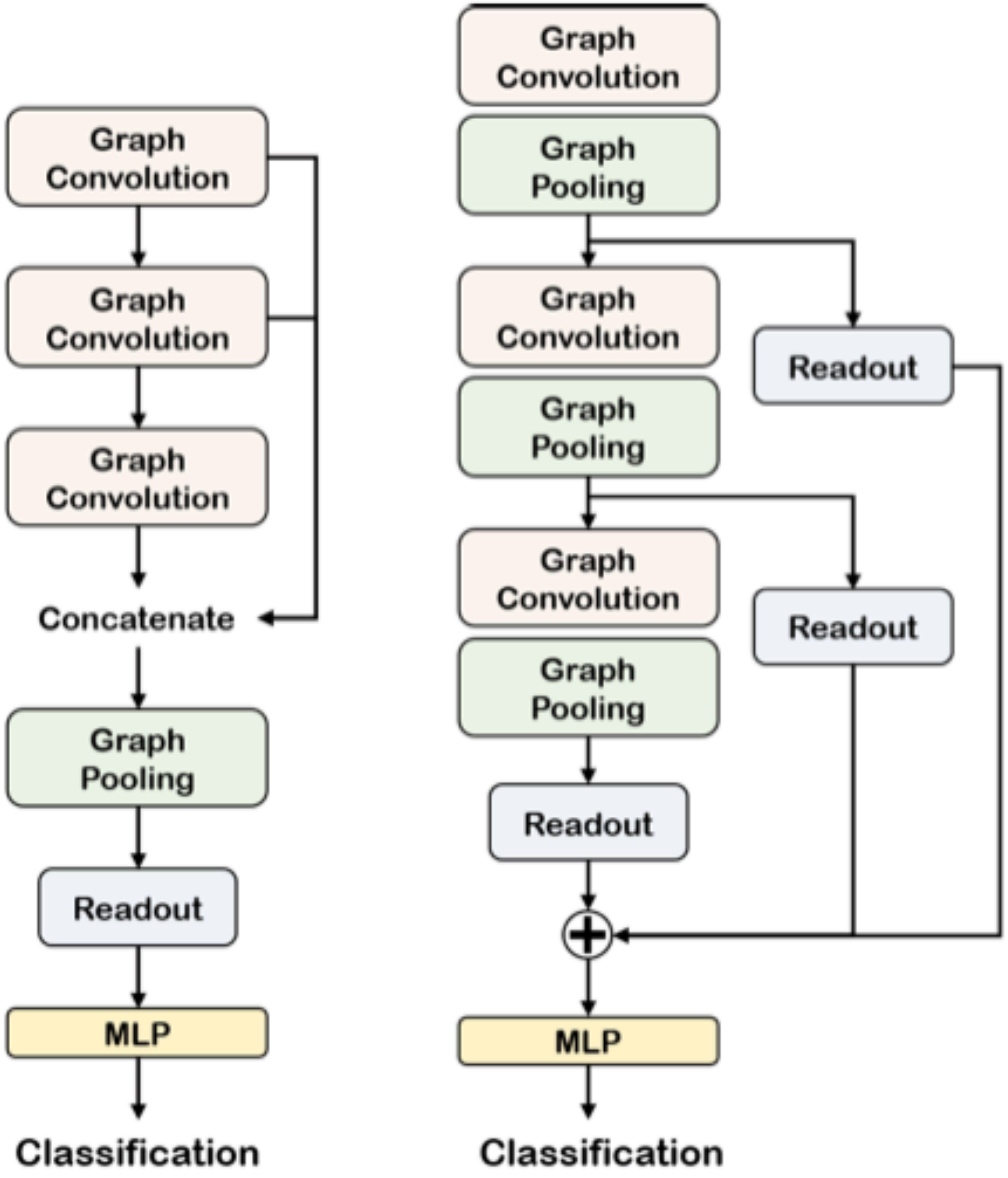
$$Z = \frac{1}{M}\sum_{m}\sigma(\mathrm{GNN}_m(X, A))$$

Method

Left: global pooling architecture SAGPool$_g$
Right: hierarchical pooling architecture SAGPool$_g$

Readout layer: aggregates node features to make a fixed size representation

$$s = \frac{1}{N}\sum_{i=1}^{N} x_i \,||\, \max_{i=1}^{N} x_i$$



Dataset

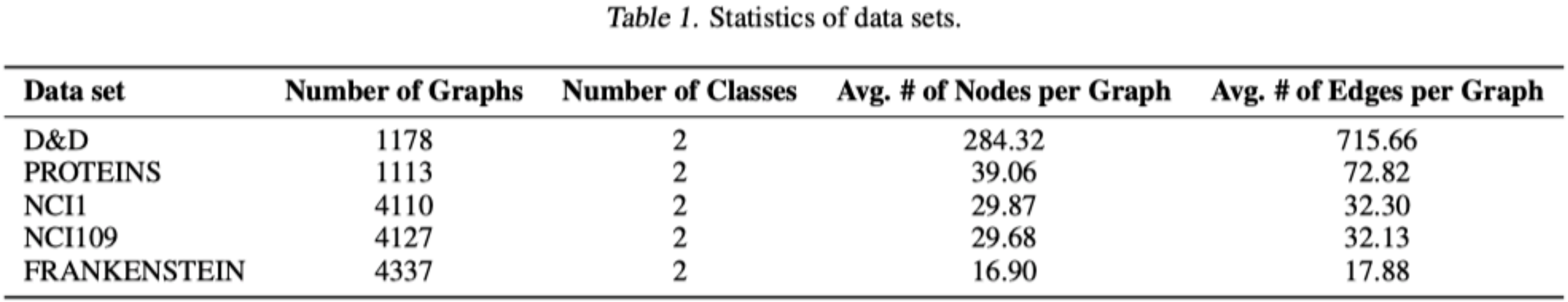


Table 1. Statistics of data sets.

Data set	Number of Graphs	Number of Classes	Avg. # of Nodes per Graph	Avg. # of Edges per Graph
D&D	1178	2	284.32	715.66
PROTEINS	1113	2	39.06	72.82
NCI1	4110	2	29.87	32.30
NCI109	4127	2	29.68	32.13
FRANKENSTEIN	4337	2	16.90	17.88

Result

Comparison with other methods

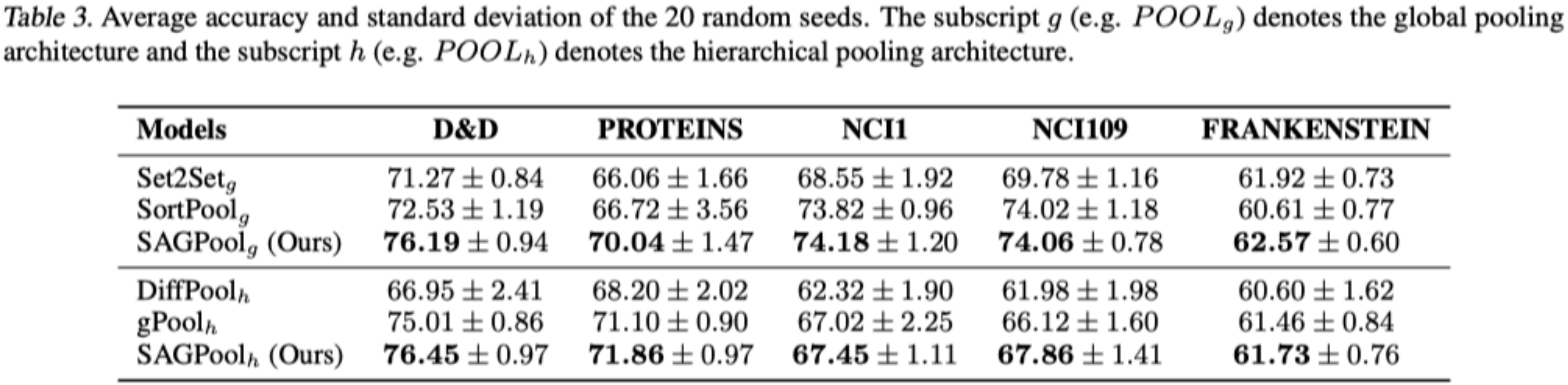


Table 3. Average accuracy and standard deviation of the 20 random seeds. The subscript g (e.g. $POOL_g$) denotes the global pooling architecture and the subscript h (e.g. $POOL_h$) denotes the hierarchical pooling architecture.

Models	**D&D**	**PROTEINS**	**NCI1**	**NCI109**	**FRANKENSTEIN**
Set2Set$_g$	71.27 ± 0.84	66.06 ± 1.66	68.55 ± 1.92	69.78 ± 1.16	61.92 ± 0.73
SortPool$_g$	72.53 ± 1.19	66.72 ± 3.56	73.82 ± 0.96	74.02 ± 1.18	60.61 ± 0.77
SAGPool$_g$ (Ours)	$\mathbf{76.19} \pm 0.94$	$\mathbf{70.04} \pm 1.47$	$\mathbf{74.18} \pm 1.20$	$\mathbf{74.06} \pm 0.78$	$\mathbf{62.57} \pm 0.60$
DiffPool$_h$	66.95 ± 2.41	68.20 ± 2.02	62.32 ± 1.90	61.98 ± 1.98	60.60 ± 1.62
gPool$_h$	75.01 ± 0.86	71.10 ± 0.90	67.02 ± 2.25	66.12 ± 1.60	61.46 ± 0.84
SAGPool$_h$ (Ours)	$\mathbf{76.45} \pm 0.97$	$\mathbf{71.86} \pm 0.97$	$\mathbf{67.45} \pm 1.11$	$\mathbf{67.86} \pm 1.41$	$\mathbf{61.73} \pm 0.76$

Result

Comparison with different SAGPools

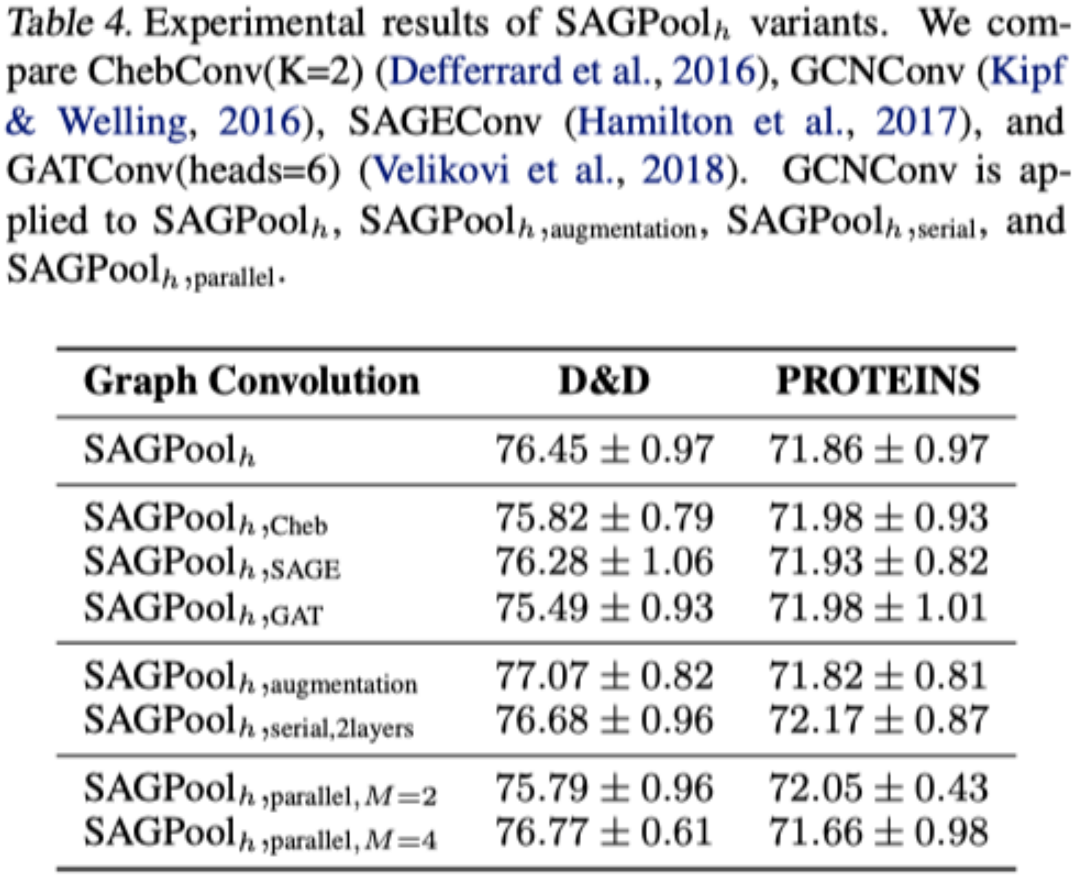
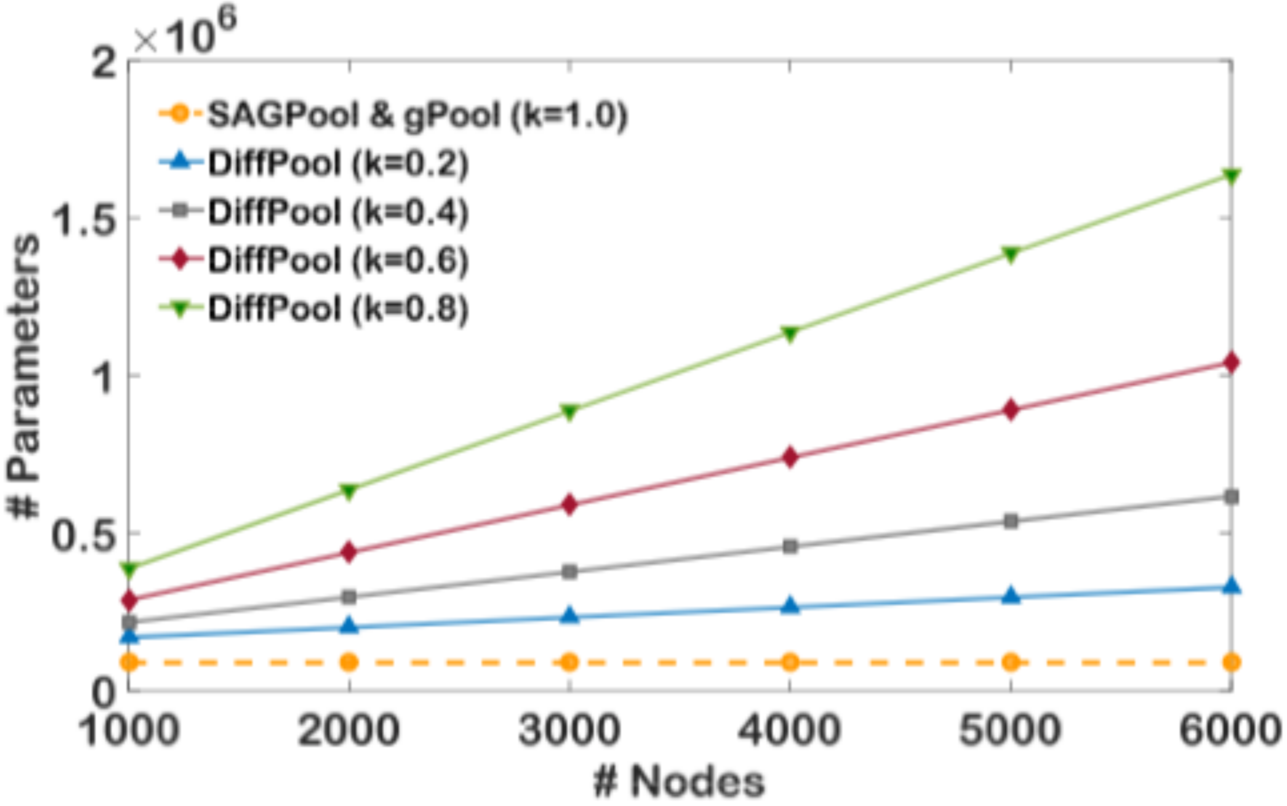


Table 4. Experimental results of SAGPool$_h$ variants. We compare ChebConv(K=2) (Defferrard et al., 2016), GCNConv (Kipf & Welling, 2016), SAGEConv (Hamilton et al., 2017), and GATConv(heads=6) (Velikovi et al., 2018). GCNConv is applied to SAGPool$_h$, SAGPool$_{h\text{ ,augmentation}}$, SAGPool$_{h\text{ ,serial}}$, and SAGPool$_{h\text{ ,parallel}}$.

Graph Convolution	**D&D**	**PROTEINS**
SAGPool$_h$	76.45 ± 0.97	71.86 ± 0.97
SAGPool$_{h\text{ ,Cheb}}$	75.82 ± 0.79	71.98 ± 0.93
SAGPool$_{h\text{ ,SAGE}}$	76.28 ± 1.06	71.93 ± 0.82
SAGPool$_{h\text{ ,GAT}}$	75.49 ± 0.93	71.98 ± 1.01
SAGPool$_{h\text{ ,augmentation}}$	77.07 ± 0.82	71.82 ± 0.81
SAGPool$_{h\text{ ,serial,2layers}}$	76.68 ± 0.96	72.17 ± 0.87
SAGPool$_{h\text{ ,parallel}, M=2}$	75.79 ± 0.96	72.05 ± 0.43
SAGPool$_{h\text{ ,parallel}, M=4}$	76.77 ± 0.61	71.66 ± 0.98

Result

Parameter comparison



Thanks