Graph Neural Networks

Lecture 8

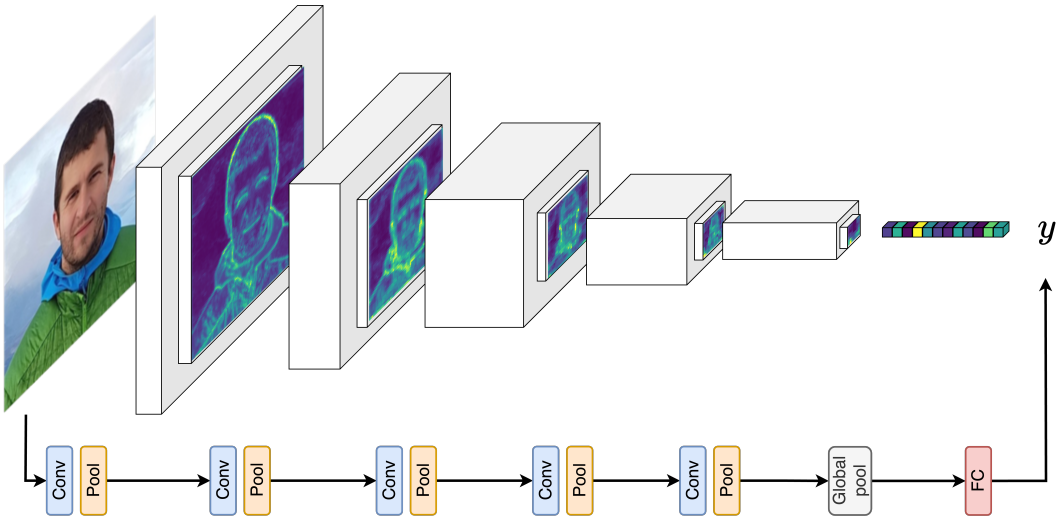
Danila Biktimirov
Applied Computer Science
Neapolis University Pafos

21 March 2025

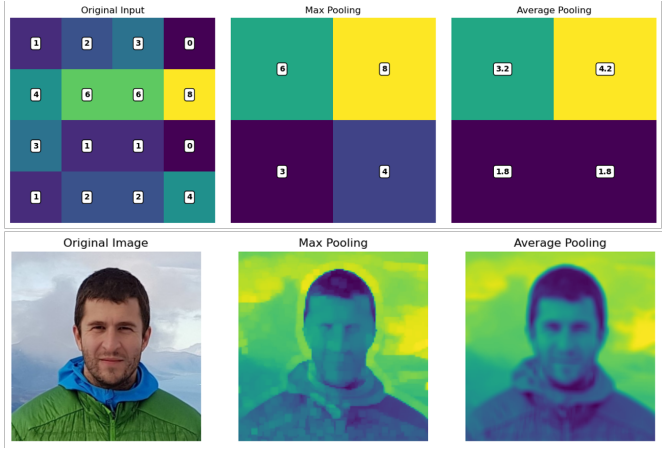
Overview

1. An introduction to pooling in GNNs
2. Flat GNNs
3. Select-Reduce-Connect (SRC)
4. Soft clustering pooling methods
5. One-over-K pooling methods
6. Score-based pooling methods
7. Evaluation procedures

An introduction to pooling in GNNs



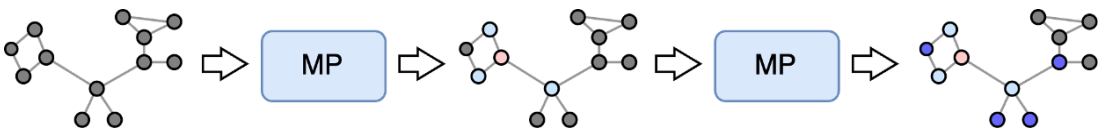
An introduction to pooling in GNNs



Graph Neural Networks

$$f\left(\text{  }\right) = \phi\left(\psi(\text{}), \gamma(\text{  })\right) = \text{  }$$

Flat GNNs



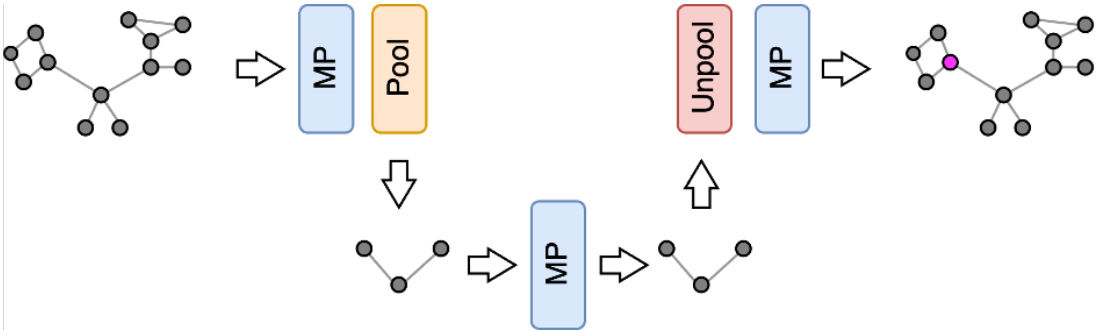
Flat GNNs

Hierarchical GNNs

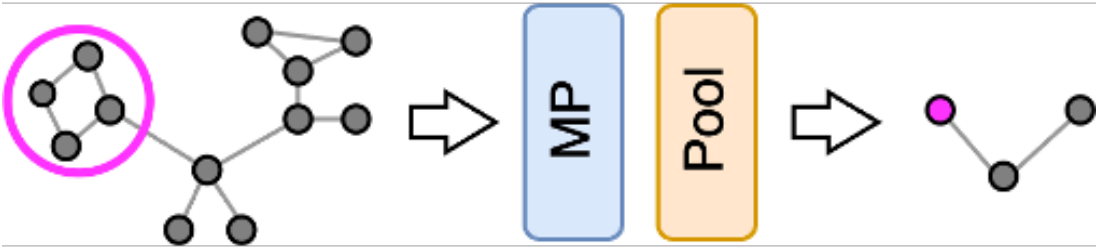
What kind of downstream tasks can be solved with hierarchical GNNs?

Graph-level Tasks

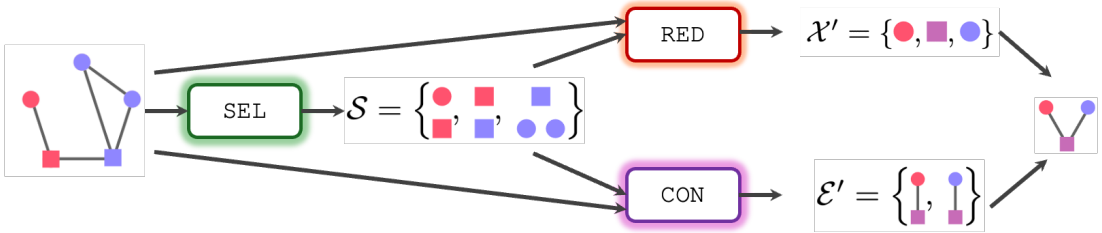
Node-level Tasks



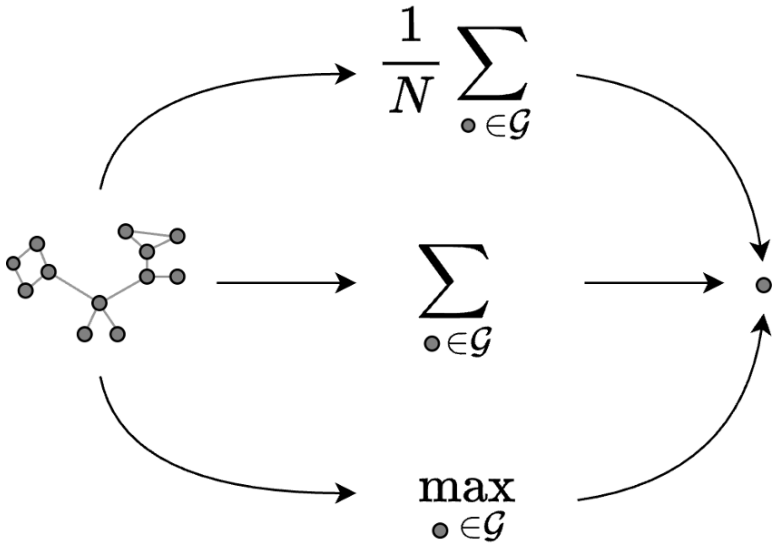
Community-level Tasks



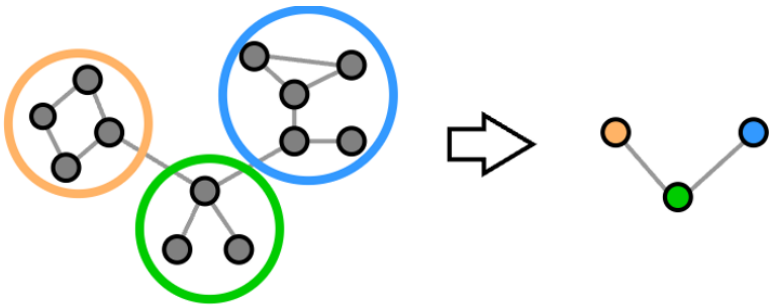
Select-Reduce-Connect (SRC)



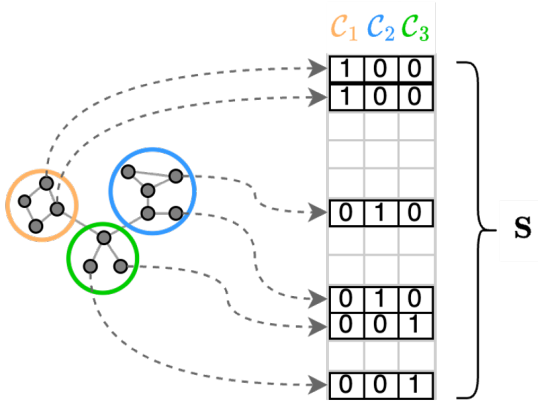
Global pooling



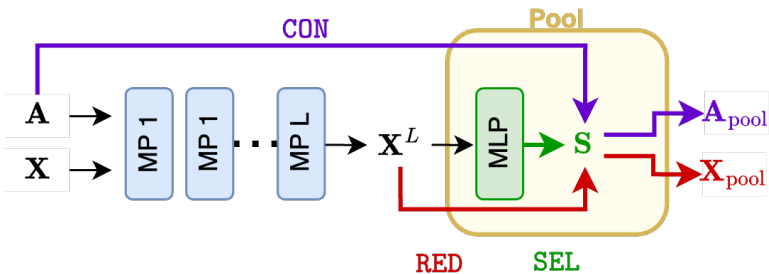
Soft clustering pooling methods



Soft clustering pooling methods



Soft clustering pooling methods



Soft clustering pooling methods

$$\mathbf{X}_{\text{pool}} = \mathbf{S}^\top \mathbf{X}$$

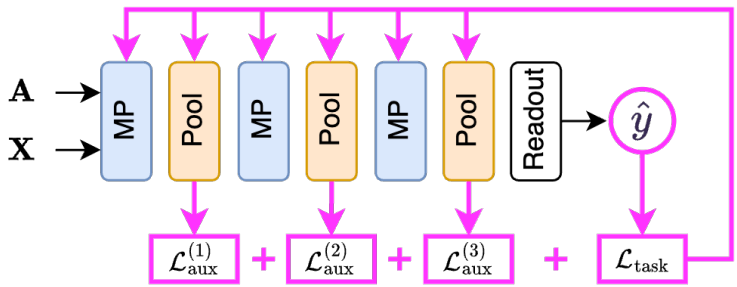
$$\mathbf{A}_{\text{pool}} = \mathbf{S}^\top \mathbf{A}\mathbf{S} \quad \text{or} \quad \mathbf{A}_{\text{pool}} = \mathbf{S}^+ \mathbf{A}\mathbf{S}$$

Degenerate solutions

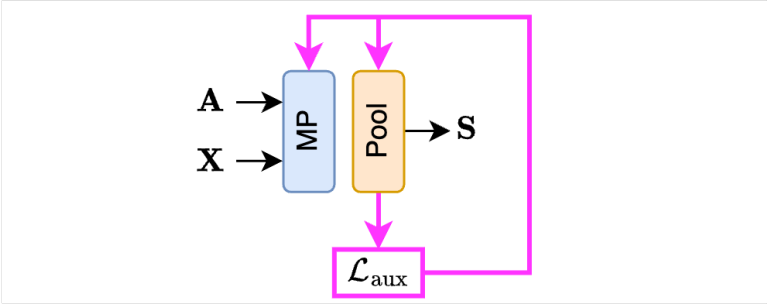
$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}$$

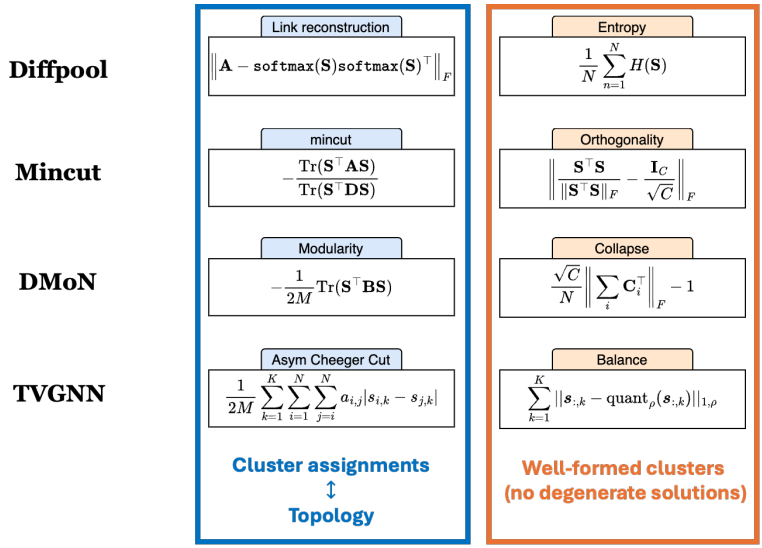
Auxiliary Losses



Auxiliary Losses

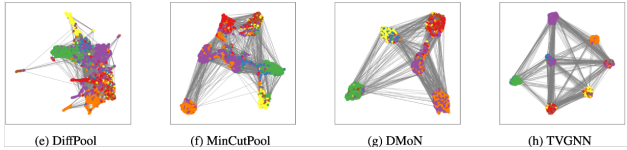


Auxiliary Losses



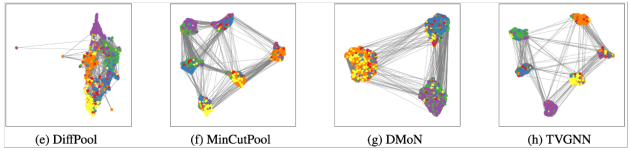
Auxiliary Losses

Cora



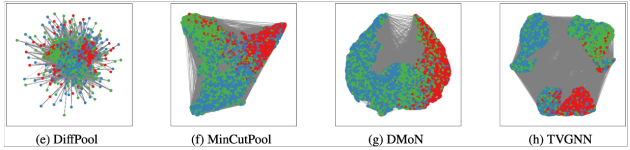
(e) DiffPool (f) MinCutPool (g) DMoN (h) TVGNN

Citeseer



(e) DiffPool (f) MinCutPool (g) DMoN (h) TVGNN

Pubmed



(e) DiffPool (f) MinCutPool (g) DMoN (h) TVGNN

Pros and Cons

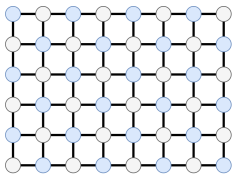
Pros:

- Trainable soft pooling adapts to the downstream task.
- Preserves much of the original graph's information via supernodes.

Cons:

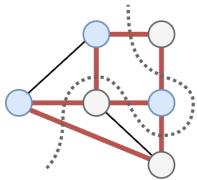
- $\mathbf{S}$ is dense and large; computing $\mathbf{A}' = \mathbf{S}^\top \mathbf{A}\mathbf{S}$ is expensive and results in dense $\mathbf{A}'$.
- Requires fixed K clusters; not ideal for graphs with varying sizes and may even *upscale* the graph.

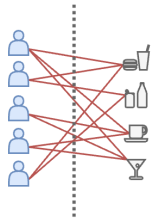
One-over-K pooling methods



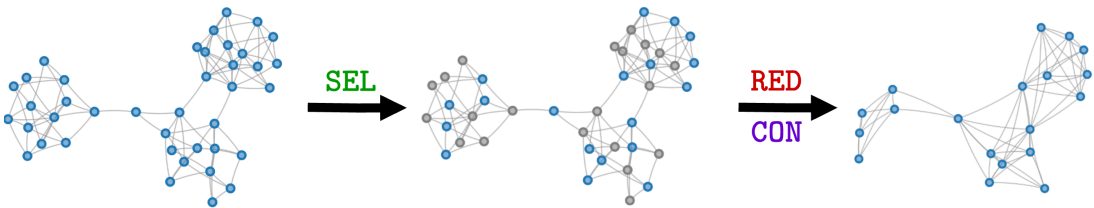
Graclus

Node Decimation Pool

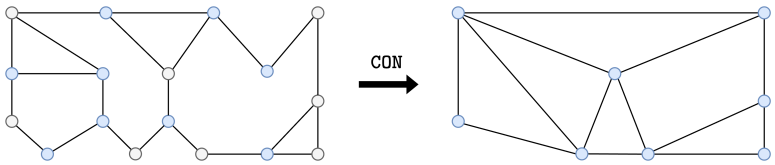




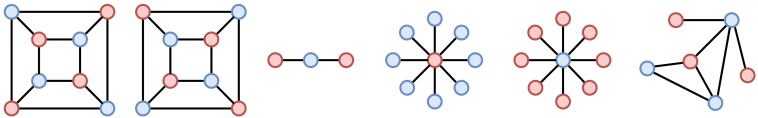
Node Decimation Pool



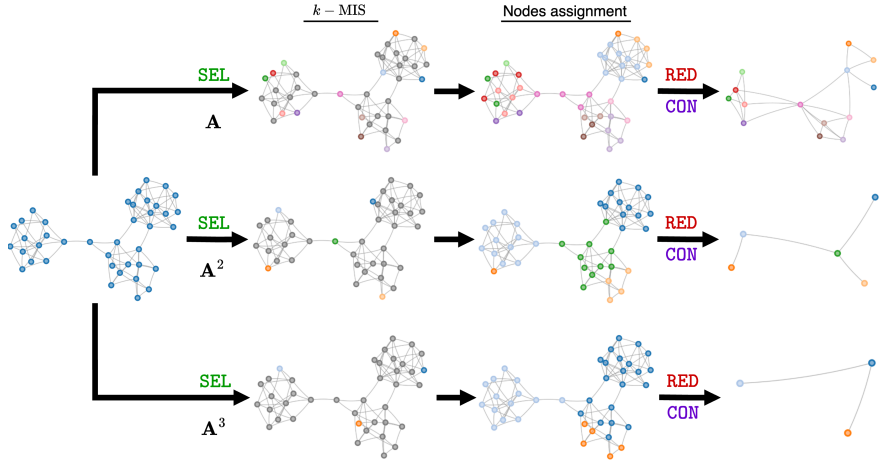
Node Decimation Pool



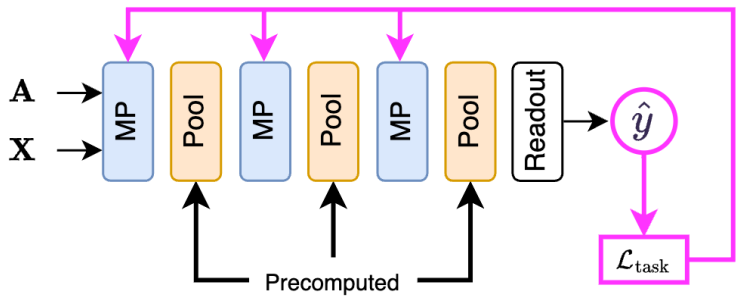
k-Maximal Independent Sets



k-Maximal Independent Sets



k-Maximal Independent Sets



Pros and Cons

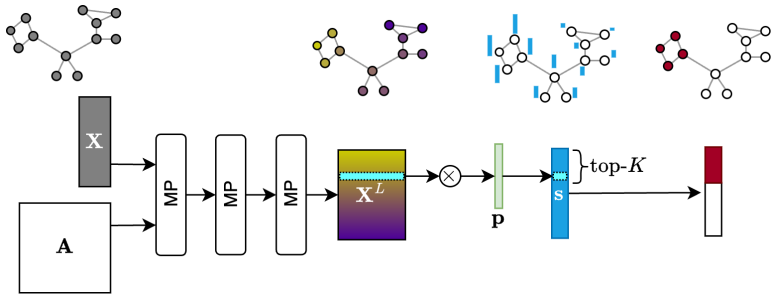
Pros:

- Precomputed pooling $\rightarrow$ fast and efficient, ideal for large graphs or limited compute.
- No trainable params $\rightarrow$ easier training and better for small datasets.

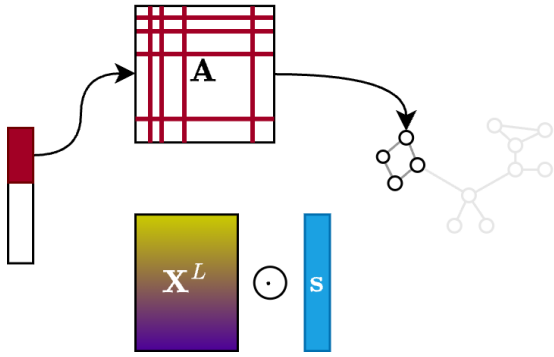
Cons:

- Cannot adapt pooling to specific tasks due to lack of learning.
- Ignores node features $\rightarrow$ risks poor aggregation decisions.

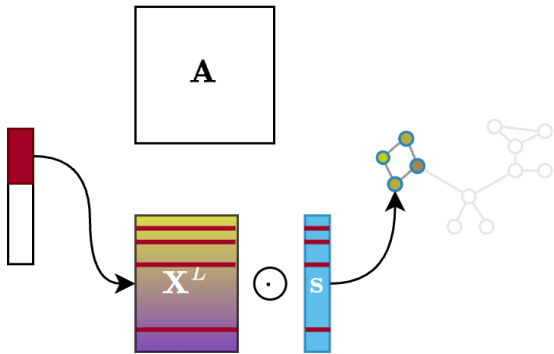
Score-based pooling methods



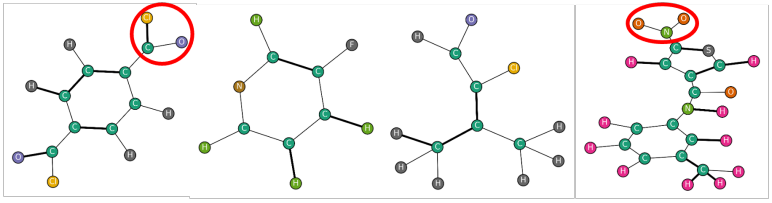
Score-based pooling methods



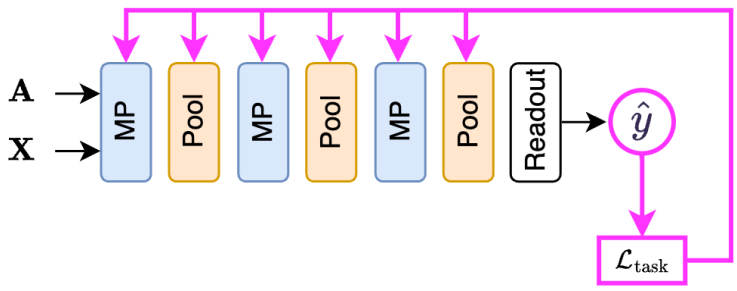
Score-based pooling methods



Node selection and training



Node selection and training



Pros and Cons

Pros:

- Flexible — score vector $\mathbf{s}$ depends on node features and the task.
- Fewer parameters and lower cost than soft clustering.

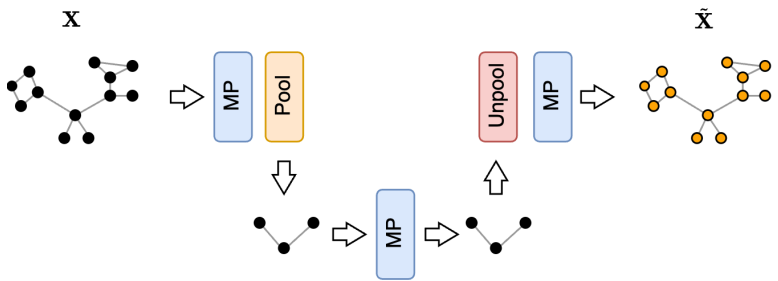
Cons:

- Discards parts of the graph — bad for tasks needing full structure preservation.

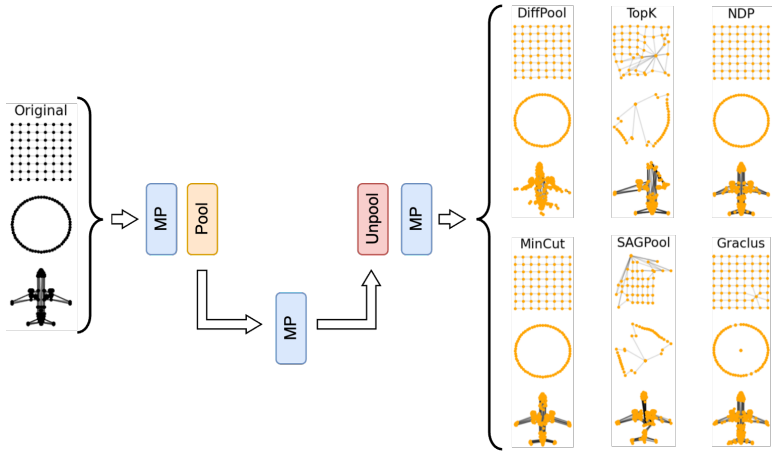
Evaluation procedures

Which pooler to choose usually depends on the data, the resources, and the task at hand.

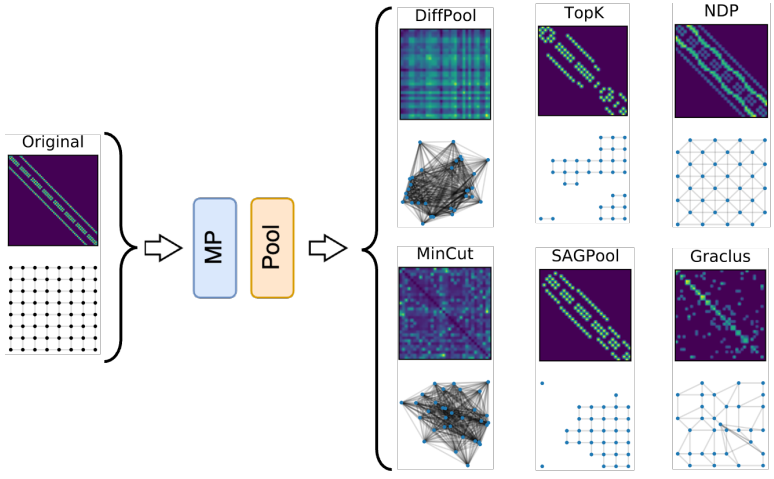
Preserving node information



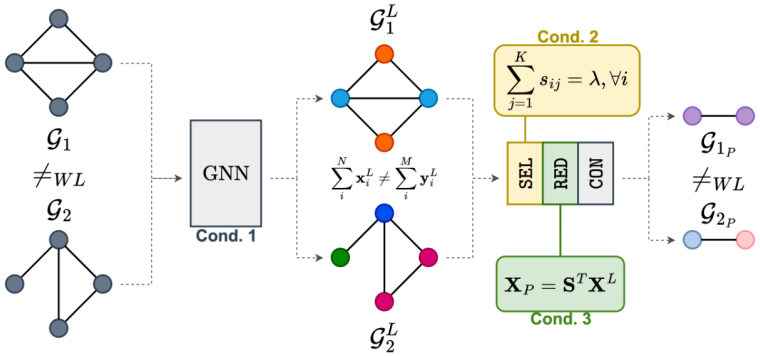
Preserving node information



Preserving topology



Expressiveness



The End?