# Towards Sparse Hierarchical Graph Classifiers

Cătălina Cangea* [1] Petar Veličković* [1] Nikola Jovanović [2] Thomas Kipf [3] Pietro Liò [1]

[1]University of Cambridge [2]Faculty of Computing, Union University Belgrade [3]University of Amsterdam

## Overview

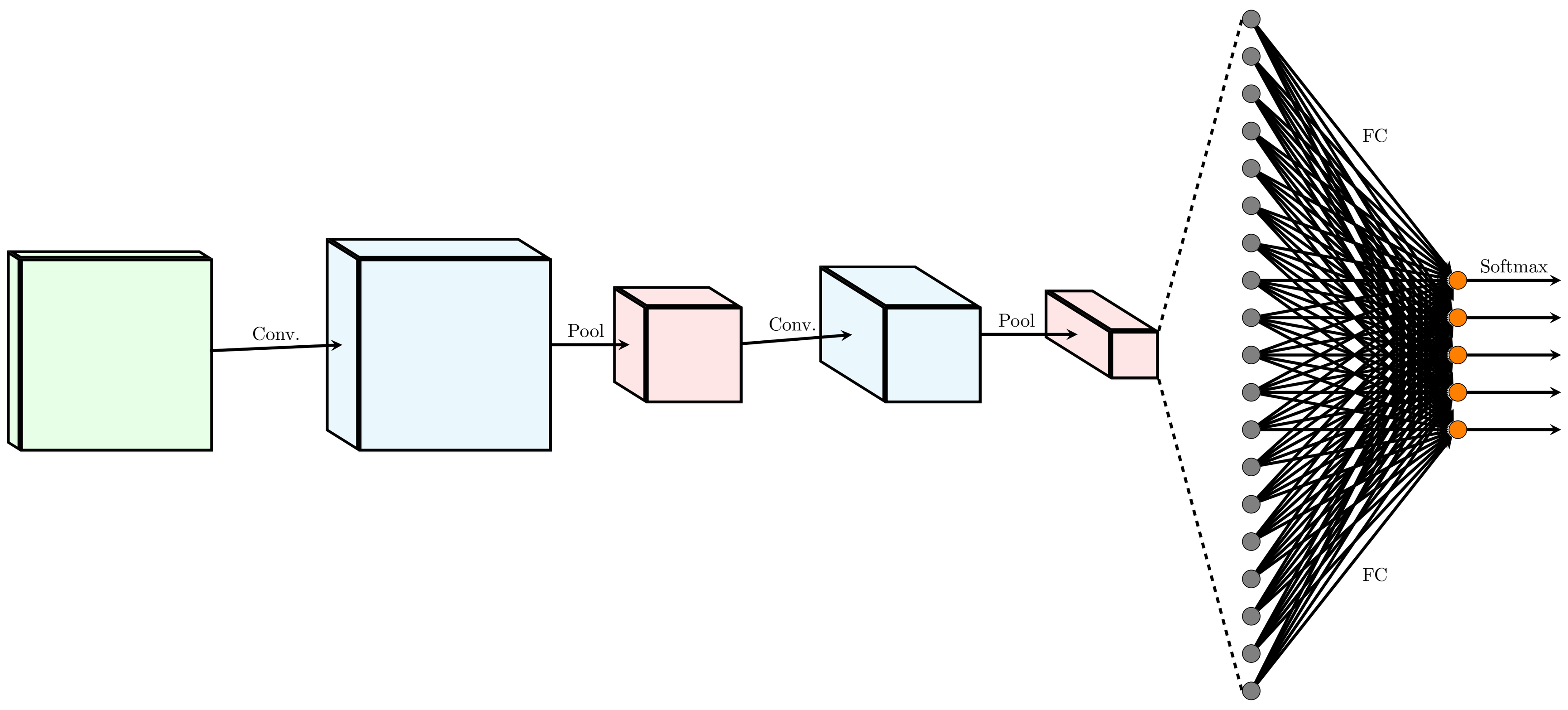
Here we study the problem of **graph classification**; the task of learning to categorise graphs into classes.

This is a direct generalisation of **image classification**, as images may be easily cast as a special case of a "grid graph" (with each pixel of an image connected to its eight immediate neighbours).

Therefore, it is natural to investigate and generalise CNN elements to graphs. However, while generalising the **convolutional** layer has received widespread attention, the **graph pooling** layer still lacks some desirable properties; such as **sparsity**.

Here we combine several recent advances in graph neural network design to demonstrate that **competitive** hierarchical graph classification results are possible **without sacrificing sparsity**.

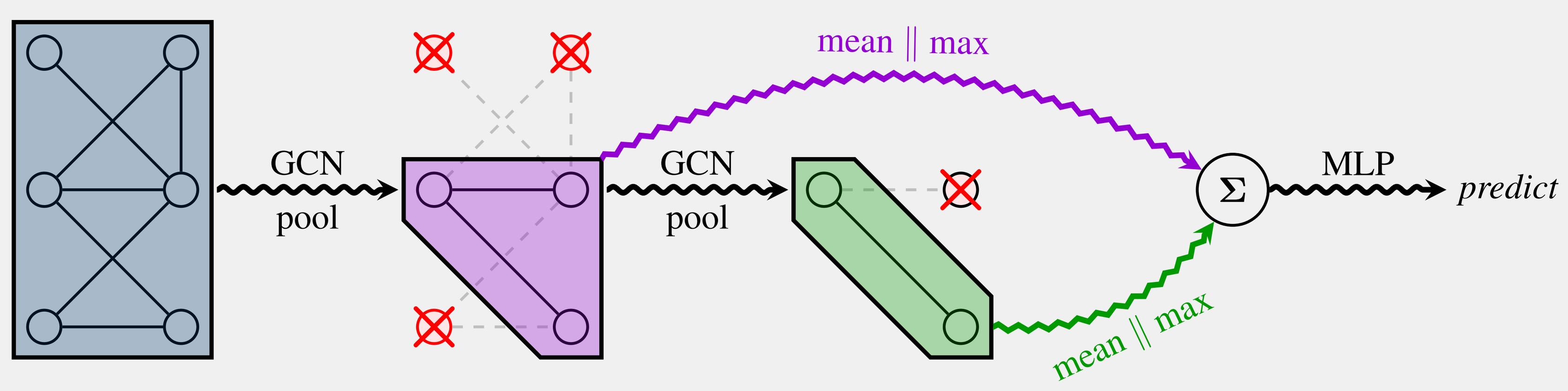
## A CNN-style *graph classifier?*




**Convolution**: *Many* choices with suitable properties, e.g. *graph convolutional networks*, *graph attention networks*, *message passing neural networks*...

**Pooling**: Either *fixed* (e.g. computed by *GraClus*) or *learnable* (*DiffPool* [5]). Fixed pooling is not **end-to-end**; DiffPool is not **sparse**.

## Architecture [2]




**Convolution**: We use the **mean-pooling** [3] propagation rule (to preserve inductive properties), with a **skip connection**:

$$\mathrm{MP}(\mathbf{X}, \mathbf{A}) = \sigma\left(\hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}\mathbf{X}\mathbf{\Theta} + \mathbf{X}\mathbf{\Theta}'\right)$$

**Pooling**: We use the recently proposed **Graph U-Net** [1] architecture, which pools by *dropping* $N - \lceil kN \rceil$ nodes, trivially preserving sparsity:

$$\vec{y} = \frac{\mathbf{X}\vec{p}}{\|\vec{p}\|} \qquad \vec{i} = \text{top-}k(\vec{y}, k) \qquad \mathbf{X}' = (\mathbf{X} \odot \tanh(\vec{y}))_{\vec{i}} \qquad \mathbf{A}' = \mathbf{A}_{\vec{i},\vec{i}}$$

**Readout**: Inspired by the **jumping knowledge network** [4] architecture, we perform *global average- and max-pooling* after every conv → pool block, *aggregating* them to obtain the final representation:

$$\vec{s}^{(l)} = \frac{1}{N^{(l)}}\sum_{i=1}^{N^{(l)}} \vec{x}_i^{(l)} \,\|\, \max_{i=1}^{N^{(l)}} \vec{x}_i^{(l)}$$

$$\vec{s} = \sum_{l=1}^{L} \vec{s}^{(l)}$$

## Quantitative results

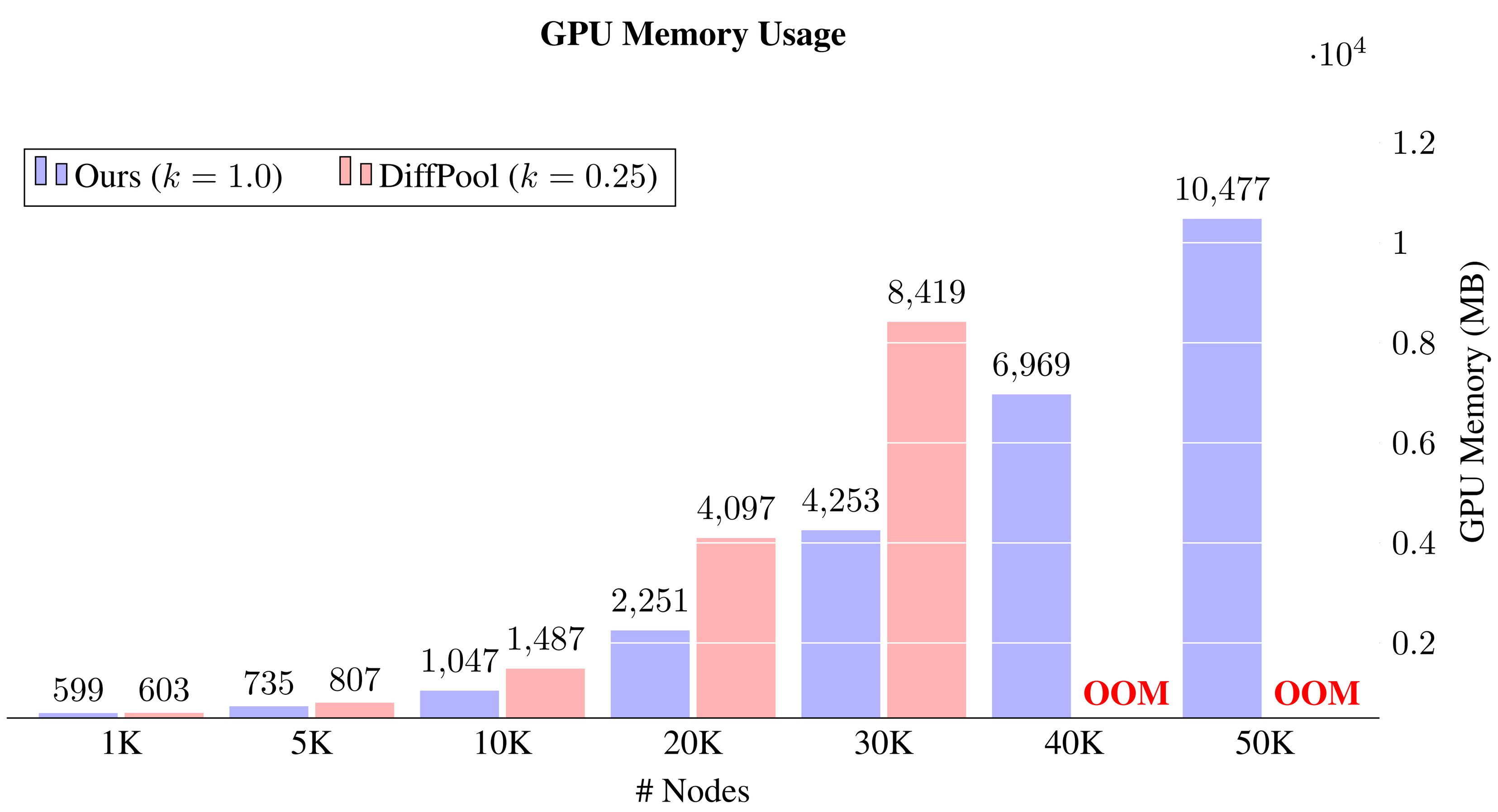
Classification accuracy percentages after 10-fold crossvalidation.

| Model | Datasets | | | |
|---|---|---|---|---|
| | *Enzymes* | *D&D* | *Collab* | *Proteins* |
| Graphlet | 41.03 | 74.85 | 64.66 | 72.91 |
| Shortest-path | 42.32 | 78.86 | 59.10 | 76.43 |
| 1-WL | 53.43 | 74.02 | 78.61 | 73.76 |
| WL-QA | 60.13 | 79.04 | 80.74 | 75.26 |
| PatchySAN | – | 76.27 | 72.60 | 75.00 |
| GraphSAGE | 54.25 | 75.42 | 68.25 | 70.48 |
| ECC | 53.50 | 74.10 | 67.79 | 72.65 |
| Set2Set | 60.15 | 78.12 | 71.75 | 74.29 |
| SortPool | 57.12 | 79.37 | 73.76 | 75.54 |
| DiffPool-Det | 58.33 | 75.47 | **82.13** | 75.62 |
| DiffPool-NoLP | 61.95 | 79.98 | 75.58 | 76.22 |
| DiffPool | 62.53 | **80.64** | 75.48 | **76.25** |
| Ours | **64.17** | 78.59 | 74.54 | 75.46 |

Our model successfully outperforms the sparse aggregation-based GraphSAGE baseline, while being a close competitor to DiffPool variants, across all datasets. This confirms the effectiveness of leveraging learnable pooling while preserving sparsity.

## Memory consumption

GPU memory usage of our method (with no pooling; $k = 1.0$) and DiffPool ($k = 0.25$) during training on Erdős-Rényi graphs of varying node sizes (and $|E| = 2|V|$). Both methods ran with 128 input and hidden features, and three Conv-Pool layers. "OOM" denotes out-of-memory.




## References

[1] Anonymous.
Graph u-net.
In *Submitted to the Seventh International Conference on Learning Representations (ICLR)*, 2018.

[2] Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò.
Towards sparse hierarchical graph classifiers.
*arXiv preprint arXiv:1811.01287*, 2018.

[3] Thomas N Kipf and Max Welling.
Semi-Supervised Classification with Graph Convolutional Networks.
In *International Conference on Learning Representations*, 2017.

[4] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka.
Representation learning on graphs with jumping knowledge networks.
In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5449--5458, 2018.

[5] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec.
Hierarchical graph representation learning with differentiable pooling.
In *Advances in Neural Information Processing Systems*, pages 4801--4811, 2018.