# Clique Pooling for Graph Classification

Enxhell Luzhnica*, Ben Day* and Pietro Lió
Department of Computer Science & Technology, University of Cambridge
Cambridge, United Kingdom.
el476@cam.ac.uk,ben.day@cl.cam.ac.uk

UNIVERSITY OF CAMBRIDGE

## 1. Introduction

Here we study the problem of **pooling** in the context of **graph classification**.

Current approaches (Ying et al., 2018; Cangea et al., 2018) require preset hyperparameters and are not purely **topological**, unlike the pooling in **CNN**s.

We show a purely **non-parametric, topological, static** pooling method that has a clear correspondence to regular graphs (images) and achieves competitive performance in both **regular** and **irregular graphs**.

## 2. Maximal Cliques Efficiency

The algorithm for finding maximal cliques presented by Bron and Kerbosch (1973) which was later on adapted to an iterative procedure and even parallelized (Zhang et al., 2005).

Eppstein et al. (2010) showed a method that can find all maximal cliques in $O(dn3^{d/3})$ where $d$ is the **degeneracy** of the graph.

The degeneracy is also known as *width, k-core number, linkeage* of the graph, and is also one less than the **chromatic number** of the graph which is known to be small in practice. For instance, for the **planar graphs**, $d = 3$ (at most) in which case the method is **linear**. Conte et al. (2016) showed that the maximal clique finding can be made efficient even on **large-scale social graphs**.

## 3. Architecture

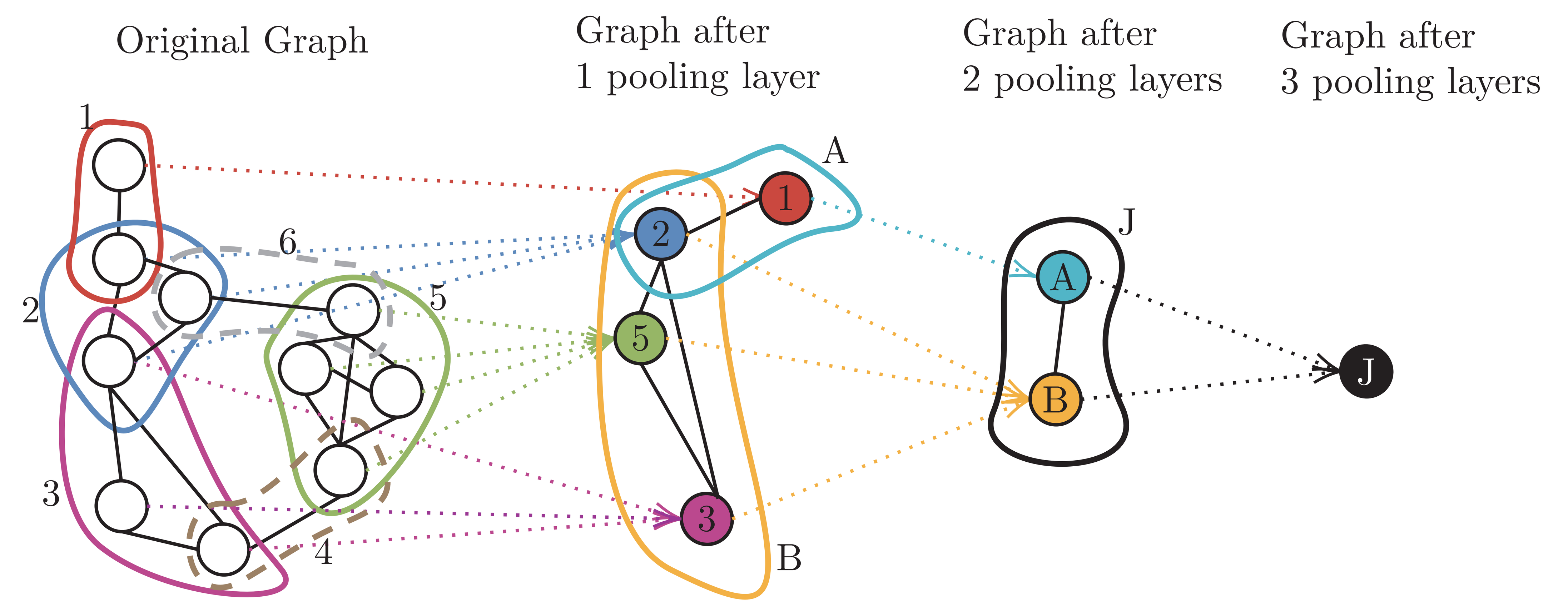
**Irregular Graphs** We use the **mean** option of GraphSAGE as the **convolution**.

The **readout function** of the whole graph is the **max** and **sum**. These readouts are then used in a **jumping knowledge** style - concatenating all the readouts after each layer and feeding them to an MLP.

The general architecture is: (GCN-Pool)-(GCN-Pool)-(GCN-Pool)-MLP.

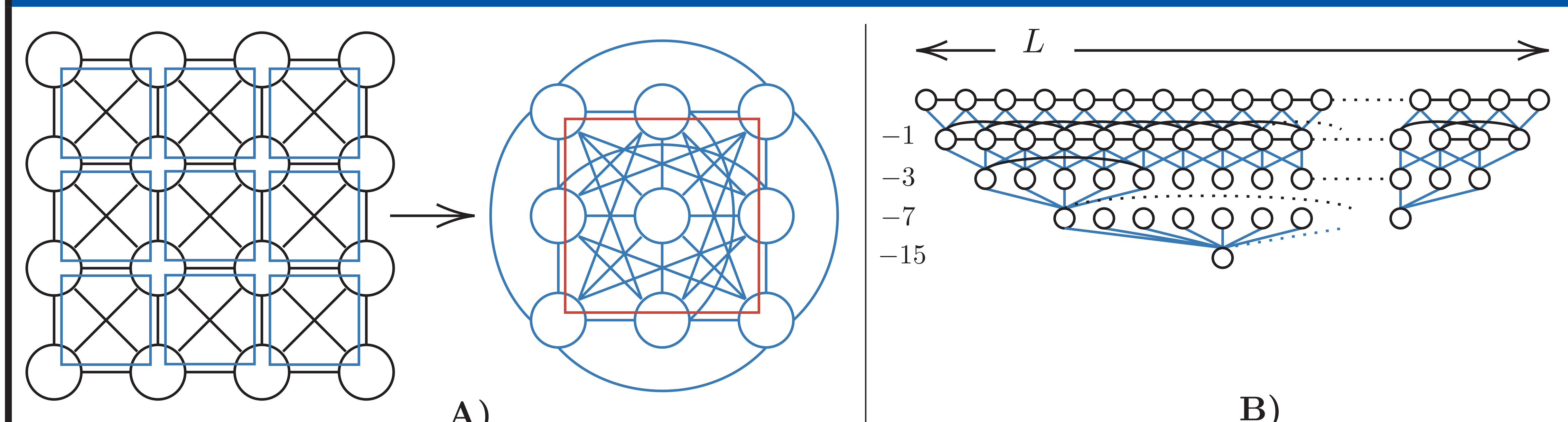
**Regular Graphs** We use a VGG architecture where all the pools have been substituted with the analog **clique pooling**.

## 4. Irregular Graphs




The colored borders represent the maximal cliques, dotted arrows indicate the cliques to which the nodes are assigned. Since the blue clique (2) is bigger (in terms of nodes), the node belonging to red (1) and blue (2) maximal clique is assigned to the **blue (2) cluster only**. In the case of the node intersecting the blue (2) and purple (3) maximal cliques, the node is assigned to **both cliques** since the cliques have the same size. The grey maximal clique (6) is **not represented** in the new coarsened graph since the nodes in that clique have already been assigned to larger cliques: the green (5) and blue (2) cliques. The nodes is the coarsened graph are connected if any two nodes in the respective cliques are connected.

## 5. Regular Graphs - Images




**A)** The use of 3-by-3 convolutions implies the graph structure as shown in the figure - pixels are connected to their eight immediate neighbours. The first pool will be 2-by-2 with a stride of 1. The second pool will then be 3-by-3 with a stride of 1.

**B)** In the right diagram we consider the **reduction** $r_n$ in nodes when the pooling operation is applied $n = 4$ times to a **1D chain**. Initially the cliques are of size two and the chain reduces in length by one. After that the cliques and reduction grow in size.

$$r_n = \sum_{i=1}^{n} 2^{n-1} = 2^n - 1.$$

## 6. Results

Our non-parametric approach is competitive with the parametric approaches. The method **outperforms** the GraphSAGE **baseline** and most of the **kernel-based** and GNN approaches. Moreover, because we do not introduce any additional parameters our method is fast to train and does not suffer the instabilities associated with DiffPool. It also **outperforms** the DiffPool with deterministic clustering on two datasets.
The image investigation found a small, but significant ($p = 0.02$), reduction of in mean accuracy over the 2-by-2 pool baseline.

| | Datasets | | | |
|---|---|---|---|---|
| | Enzymes | DD | 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 | 62.67 | 79.98 | 75.63 | 77.42 |
| DiffPool | 64.23 | **81.15** | 75.50 | **78.10** |
| Sparse HGC | **64.17** | 78.59 | 74.54 | 75.46 |
| CliquePool | 60.71 | 79.03 | 73.30 | 74.21 |

| Model | Mean | Std. Dev. |
|---|---|---|
| 2-by-2 pool | **92.4** | 0.3 |
| Clique pool | 92.0 | 0.5 |

Table 1: Results of benchmarking on irregular graphs (left), and CIFAR-10 for regular graphs (right).

## 7. References

Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 9 1973. ISSN 00010782. doi: 10.1145/362342.362367. URL http://portal.acm.org/citation.cfm?doid=362342.362367.

Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò. Towards Sparse Hierarchical Graph Classifiers. 11 2018. URL http://arxiv.org/abs/1811.01287.

Alessio Conte, Roberto De Virgilio, Antonio Maccioni, Maurizio Patrignani, and Riccardo Torlone. Finding all maximal cliques in very large social networks. In *Advances in Database Technology - EDBT*, volume 2016-March, 2016. ISBN 9783893180707. doi: 10.5441/002/edbt.2016.18. URL https://openproceedings.org/2016/conf/edbt/paper-54.pdf.

David Eppstein, Maarten Löffler, and Darren Strash. Listing All Maximal Cliques in Sparse Graphs in Near-optimal Time. 6 2010. URL http://arxiv.org/abs/1006.5440.

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling. 2018. ISSN 18160948. doi: arXiv:1806.08804v3. URL https://arxiv.org/pdf/1806.08804.pdf http://arxiv.org/abs/1806.08804.

Yun Zhang, Faisal N. Abu-Khzam, Nicole E. Baldwin, Elissa J. Chesler, Michael A. Langston, and Nagiza F. Samatova. Genome-scale computational approaches to memory-intensive applications in systems biology. In *Proceedings of the ACM/IEEE 2005 Supercomputing Conference, SC'05*, 2005. ISBN 1595930612. doi: 10.1109/SC.2005.29.