Weisfeiler and Leman Go Loopy

Raffaele Paolino*, Sohir Maskey*, Pascal Welke, Gitta Kutyniok

1. Motivation

↪ Need for more expressive and scalable neural architectures:

- ▲ Message Passing Neural Networks expressive power bounded by Weisfeiler-Leman test.
- ▲ Neural Networks based on higher-order Weisfeiler-Leman test present scalability issues.

↪ Ability to count important substructures:

- ▲ Other methods have limited cycle-counting powers.
- ▲ Inputting substructures counting is not flexible as substructures need to be defined by the user.

2. Proposed Solution

↪ We increase neighborhood of nodes by considering “nearby” paths:

- ▲ Provably more expressive than other methods.
- ▲ Scalable as it retains the initial sparsity of the graph.

3. Preliminaries

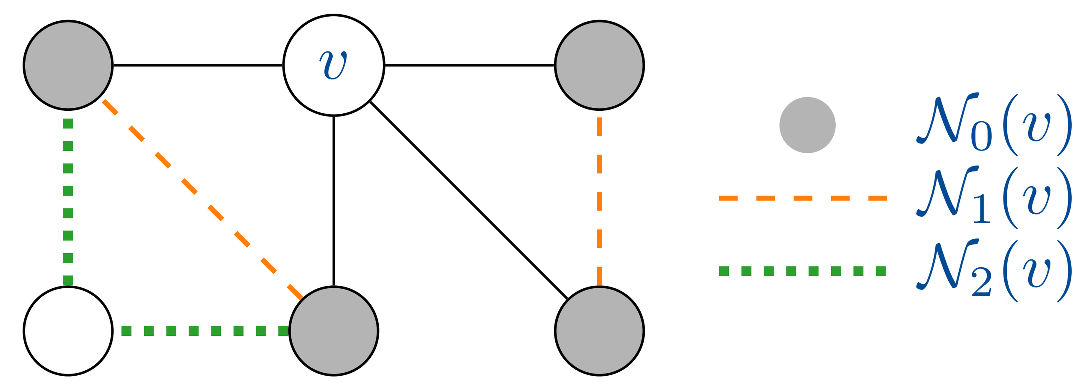
Definition. Given a graph G, a simple path of length r is a collection $\mathbf{p} = \{p_i\}_{i=1}^{r+1}$ of $r+1$ nodes such that consecutive nodes are adjacent, i.e.,

$$p_i, p_{i+1} \in E(G),\ \forall i \in \{1, \dots, r\},$$

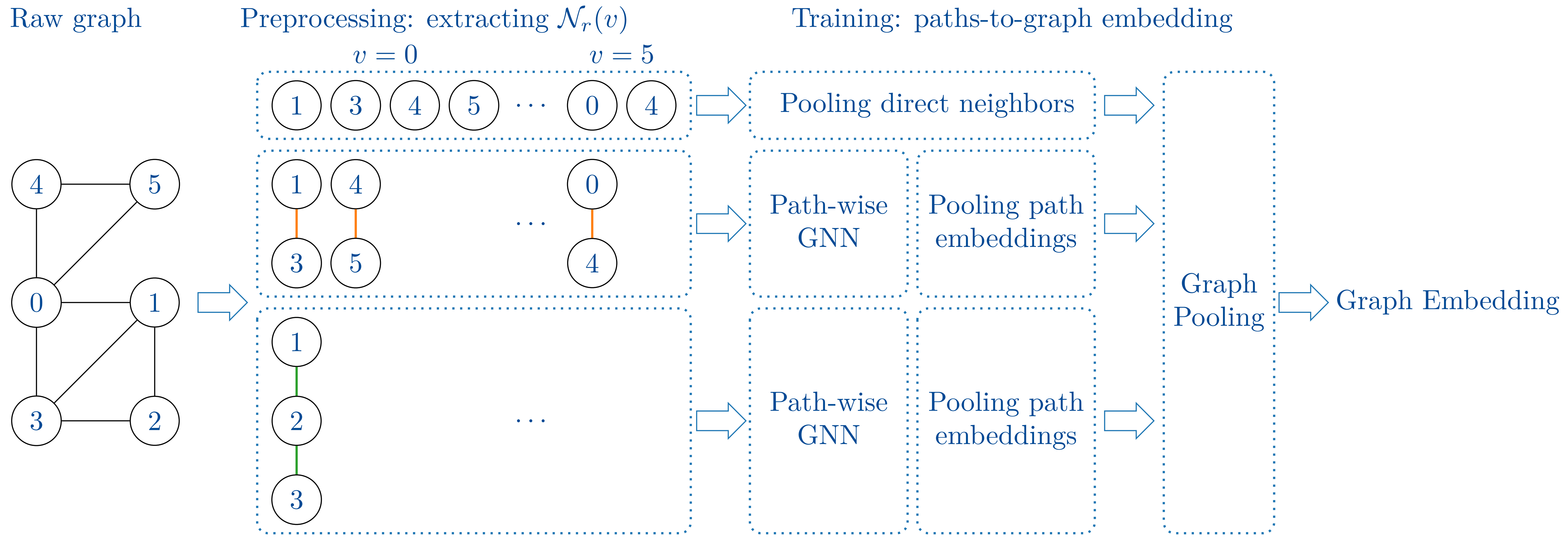
and there are no repeated nodes, i.e., $i \neq j \implies p_i \neq p_j$.

Definition. Given a graph G and an integer $r \geq 1$, we define the r-neighborhood $\mathcal{N}_r(v)$ of $v \in V(G)$ as the set of all simple paths of length r between distinct direct neighbors of v which do not contain v, i.e.,

$$\mathcal{N}_r(v) \coloneqq \{\mathbf{p} \mid \mathbf{p}\ r\text{-path},\ p_1, p_{r+1} \in \mathcal{N}(v), v \notin \mathbf{p}\}.$$



4. Loopy Weisfeiler Leman



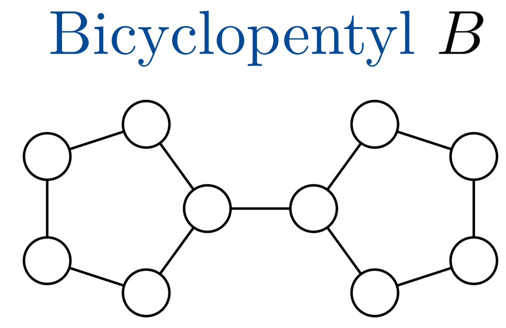
Theorem 1. Let $r \geq 1$, r-ℓWL can subgraph-count all cycles with at most $r+2$ nodes.

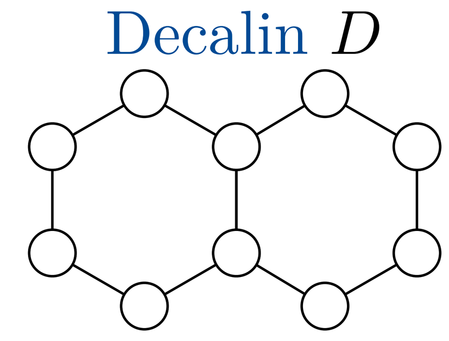
Intuitive idea of counting how many times a motif appears in the graph.

Theorem 2. Let $r \geq 1$. Then, r-ℓWL can homomorphism-count any graph in which every edge lies on at most one simple cycle of length at most r.

It is a complete measure: knowing the homomorphism count of each possible motif in a graph means knowing the graph itself

Bicyclopentyl B Decalin D





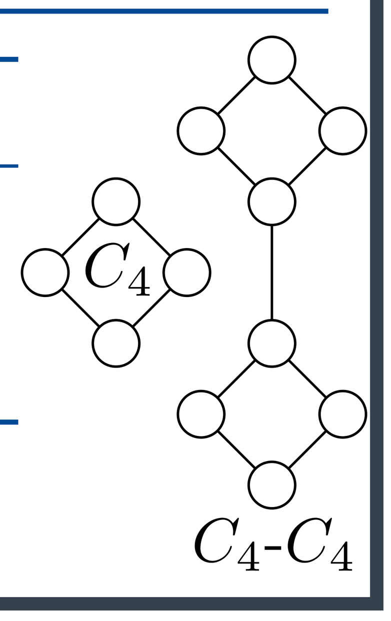
F										
B	10	22	50	118	114	78	262	270	290	20
D	10	22	50	118	114	78	262	270	290	0

5. Experiments

	Model	ZINC12K
1	5-ℓGIN	0.072 ± 0.002
2	DRFWL	0.077 ± 0.002
3	CIN	0.079 ± 0.006

	Model	ZINC250K
1	5-ℓGIN	0.022 ± 0.001
2	CIN	0.022 ± 0.002
3	I2-GNN	0.023 ± 0.001

Model	hom(C_4, G)	hom(C_4–C_4, G)
0-ℓGIN	$(2.48 \pm 0.01)\,10^{-1}$	$(1.14 \pm 0.01)\,10^{-1}$
1-ℓGIN	$(1.91 \pm 0.03)\,10^{-1}$	$(7.9 \pm 0.1)\,10^{-2}$
2-ℓGIN	$(2.56 \pm 0.49)\,10^{-4}$	$(1.8 \pm 0.6)\,10^{-2}$



6. Conclusions

GitHub

↪ Expressive:

- ▲ strictly more powerful than 1-WL;
- ▲ incomparable to k-WL and subgraph GNNs;
- ▲ more powerful than injecting subgraph-counts and homomorphism-counts as features;

↪ Scalable:

- ▲ preprocessing complexity $\mathcal{O}(N\, d^{r+2})$;
- ▲ linear complexity in the forward pass w.r.t. the number of edges and the number of paths in the r-neighborhoods;