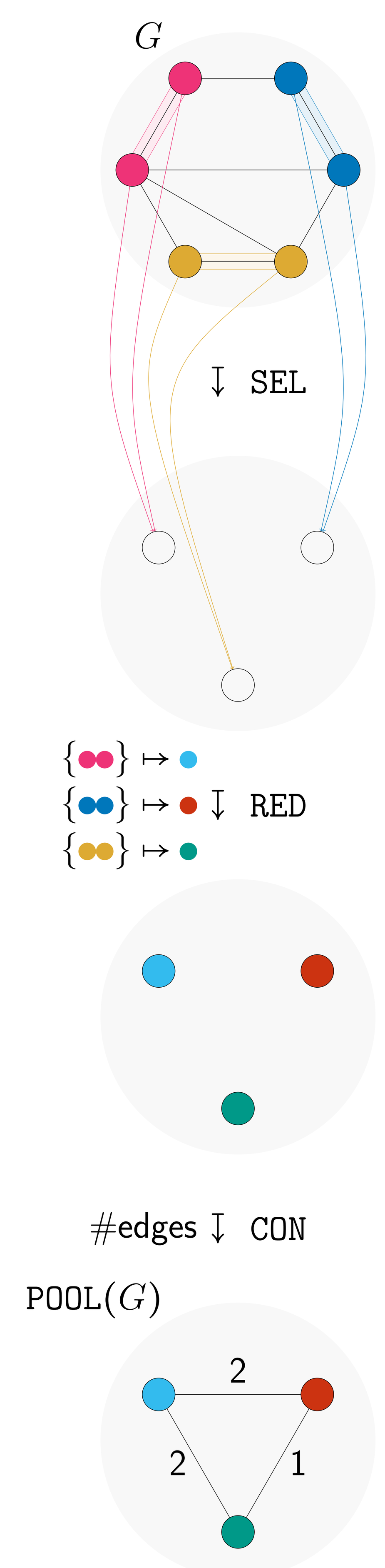
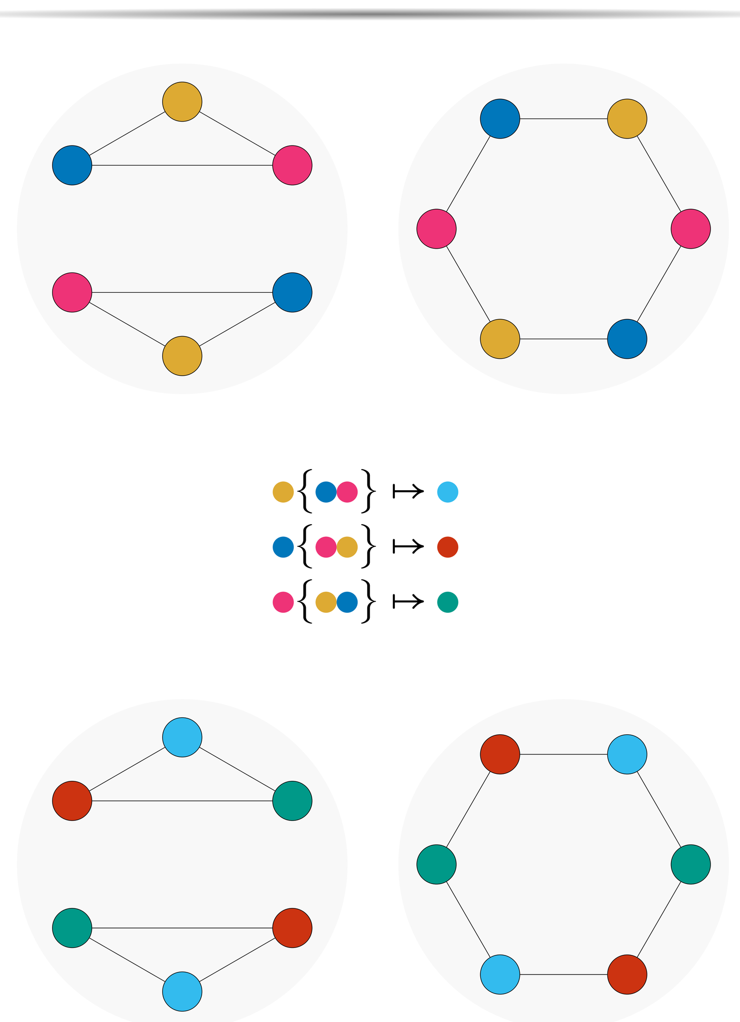
Graph Pooling Provably Improves Expressivity

Veronica Lachi, Alice Moallemy-Oureh, Andreas Roth, Pascal Welke
University of Siena, University of Kassel, TU Dortmund University, TU Wien
veronica.lachi@student.unisi.it, amoallemy@uni-kassel.de, andreas.roth@tu-dortmund.de, pascal.welke@tuwien.ac.at

Graph Pooling

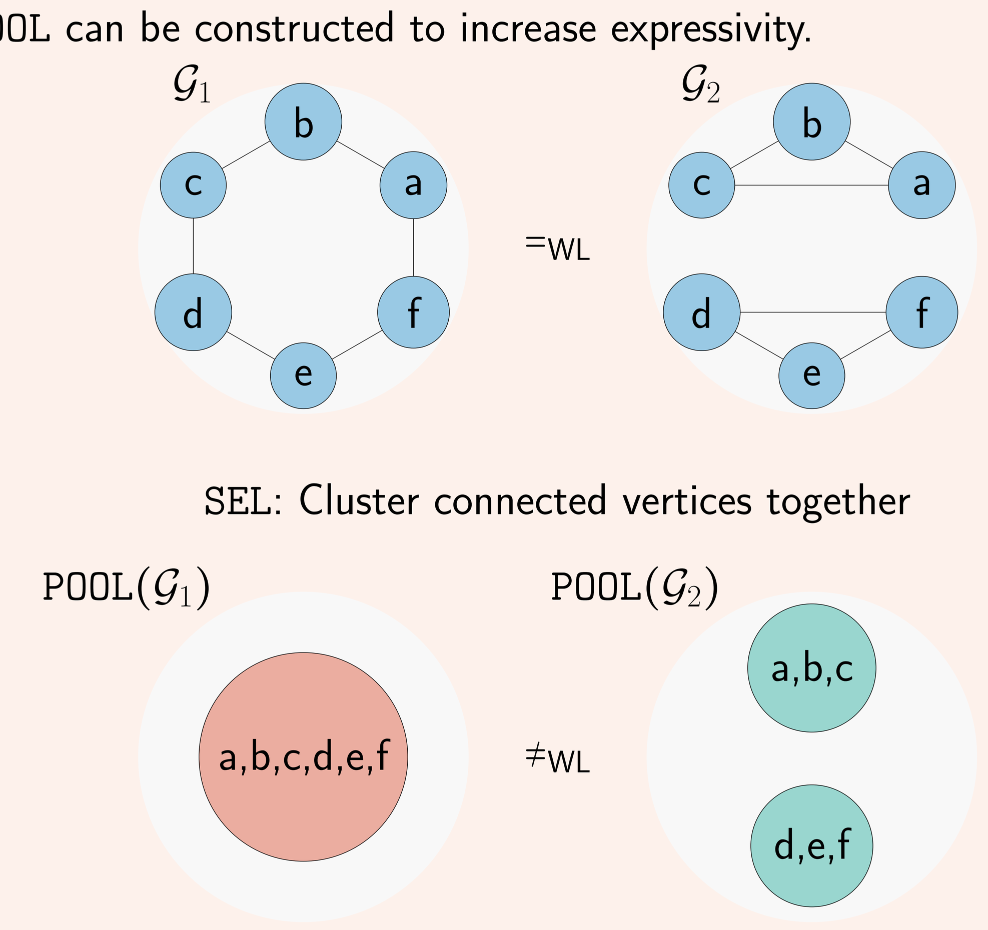


WL Test



Select can Increase Expressivity

Let SEL distinguish some graphs that WL does not distinguish. Then, POOL can be constructed to increase expressivity.



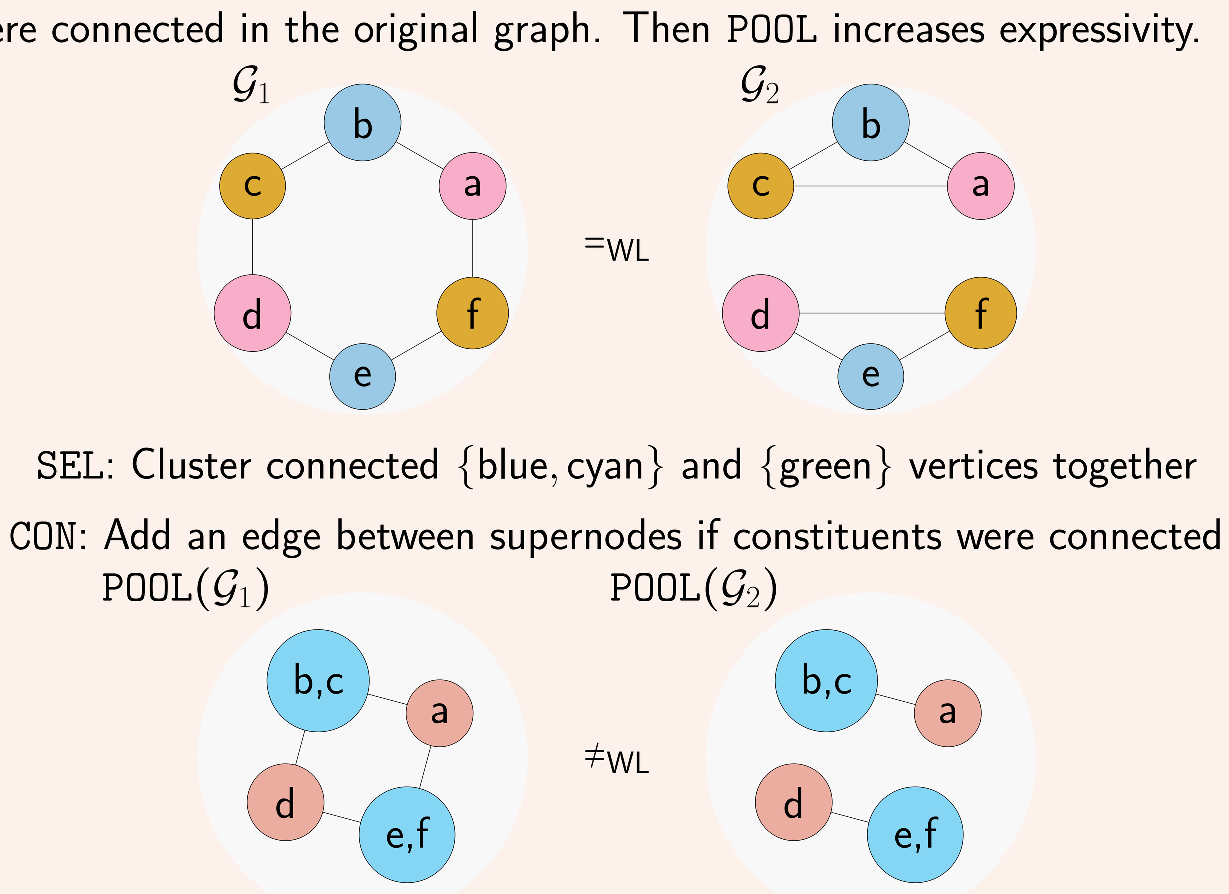
Reduce should be Injective

Let RED be an injective function. For any two graphs $\mathcal{G}_1, \mathcal{G}_2$ we have

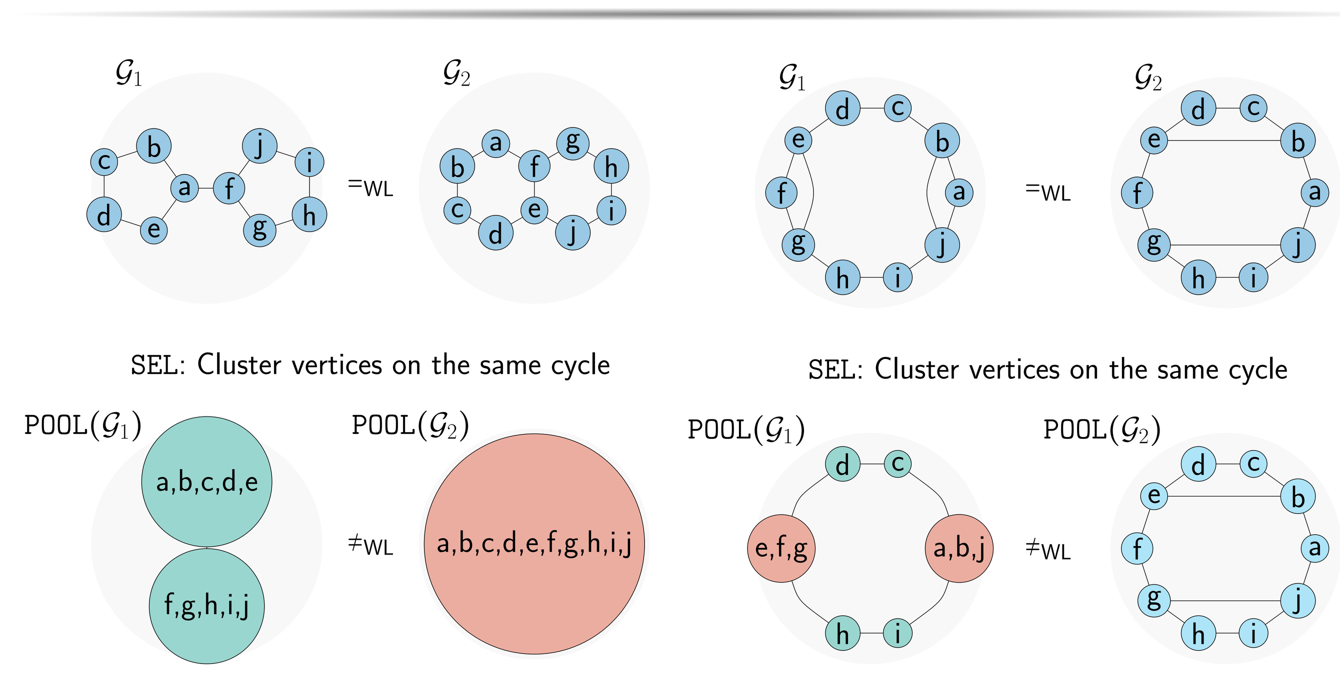
$$\mathrm{SEL}(\mathcal{G}_1) \neq \mathrm{SEL}(\mathcal{G}_2) \implies \mathrm{POOL}(\mathcal{G}_1) \neq_{\mathrm{WL}} \mathrm{POOL}(\mathcal{G}_2).$$

Connect can Increase Expressivity

Let CON create an edge between two supernodes if any pair of their nodes were connected in the original graph. Then POOL increases expressivity.



Alternative Select Functions



Motivation

- **Graph Pooling** is a powerful technique for reducing a graph's size, enabling GNNs to gradually learn more global information.
- Existing graph pooling methods are observed to either compromise or, at best, maintain the **expressivity** of GNNs, highlighting a critical challenge in the field.
- Are there graph pooling methods that not only preserve but actively **increase** the expressivity of GNNs?

Definitions

Pooling operator POOL
- maps a graph to a potentially smaller graph.
- is permutation invariant: $\mathcal{G}_1 \simeq \mathcal{G}_2 \Rightarrow \mathrm{POOL}(\mathcal{G}_1) \simeq \mathrm{POOL}(\mathcal{G}_2)$
- can be written as a triplet (SEL, RED, CON) of Select-Reduce-Connect functions.

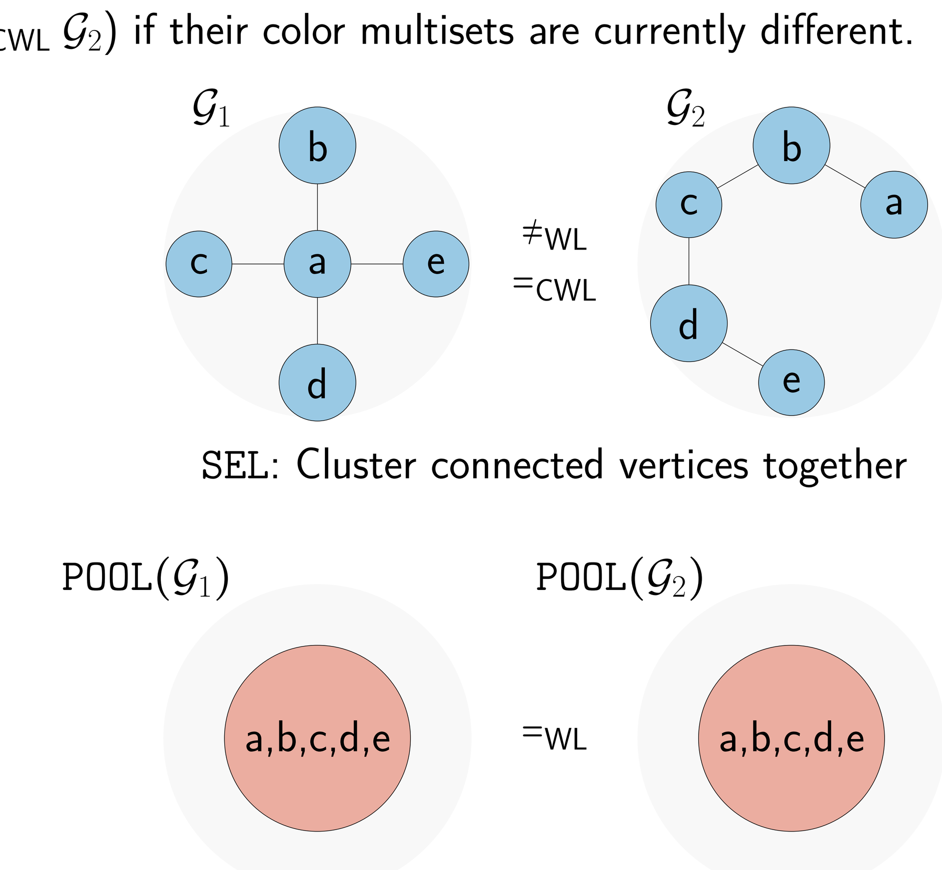
Select SEL: $\mathcal{G} \mapsto \mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ clusters the input graph nodes into so-called supernodes $\mathcal{S}_j = \{s_i^j\}_{i=1}^N$ where s_i^j indicates the contribution of node i on supernode j.

Reduce RED aggregates the features of the nodes assigned to the same supernode.

Connect CON generates the edges and edge features of the resulting graph POOL(G), if applicable, by connecting the supernodes.

Current Distinguishability Matters

- $\mathcal{G}_1$ and $\mathcal{G}_2$ are **WL distinguishable** ($\mathcal{G}_1 \neq_{\mathrm{WL}} \mathcal{G}_2$) if there exists an iteration t for which $\{c_n^{(t)} : n \in \mathcal{V}_1\} \neq \{c_n^{(t)} : n \in \mathcal{V}_2\}$.
- Two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are **currently WL distinguishable** ($\mathcal{G}_1 \neq_{\mathrm{CWL}} \mathcal{G}_2$) if their color multisets are currently different.



Expressive Pooling Operators

A pooling operator POOL = (SEL, RED, CON)

- **Maintains Expressivity**
  - if it maps any pair of currently WL-distinguishable graphs to a pair of WL-distinguishable graphs:
    $$\mathcal{G}_1 \neq_{CWL} \mathcal{G}_2 \Rightarrow \mathrm{POOL}(\mathcal{G}_1) \neq_{WL} \mathrm{POOL}(\mathcal{G}_2)\ .$$
- **Increases Expressivity**
  - if it maintains expressivity
  - if there is a pair of graphs that are WL indistinguishable which become WL-distinguishable after pooling:
    $$\text{there exist } \mathcal{G}_1 =_{WL} \mathcal{G}_2 \text{ with } \mathrm{POOL}(\mathcal{G}_1) \neq_{WL} \mathrm{POOL}(\mathcal{G}_2)\ .$$

Conclusion

- Strategic pool assignment and topology-aware edge pruning lead to reduced graph size while enhancing GNN expressivity.
- Our findings establish a theoretical basis for existing methods, providing practical guidance for designing more expressive GNNs with hierarchical pooling operators.

Future Work

- Measuring the actual gains in expressivity beyond general improvement to provide a more accurate understanding of enhancements.
- Applying our insights to modify existing pooling methods and devise novel approaches, aiming for improved predictive performance.
- Evaluating the methods thoroughly on both synthetic and real datasets to ensure a comprehensive assessment of their effectiveness.