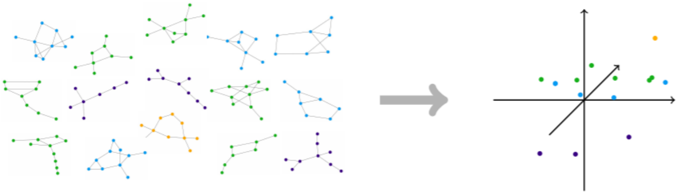
Graph Neural Networks with Local Graph Parameters

Pablo Barcelo*, Floris Geerts**, Juan L. Reutter*, Maksimilian Ryschkov**

*Pontificia Universidad Católica de Chile, Santiago, Chile; **University of Antwerp, Belgium

Deep learning on graph data: Graph Neural Networks

GNNs generate vertex embeddings $\mathbf{h}_u$ for every $u \in V$ which fit the graph learning task.
graph embeddings for every graph



We focus on MPNNs **(=Message Passing Neural Networks)**

MPNNs: Idea

Given a labeled graph $G = (V, E, \chi)$

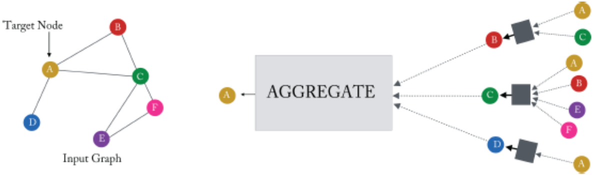
1. Start with an initial vertex embedding for all $u \in V$
2. Iteratively update the embeddings using the neighborhood information.

Two types of information get passed:

- Structural information
- Feature information

$$h_u^0 = \text{ initial label } \chi_u$$

$$h_u^{(l+1)} = \texttt{Update}^{(l)}\left(h_u^{(l)}, \texttt{Aggregate}\left(\{h_v^{(l)}, \forall v \in \mathcal{N}(u)\}\right)\right)$$



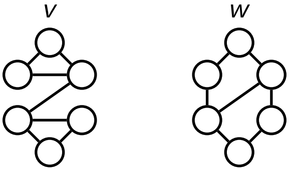
MPNNs: Strengths and limitations

Strengths

- Efficient
- Number of model parameters independent of graph size: generalizes to graphs of any size
- Perform well on variety of graph tasks

Limitations: MPNNs cannot learn functions that depend on:

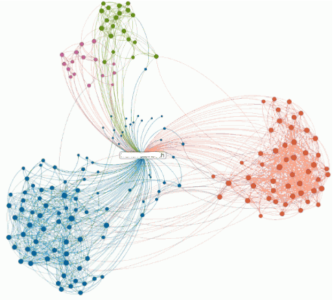
- The number of connected components in a graph
- Presence of cliques, cycles and other not-tree like structures



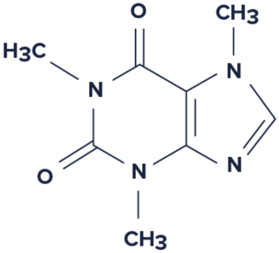
A need for more powerful MPNNs

Graph patterns are important indicators in graph data.

In social networks, cliques indicate communities.



In molecules, cycles can indicate chemical properties.



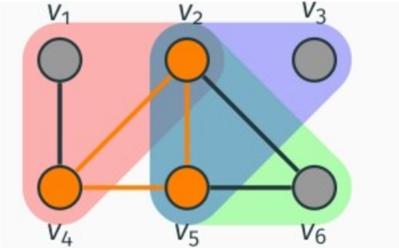
Possible approaches:

1. Higher-order GNNs
2. GNNs with extended features ← **our contribution**

1. Higher-order GNNs

Instead of updating vertex embeddings, embeddings of k-tuples of vertices are iteratively computed.

- Isomorphism types of subgraphs induced by k-tuples are included in the initial labels
- Patterns up to treewidth k can be detected
- Inefficient: require at least quadratic number of features



2. GNNs with extended features

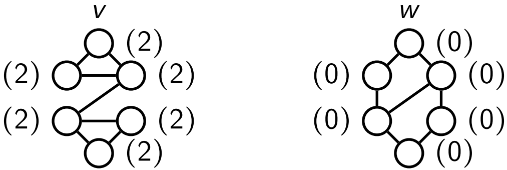
Idea: Adding features containing information that MPNNs cannot learn.

We propose: $\mathcal{F}$-MPNN**s** $=$ a type of Graph Neural Networks infused with **local higher-order graph structure information**

- Efficiency on par with MPNNs.
- Stronger in expressive power than MPNNs.

$\mathcal{F}$ – MPNNs

1. Select a pattern set $\mathcal{F} = \{P_1^r, \ldots, P_l^r\}$. E.g. $\mathcal{F} = \{$, , , $\}$
2. Count the local occurrences of these patterns v: $\hom(P^r, G^v)$



Example: $\mathcal{F} = \{$ $\}$

Why homomorphisms?

- Easier to compute than subgraph isomorphisms
- Homomorphisms counts underly the expressive power of MPNNs.
- Theoretically interchangeable with isomorphism counts

3. Extend the inital vertex labels with these additional features.

$$h_u^0 = (\text{ initial label } \chi_u, \hom(P_1^r, G^v), \ldots, \hom(P_l^r, G^v))$$

4. Apply a MPNN model on these new, extended, vertex labels.

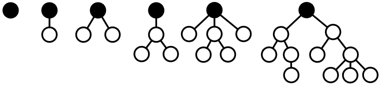
$$h_u^{(l+1)} = \texttt{Update}^{(l)}\left(h_u^{(l)}, \texttt{Aggregate}\left(\{h_v^{(l)}, \forall v \in \mathcal{N}(u)\}\right)\right)$$

Expressive Power of $\mathcal{F}$-MPNNs

In graphs G and H, vertices v and w are indistinguishable (embedded in the same way) if:

MPNNs	
Homomorphism counts are equal for every **rooted tree** (Dell et al, 2018).	

with:

Rooted trees S^r: a graph without cycles and a designated root 	

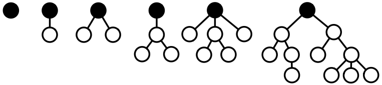
Expressive Power of $\mathcal{F}$-MPNNs

In graphs G and H, vertices v and w are indistinguishable (embedded in the same way) if:

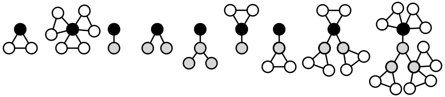
MPNNs	$\mathcal{F}$-MPNNs
Homomorphism counts are equal for every **rooted tree** (Dell et al., 2018).	Homomorphism counts are equal for every $\mathcal{F}$-**pattern tree** (our contribution).

with:

Rooted trees S^r: a graph without cycles and a designated root



$\mathcal{F}$*-pattern tree* T^r: backbone tree S^r with vertices $s \in V_S$ joined with copies of patterns in $\mathcal{F}$



Comparison with higher-order GNNs

1. **Highest treewidth of patterns in $\mathcal{F} \leq k$:**
   $\mathcal{F}$-MPNNs cannot distinguish any pair of graphs indistinguishable by kth-order GNNs

   Example: $\{K_3, K_4\}$-MPNNs cannot distinguish any pair of graphs indistinguishable by 3th-order GNNs

2. **Highest treewidth of patterns in $\mathcal{F} > k$:**
   There exists a pair of graphs indistinguishable by kth-order GNNs that some $\mathcal{F}$-MPNNs can distinguish

   Example: There exists a pair of graphs indistinguishable by 3th-order GNNs that some $\{K_3, K_4, K_5\}$-MPNN can distinguish

Pattern choice

- The choice of patterns in $\mathcal{F}$: important & application-dependent.
- We prove several results offering possible choices for patterns.

  Examples:

  - $\{K_3, \ldots, K_k\}$-MPNN is more expressive than $\{K_3, \ldots, K_{k-1}\}$-MPNN for any $k > 3$
  - $\{C_3, \ldots, C_{2k-1}, C_{2k+1}\}$-MPNN is more expressive than $\{C_3, \ldots, C_{2k-1}\}$-MPNN for any $k > 3$

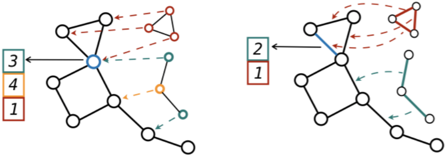
Experiments: Methodology

We use the benchmark study for GNNs by Dwivedi et al. (2020).

Datasets	ZINC	PATTERN	COLLAB
Learning Tasks	Graph Regression	Node Classification	Link Prediction
Pattern sets	$\{C_l \mid 3 \leq k \leq 10\}$	$\{K_l \mid 3 \leq k \leq 5\}$	$\{K_l \mid 3 \leq k \leq 5\}$

Comparison to:

- Baseline models with same parameters
- For ZINC: the similar GSN (Bouritsas et al, 2020) approach, where isomorphisms instead of homomorphisms are computed:



Experiments: Some Results

GAT on ZINC with varying $\mathcal{F}$

$\mathcal{F}$	MAE
None	0.47±0.02
$\{C_3\}$	0.45±0.01
$\{C_4\}$	0.34±0.02
$\{C_6\}$	0.31±0.01
$\{C_5, C_6\}$	0.28±0.01
$\{C_3 \ldots C_6\}$	0.23±0.01
$\{C_3 \ldots C_{10}\}$	**0.22±0.01**

Results on the ZINC dataset with $\mathcal{F} = \{C_l \mid 3 \leq k \leq 10\}$

Dataset	ZINC		
Model	MAE (base)	MAE (hom)	MAE (iso)
GAT	0.47±0.02	0.22±0.01	0.24±0.01
GCN	0.35±0.01	0.20±0.01	0.22±0.01
GraphSage	0.44±0.01	0.24±0.01	0.22±0.01
MoNet	0.25±0.01	0.19±0.01	0.16±0.01
GatedGCN	0.34±0.05	**0.14±0.01**	0.14±0.01

Experiments: More Results

Results on the COLLAB and PATTERN datasets

Dataset	COLLAB		PATTERN	
Model	Hits@50 (base)	Hits@50 (hom)	Accuracy (base)	Accuracy (hom)
GAT	50.32±0.55	52.87±0.87	78.83 ± 0.60	85.50±0.23
GCN	51.35±1.30	**54.60±1.01**	71.42 ± 1.38	82.49 ± 0.48
GraphSage	50.33±0.68	51.39±1.23	70.78 ± 0.19	85.85 ± 0.15
MoNet	49.81±1.56	51.76±1.38	85.90 ± 0.03	**86.63 ± 0.03**
GatedGCN	51.00 ± 2.54	51.57 ± 0.68	86.15±0.08	85.56±0.33

Conclusions

- $\mathcal{F}$-MPNN beat MPNNs in expressive power and are more efficient than higher-order GNNs

- Adding patterns to MPNNs is a low-cost strategy for improving the learning power of MPNNs.

- Pattern choice is important, but simple sets of cliques or cycles are shown to work.

- Experimental study shows that the performance of various MPNN models is enhanced by additional structural vertex features