ICML
International Conference
On Machine Learning

Breaking the Limits of Message Passing Graph Neural Networks

Muhammet Balcilar, Pierre Héroux, Benoit Gaüzère, Pascal Vasseur,
Sébastien Adam, Paul Honeine

UNIVERSITÉ DE ROUEN NORMANDIE

litis

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

interdigital™

UNIVERSITÉ de Picardie Jules Verne

Message Passing (Graph) Neural Network

- In our previous research[1], we generalize spatial and spectral GNN by

$$H^{(l+1)} = \sigma\Big(\sum_s C^{(s)} H^{(l)} W^{(l,s)}\Big),$$

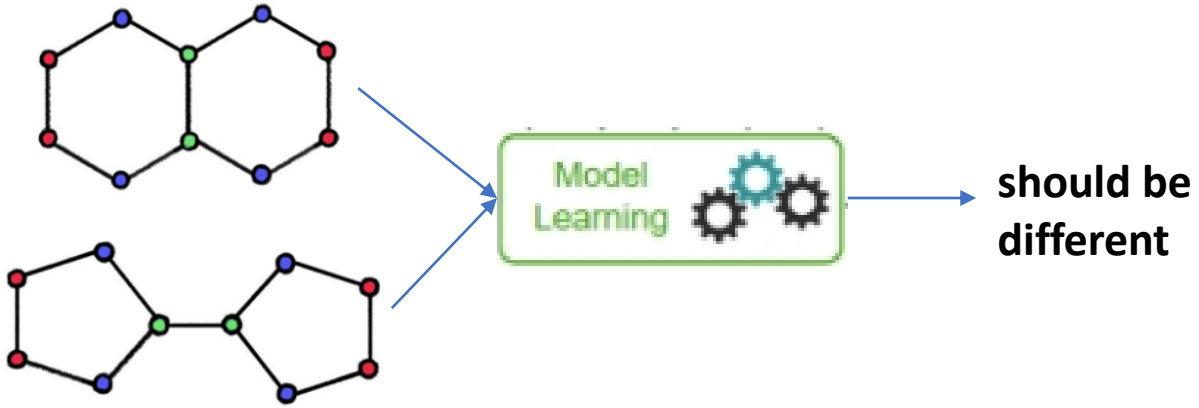
Convolution Support — Node Features — Trainable Parameters

- Spatial Methods are defined by C matrices
- Spectral Method defined by $B_{i,j} = \Phi_j(\lambda_i)$
- Where transition can be written by $C^{(s)} = U\,\mathrm{diag}(\Phi_s(\boldsymbol{\lambda}))U^\top$.

[1] Balcilar et al. Analyzing the expressive power of graph neural networks in a spectral perspective. ICLR2021.

Expressive Power of GNN

- Universality of the GNN depends on
  - ability to produce different output for non-isomorphic graphs.

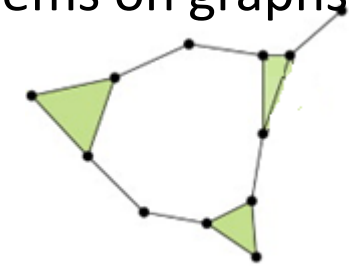


- 1-WL=2-WL <3-WL<4-WL<.....<k-WL
- We can classify GNN by equivalence of WL test order
- k>2, k-WL GNN needs
  - O(n^(k-1)) memory, O(n^k) CPU time

1-WL GNN (MPNN) versus k-WL GNN

- Pros:
  - Linear memory&time complexity.
  - Local update schema.
  - Natural problems consist of graphs can be distinguishable by 1-WL.
  - Their results are still competitive!

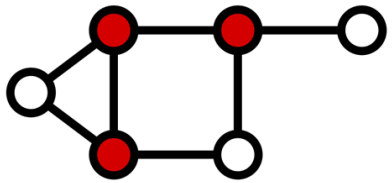
- Cons:
  - Maps 1-WL equivalent graphs to the exact the same point on latent space.
  - Cannot count some substructures that is informative many graph problems.
  - Cannot solve many combinatorial problems on graphs that may needed.



Counting triangles

- Pros:
  - Can distinguish up to k-WL equivalent graphs.
  - Can count some substructures related to k.
  - Can solve some combinatorial problems.

- Cons:
  - O(n^(k-1)) memory, O(n^k) CPU time
  - Non-local update schema.
  - Unable to learn frequency relations.
  - Their results are not better than 1-WL GNN on many realistic problems.



Minimum vertex covering

How to Increase the Expressive Power of MPNN

- Methods need massive data augmentation, they diverge slow.
  - Add random noise to the nodes as extra node feature.
  - Add unique identifier to the nodes as extra node feature.

- Methods need feature engineering.
  - Add features that cannot be obtained by MPNN as extra node feature.
  - Weight sharing w.r.t some predefined substructures.

Characterization of WL Test with MATLANG

- Recently, the connection between Matrix Language and WL-test was found.

Definition 1. *$ML(\mathcal{L})$ is a matrix language with an allowed operation set $\mathcal{L} = \{op_1, \dots op_n\}$, where $op_i \in \{., +, ^\top, diag, tr, \mathbf{1}, \odot, \times, f\}$. The possible operations are matrices multiplication and addition, matrix transpose, vector diagonalization, matrix trace computation, column vector full of 1, element-wise matrix multiplication, matrix/scalar multiplication and element-wise custom function operating on scalars or vectors.*

Definition 2. *$e(X) \in \mathbb{R}$ is a sentence in $ML(\mathcal{L})$ if it consists of any possible consecutive operations in $\mathcal{L}$, operating on a given matrix X and resulting in a scalar value.*

As an example, $e(X) = \mathbf{1}^\top X^2 \mathbf{1}$ is a sentence of $ML(\mathcal{L})$ with $\mathcal{L} = \{., ^\top, \mathbf{1}\}$, computing the sum of all elements of square matrix X.

Characterization of WL Test with MATLANG

- Three different Matrix Language and their connection to the WL test are given by:

Remark 1. *Two adjacency matrices are indistinguishable by the 1-WL test if and only if* $e(A_G) = e(A_H)$ *for all* $e \in \mathcal{L}_1$ *with* $\mathcal{L}_1 = \{., ^\top, \mathbf{1}, diag\}$. *Hence, all possible sentences in* $\mathcal{L}_1$ *are the same for 1-WL equivalent adjacency matrices. Thus,* $A_G \equiv_{1-WL} A_H \leftrightarrow A_G \equiv_{ML(\mathcal{L}_1)} A_H$. *(see Theorem 7.1 in (Geerts, 2020))*

Remark 2. $ML(\mathcal{L}_2)$ *with* $\mathcal{L}_2 = \{., ^\top, \mathbf{1}, diag, tr\}$ *is strictly more powerful than* $\mathcal{L}_1$, *i.e., than the 1-WL test, but less powerful than the 3-WL test. (see Theorem 7.2 and Example 7.3 in (Geerts, 2020))*

Remark 3. *Two adjacency matrices are indistinguishable by the 3-WL test if and only if they are indistinguishable by any sentence in* $ML(\mathcal{L}_3)$ *with* $\mathcal{L}_3 = \{., ^\top, \mathbf{1}, diag, tr, \odot\}$. *Thus,* $A_G \equiv_{3-WL} A_H \leftrightarrow A_G \equiv_{ML(\mathcal{L}_3)} A_H$. *(see Theorem 9.2 in (Geerts, 2020))*

Remark 4. *Enriching the operation set to* $\mathcal{L}^+ = \mathcal{L} \cup \{+, \times, f\}$ *where* $\mathcal{L} \in (\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3)$ *does not improve the expressive power of the language. Thus,* $A_G \equiv_{ML(\mathcal{L})} A_H \leftrightarrow A_G \equiv_{ML(\mathcal{L}^+)} A_H$. *(see Proposition 7.5 in (Geerts, 2020))*

Theorem 1. *MPNNs such as GCN, GAT, GraphSage, GIN (defined in Appendix H) cannot go further than operations in $\mathcal{L}_1^+$. Thus, they are not more powerful than the 1-WL test.*

Theorem 2. *Chebnet is more powerful than the 1-WL test if the Laplacian maximum eigenvalues of the non-regular graphs to be compared are not the same. Otherwise Chebnet is not more powerful than 1-WL.*

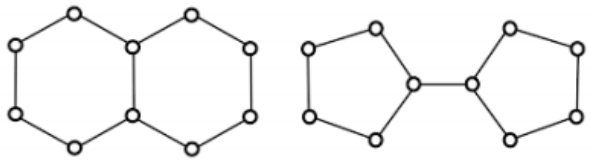


Figure 1. Decalin (G) and Bicyclopentyl (H) graphs are $\mathcal{L}_1$ and also 1-WL equivalent, but Chebnet can distinguish them.

How Powerful are MPNNs?

- Using connection between MATLANG and WL test, we proved these theorems.

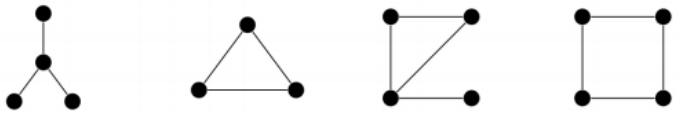


Figure 2. Sample of patterns: 3-star, triangle, tailed triangle and 4-cycle graphlets used in our analysis.

Theorem 3. *3-star graphlets can be counted by sentences in $\mathcal{L}_1^+$.*

Theorem 4. *Triangle and 4-cycle graphlets can be counted by sentences in $\mathcal{L}_2^+$.*

Theorem 5. *Tailed triangle graphlets can be counted by sentences in $\mathcal{L}_3^+$.*

New 1-WL MPNN with MATLANG

- Any GNN which can produce all sentences in $\mathcal{L}_1 = \{., ^\top, \mathbf{1}, diag\}$ have exact the same power of 1-WL test.
- GNNML1:

$$H^{(l+1)} = \sigma\left(H^{(l)} W^{(l,1)} + A H^{(l)} W^{(l,2)} + H^{(l)} W^{(l,3)} \odot H^{(l)} W^{(l,4)} \right)$$

Theorem 6. *GNNML1 can produce every possible sentences in $ML(\mathcal{L}_1)$ for undirected graph adjacency A with monochromatic edges and nodes. Thus, GNNML1 is exactly as powerful as the 1-WL test.*

Beyond 1-WL MPNN with MATLANG

- Any GNN which can produce all sentences in

$$\mathcal{L}_3 = \{., ^\top, \mathbf{1}, diag, tr, \odot\}$$

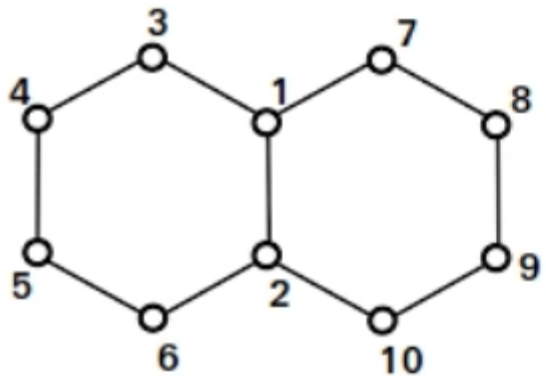
have exact the same power of 3-WL test.

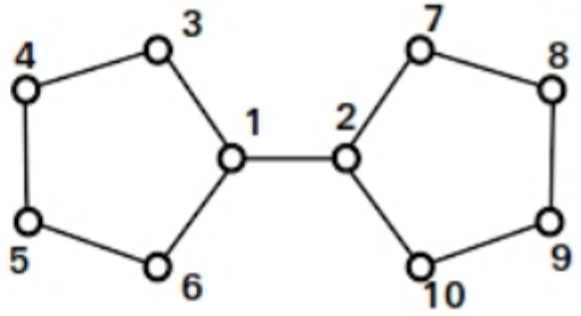
- If we add ability to calculate

$$\{tr, \odot\}$$

on to GNNML1, we can go beyond 1-WL.

How Trace operator helps?

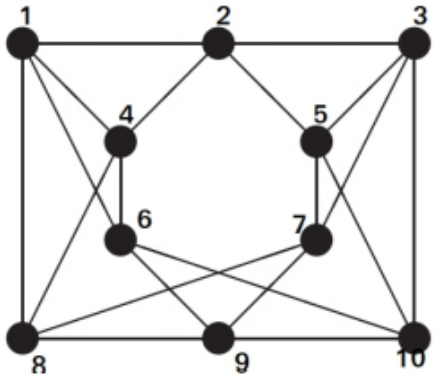


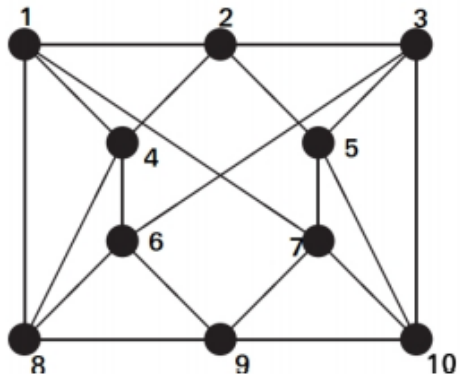


$$A_G = \begin{pmatrix} 0&1&1&0&0&0&1&0&0&0 \\ 1&0&0&0&0&1&0&0&0&1 \\ 1&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0 \\ 0&1&0&0&1&0&0&0&0&0 \\ 1&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&0&0&1&0 \end{pmatrix} \text{ and } A_H = \begin{pmatrix} 0&1&1&0&0&1&0&0&0&0 \\ 1&0&0&0&0&0&1&0&0&1 \\ 1&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&0&0&1&0 \end{pmatrix}$$

$$tr(A_G^5) = 0 \qquad tr(A_H^5) = 20$$

How Elementwise Martix Mul operator helps?





$$A_G = \begin{pmatrix} 0&1&0&1&0&1&0&1&0&0 \\ 1&0&1&1&1&0&0&0&0&0 \\ 0&1&0&0&1&0&1&0&0&1 \\ 1&1&0&0&0&1&0&1&0&0 \\ 0&1&1&0&0&0&1&0&0&1 \\ 1&0&0&1&0&0&0&0&1&1 \\ 0&0&1&0&1&0&0&1&1&0 \\ 1&0&0&1&0&0&1&0&1&0 \\ 0&0&0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&0&0&1&0 \end{pmatrix} \text{ and } A_H = \begin{pmatrix} 0&1&0&1&0&0&1&1&0&0 \\ 1&0&1&1&1&0&0&0&0&0 \\ 0&1&0&0&1&1&0&0&0&1 \\ 1&1&0&0&0&1&0&1&0&0 \\ 0&1&1&0&0&0&1&0&0&1 \\ 0&0&1&1&0&0&0&1&1&0 \\ 1&0&0&0&1&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&1&0 \\ 0&0&0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&0&1&0&1&0 \end{pmatrix}$$

$$tr(A_G^2) = tr(A_H^2) = 40,$$
$$tr(A_G^3) = tr(A_H^3) = 48$$
$$tr(A_G^4) = tr(A_H^4) = 360,$$
$$tr(A_G^5) = tr(A_H^5) = 920$$

$$e(X) = \mathbf{1}^\top((X \odot X^2)^2\mathbf{1})^2$$

$$\mathbf{1}^\top((A_G \odot A_G^2)^2\mathbf{1})^2 = 6032 \qquad \mathbf{1}^\top((A_H \odot A_H^2)^2\mathbf{1})^2 = 5872$$

Beyond 1-WL MPNN with MATLANG

- Any GNN which can produce all sentences in

$$\mathcal{L}_3 = \{.,^\top, \mathbf{1}, diag, tr, \odot\}$$

have exact the same power of 3-WL test.

- If we add ability to calculate $\{tr, \odot\}$

on GNNML1, we can go beyond 1-WL.

- However, MPNN does not keep power of adjacencies explicitly, thus cannot have its trace or elementwise multiplication.
- For instance, MPNN can have $C^3 1$, with 3 layers of network by C(C(C1))

Beyond 1-WL MPNN with MATLANG

- Our solution is to design graph convolution supports which can be written by power series of graph adjacency (or graph laplacien).
- We have S+1 number of predefined initial graph convolution matrix in preprocessing step such as;

$$C'^{(0)} = I$$

$$C'^{(1)} = M \odot Udiag(\Phi_1(\lambda))U^\top$$

$$\dots$$

$$C'^{(S)} = M \odot Udiag(\Phi_S(\lambda))U^\top$$

Theorem 7. *A convolution support given by*

$$C'^{(s)} = Udiag(\Phi_s(\lambda))U^\top, \tag{3}$$

where $\Phi_s(\lambda) = exp(-b(\lambda - f_s)^2)$, $f_s \in [\lambda_{min}, \lambda_{max}]$ *is a scalar design parameter of each convolution support and* $b > 0$ *is a general scalar design parameter, can be expressed as a linear combination of all powers of graph Laplacian (or adjacency) as follows, with* $\alpha_{s,i} = \frac{\Phi_s^{(i)}(0)}{i!}$*:*

$$C'^{(s)} = \alpha_{s,0}L^0 + \alpha_{s,1}L^1 + \alpha_{s,2}L^2 + \dots. \tag{4}$$

We used fixed 1-hop receptive field
M=A+I

Beyond 1-WL MPNN with MATLANG

- We can learn necessary power of convolution support by MLP as in;

$$C = mlp_4\left(mlp_1(C')|mlp_2(C') \odot mlp_3(C')\right)$$

Initial sparse convolution supports

$$C' = [C'^{(1)}|\ldots|C'^{(s)}] \in \mathbb{R}^{n\times n\times S}$$

Learned sparse convolution supports

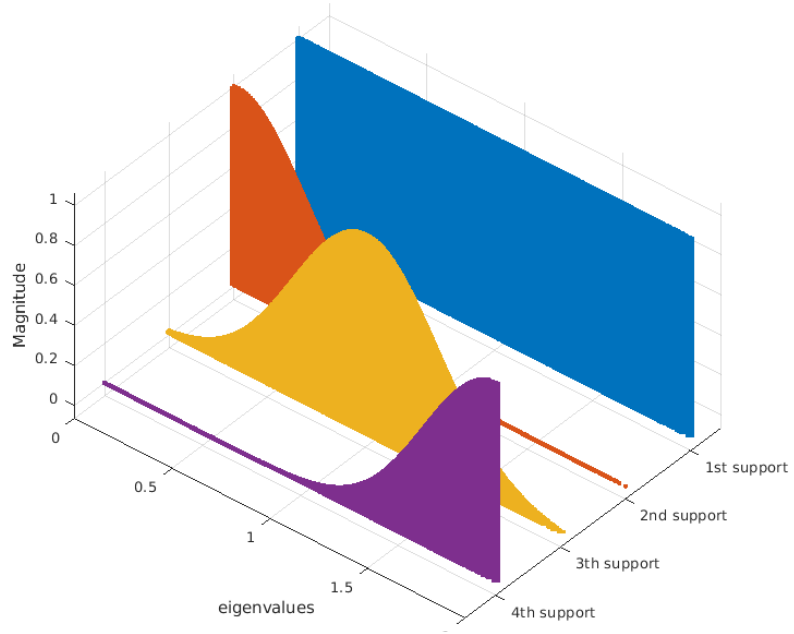
$$C = [C^{(1)}|\ldots|C^{(S)}] \in \mathbb{R}^{n\times n\times S}$$

- Then define GNNML3 forward calculations as;

$$H^{(l+1)} = \sigma\Big(\sum_s (C^{(s)}H^{(l)}W^{(l,s)})|mlp_5(H^{(l)})\odot mlp_6(H^{(l)})\Big)$$

Pros and Cons of GNNML3

- Pros
  - Except pre-procesing step, it needs linear time&memory complexity.
  - Graph convolution supports are aware of frequency of signal on graph.
  - Since it can produce elementwise mul and trace of necessary power of adjacency, it is theoretically more powerful than 1-WL, experimentally equal to 3-WL.
  - Because of receptive field mask, it has local update schema.
- Cons
  - Needs eigendecomposition in preprocessing step.
  - Needs predefined frequency responses of graph convolution.



$\Phi_1(\lambda) = 1$

$\Phi_2(\lambda) = exp(-4(\lambda - 0)^2)$

$\Phi_3(\lambda) = exp(-4(\lambda - 1)^2)$

$\Phi_4(\lambda) = exp(-4(\lambda - 2)^2)$

Results

- Can the models generalize the counting of some substructures in a given graph?

Table 2. Median of test set MSE error for graphlet counting problem on random graph dataset over 10 random runs.

MODEL	3-STARS	CUSTOM	TRIANGLE	TAILED-TRI	4-CYCLES
MLP	**1.0E-4**	4.58E-1	3.13E-1	2.22E-1	1.73E-1
GCN	**1.0E-4**	3.22E-3	2.43E-1	1.42E-1	1.14E-1
GAT	**1.0E-4**	4.57E-3	2.47E-1	1.44E-1	1.12E-1
GIN	**1.0E-4**	1.47E-3	2.06E-1	1.18E-1	1.21E-1
CHEBNET	**1.0E-4**	**7.68E-4**	2.01E-1	1.15E-1	9.60E-2
PPGN	**1.0E-4**	**9.19E-4**	**1.00E-4**	**2.61E-4**	**3.30E-4**
GNNML1	**1.0E-4**	**2.75E-4**	2.45E-1	1.32E-1	1.14E-1
GNNML3	**1.0E-4**	**7.24E-4**	**4.44E-4**	**3.18E-4**	**6.62E-4**

- How many pairs of non-isomorphic simple graphs that are either 1-WL or 3-WL equivalent are not distinguished by the models?

Table 1. Number of undistinguished pairs of graphs in graph8c, sr25 and EXP datasets and binary classification accuracy on EXP dataset. An ideal method does not find any pair similar and classifies graphs with 100% accuracy. The number of pairs is 61M for graph8c, 105 pairs for sr25 and 600 for EXP.

MODEL	GRAPH8C	SR25	EXP	EXP-CLASSIFY
MLP	293K	105	600	50%
GCN	4775	105	600	50%
GAT	1828	105	600	50%
GIN	386	105	600	50%
CHEBNET	44	105	71	82%
PPGN	**0**	105	**0**	**100%**
GNNML1	333	105	600	50%
GNNML3	**0**	105	**0**	**100%**

Results

- Can the models learn low-pass, high-pass and band-pass filtering effects and generalize the classification problem according to the frequency of the signal?

Table 3. Spectral expressive analysis results. R^2 for LowPass, HighPass and BandPass node regression tasks, accuracy on graph classification task. Results are median of 10 different runs.

MODEL	LOW-PASS	HIGH-PASS	BAND-PASS	CLASSIFY
MLP	0.9749	0.0167	0.0027	50.0%
GCN	0.9858	0.0863	0.0051	77.9%
GAT	0.9811	0.0879	0.0044	85.3%
GIN	0.9824	0.2934	0.0629	87.6%
CHEBNET	**0.9995**	**0.9901**	**0.8217**	**98.2%**
PPGN	**0.9991**	**0.9925**	0.1041	91.2%
GNNML1	**0.9994**	**0.9833**	0.3802	92.8%
GNNML3	**0.9995**	**0.9909**	**0.8189**	**97.8%**

- Can the models generalize downstream graph classification and regression tasks?

Table 4. Results on Zinc12K and MNIST-75 datasets. The values are the MSE for Zinc12K and the accuracy for MNIST-75. Edge features are not used even if they are available in the datasets. For Zinc12K, all models use node labels. For MNIST-75, the model uses superpixel intensive values and node degree as node features.

MODEL	ZINC12K	MNIST-75
MLP	0.5869 ± 0.025	$25.10\% \pm 0.12$
GCN	0.3322 ± 0.010	$52.80\% \pm 0.31$
GAT	0.3977 ± 0.007	$82.73\% \pm 0.21$
GIN	0.3044 ± 0.010	$75.23\% \pm 0.41$
CHEBNET	0.3569 ± 0.012	$\mathbf{92.08\% \pm 0.22}$
PPGN	$\mathbf{0.1589 \pm 0.007}$	$90.04\% \pm 0.54$
GNNML1	0.3140 ± 0.015	$84.21\% \pm 1.75$
GNNML3	$\mathbf{0.1612 \pm 0.006}$	$\mathbf{91.98\% \pm 0.18}$

Conclusion

- Except preprocessing step, we reach 3-WL expressive power with MPNN.
- GNNML3 is as good as spectral graph convolution on problem depends on graph signal frequency.
- GNNML3 is as good as 3-WL equivalent GNN on problems depends on graph substructure counting.
- GNNML3 provides compromises between frequency awareness and structural awareness.
- It would give better result on mix problems (problem agnostic)