Provably Powerful Graph Networks

Haggai Maron[1][2]*, Heli Ben-Hamu[1]*, Hadar Serviansky[1]* & Yaron Lipman[1]

[1]Weizmann Institute of Science, [2]NVIDIA Research, (*Equal contribution)

Introduction

There is an ongoing research effort aimed at finding Graph Neural Network (GNN) architectures that are both *expressive* and *practical*. Recently, [2, 3] have analysed the expressive power of Message-Passing neural networks by comparing it to the Weisfeiler-Lehman (WL) hierarchy: a hierarchy of graph isomorphism tests with increasing complexity and expressive power. The main result is that message-passing networks have limited expressive power and are at most as powerful as the first WL test (1-WL).

Goal

Find GNN architectures that are provably more powerful than message passing GNNs:

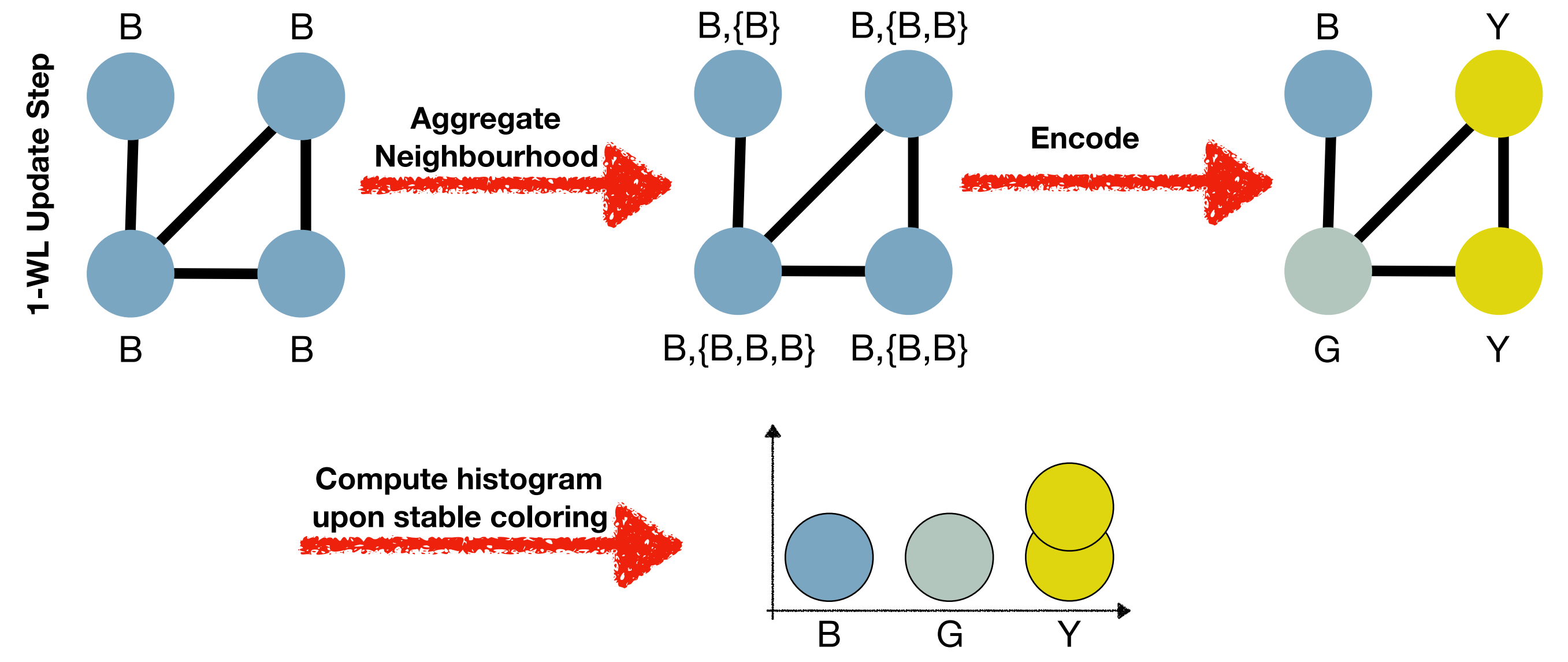
1. Find neural network architectures that are as powerful as k-WL test.
2. Due to the increasing complexity in the hierarchy, find a simple and scalable architecture that is provably more powerful than message passing.

Weisfeiler-Lehman graph isomorphism tests

The WL tests are a family of iterative algorithms used for testing graph isomorphism. Let:

- $G = (V, E, d)$ be a colored graph, $|V| = n$.
- Σ is a set of colors.
- $d : V \to \Sigma$ is a function that assigns a color to each vertex in V.

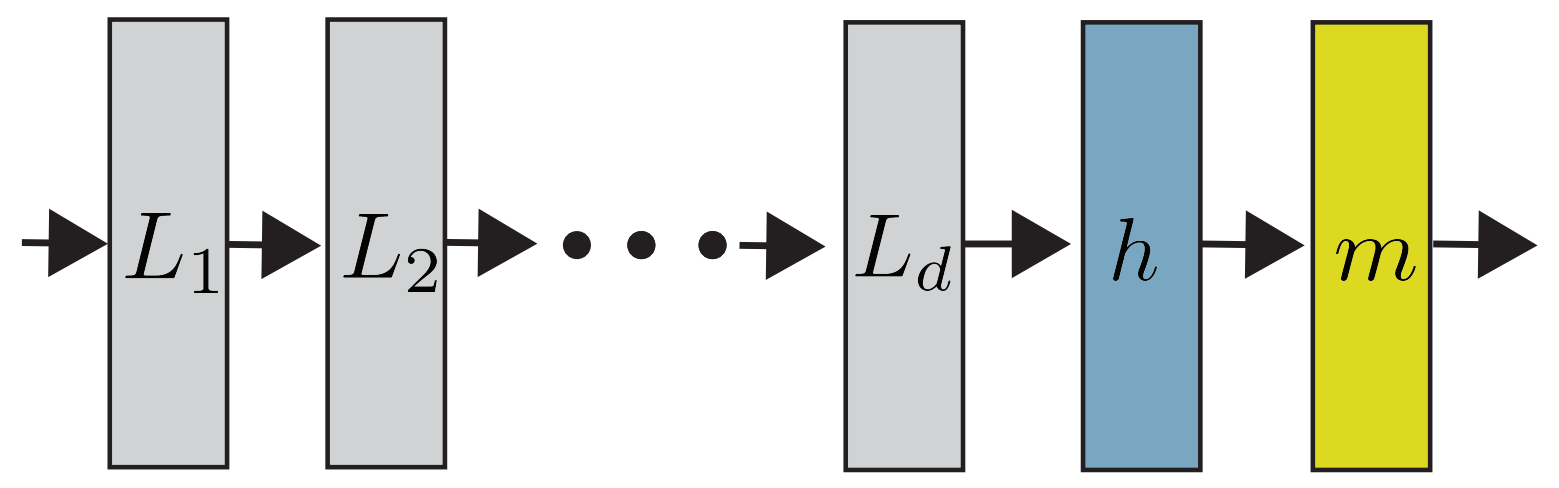
1-WL. At each iteration the coloring of each node is updated according to its current color and the colors of its neighbors. Upon reaching a stable coloring a global feature of the graph is calculated in the form of a color histogram.



k-WL. The k-WL test is a generalization of 1-WL: it re-colors each k-tuple of nodes in the graph at each iteration according to its neighborhood's coloring. For $k \geq 2$, $k+1$-WL is strictly more powerful than k-WL.

k-order graph networks (k-GN)

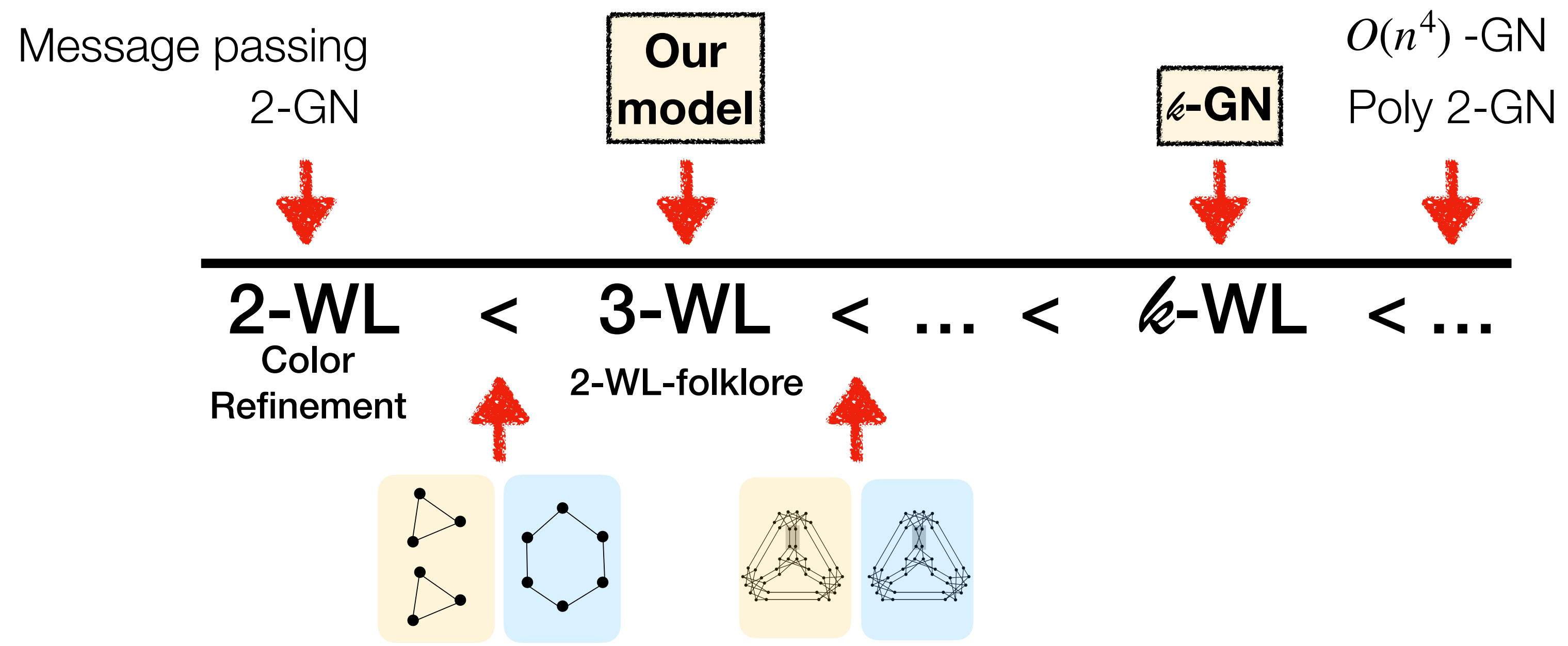
A permutation invariant family of neural network architectures devised in [1]:



$L_i : \mathbb{R}^{n^{k_i} \times a_i} \to \mathbb{R}^{n^{k_{i+1}} \times a_{i+1}}$ are S_n-equivariant linear layers (mapping k_i degree tensors to k_{i+1} degree tensors) composed of the full linear S_n-equivariant basis characterized in [1], h is an S_n-invariant layer and m is a multi-layer perceptron (MLP). The order of the network is the maximal tensor degree $\max_{i \in [d+1]} k_i = k$.

Contributions

1. We show that the discriminative power of k-GNs is at least as the power of k-WL. This implies that for $k \geq 3$ these models are more powerful than message-passing neural networks. However, this expressiveness comes at the cost of having to compute and store high order tensors.
2. We propose a *simple* and *scalable* model architecture that is provably more powerful than message-passing networks, having the discriminative power of at least 3-WL.

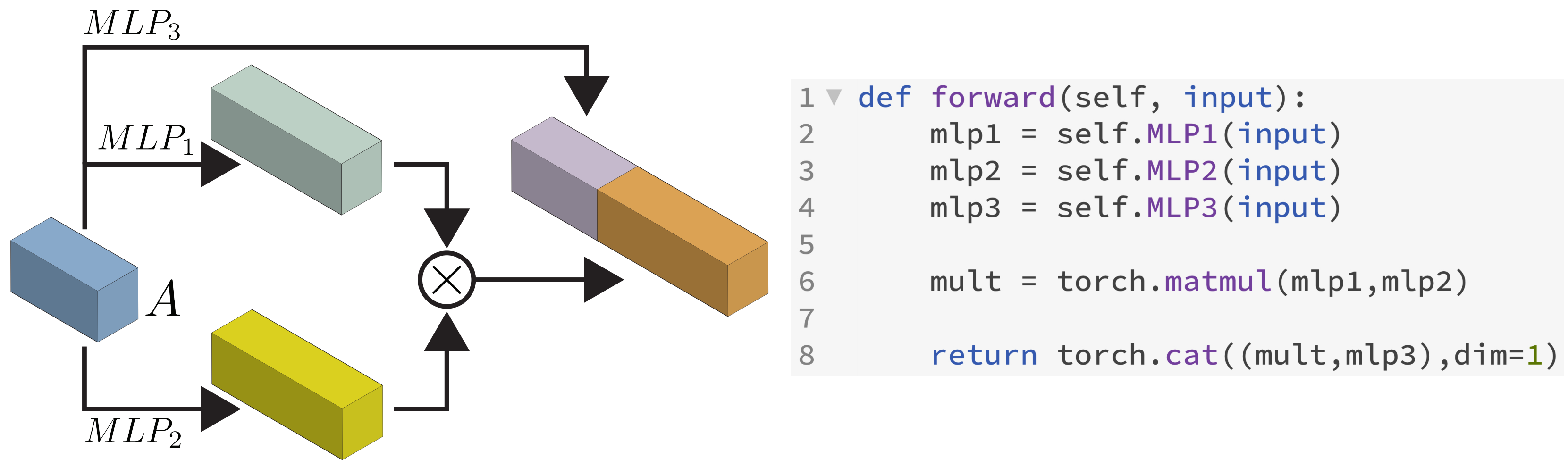


Simple network architecture with 3-WL power

We propose the following neural network architecture:

$$F = m \circ h \circ B_d \circ B_{d-1} \cdots \circ B_1 \qquad (1)$$

where m is an MLP, h is an invariant layer, and $B_1, \dots, B_d$ are blocks with the structure and code as in Figure 1. The block consists of (i) applying 3 different MLPs to the feature dimension of the input (ii) multiplying the output of two MLPs feature-wise and (iii) concatenating the output of the last MLP.



```
def forward(self, input):
    mlp1 = self.MLP1(input)
    mlp2 = self.MLP2(input)
    mlp3 = self.MLP3(input)

    mult = torch.matmul(mlp1,mlp2)

    return torch.cat((mult,mlp3),dim=1)
```

Figure 1. A basic block of a simple and powerful (3-WL) graph neural network.

Theorem. Given two graphs $G = (V, E, d)$, $G' = (V', E', d')$ that can be distinguished by the 3-WL graph isomorphism test (i.e., the test generates different global features for G and G'), there exists a network F (equation 1) so that $F(G) \neq F(G')$. On the other direction for every two isomorphic graphs $G \cong G'$ and F (equation 1), $F(G) = F(G')$.

Intuition. We give intuition to the improved power gained by the addition of matrix multiplication by showing that this model can distinguish between the two regular graphs that appear above, which are 2-WL indistinguishable. We consider a constant initial coloring of the graph G with adjacency matrix A. A network with 2 blocks can compute A^3 and then take its trace (using the invariant layer); trace(A^3) computes the number of cycles of length 3. Now, the graph on the right has 0 such cycles while the graph on the left has 12.

k-GNs are as powerful as k-WL

Theorem. Given two graphs $G = (V, E, d)$, $G' = (V', E', d')$ that can be distinguished by the k-WL graph isomorphism test, there exists a k-order network F so that $F(G) \neq F(G')$. On the other direction for every two isomorphic graphs $G \cong G'$ and k-order network F, $F(G) = F(G')$.

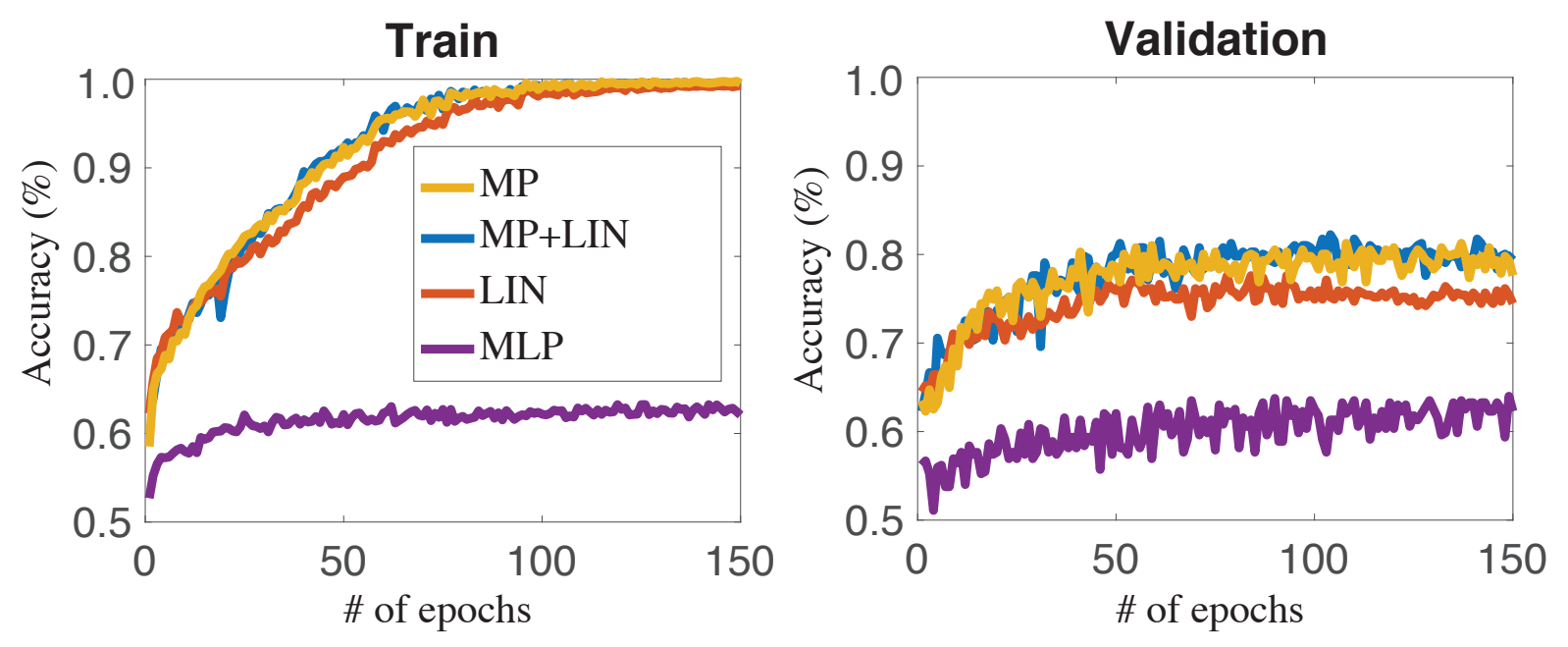
Proof idea. The proof is based on showing that k-GNs can implement the different parts of the k-WL algorithm. Namely, neighborhood aggregation and color encoding. This can be done by learning an injective function on multisets. Each k-tuple has k neighborhoods, where each neighborhood coloring is represented as a multiset. We show that a family of polynomials composed with summation and tiling over the nodes dimension is an injective multiset function that can be approximated by k-GNs. Finally, the encoding is simply concatenating the output of the above function for each neighborhood.

Experiments

Table 1. Graph Classification Results on the datasets from [4]

dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	COLLAB	IMDB-B	IMDB-M
size	188	344	1113	4110	4127	5000	1000	1500
classes	2	2	2	2	2	3	2	3
avg node #	17.9	25.5	39.1	29.8	29.6	74.4	19.7	13
			Results					
GK	81.39±1.7	55.65±0.5	71.39±0.3	62.49±0.3	62.35±0.3	NA	NA	NA
RW	79.17±2.1	55.91±0.3	59.57±0.1	> 3 days	NA	NA	NA	NA
PK	76±2.7	59.5±2.4	73.68±0.7	82.54±0.5	NA	NA	NA	NA
WL	84.11±1.9	57.97±2.5	**74.68±0.5**	**84.46±0.5**	**85.12±0.3**	NA	NA	NA
FGSD	**92.12**	**62.80**	73.42	79.80	78.84	**80.02**	73.62	**52.41**
AWE-DD	NA	NA	NA	NA	NA	73.93±1.9	**74.45 ± 5.8**	51.54 ±3.6
AWE-FB	87.87±9.7	NA	NA	NA	NA	70.99 ± 1.4	73.13 ±3.2	51.58 ± 4.6
DGCNN	85.83±1.7	58.59±2.5	75.54±0.9	74.44±0.5	NA	73.76±0.5	70.03±0.9	47.83±0.9
PSCN (k=10)	88.95±4.4	62.29±5.7	75±2.5	76.34±1.7	NA	72.6±2.2	71±2.3	45.23±2.8
DCNN	NA	NA	61.29±1.6	56.61± 1.0	NA	52.11±0.7	49.06±1.4	33.49±1.4
ECC	76.11	NA	NA	76.82	75.03	NA	NA	NA
DGK	87.44±2.7	60.08±2.6	75.68±0.5	80.31±0.5	80.32±0.3	73.09±0.3	66.96±0.6	44.55±0.5
DiffPool	NA	NA	**78.1**	NA	NA	75.5	NA	NA
CCN	**91.64±7.2**	**70.62±7.0**	NA	76.27±4.1	75.54±3.4	NA	NA	NA
Invariant Graph Networks	83.89±12.95	58.53±6.86	76.58±5.49	74.33±2.71	72.82±1.45	78.36±2.47	72.0±5.54	48.73±3.41
GIN	89.4±5.6	64.6±7.0	76.2±2.8	82.7±1.7	NA	80.2±1.9	**75.1±5.1**	**52.3±2.8**
1-2-3 GNN	86.1±	60.9±	75.5±	76.2±	NA	NA	74.2±	49.5±
Ours 1	90.55±8.7	66.17±6.54	77.2±4.73	**83.19±1.11**	**81.84±1.85**	80.16±1.11	72.6±4.9	50±3.15
Ours 2	88.88±7.4	64.7±7.46	76.39±5.03	81.21±2.14	**81.77±1.26**	**81.38±1.42**	72.2±4.26	44.73±7.89
Ours 3	89.44±8.05	62.94±6.96	76.66±5.59	80.97±1.91	**82.23±1.42**	**80.68±1.71**	73±5.77	50.46±3.59
Rank	**3rd**	**2nd**	**2nd**	**2nd**	**2nd**	**1st**	**6th**	**5th**

Methods above the line are not based on neural networks. Ours 1, Ours 2, Ours 3 stand for three slightly different architectures that use our basic block (details in the paper).



Equivariant layers evaluation. We performed a baseline comparison on the NCI1 dataset [4]: (i) our suggested model, denoted Matrix Product (MP); (ii) matrix product + full linear basis from [1] (MP+LIN); (iii) only full linear basis (LIN); and (iv) MLP applied to the feature dimension. Although all methods (excluding MLP) are able to achieve a zero training error, the (MP) and (MP+LIN) enjoy better generalization than the linear basis of [1]. Note that (MP) and (MP+LIN) are comparable, however (MP) is considerably more efficient.

References

[1] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman.
Invariant and equivariant graph networks.
In *International Conference on Learning Representations*, 2019.

[2] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe.
Weisfeiler and leman go neural: Higher-order graph neural networks.
arXiv preprint arXiv:1810.02244, 2018.

[3] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka.
How powerful are graph neural networks?
In *International Conference on Learning Representations*, 2019.

[4] Pinar Yanardag and S.V.N. Vishwanathan.
Deep Graph Kernels.
In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 2015.