NEW YORK UNIVERSITY

Exploring the expressive power of graph neural networks

Zhengdao Chen
New York University

Based on joint work with Joan Bruna (NYU), Lei Chen (NYU), and Soledad Villar (JHU)

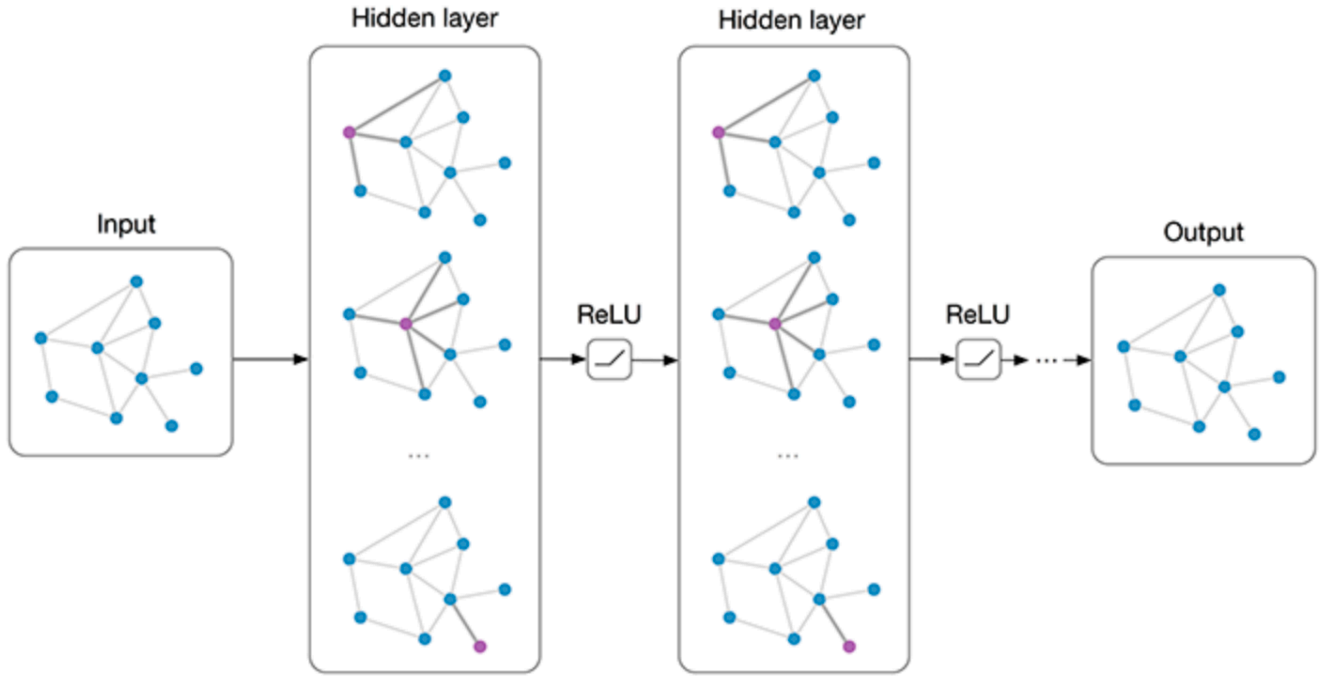
Learning on Graphs and Geometry Reading Group
2021.11.16

Graph Neural Network (GNN)

GNN: neural network model for graph-structured inputs

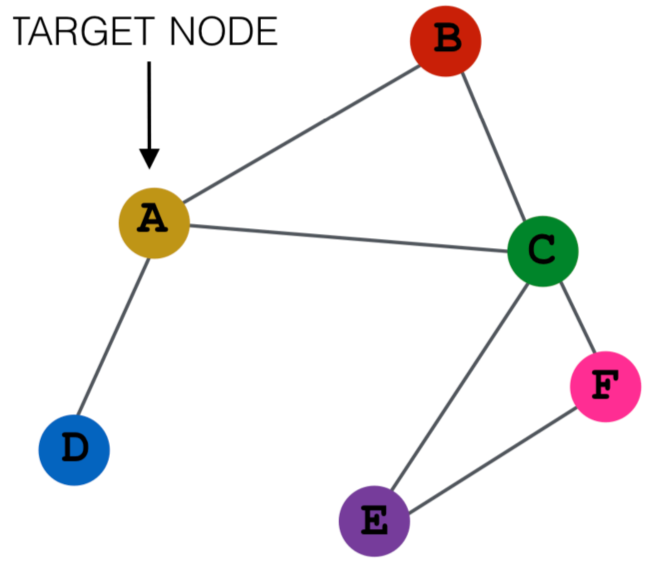
Applicable to different scenarios:

- graph-level prediction
- node-level prediction
- edge-level prediction

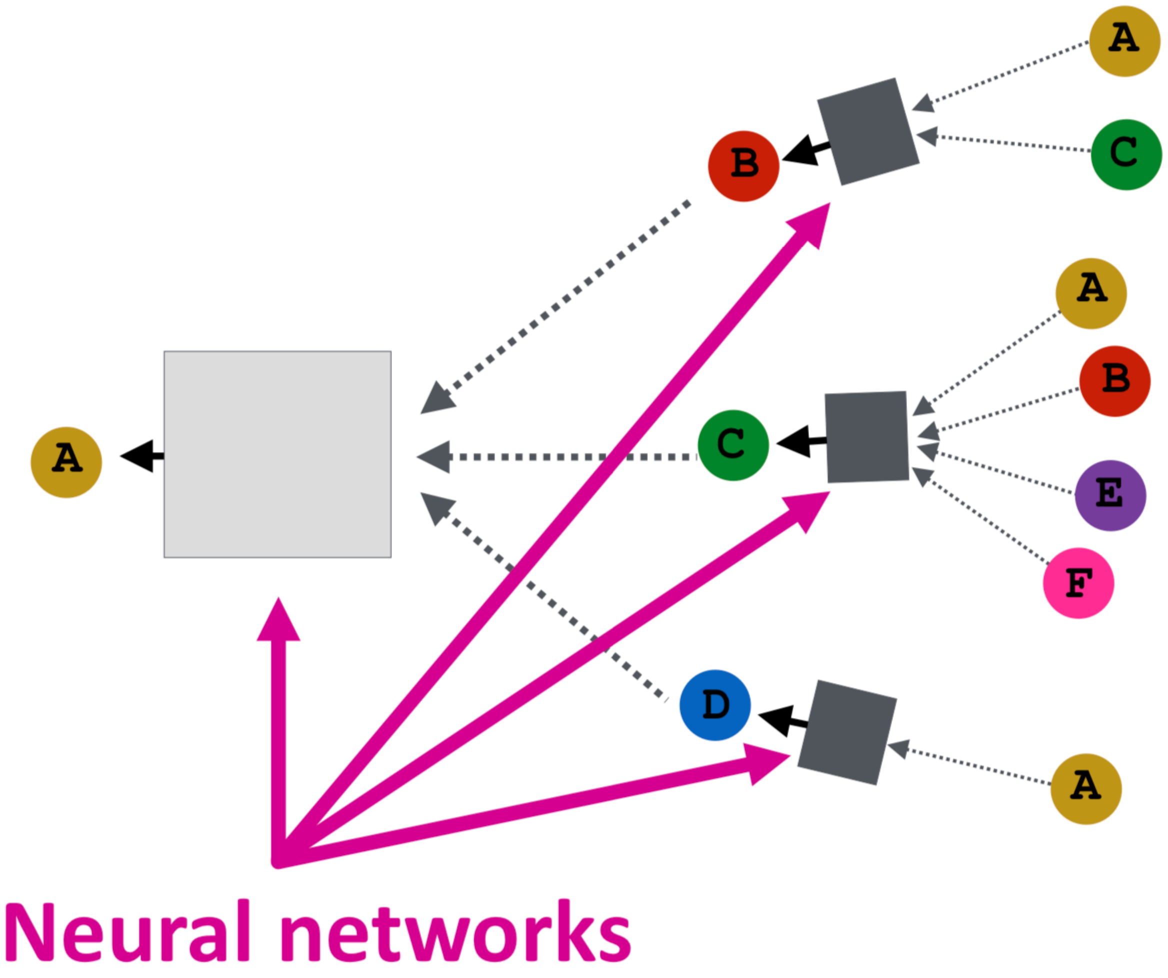


Message Passing NN (MPNN)

Based on message passing / neighborhood aggregation



INPUT GRAPH



Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, http://cs224w.stanford.edu

Understanding GNNs

Properties of GNNs：

- “G”:
  - Symmetry: invariance/equivariance to node permutations
  - Computational efficiency
  - Inductive generalization

- “NN”:
  - Trainability
  - **Expressiveness**

How expressive are GNNs?
How to even measure their expressive power?

Classical perspective: function approximation

Criterion 0: Ability to approximate functions on graphs

GNN for graph-level prediction tasks $f : \mathcal{G} \to \mathbb{R}$

Isomorphic graphs are typically considered equivalent

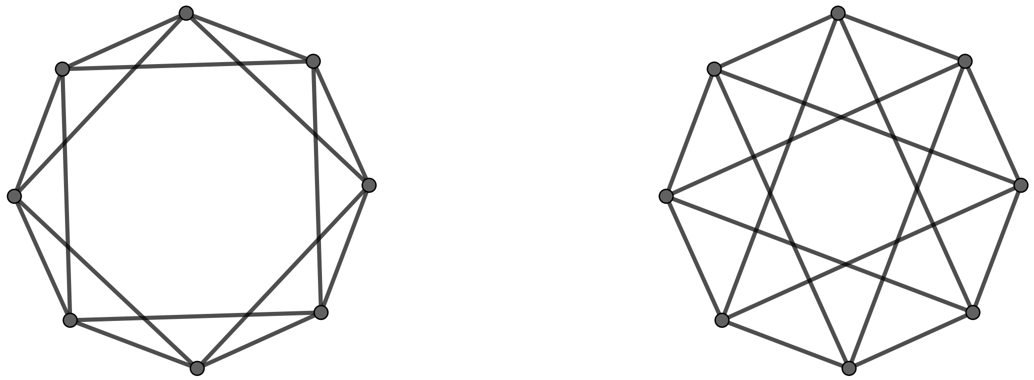
➡ Interested in *permutation-invariant* functions on graphs

Question: Given enough width and depth, can GNNs approximate arbitrary permutation-invariant function on graphs?

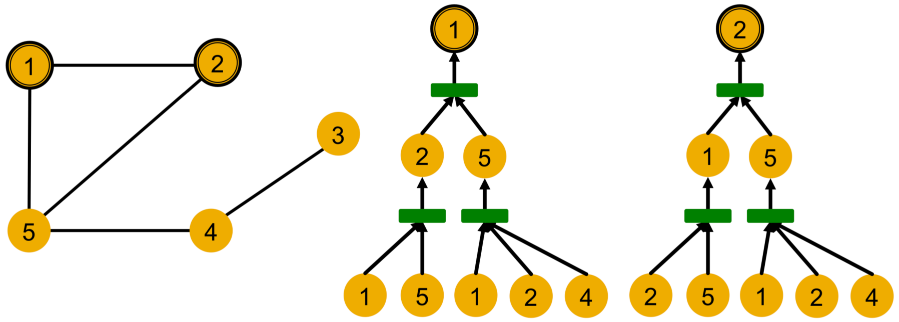
Another perspective: graph isomorphism tests

Criterion 1: Ability to distinguish *non-isomorphic* graphs

e.g. MPNNs cannot distinguish this pair of graphs



More generally, MPNNs are no more expressive than the **Weisfeiler-Lehman (WL)** test in distinguishing non-isomorphic graphs



Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, http://cs224w.stanford.edu

Graph Isomorphism Network & k-GNN: as expressive as (k-)WL

Xu et al. (2019); Morris et al. (2019)

The two perspectives are equivalent

⇨ If a function class can *universally approximate* permutation-invariant functions on graphs, it can also *distinguish any pair of non-isomorphic graphs*

⇦ If a function class can *distinguish any pair of non-isomorphic graphs*, then (with a small augmentation) it can also *universally approximate* permutation-invariant functions on graphs

Measuring expressive power of a function class by the **sigma-algebra of graph equivalence classes** that it induces

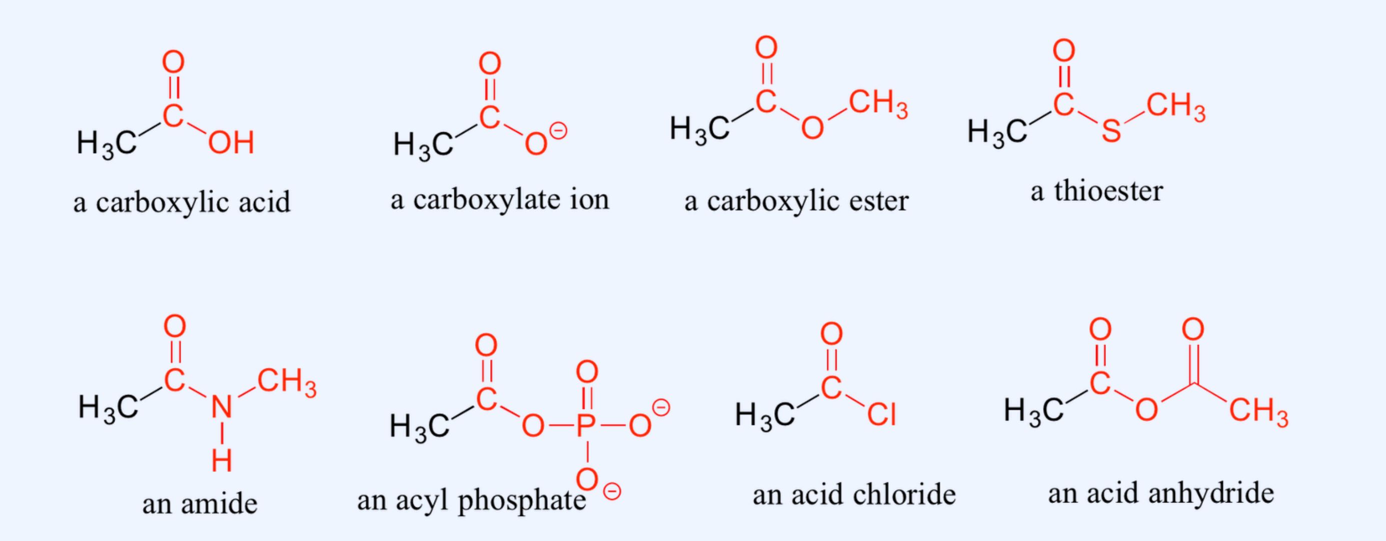
Z. Chen, S. Villar, L. Chen and J. Bruna (NeurIPS 2019)
"On the Equivalence between Graph Isomorphism Testing and Function Approximation by GNNs"

Other criteria for expressive power?

We want to find a criterion that is intuitive, concrete and relevant for practice

Criterion 2: Ability to count substructures

e.g. *functional groups* in organic molecules are often important indicators of chemical properties

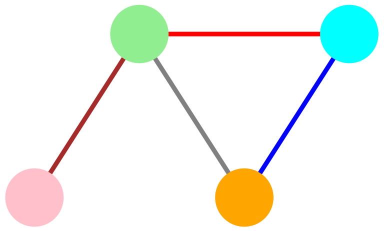


Z. Chen, S. Villar, L. Chen and J. Bruna (NeurIPS 2019)

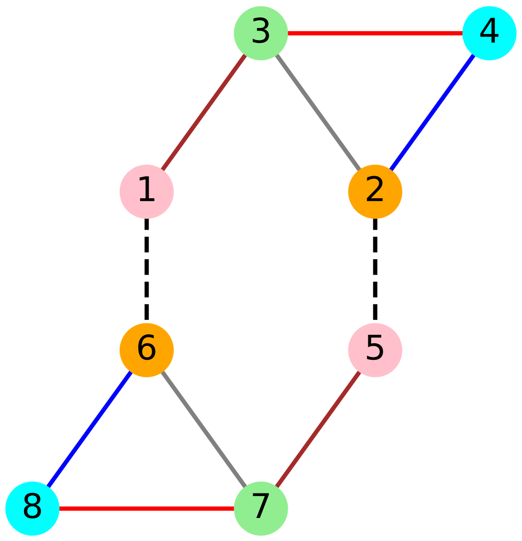
MPNNs unable to count substructures

For any connected pattern consisting of 3 or more nodes, no MPNN can count it exactly in all graphs

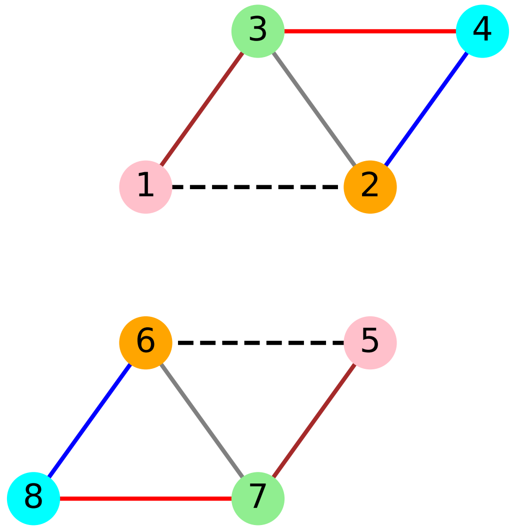
e.g.



pattern



graph A



graph B

Z. Chen, L. Chen, S. Villar, and J. Bruna (NeurIPS 2020)
"Can Graph Neural Networks Count Substructures?"

Build models able to count substructures

Background： Relational Pooling (RP) model Murphy et al. (2020)

$$f_{\mathrm{RP}}(G) = \sum_{\pi \in S_n} \bar{f}(\boldsymbol{A}_{\pi \circ G})$$

but O(n!)

We can universally approximate *permutation-invariant* functions by symmetrizing (summing or averaging) *permutation-sensitive* functions

Local Relational Pooling (LRP) **Z. Chen**, L. Chen, S. Villar, and J. Bruna (NeurIPS 2020)

Apply RP to each *egonet* (local neighborhood), then aggregate across all neighborhoods

$$f^{r}_{\mathrm{LRP}}(G) = \sum_{i \in V} f_{\mathrm{RP}}(G^{\mathrm{ego}}_{i,r})$$

only O(nr!)

Can also be applied iteratively —> **Deep LRP**

Other works on GNNs & substructures: Bouritsas et al. (2020); Tahmasebi et al. (2020)

LRP model in practice

1. Molecular predictions

Z. Chen, L. Chen, S. Villar, and J. Bruna (NeurIPS 2020)

QM9

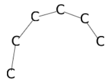
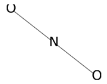
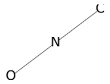
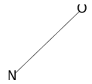
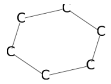
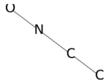
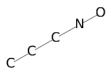
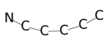
Target	DTNN	MPNN	123-gnn	Powerful-IGN	Deep LRP-1-4
μ	0.244	0.358	0.476	0.231	0.399
α	0.95	0.89	0.27	0.382	0.337
ϵ_{homo}	0.00388	0.00541	0.00337	0.00276	0.00287
ϵ_{lumo}	0.00512	0.00623	0.00351	0.00287	0.00309
Δ_ϵ	0.0112	0.0066	0.0048	0.00406	**0.00396**
$\langle R^2 \rangle$	17	28.5	22.9	16.07	20.4
ZPVE	0.00172	0.00216	0.00019	0.00064	0.00067
U_0	2.43	2.05	0.0427	0.234	0.590
U	2.43	2	0.111	0.234	0.588
H	2.43	2.02	0.0419	0.229	0.587
G	2.43	2.02	0.0469	0.238	0.591
C_v	0.27	0.42	0.0944	0.184	0.149
Loss	0.1014	0.1108	0.0657	0.0512	0.0641

molhiv

Model	Training	Validation	**Testing**
Deep LRP-1-4	89.81±2.90	81.31±0.88	76.87±1.80
Deep LRP-1-4 (ES)	87.56±2.11	82.09±1.16	**77.19**±**1.40**
GIN†	88.64±2.54	82.32±0.90	75.58±1.40
GIN + VN†	92.73±3.80	84.79±0.68	77.07±1.49
GCN†	88.54±2.19	82.04±1.41	76.06±0.97
GCN + VN†	90.07±4.69	83.84±0.91	75.99±1.19
GAT ‡	-	-	72.9±1.8
GraphSAGE ‡	-	-	74.4±0.7

2. Explaining a task

L. Chen*, **Z. Chen*** and J. Bruna (ICASSP 2021)
"Learning the Relevant Substructures for Tasks on Graph Data"

	pattern	rank	pattern	rank
Class 0		1		2
		3		4
Class 1		1		2
		3		4

Related work: GNNExplainer (Ying et al., 2019)

“Simplified GNN”： Graph-Augmented MLP

Idea: Decoupling propagation from transformation

Step 1: Augmenting node features with ***multi-hop graph operators***

e.g. $X_{\text{aug}} = \left[X, \tilde{A}X, \tilde{A}^2X, ..., \tilde{A}^{K-1}X\right] \in \mathbb{R}^{n \times (Kd)}$

Step 2: Apply node-wise MLP to the augmented node features

e.g. SGC (Wu et al. 2019), GFN (Chen et al. 2020), SIGN (Frasca et al. 2020)
Strengthened with attention: Zhang et al. (2021)

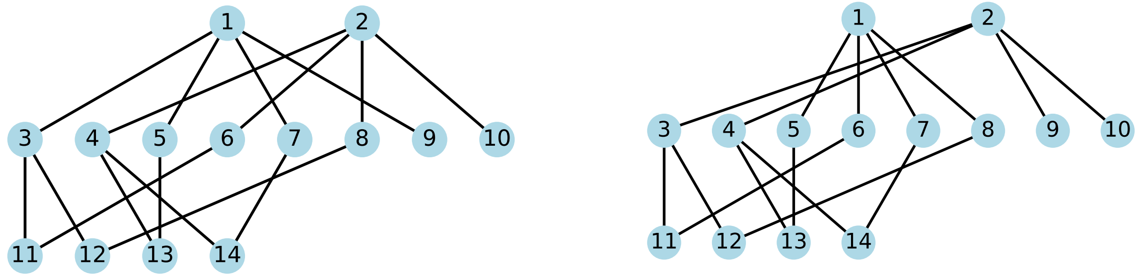
Compared to deep GNNs:

- Pros: scalability; mitigating over-smoothing
- Cons: expressive power **?**

GNN vs Graph-Augmented MLP

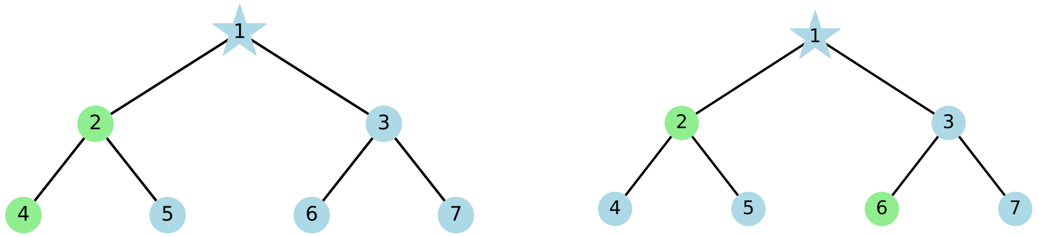
Criterion 1: Ability to distinguish non-isomorphic graphs

Distinguishable by MPNN
but not GA-MLP



Criterion 3: Ability to count attributed walks

MPNN can count attributed walks, but GA-MLP cannot



	Cora		RRG	
Model	Train	Test	Train	Test
GIN	**3.98E-6**	**9.72E-7**	**3.39E-5**	**2.61E-4**
GA-MLP-A	1.23E-1	1.56E-1	1.75E-2	2.13E-2
GA-MLP-$A+$	1.87E-2	6.44E-2	1.69E-2	2.13E-2
GA-MLP-$\tilde{A}_{(1)}$	4.22E-1	5.79E-1	1.02E-1	1.58E-1
GA-MLP-$\tilde{A}_{(1)}+$	4.00E-1	5.79E-1	1.12E-1	1.52E-1

L. Chen*, **Z. Chen*** and J. Bruna (ICLR 2021)
“On Graph Neural Networks versus Graph-Augmented MLPs”

Choice of operator matters

Task: supervised community detection for stochastic block models

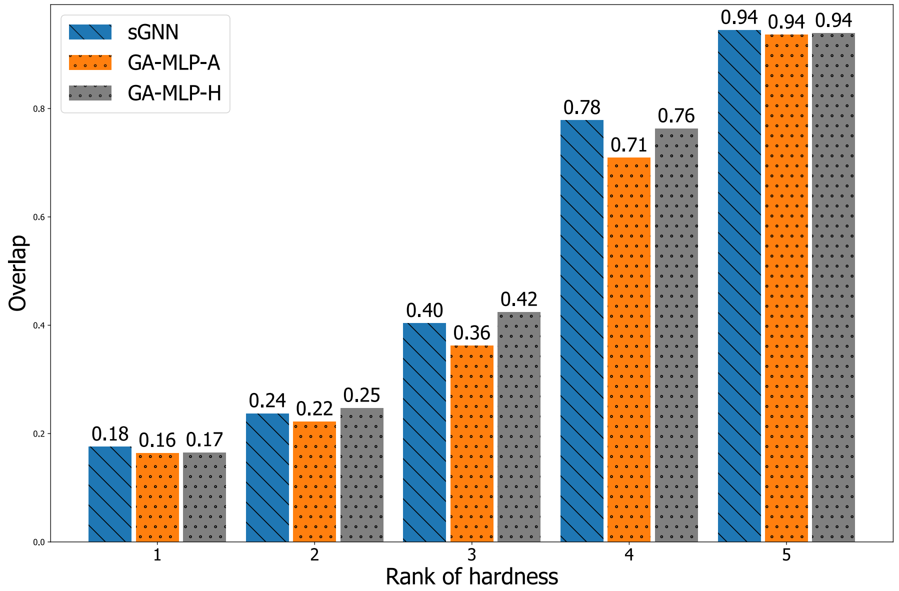
Z. Chen, L. Li and J. Bruna (ICLR 2019)
"Supervised Community Detection with Line Graph Neural Networks"

The performance of GA-MLP depends on what graph operator is used

But the optimal operator is not known a priori

GNN does not need this information to achieve good performance

L. Chen*, **Z. Chen*** and J. Bruna. (ICLR 2021)



Beyond message passing - Geometric DL

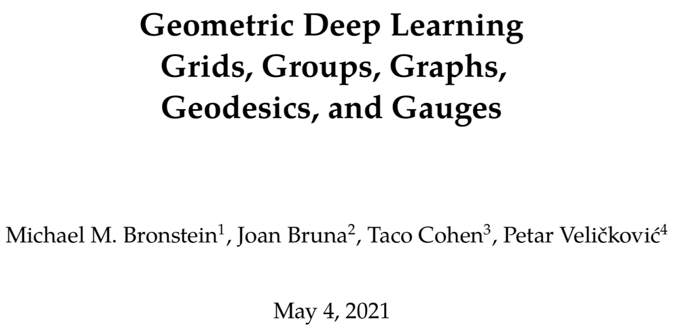
For graph-level prediction, we want to learn a function

$$f : \mathcal{G} \to \mathbb{R}$$

or, it can be written as a ***permutation-invariant*** function

$$f : \mathbb{R}^{n \times n} \to \mathbb{R}$$

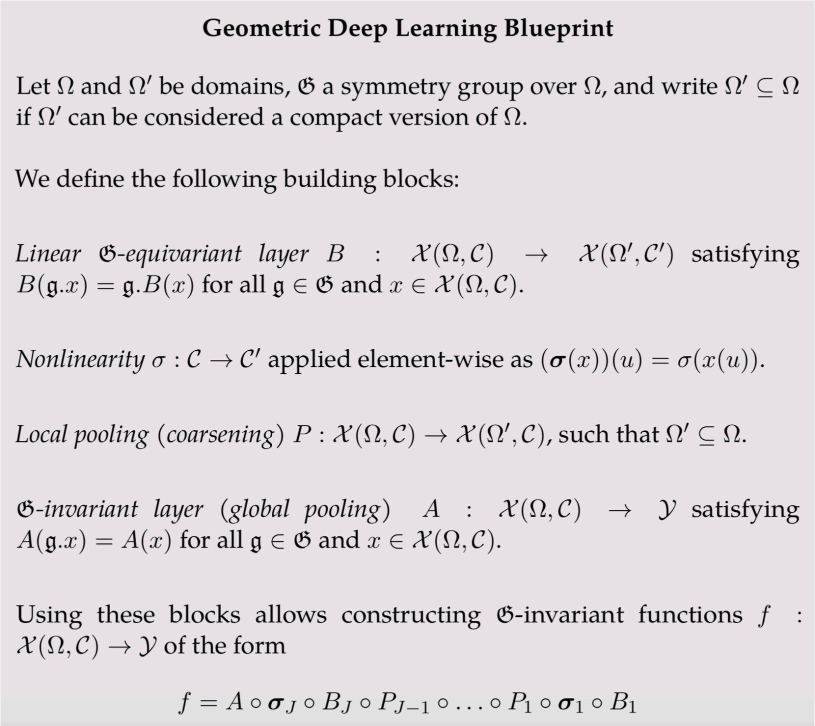
"**Geometric Deep Learning Blueprint**": interleaving *linear equivariant layers* with *point-wise nonlinearities*



Geometric Deep Learning
Grids, Groups, Graphs,
Geodesics, and Gauges

Michael M. Bronstein[1], Joan Bruna[2], Taco Cohen[3], Petar Veličković[4]

May 4, 2021



Geometric Deep Learning Blueprint

Let Ω and Ω' be domains, $\mathfrak{G}$ a symmetry group over Ω, and write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω.

We define the following building blocks:

Linear $\mathfrak{G}$-equivariant layer $B : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{X}(\Omega', \mathcal{C}')$ satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Nonlinearity $\sigma : \mathcal{C} \to \mathcal{C}'$ applied element-wise as $(\boldsymbol{\sigma}(x))(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

$\mathfrak{G}$-invariant layer (global pooling) $A : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{Y}$ satisfying $A(\mathfrak{g}.x) = A(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Using these blocks allows constructing $\mathfrak{G}$-invariant functions $f : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{Y}$ of the form

$$f = A \circ \boldsymbol{\sigma}_J \circ B_J \circ P_{J-1} \circ \ldots \circ P_1 \circ \boldsymbol{\sigma}_1 \circ B_1$$

2-Invariant Graph Network (2-IGN)

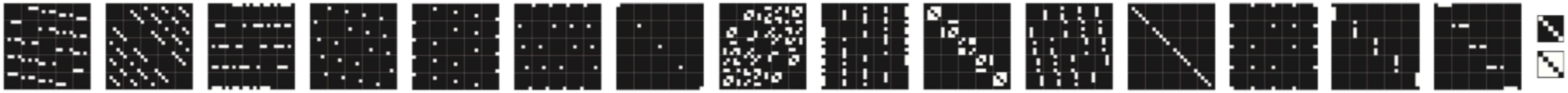
2-IGN: Instantiating Geometric DL in graphs

In every layer: $\boldsymbol{A}^{(l+1)} = \sigma\left(L^{(l+1)}\left(\boldsymbol{A}^{(l)}\right)\right)$

with equivariant linear maps: $L : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$

$$L(\mathbf{A})_{i'} = \sum_{\boldsymbol{a},i} \mathbf{T}_{\boldsymbol{a},i,i'} \mathbf{A}_{\boldsymbol{a},i} + \mathbf{Y}_{i'}; \quad \mathbf{T} = \sum_{\lambda,j,j'} w_{\lambda,j,j'} \mathbf{B}^{\lambda,j,j'}; \mathbf{Y} = \sum_{j'} b_{j'} \mathbf{C}^{j'}$$

$$L(\mathbf{A})_{\boldsymbol{b},i'} = \sum_{\boldsymbol{a},i} \mathbf{T}_{\boldsymbol{a},i,\boldsymbol{b},i'} \mathbf{A}_{\boldsymbol{a},i} + \mathbf{Y}_{\boldsymbol{b},i'}; \quad \mathbf{T} = \sum_{\mu,j,j'} w_{\mu,j,j'} \mathbf{B}^{\mu,j,j'}; \mathbf{Y} = \sum_{\lambda,j'} b_{\lambda,j'} \mathbf{C}^{\lambda,j'}$$



Maron et al. (2019)
Maron et al. (2020)

Is this model universal?

2-IGNs go beyond message passing / WL-like structure

But, 2-IGNs are also no more powerful than the 2-WL test

Z. Chen, S. Villar, L. Chen and J. Bruna (NeurIPS 2019)
Z. Chen, L. Chen, S. Villar, and J. Bruna (NeurIPS 2020)

More generally, k-IGNs are no more powerful than the k-WL test

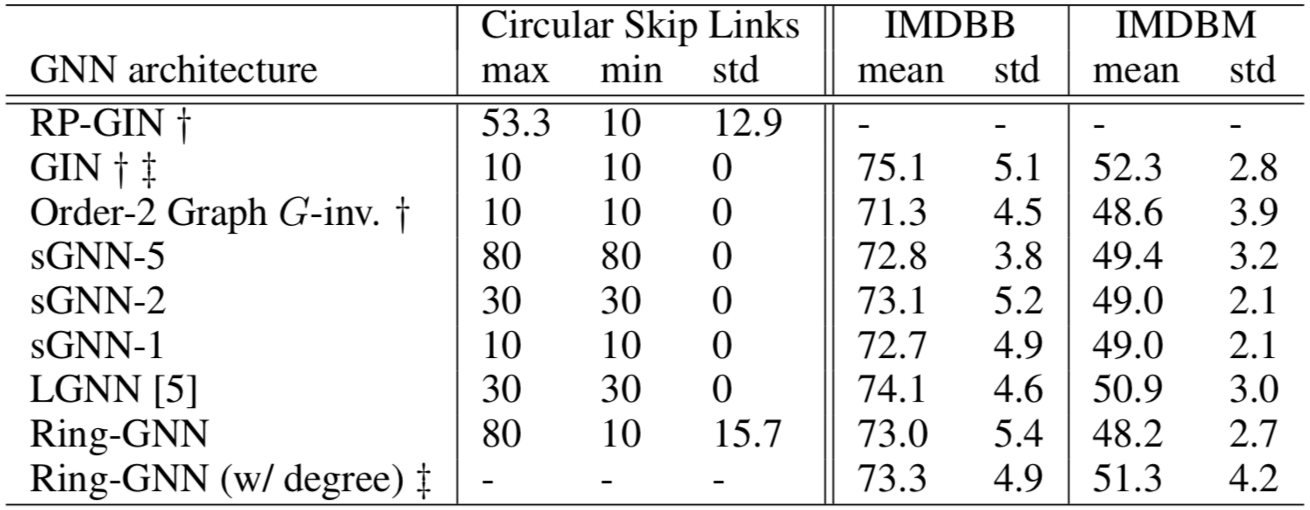
F. Geerts et al. (2020)

A model with stronger expressive power

Ring-GNN

Z. Chen, S. Villar, L. Chen and J. Bruna
(NeurIPS 2019)

$$\begin{aligned} \boldsymbol{A}_1^{(l+1)} &= \sigma\Big(L_a^{(l+1)}(\boldsymbol{A}^{(l)})\Big) \\ \boldsymbol{A}_2^{(l+1)} &= \sigma\Big(L_b^{(l+1)}(\boldsymbol{A}^{(l)}) \cdot L_c^{(l+1)}(\boldsymbol{A}^{(l)})\Big) \\ \boldsymbol{A}^{(l+1)} &= k_1^{(l)}\boldsymbol{A}_1^{(l+1)} + k_2^{(l)}\boldsymbol{A}_2^{(l+1)} \end{aligned}$$



	Circular Skip Links			IMDBB		IMDBM	
GNN architecture	max	min	std	mean	std	mean	std
RP-GIN †	53.3	10	12.9	-	-	-	-
GIN † ‡	10	10	0	75.1	5.1	52.3	2.8
Order-2 Graph G-inv. †	10	10	0	71.3	4.5	48.6	3.9
sGNN-5	80	80	0	72.8	3.8	49.4	3.2
sGNN-2	30	30	0	73.1	5.2	49.0	2.1
sGNN-1	10	10	0	72.7	4.9	49.0	2.1
LGNN [5]	30	30	0	74.1	4.6	50.9	3.0
Ring-GNN	80	10	15.7	73.0	5.4	48.2	2.7
Ring-GNN (w/ degree) ‡	-	-	-	73.3	4.9	51.3	4.2

Other models: PPGN (Maron et al. 2019), DE-GNN (Li et al. 2020), SMP (Vignac et al. 2020), ID-GNN (You et al. 2021), …

In summary

1. Four criteria for measuring the expressive power of GNNs
   - Ability to approximate permutation-invariant functions on graphs
   - Ability to distinguish non-isomorphic graphs
   - Ability to count substructures
   - Ability to count attributed walks

2. Existing GNN models (MPNNs and 2-IGNs) are limited in expressive power

3. Two novel models with stronger expressive power
   - Ring-GNN
   - (Deep) Local Relational Pooling

References

1. **Z. Chen,** L. Li, J. Bruna (2019), “Supervised Community Detection with Line Graph Neural Networks”, ICLR 2019.
2. **Z. Chen**, S. Villar, L. Chen, J. Bruna (2019), “On the Equivalence between Graph Isomorphism Testing and Function Approximation by GNNs”, NeurIPS 2019.
3. **Z. Chen**, L. Chen, S. Villar, J. Bruna (2020), “Can Graph Neural Networks Count Substructures?”, NeurIPS 2020.
4. L. Chen, **Z. Chen**, J. Bruna (2021), “Learning the Relevant Substructures for Tasks on Graph Data”, ICASSP 2021.
5. L. Chen*, **Z. Chen***, J. Bruna (2021), “On Graph Neural Networks versus Graph-Augmented MLPs”, ICLR 2021.
6. T. Kipf and M. Welling (2017), “Semi-Supervised Classification with Graph Convolutional Networks”, ICLR 2017.
7. K. Xu, et al. (2019), “How Powerful are Graph Neural Networks?”, ICLR 2019.
8. Morris et al. (2019), “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”, AAAI 2019.
9. H. Maron et al. (2018), “Invariant and Equivariant Graph Networks”, ICLR 2019.
10. H. Maron et al. (2019), “Provably Powerful Graph Networks”, NeurIPS 2020.
11. P. Li et al. (2020), “Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning”, NeurIPS 2020.
12. C. Vignac et al. (2020), “Building powerful and equivariant graph neural networks with structural message-passing”, NeurIPS 2020.
13. J. You et al. (2021), “Identity-aware Graph Neural Networks”, AAAI 2021.
14. R. Murphy et al. (2019), “Relational Pooling for Graph Representations”, ICML 2019.
15. G. Bouritsas et al. (2020), “Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting”, arxiv.
16. B. Tahmasebi et al. (2020), “Counting Substructures with Higher-Order Graph Neural Networks: Possibility and Impossibility Results”, arxiv.
17. R. Ying et al. (2019), “GNNExplainer: Generating Explanations for Graph Neural Networks”,
18. F. Wu et al. (2019), “Simplifying Graph Convolutional Networks”, ICML 2019.
19. T. Chen et al. (2020), “Are Powerful Graph Neural Nets Necessary? A Dissection on Graph Classification”, arxiv.
20. F. Frasca et al. (2020), “SIGN: Scalable Inception Graph Neural Networks”, arxiv.
21. F. Geerts (2020), “The expressive power of kth-order invariant graph networks”, arxiv.

Thanks!