Oct 6th, 2021
MSU TDA Seminar

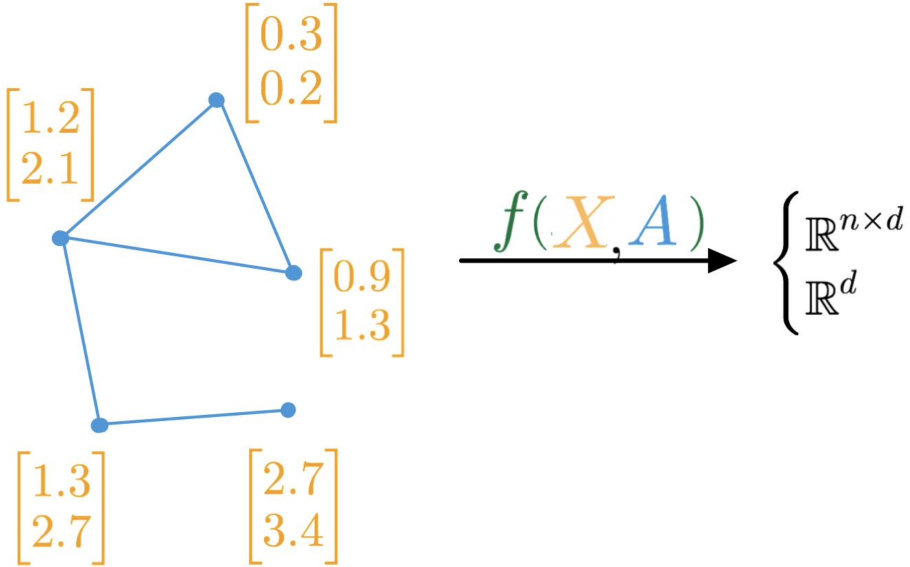
Neural Topological Representation Learning

Cristian Bodnar

UNIVERSITY OF CAMBRIDGE

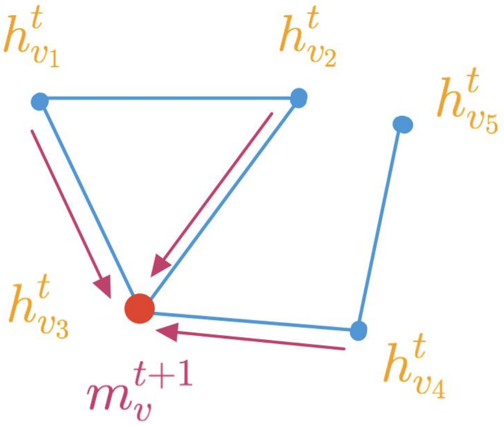
Graph Representation Learning

The aim is to learn **data-driven representations** of nodes or entire graphs. These can be used for node/graph regression/classification.

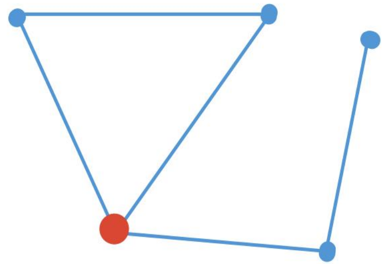


Graph Neural Networks (GNNs)

Message from neighborhood $\longrightarrow m_v^{t+1} = \mathrm{AGG}_{w \in \mathcal{N}(v)}\Big(M\big(h_v^t, h_w^t, h_{e(v,w)}^t\big)\Big)$



Updated node representation $\longrightarrow h_v^{t+1} = U\Big(h_v^t, m_v^{t+1}\Big)$



Graph-level representation $\longrightarrow h_G = \mathrm{READOUT}\big(\{\!\{h_v^L | v \in G\}\!\}\big)$

$$h_{v_3}^{t+1} := U(h_{v_3}^t, m_v^{t+1})$$

Pooling in GNNs

GNN layers can be interleaved with graph coarsening (i.e. pooling) layers. But what does it mean to perform pooling in a graph?

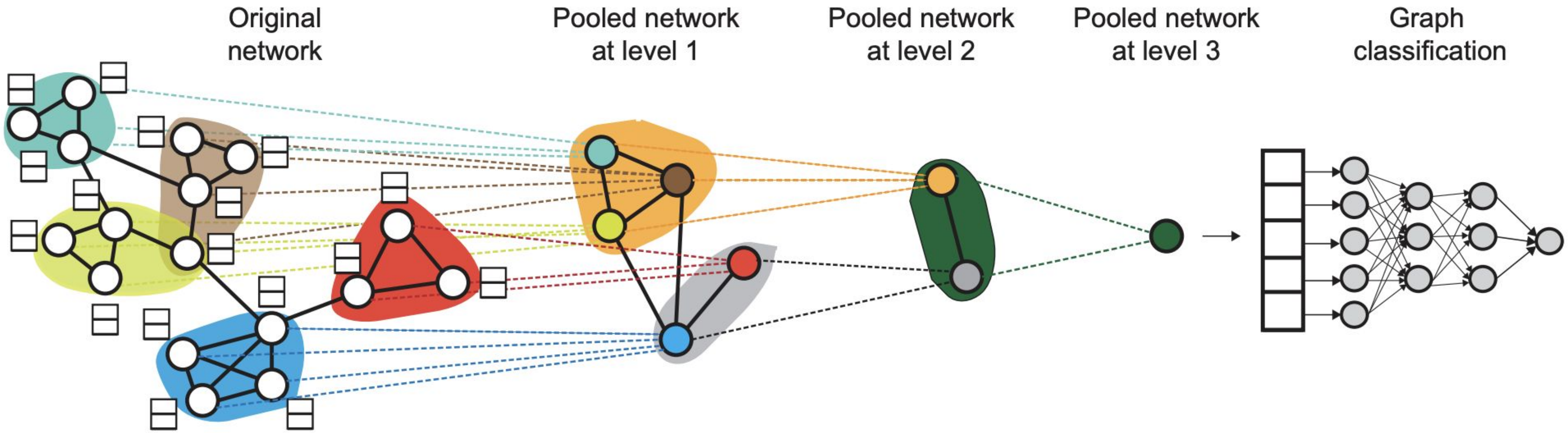


Image is taken from [1]

[1]: Hierarchical Graph Representation Learning with Differentiable Pooling (Ying et al., 2018)

Deep Graph Mapper: Seeing Graphs through the Neural Lens

Frontiers in Big Data, Topology in Real-World Machine Learning and Data Analysis Research Topic

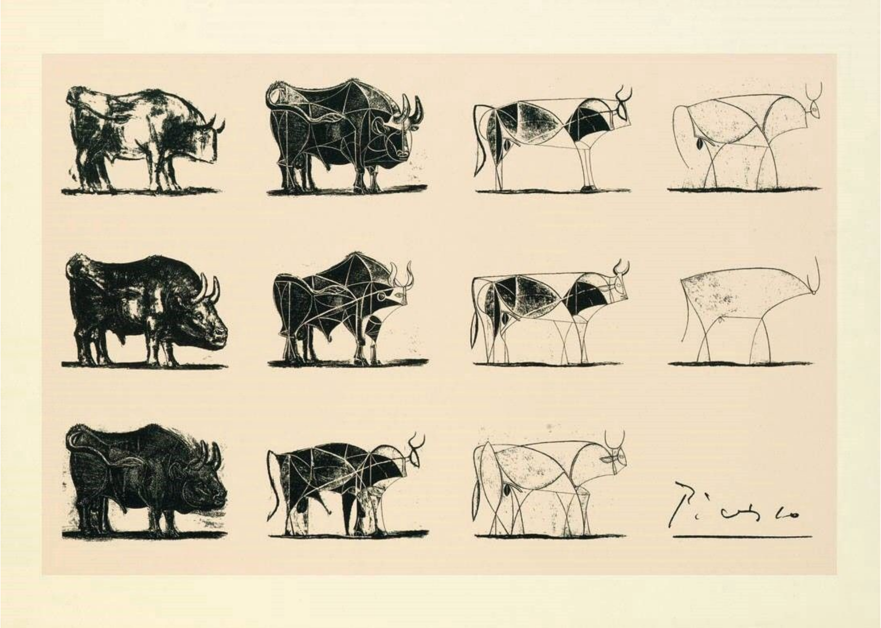
Cristian Bodnar*, Cătălina Cangea*, Pietro Liò

UNIVERSITY OF CAMBRIDGE

Based on **https://www.frontiersin.org/articles/10.3389/fdata.2021.680535/full**

Desirable properties of pooling

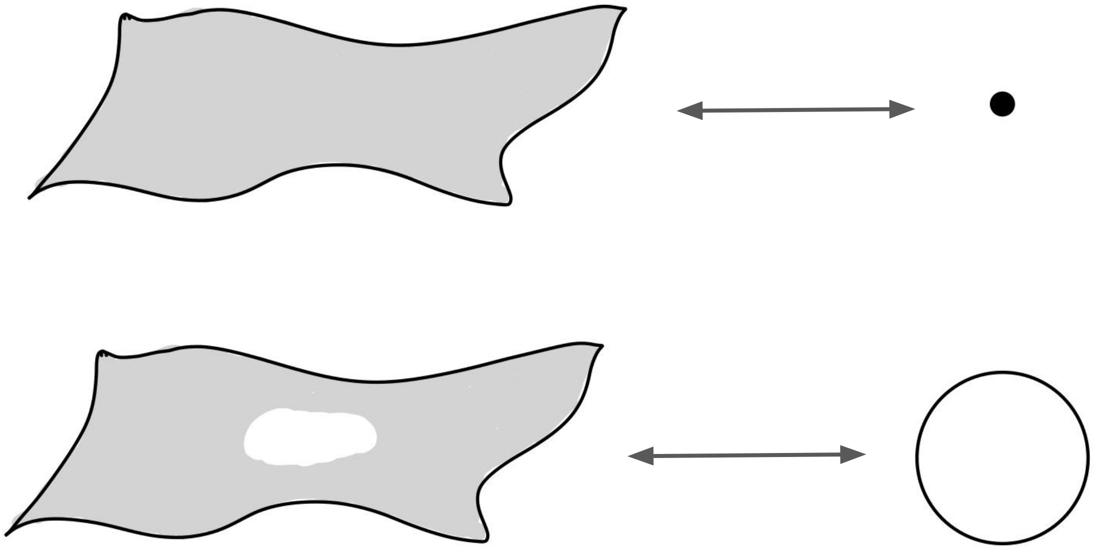
A **simplified** space that maintains some **resemblance** to the original space.



Pablo Picasso, The Bull — 1945

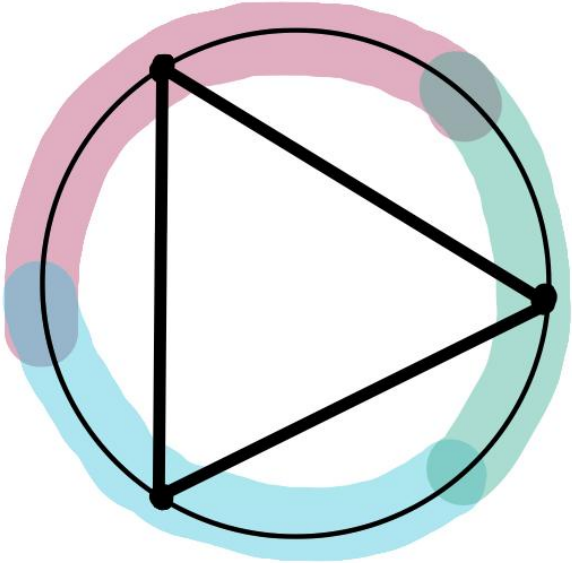
Pooling and Homotopy Equivalence

Spaces that can be transformed into each other by bending, stretching and **contracting**.

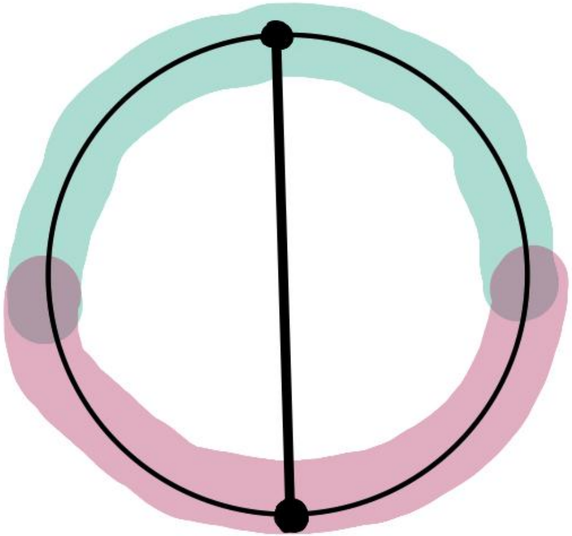


Nerve Theorem (simplified)

If we have a **good cover** of a topological space, then we can construct a simplicial complex based on this cover that is homotopy equivalent to that space.

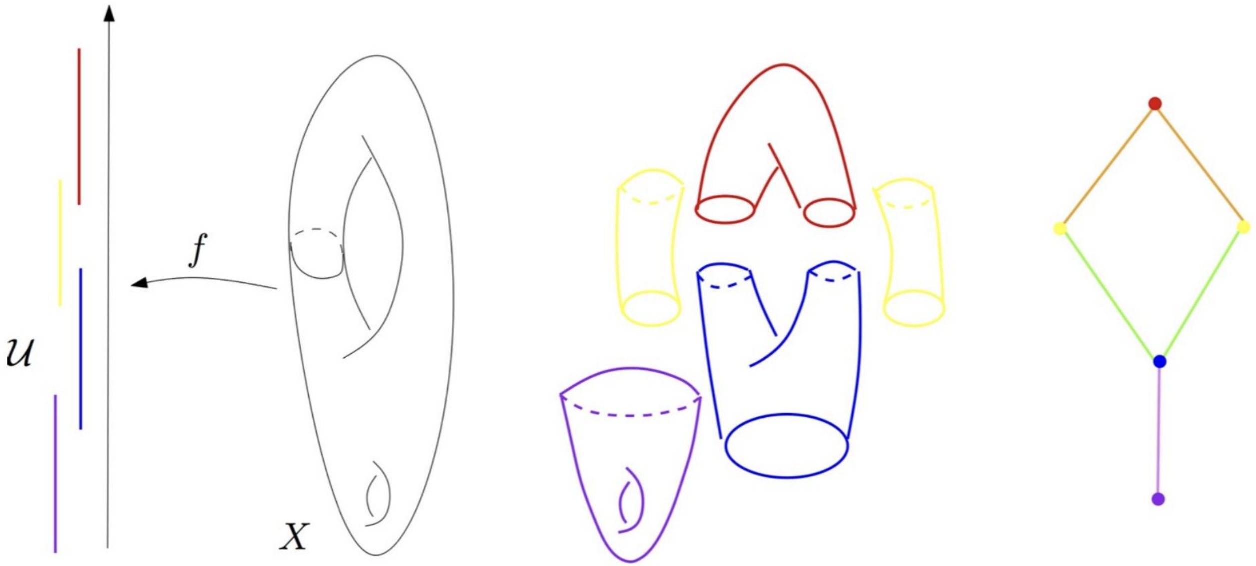


A good cover



A bad cover

Mapper [1]



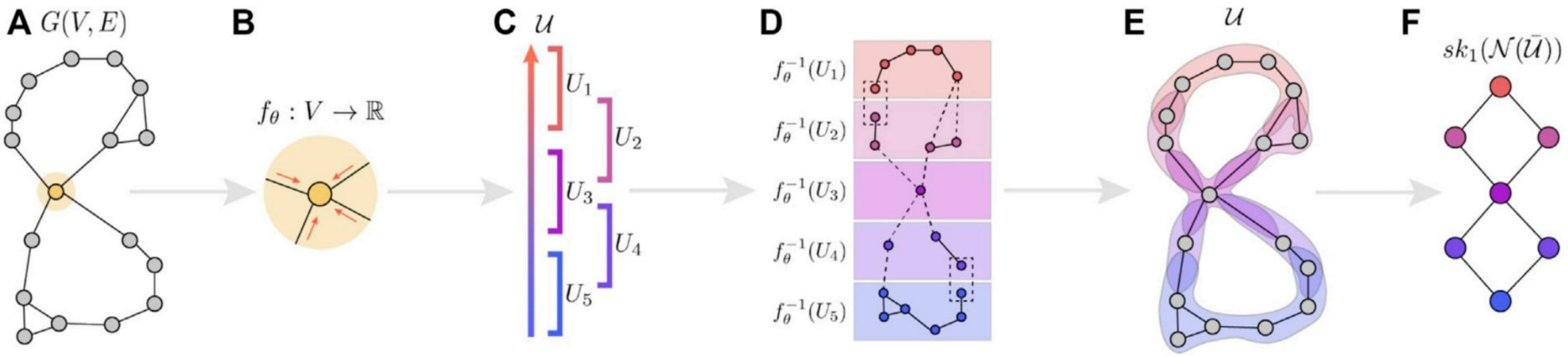
(a)

Image is taken from [2]

[1]: Topological methods for the analysis of high dimensional data sets and 3d object recognition (Singh et al., 2007)
[2]: An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists (Chazal et al., 2017)

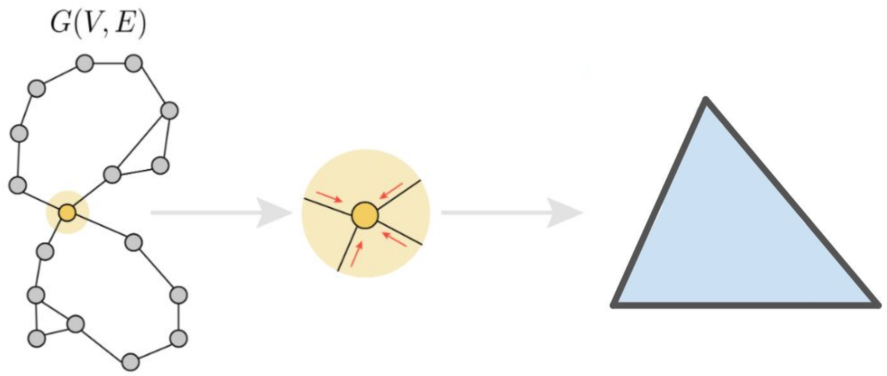
Structural Deep Graph Mapper

We use **GNNs** to compute a **pull-back cover** of the nodes and edges. The output is the **nerve** of the pull-back cover.



A framework for pooling

Proposition. SDGM generalises all approaches based on soft-cluster assignments (e.g. DiffPool [1], minCUT [2]).



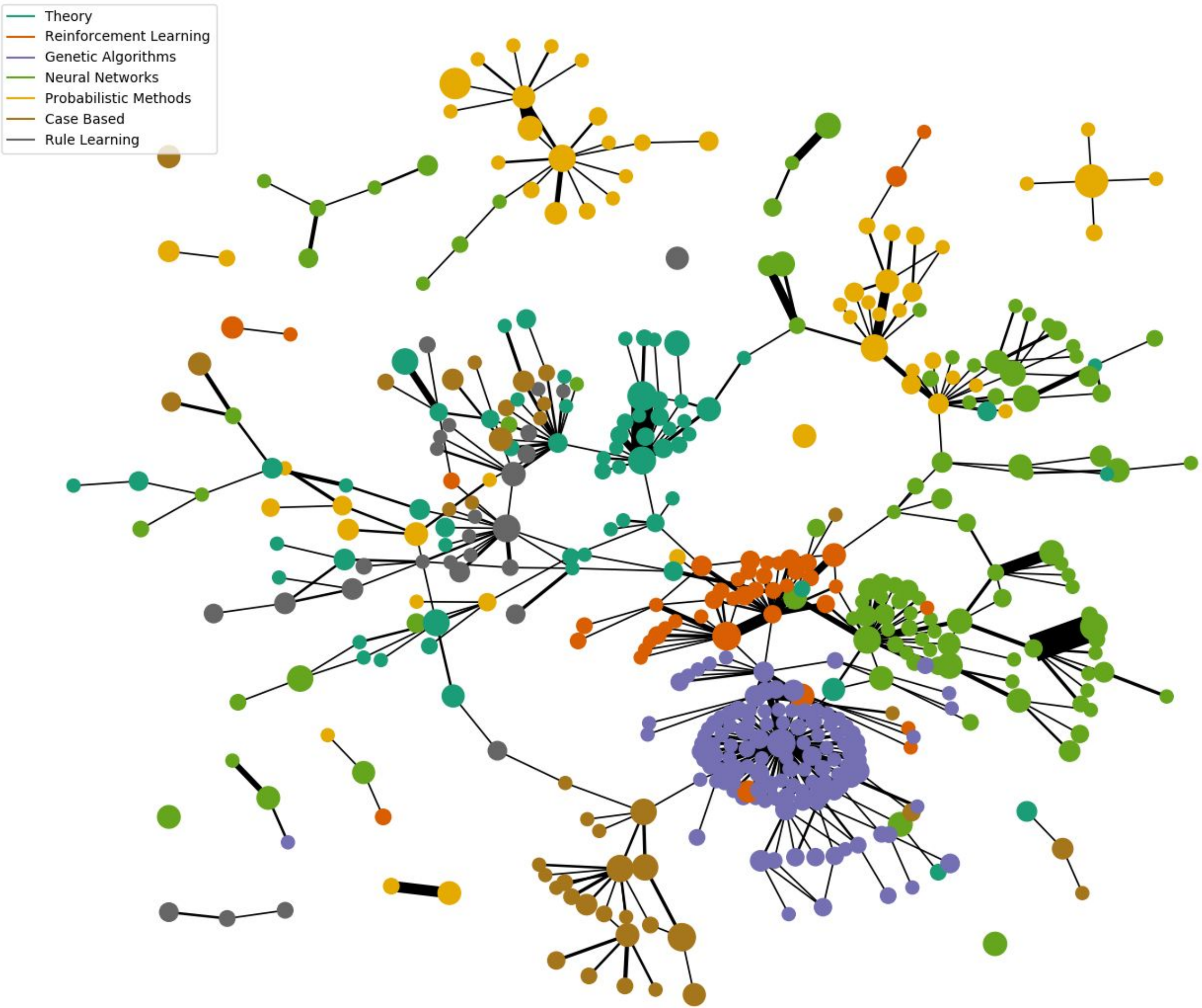
[1]: Hierarchical Graph Representation Learning with Differentiable Pooling (Ying et al., 2018).

[2]: Spectral Clustering with Graph Neural Networks for Graph Pooling (Bianchi et al., 2019).

Graph Classification Results

We propose a **differentiable** pooling operator (DMP) and a **fixed** one based on PageRank (MPR).

	Molecular				Social				
Model	D&D	Mutag	NCI1	Proteins	Collab	IMDB-B	IMDB-M	Reddit-B	Reddit-5k
DMP (Ours)	77.3 ± 3.6	84.0 ± 8.6	70.4 ± 4.2	$\mathbf{75.3 \pm 3.3}$	81.4 ± 1.2	$\mathbf{73.8 \pm 4.5}$	$\mathbf{50.9 \pm 2.5}$	86.2 ± 6.8	51.9 ± 2.1
MPR (Ours)	$\mathbf{78.2 \pm 3.4}$	80.3 ± 6.0	69.8 ± 1.8	75.2 ± 2.2	$\mathbf{81.5 \pm 1.0}$	73.4 ± 2.7	50.6 ± 2.0	86.3 ± 4.8	52.3 ± 1.6
Top-k	75.1 ± 2.2	82.5 ± 6.8	67.9 ± 2.3	74.8 ± 3.0	75.0 ± 1.1	69.6 ± 3.8	45.0 ± 2.8	79.4 ± 7.4	48.5 ± 1.1
minCUT	77.6 ± 3.1	82.9 ± 6.0	68.8 ± 2.1	73.5 ± 2.9	79.9 ± 0.8	70.7 ± 3.5	50.6 ± 2.1	$\mathbf{87.2 \pm 5.0}$	$\mathbf{52.9 \pm 1.3}$
DiffPool	77.9 ± 2.4	$\mathbf{94.7 \pm 7.1}$	68.1 ± 2.1	74.2 ± 0.3	81.3 ± 0.1	72.4 ± 3.1	50.3 ± 1.8	79.0 ± 1.1	50.4 ± 1.7
WL	77.4 ± 2.6	74.5 ± 6.5	$\mathbf{76.4 \pm 2.7}$	74.7 ± 3.2	78.5 ± 1.1	72.1 ± 3.1	50.7 ± 2.9	66.7 ± 10.4	49.2 ± 1.4
Flat	69.9 ± 2.2	71.8 ± 4.3	65.5 ± 1.7	70.2 ± 2.6	80.9 ± 1.4	73.6 ± 4.2	48.5 ± 2.4	70.0 ± 10.8	49.5 ± 1.7
avg-MLP	63.7 ± 1.4	69.1 ± 5.8	55.7 ± 2.8	61.8 ± 1.7	74.8 ± 1.3	71.5 ± 2.9	49.5 ± 2.2	53.6 ± 6.2	45.9 ± 1.6


Theory
Reinforcement Learning
Genetic Algorithms
Neural Networks
Probabilistic Methods
Case Based
Rule Learning

Weisfeiler and Lehman Go Cellular: CW Networks

NeurIPS 2021

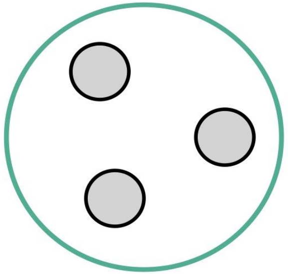
Cristian Bodnar[1,*] Fabrizio Frasca[2,3,*] Nina Otter[7]
Yu Guang Wang[4,5,6*] Pietro Liò[1] Guido Montúfar[4,7,*] Michael Bronstein[2,3]

UNIVERSITY OF CAMBRIDGE [1] [2] Imperial College London [3] MAX-PLANCK-GESELLSCHAFT [4] 上海交通大學 SHANGHAI JIAO TONG UNIVERSITY [5] UNSW SYDNEY [6] UCLA [7] erc

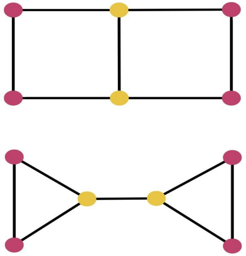
Based on **arxiv:2106.12575**, with elements from **arxiv:2103.03212**

Motivation: Limitations of GNNs

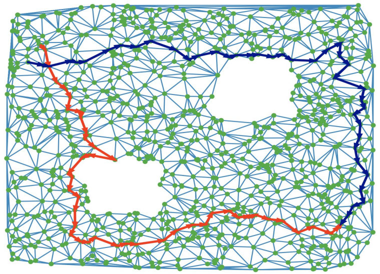
Groupwise interactions



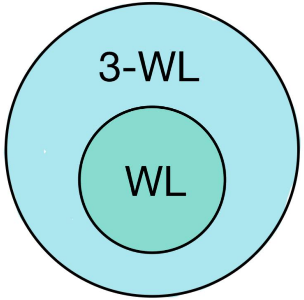
Higher-order structures



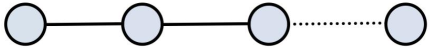
Higher-order signals



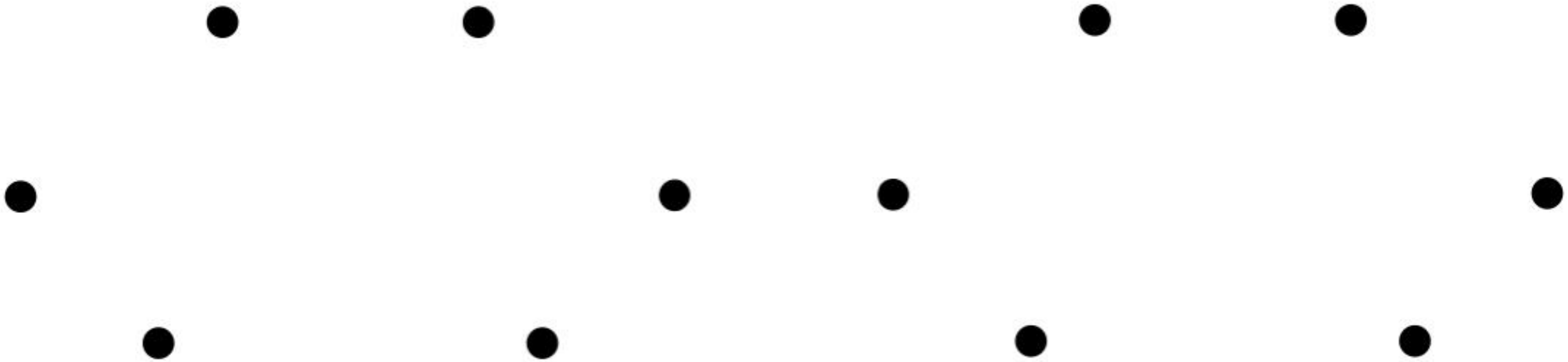
Expressive power



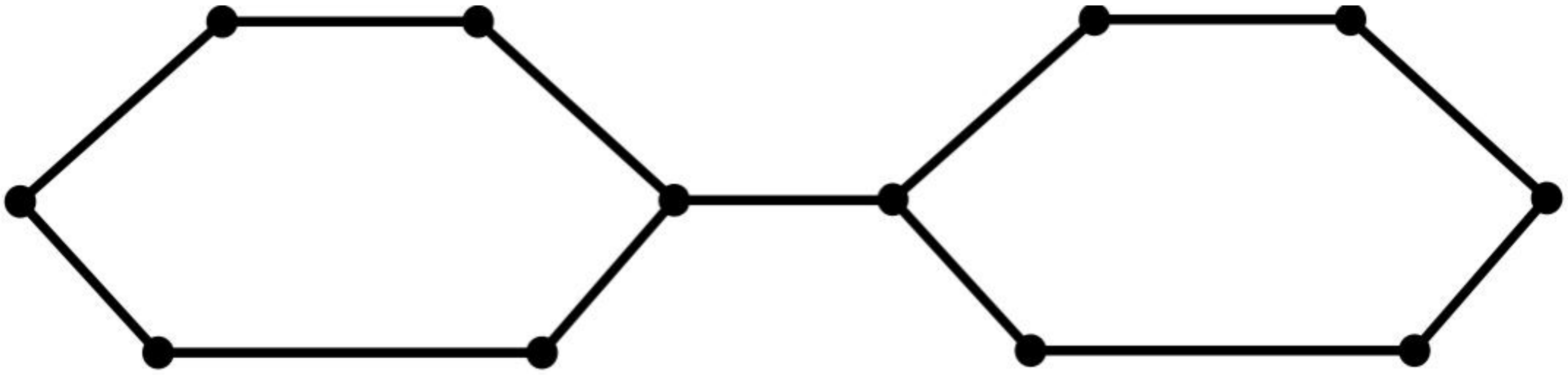
Long-range interactions



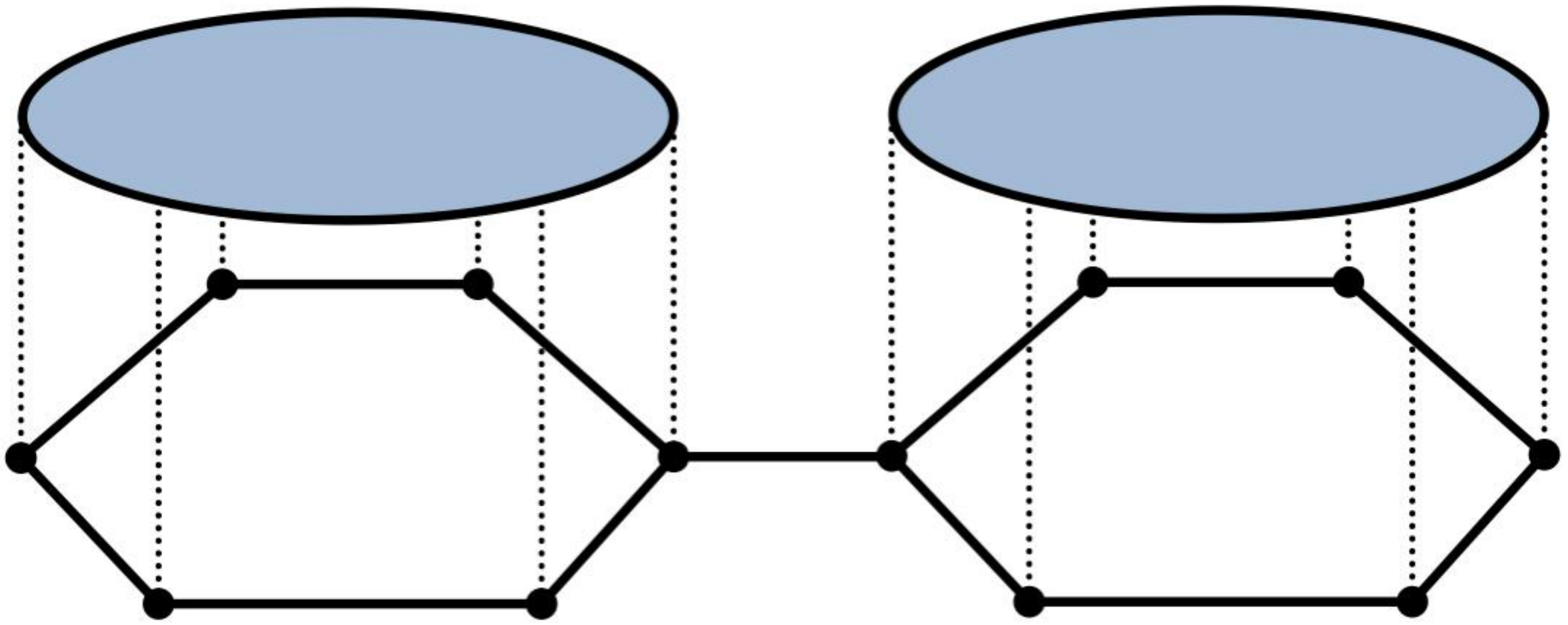
CW Complexes



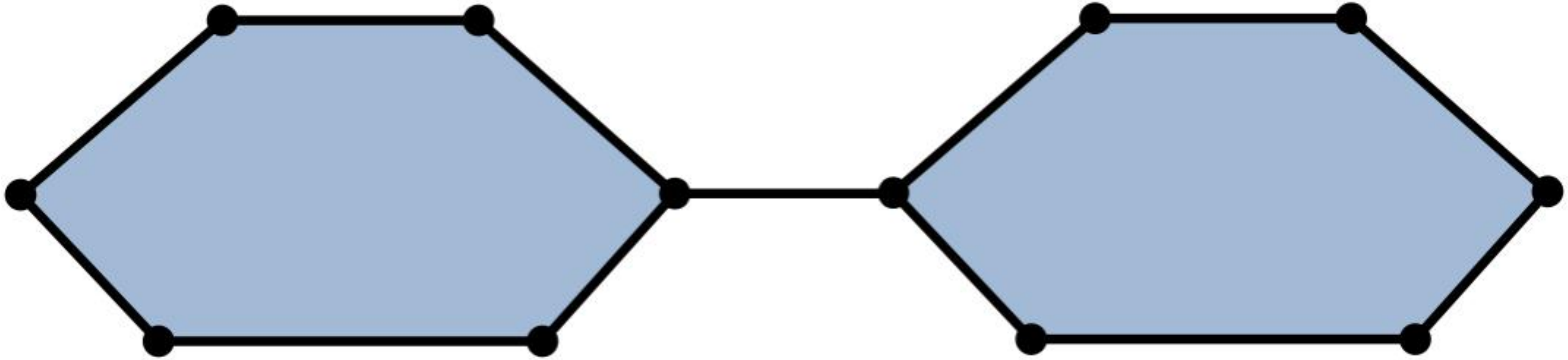
CW Complexes



CW Complexes



CW Complexes



$\tau \leq \sigma \Leftrightarrow$ Cell τ is on the boundary of cell σ

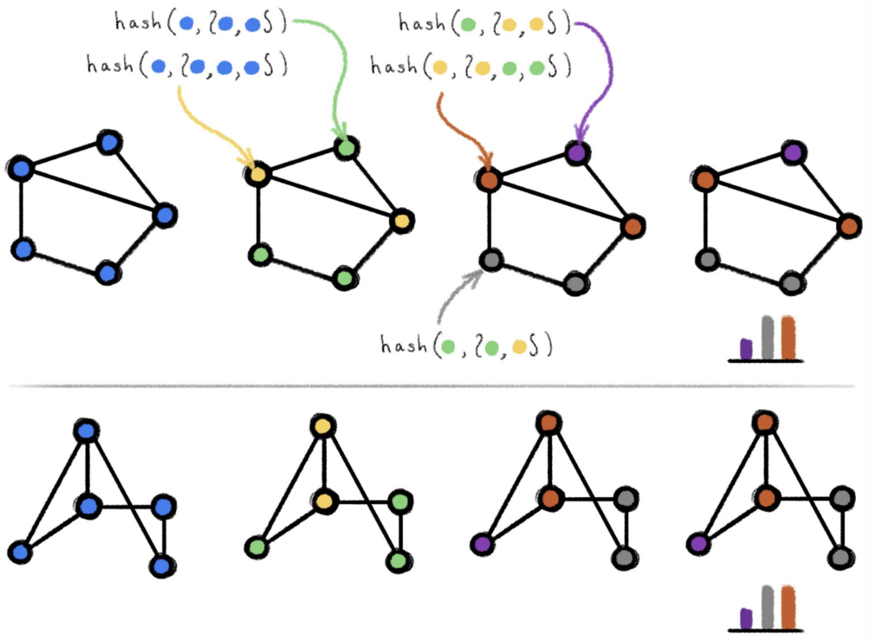
Background: The Weisfeiler-Lehman (WL) algorithm

The WL algorithm tests isomorphism between graphs [1].

It iterates *colour refinements* via hashing of neighboring colours:

$$c_v^{t+1} = \mathrm{HASH}\left(c_v^t, \{\!\{c_u^t\}\!\}_{u \in \mathcal{N}(v)}\right)$$

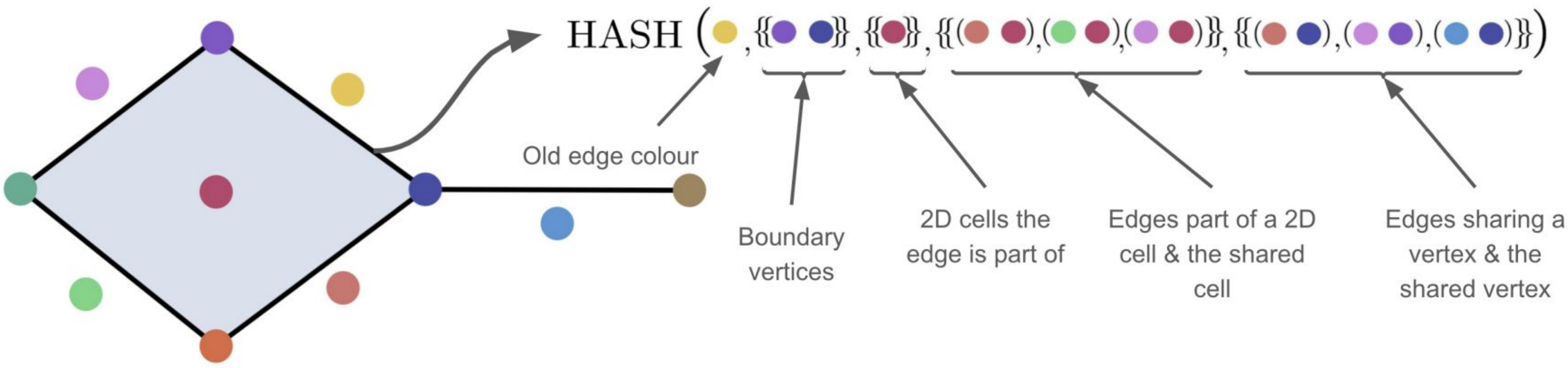
The WL test cannot distinguish all graphs.



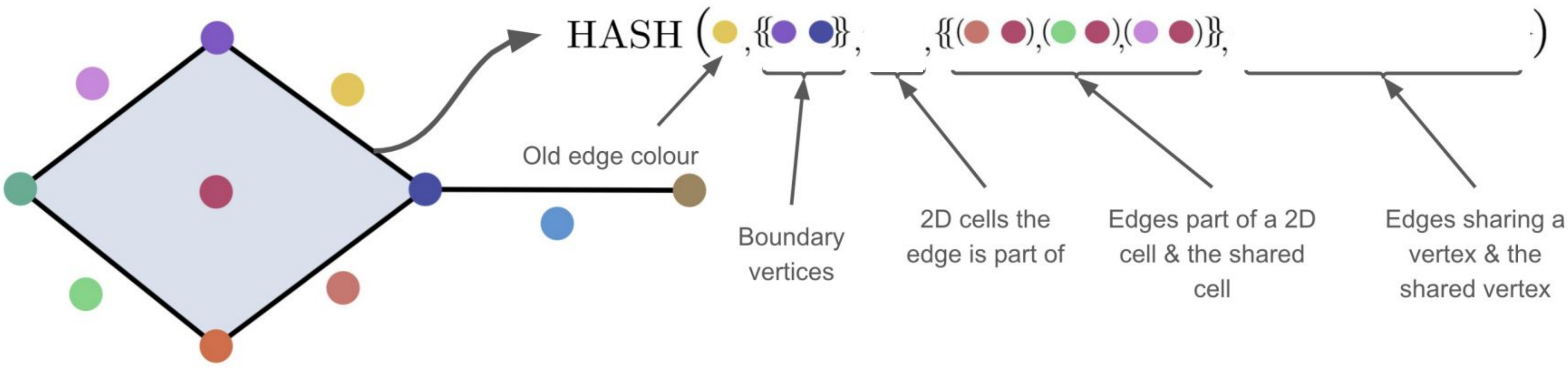
Example of execution of the WL test on two isomorphic graphs [2]

[1]: The reduction of a graph to canonical form and the algebra which appears therein (Weisfeiler and Lehman, NTI Series, 1968)
[2]: [Expressive power of graph neural networks and the Weisfeiler-Lehman test](), blogpost

Cellular Weisfeiler-Lehman

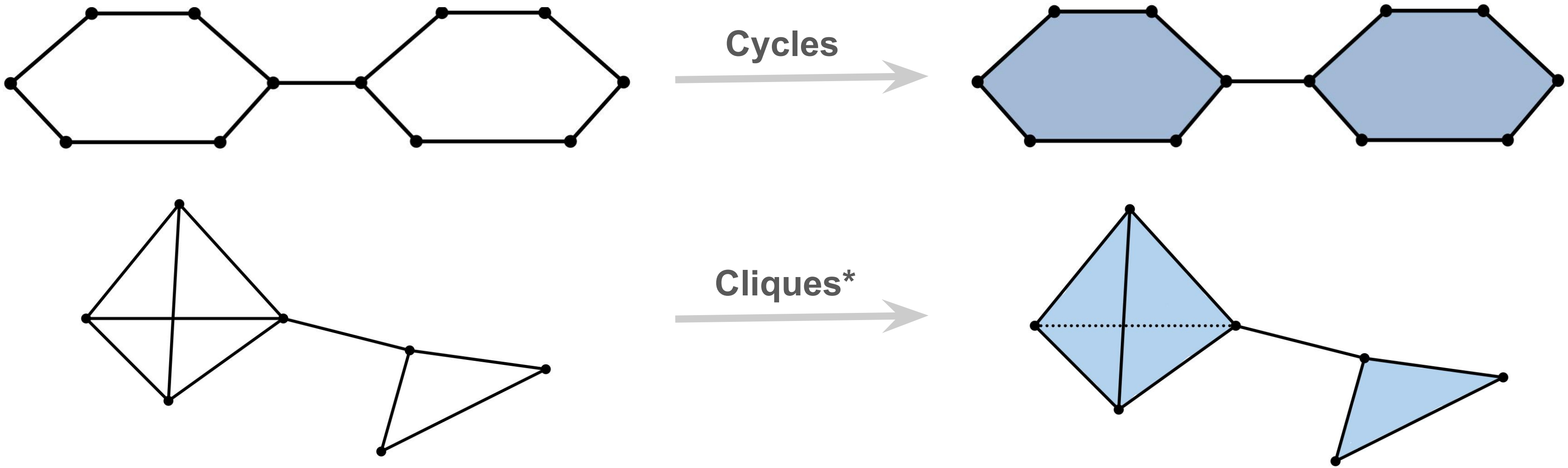


Cellular Weisfeiler-Lehman



Theorem 7. *CWL without coboundary and lower-adjacencies has the same expressive power in distinguishing non-isomorphic cell complexes as CWL with the complete set of adjacencies.*

Cellular Lifting Maps

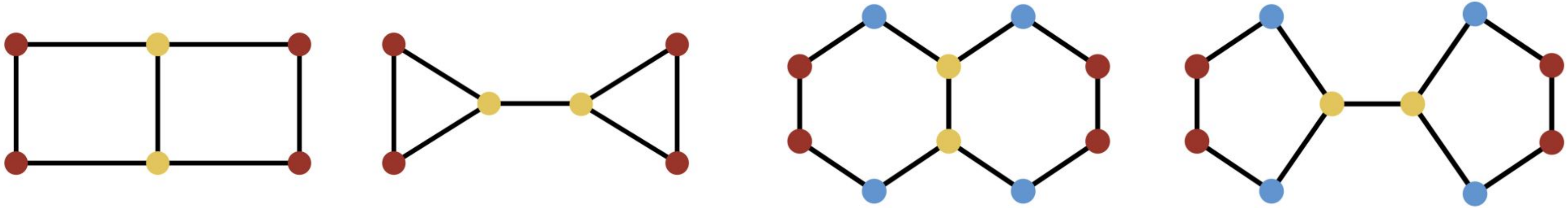


Theorem 13. *Let f be a skeleton-preserving lifting map. Then CWL using lifting f is at least as powerful as WL in distinguishing non-isomorphic graphs.*

*Studied in **arxiv:2103.03212: Weisfeiler and Lehman Go Topological**

Cellular Lifting Maps

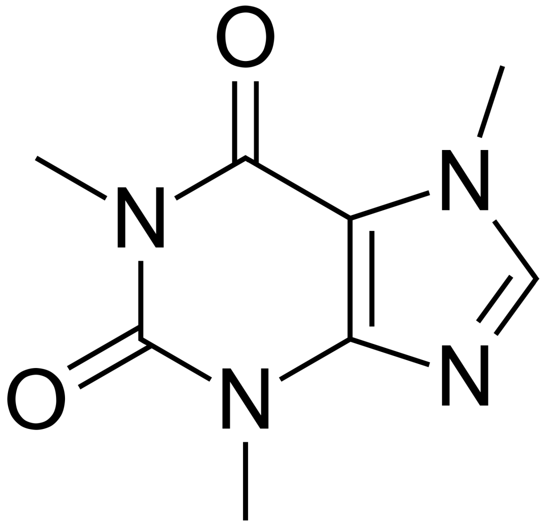
Corollary 14. *For all $k \geq 3$, the following lifting transformations make CWL strictly more powerful than the WL test. (1) The clique complex lifting considering cliques of size at most k. (2) The map that attaches 2-cells to all the simple cycles of size at most k. (3) The map that attaches 2-cells to all the induced cycles of size at most k. (4) The union of all the transformations above.*

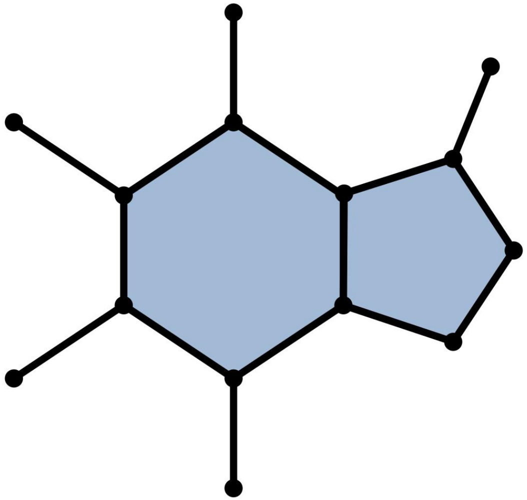


Theorem 15. *For some finite k, there exists a pair of graphs indistinguishable by 3-WL but distinguishable by CWL with the lifting maps from Corollary 14. For the clique complex and induced cycle liftings, the statement holds for $k \geq 4$. For the simple cycle based lifting, it holds for $k \geq 8$.*

Molecules as Cell Complexes

Graph representations of molecules date back to the nineteenth century [1]. However, it is not necessarily the best representation. We propose modelling **molecules as cell complexes**.





[1]: Danail Bonchev, D.H. Rouvray (eds.) (1991) "Chemical Graph Theory: Introduction and Fundamentals"

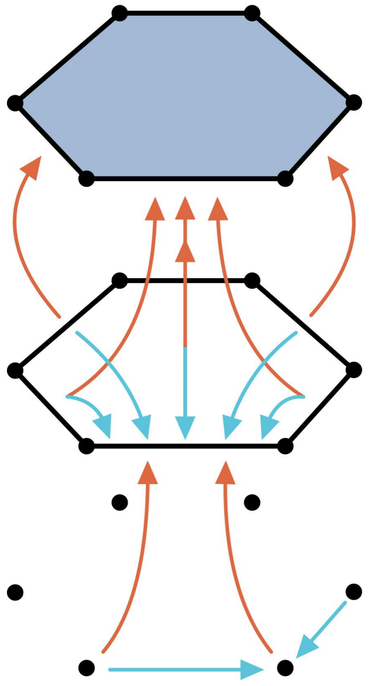
CW Networks and Molecular Message Passing

$$m_{\mathcal{B}}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{B}(\sigma)}\Big(M_{\mathcal{B}}\big(h_\sigma^t, h_\tau^t\big)\Big)$$

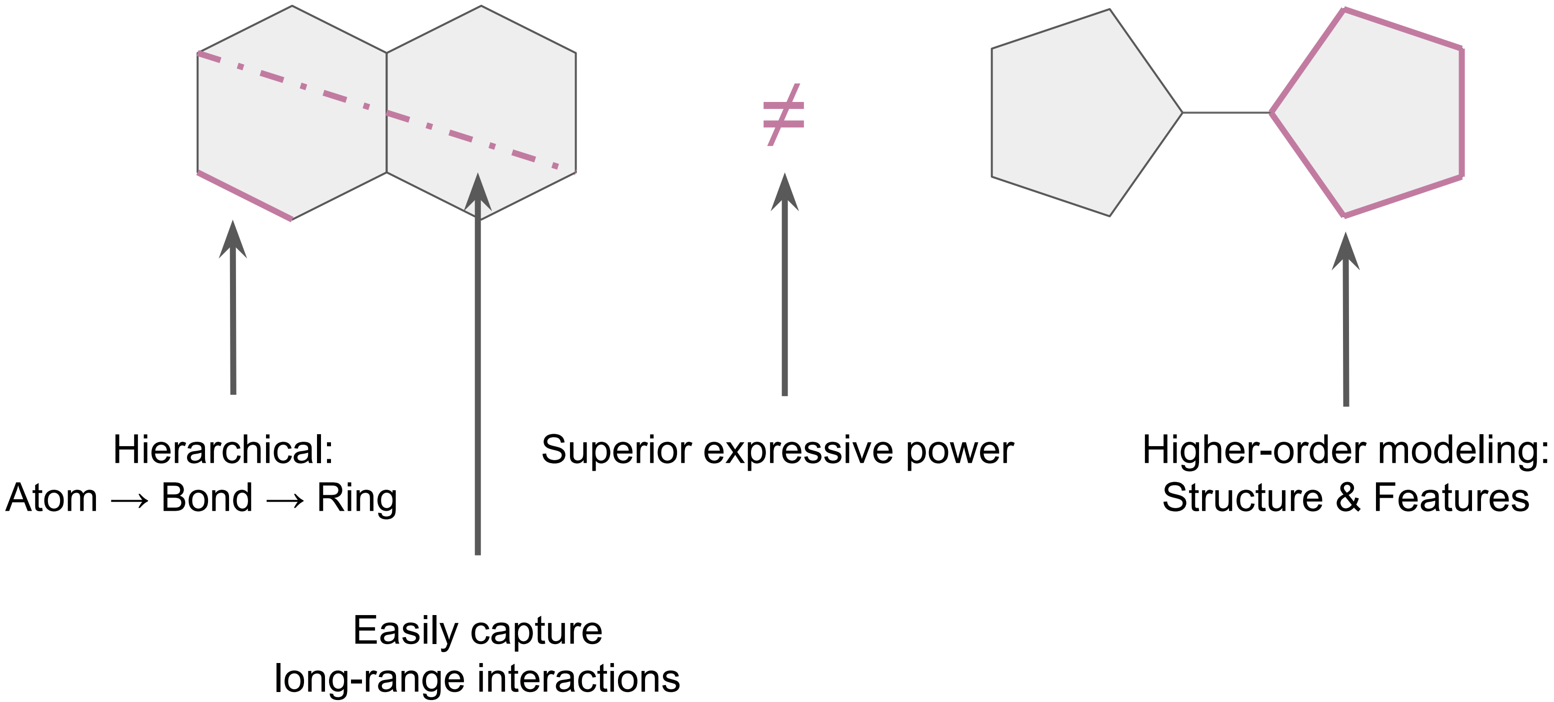
$$m_{\uparrow}^{t+1}(\sigma) = \mathrm{AGG}_{\tau \in \mathcal{N}_{\uparrow}(\sigma), \delta \in \mathcal{C}(\sigma,\tau)}\Big(M_{\uparrow}\big(h_\sigma^t, h_\tau^t, h_\delta^t\big)\Big)$$

$$h_\sigma^{t+1} = U\Big(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_{\uparrow}^{t+1}(\sigma)\Big)$$

$$h_X = \mathrm{READOUT}(\{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=0}, \{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=1}, \{\!\{h_\sigma^L\}\!\}_{dim(\sigma)=2})$$



Practical Benefits of CWNs



Computational Complexity

(Message passing)

$$\mathcal{O}\Big(\sum_{p=1}^{d} \underbrace{\overbrace{B_p}^{\text{Max boundary size}} S_p}_{\text{Boundary msgs}} + \underbrace{2 * \binom{B_p}{2} \overbrace{S_p}^{\text{Number of p-cells}}}_{\text{Upper msgs}}\Big) \dashrightarrow \Theta\Big(\sum_{p=1}^{d} S_p\Big)$$

In practice: B_p, d fixed (small) constants

Empirically, GIN is only 19% – 35% faster at inference time (ZINC)

(Lifting)

linear trend →

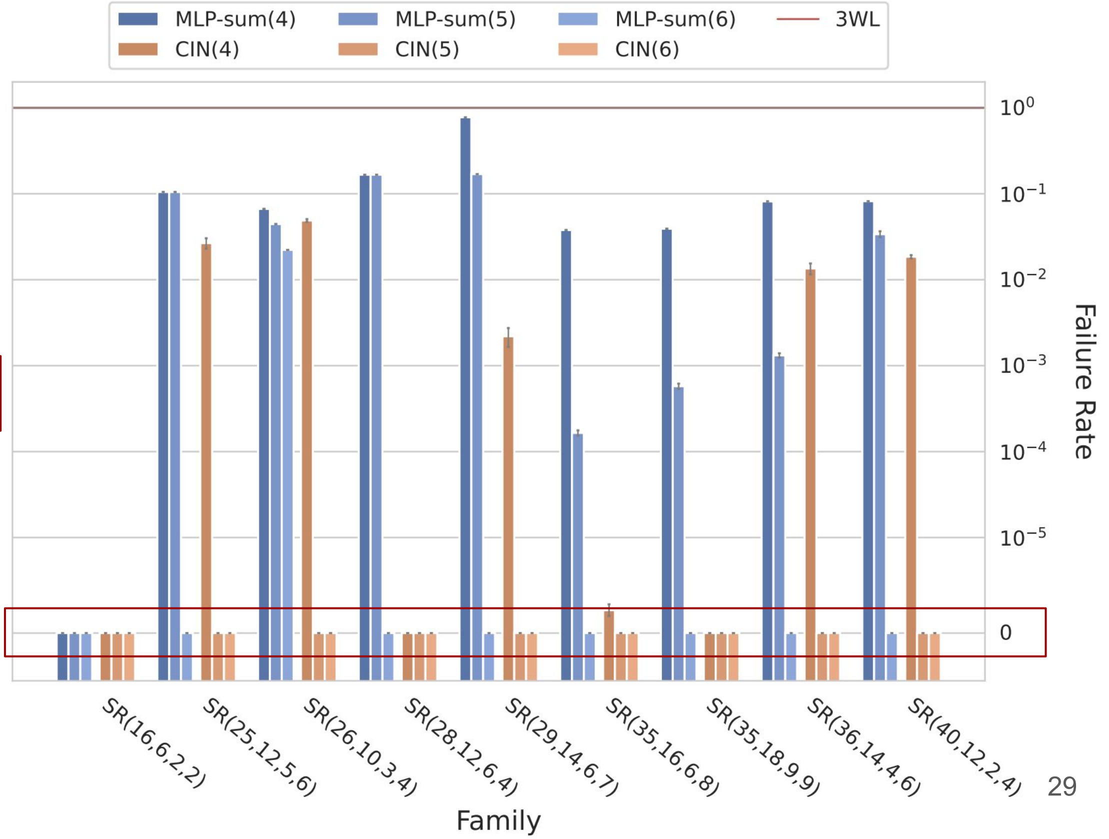
Dataset ↓ / Processes →	Seq.	2	4	8	16	32
ZINC (12k)	320.27 ± 0.54	169.95 ± 0.32	84.90 ± 0.21	43.38 ± 0.07	23.17 ± 0.68	18.59 ± 0.68
Mol-HIV (41k)	1178.98 ± 3.90	635.58 ± 0.83	319.01 ± 0.40	164.26 ± 0.52	86.92 ± 0.77	60.62 ± 2.05
ZINC-FULL (250k)	6805.35 ± 16.50	3549.16 ± 7.73	1782.41 ± 3.84	918.38 ± 3.46	492.77 ± 6.13	383.92 ± 3.30

Synthetic Experiments: Expressive Power

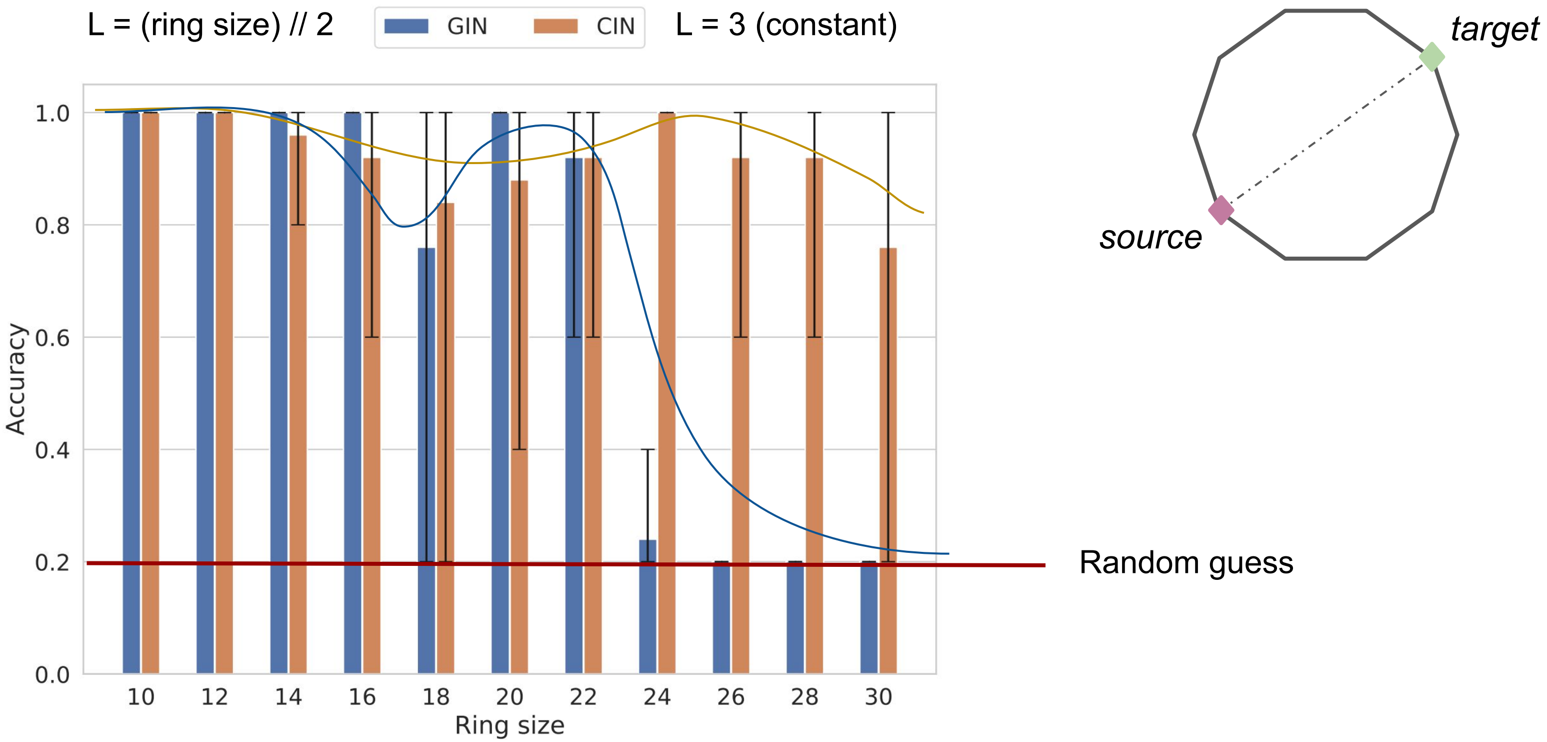
Circular Skip Links

Method	Mean	Min	Max
MP-GNNs	10.000±0.000	10.000	10.000
RingGNN	10.000±0.000	10.000	10.000
3WLGNN	97.800±10.916	30.000	100.000
CIN (Ours)	100.000±0.000	100.000	100.000

Strongly Regular



Synthetic Experiments: RingTransfer



Real-world Experiments: TUDatasets

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M	RDT-B
RWK [28]	79.2±2.1	55.9±0.3	59.6±0.1	>3 days	N/A	N/A	N/A	N/A
GK ($k = 3$) [62]	81.4±1.7	55.7±0.5	71.4±0.31	62.5±0.3	62.4±0.3	N/A	N/A	N/A
PK [56]	76.0±2.7	59.5±2.4	73.7±0.7	82.5±0.5	N/A	N/A	N/A	N/A
WL kernel [63]	90.4±5.7	59.9±4.3	75.0±3.1	**86.0**±1.8	N/A	73.8±3.9	50.9±3.8	81.0±3.1
DCNN [3]	N/A	N/A	61.3±1.6	56.6±1.0	N/A	49.1±1.4	33.5±1.4	N/A
DGCNN [74]	85.8±1.8	58.6±2.5	75.5±0.9	74.4±0.5	N/A	70.0±0.9	47.8±0.9	N/A
IGN [50]	83.9±13.0	58.5±6.9	**76.6**±5.5	74.3±2.7	**72.8**±1.5	72.0±5.5	48.7±3.4	N/A
GIN [72]	89.4±5.6	64.6±7.0	76.2±2.8	82.7±1.7	N/A	75.1±5.1	52.3±2.8	**92.4**±2.5
PPGNs [51]	**90.6**±8.7	66.2±6.6	**77.2**±4.7	83.2±1.1	**82.2**±1.4	73.0±5.8	50.5±3.6	N/A
Natural GN [20]	89.4±1.6	**66.8**±1.7	71.7±1.0	82.4±1.3	N/A	73.5±2.0	51.3±1.5	N/A
GSN [9]	**92.2** ± 7.5	**68.2** ± 7.2	**76.6** ± 5.0	**83.5** ± 2.0	N/A	**77.8** ± 3.3	**54.3** ± 3.3	N/A
SIN [7]	N/A	N/A	76.4 ± 3.3	82.7 ± 2.1	N/A	**75.6** ± 3.2	**52.4** ± 2.9	**92.2** ± 1.0
CIN (Ours)	**92.7** ± 6.1	**68.2** ± 5.6	**77.0** ± 4.3	**83.6** ± 1.4	**84.0** ± 1.6	**75.6** ± 3.7	**52.7** ± 3.1	**92.4** ± 2.1

Real-world Experiments: ZINC + MolHIV

Table 3: ZINC (MAE), ZINC-FULL (MAE) and Mol-HIV (ROC-AUC).

Method	ZINC ↓ No Edge Feat.	ZINC ↓ With Edge Feat.	ZINC-FULL ↓ All methods	MOLHIV ↑ All methods
GCN [45]	0.469±0.002	N/A	N/A	76.06±0.97
GAT [67]	0.463±0.002	N/A	N/A	N/A
GatedGCN [10]	0.422±0.006	0.363±0.009	N/A	N/A
GIN [72]	0.408±0.008	0.252±0.014	0.088±0.002	77.07±1.49
PNA [19]	0.320±0.032	0.188±0.004	N/A	79.05±1.32
DGN [5]	0.219±0.010	0.168±0.003	N/A	79.70±0.97
• HIMP [26]	N/A	0.151±0.006	0.036±0.002	78.80±0.82
• GSN [9]	0.139±0.007	0.108±0.018	N/A	77.99±1.00
CIN-small (Ours)	0.139±0.008	0.094±0.004	0.044±0.003	80.55±1.04
CIN (Ours)	**0.115±0.003**	**0.079±0.006**	**0.022±0.002**	**80.94±0.57**

Real-world Experiments: ZINC (ablation study)

Method	MAE
GatedGCN [10]	0.363±0.009
GIN [72]	0.252±0.014
PNA [19]	0.188±0.004
DGN [5]	0.168±0.003
HIMP [26]	0.151±0.006
GSN [9]	0.108±0.018
GIN-E Custom	0.196±0.007
CIN No-Rings small	0.174±0.006
CIN No-Rings	0.159±0.007
CIN-small	0.094±0.004
CIN	**0.079±0.006**

- "Standard"
- Updates edges
- Captures rings

Importance of updating edges

Importance of capturing rings

Thanks!

Cristian Bodnar
Twitter: @CristianBodnar
Email: cb2015@cam.ac.uk