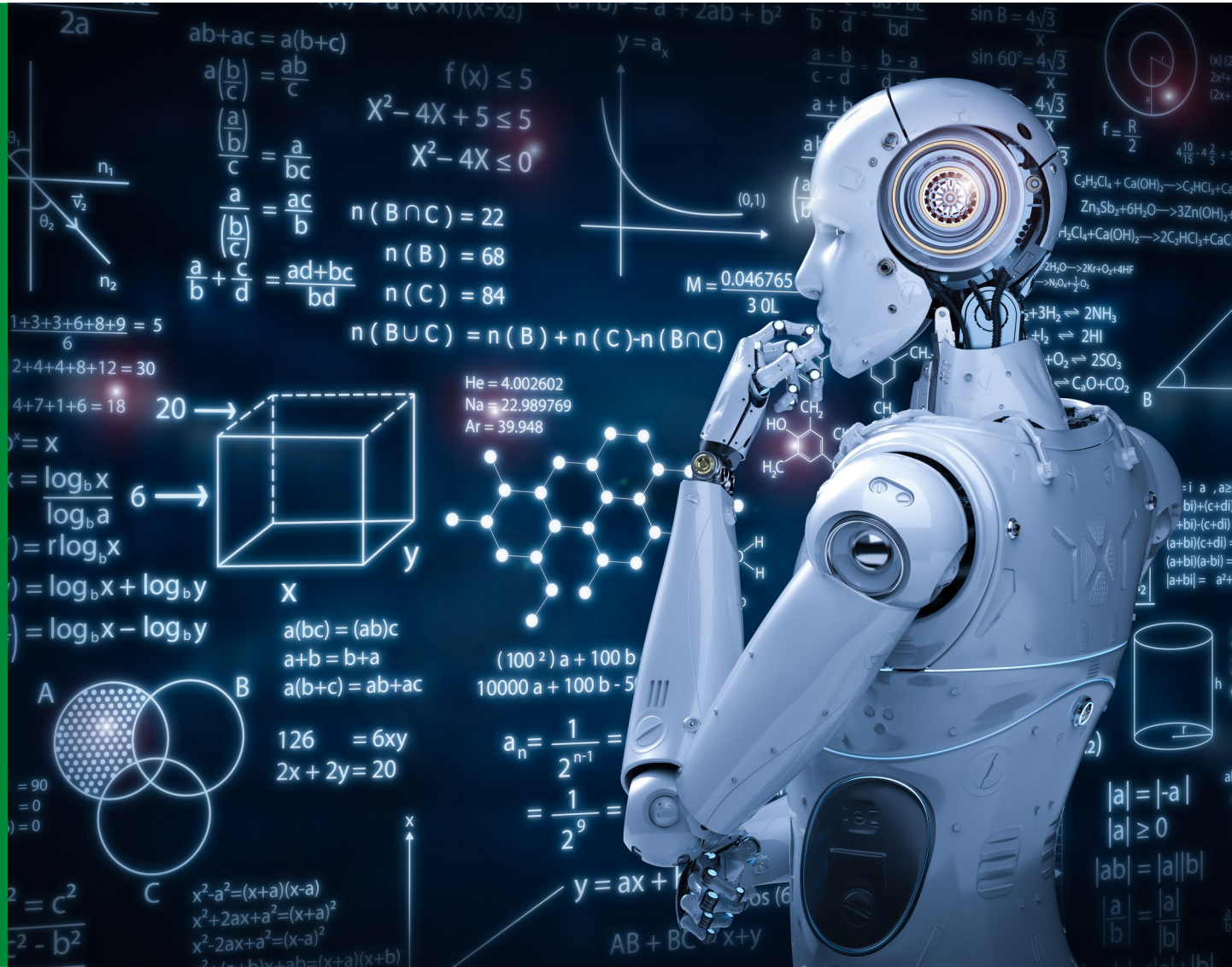
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

FACULTY FOR MATHEMATICS, INFORMATICS, AND STATISTICS
DEPARTMENT OF MATHEMATICS
BAVARIAN AI CHAIR "MATHEMATICAL FOUNDATIONS OF ARTIFICIAL INTELLIGENCE"

Weisfeiler and Leman Go Loopy

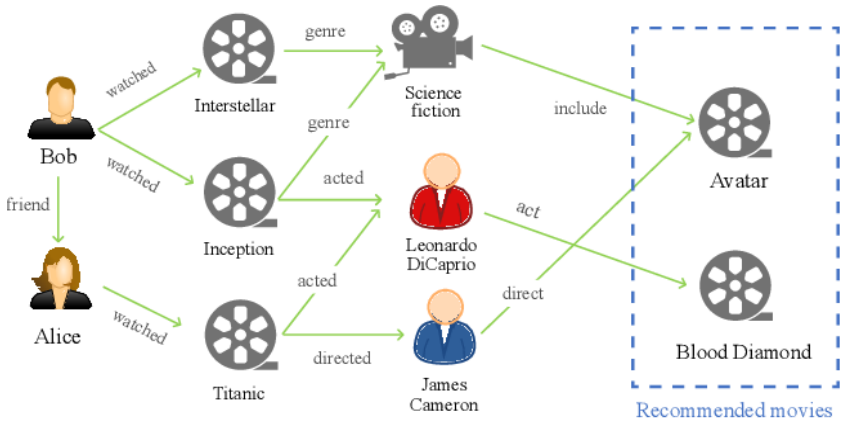
Raffaele Paolino, Sohir Maskey, Pascal Welke & Gitta Kutyniok



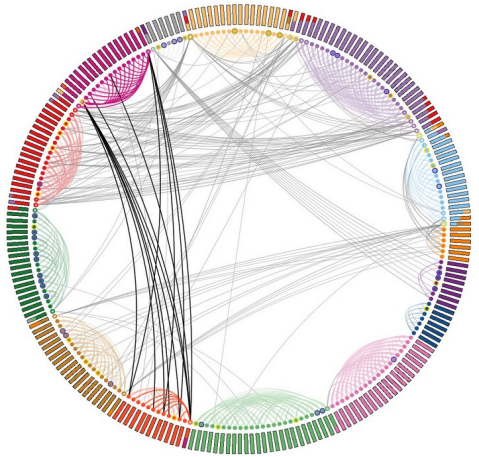
The Curret Impact of Graph Neural Networks

Social Networks

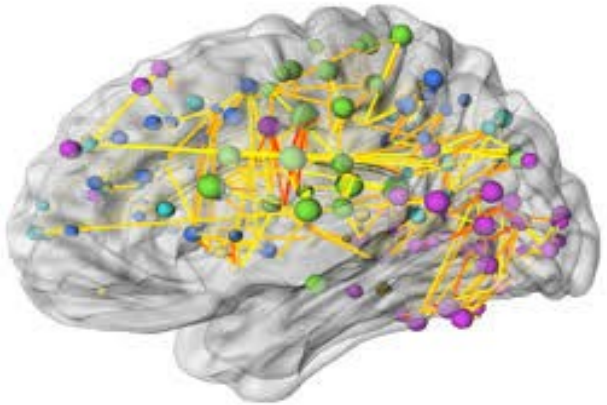
Recommender Systems



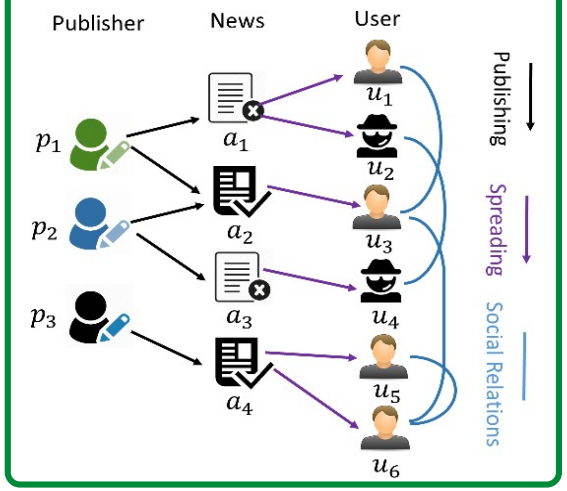
Drug Discovery



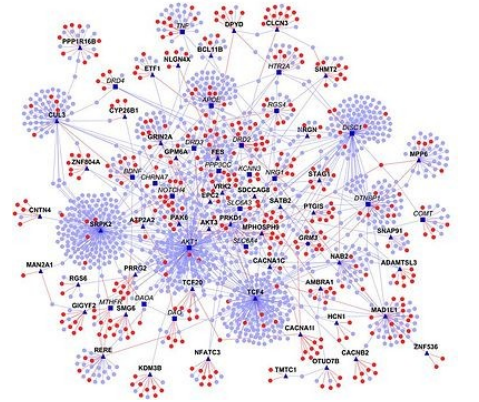
Functional Brain Networks



Fake News Detection

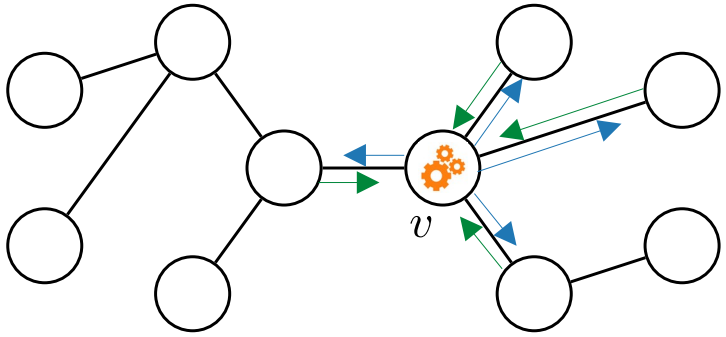


Drug Design



Gilmer, J., et al. (2017). *Neural message passing for quantum chemistry.*
Wang, J., et al. (2018). *Billion-scale commodity embedding for e-commerce recommendation in alibaba.*
Monti, F., et al. (2019). *Fake news detection on social media using geometric deep learning.*
Fan, W., et al. (2019). *Graph neural networks for social recommendation.*

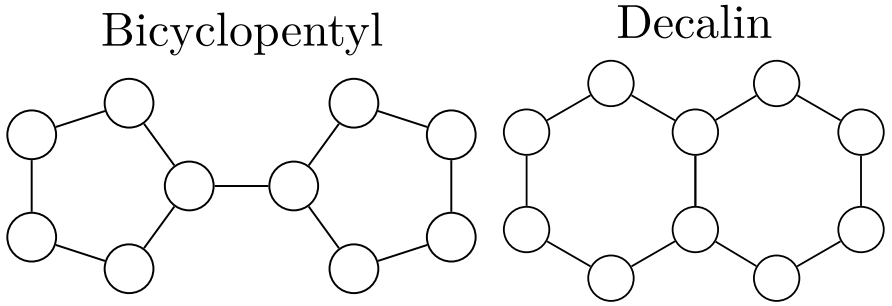
The Curret Weaknesses of Graph Neural Networks



$$x_v^{(k+1)} = \mathrm{MLP}^{(k+1)}\left(\left(1+\varepsilon^{(k+1)}\right)x_v^{(k)} + \sum_{j\in\mathcal{N}(v)} x_j^{(k)}\right)$$

→ The paradigm of GNN is **message passing**.

- Messages are collected from neighboring nodes.
- Messages are used to update the node features'.
- The new message is sent to neighboring nodes.



→ Message-passing cannot separate all non-isomorphic graphs.

- They cannot approximate all functions on graphs.
- No universal approximation theorem.

Which are the graphs that Message Passing Neural Networks can distinguish?

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). *How Powerful Are Graph Neural Networks?*

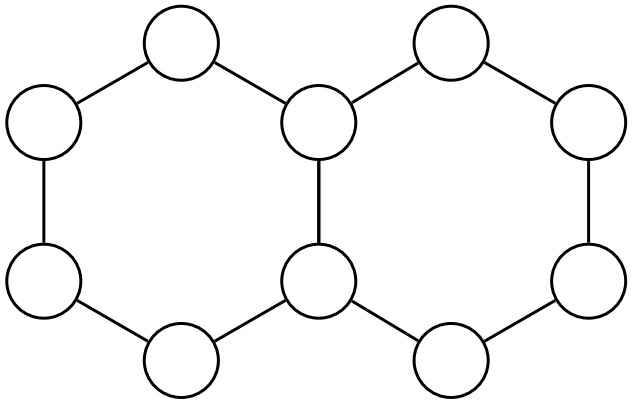
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Weisfeiler-Leman Graph Isomorphism Test

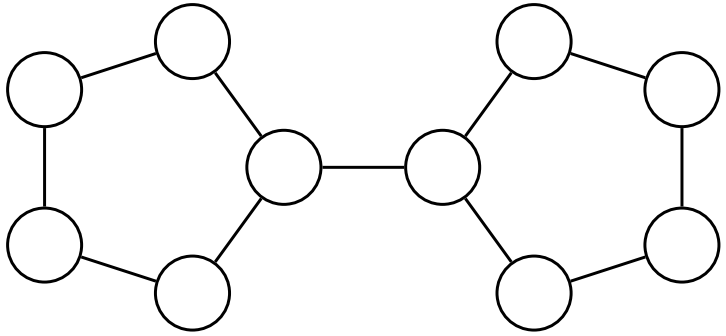
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



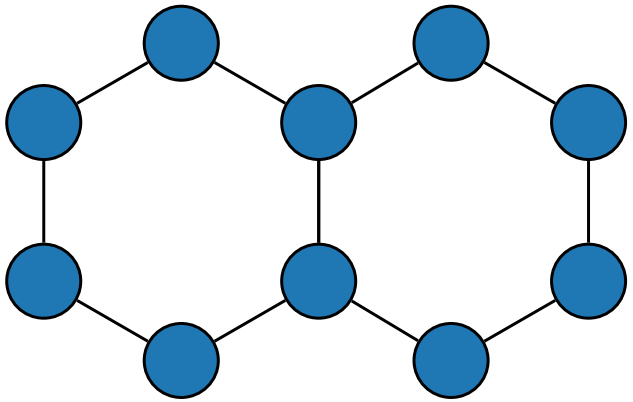
Bicyclopentyl



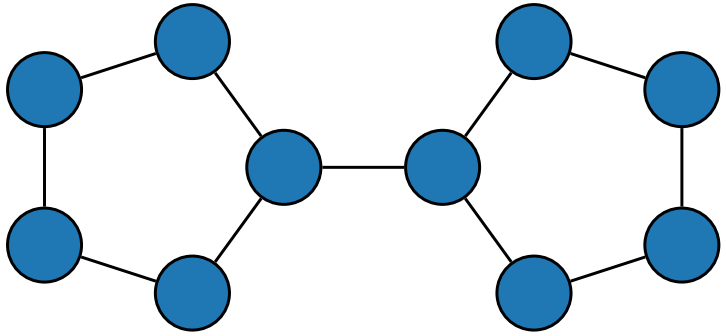
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl

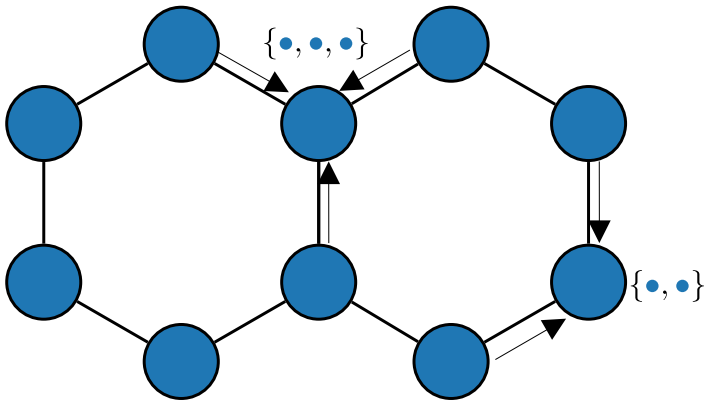


1. Assign to each node the same label •.

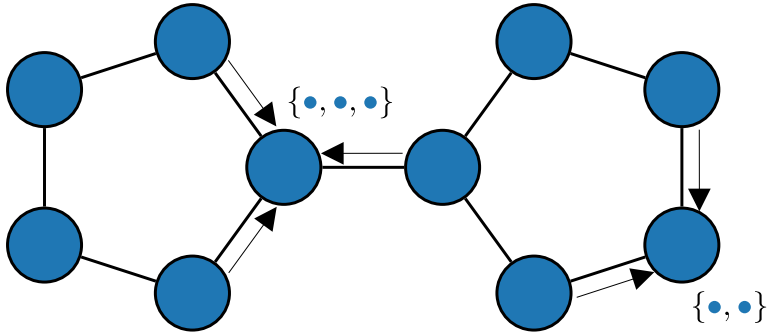
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl

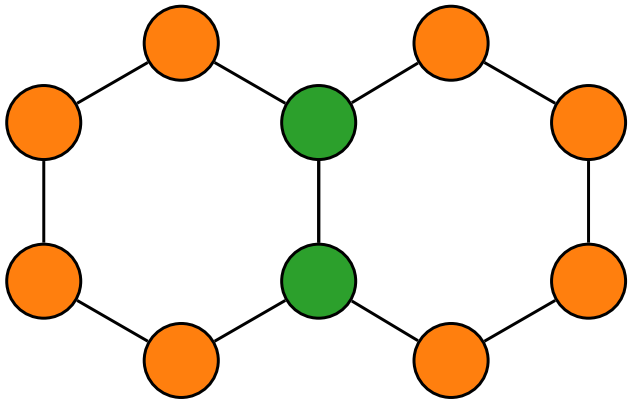


1. Assign to each node the same label •.
2. Collect all labels from direct neighbors.

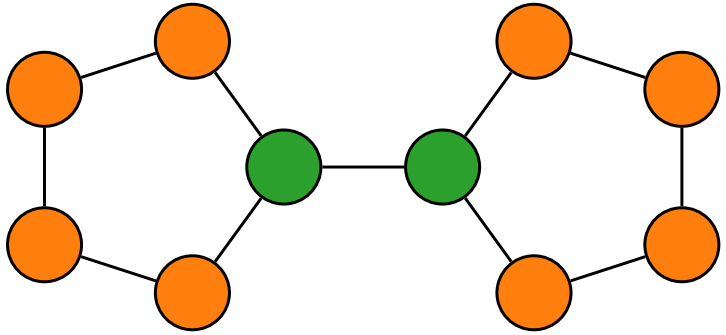
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl

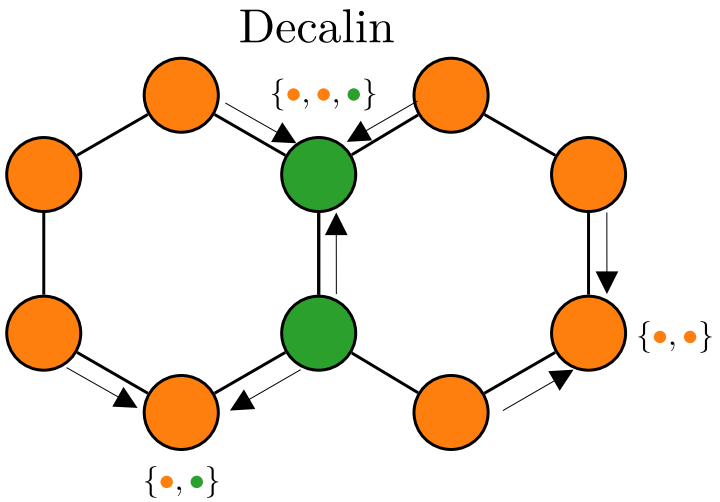


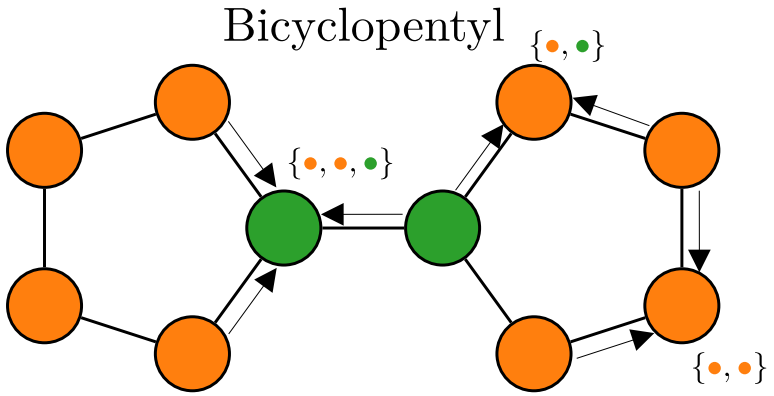
1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

$\{\bullet, \bullet\} \mapsto \bullet \qquad \{\bullet, \bullet, \bullet\} \mapsto \bullet$

Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic





1. Assign to each node the same label •.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

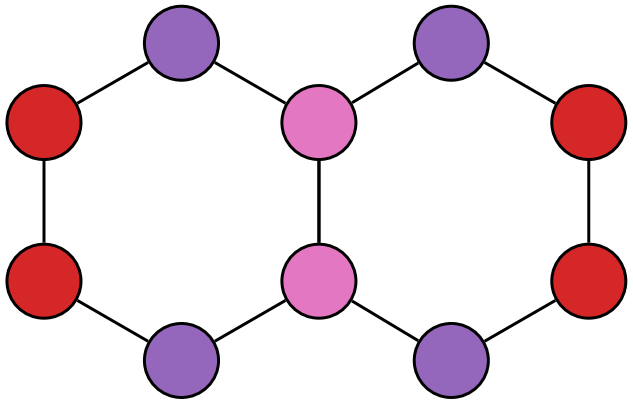
$$\{\bullet, \bullet\} \mapsto \bullet \qquad \{\bullet, \bullet, \bullet\} \mapsto \bullet$$

4. Repeat until there is no change in the partition of nodes with same label.

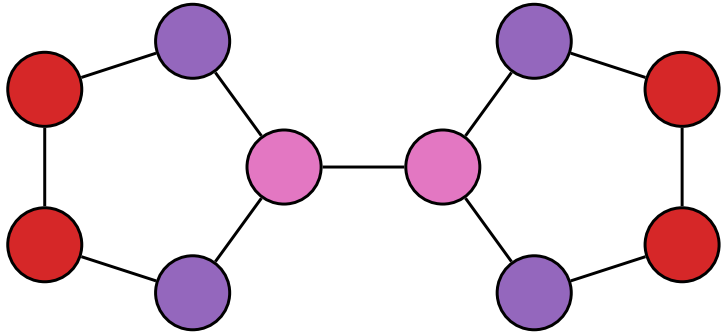
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl



1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

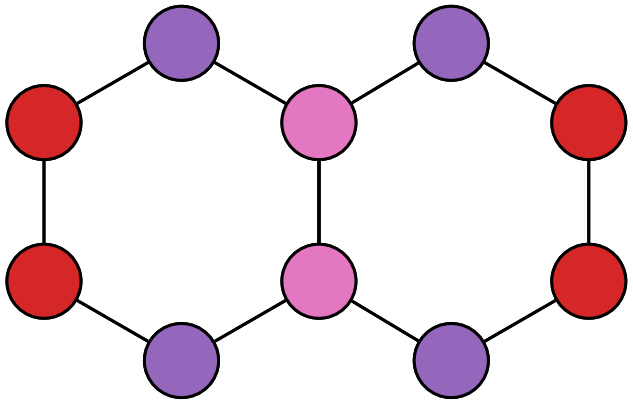
$$\{●, ●\} \mapsto ● \quad \{●, ●\} \mapsto ● \quad \{●, ●, ●\} \mapsto ●$$

4. Repeat until there is no change in the partition of nodes with same label.

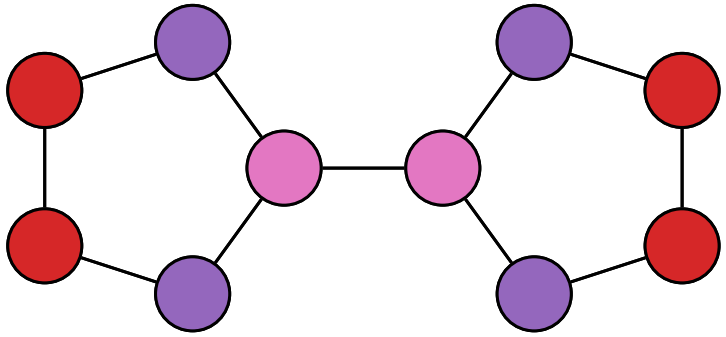
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl



1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

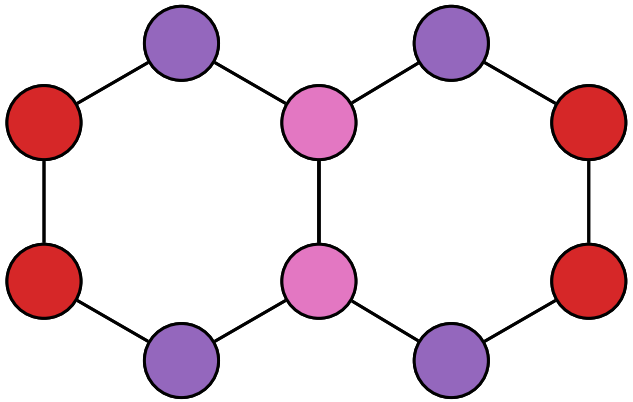
$$\{●, ●\} \mapsto ● \quad \{●, ●\} \mapsto ● \quad \{●, ●, ●\} \mapsto ●$$

4. Repeat until there is no change in the partition of nodes with same label.
5. Compare the histogram of labels: if different, the graphs are definitely non-isomorphic.

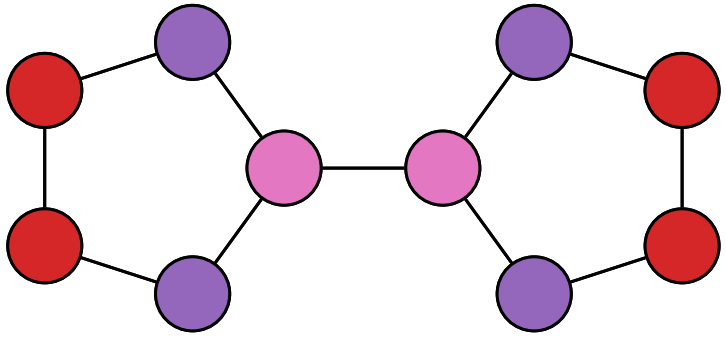
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



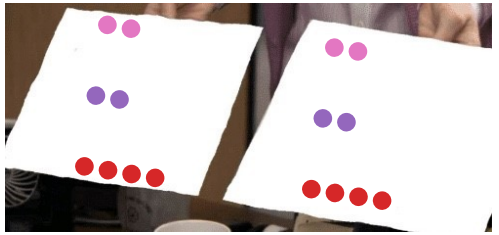
Bicyclopentyl



1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

$$\{●, ●\} \mapsto ● \quad \{●, ●\} \mapsto ● \quad \{●, ●, ●\} \mapsto ●$$

4. Repeat until there is no change in the partition of nodes with same label.
5. Compare the histogram of labels: if different, the graphs are definitely non-isomorphic.

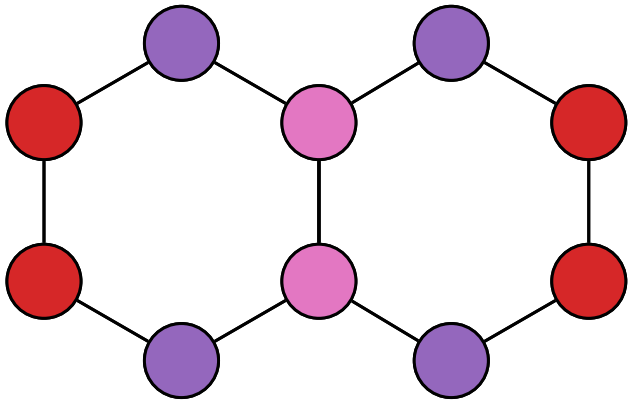


Corporate needs you to find the differences between these histograms

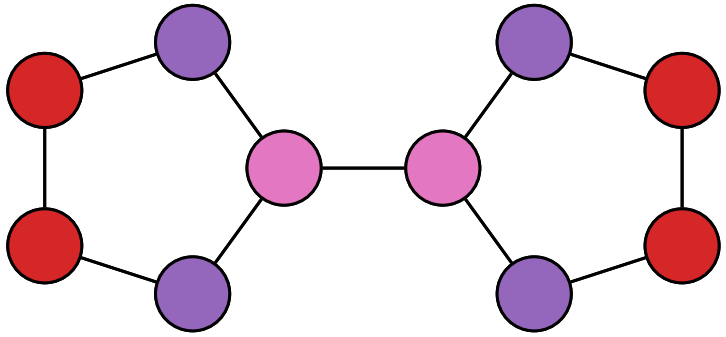
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



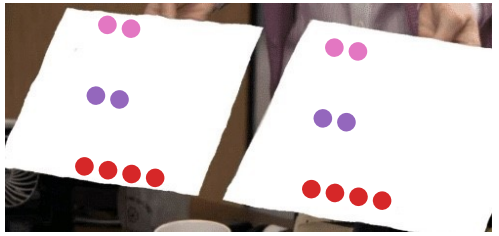
Bicyclopentyl



1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors.
3. Compress to get a new label

$$\{●, ●\} \mapsto ● \quad \{●, ●\} \mapsto ● \quad \{●, ●, ●\} \mapsto ●$$

4. Repeat until there is no change in the partition of nodes with same label.
5. Compare the histogram of labels: if different, the graphs are definitely non-isomorphic.



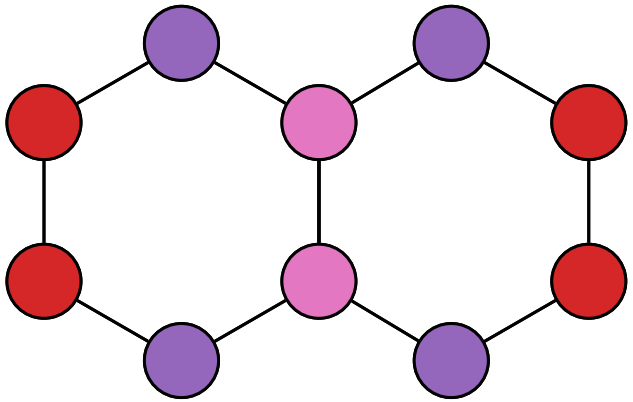
Corporate needs you to find the differences between these histograms

They are the same histograms!

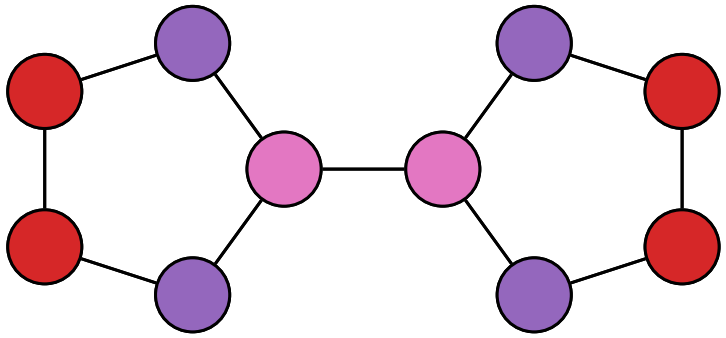
Weisfeiler-Leman Graph Isomorphism Test

Simplest algorithm to check whether two graphs are non-isomorphic

Decalin



Bicyclopentyl

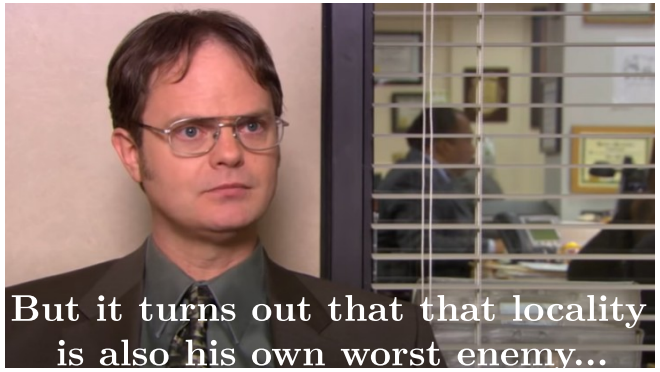


Advantage:

- Locality: only local computations needs to be performed.

Disdvantage:

- Locality: unable to capture higher-order interactions.



Proposition (Which graphs can WL distinguish?)**.** *The Weisfeiler-Leman algorithm can distinguish trees and forests (disjoint union of trees).*

Proposition (Expressive Power of GIN)**.** *GIN is as powerful as the Weisfeiler-Leman test.*

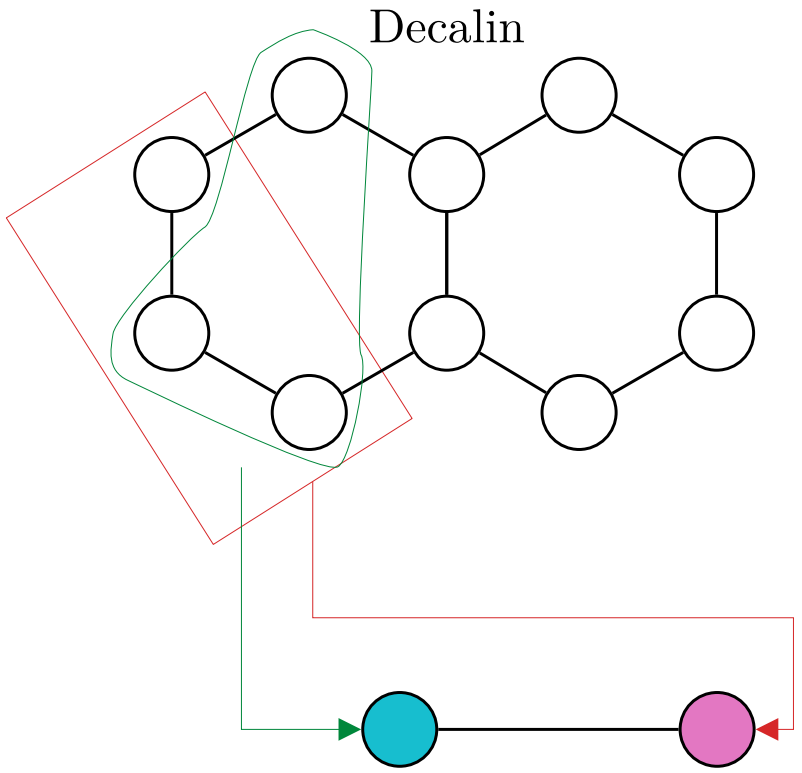
Arvind, V., Köbler, J., Rattan, G., and Verbitsky, O. (2015). *On the Power of Color Refinement.*
Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). *How Powerful Are Graph Neural Networks?*

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Higher-Order Weisfeiler-Leman Tests

Higher-Order Weisfeiler-Leman Tests

Simplest improvement of Weisfeiler-Leman Graph Isomorphism Test



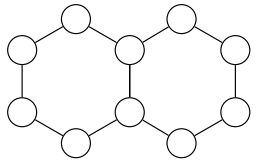
1. From the original graph G with nodes $V(G)$ and edges $E(G)$, create an auxiliary graph F.
   - $V(F) \coloneqq \left\{ \mathbf{g} = \{g_i\}_{i=1}^{k} \mid g_i \in V(G) \right\} = V(G)^k$;
   - $E(F) \coloneqq \left\{ (\mathbf{g}_1, \mathbf{g}_2) \mid d_H(\mathbf{g}_1, \mathbf{g}_2) = 1\,,\ \mathbf{g}_1, \mathbf{g}_2 \in V(G)^k \right\}$.
2. Give each k-tuple a label corresponding to the isomorphic type of the subgraph induced in the original graph.
3. Apply WL algorithm to the auxiliary graphs.

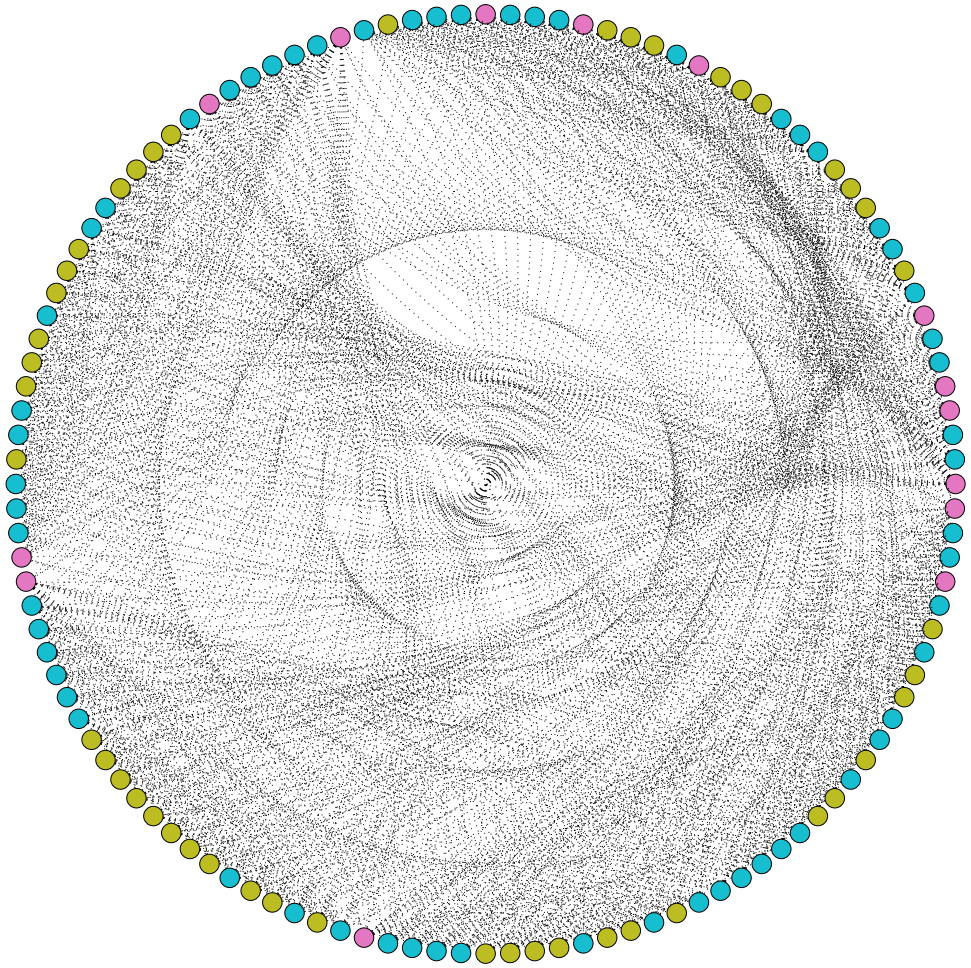
Isomorphic Types:

Higher-Order Weisfeiler-Leman Tests

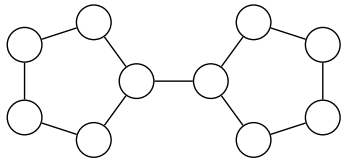
Simplest improvement of Weisfeiler-Leman Graph Isomorphism Test

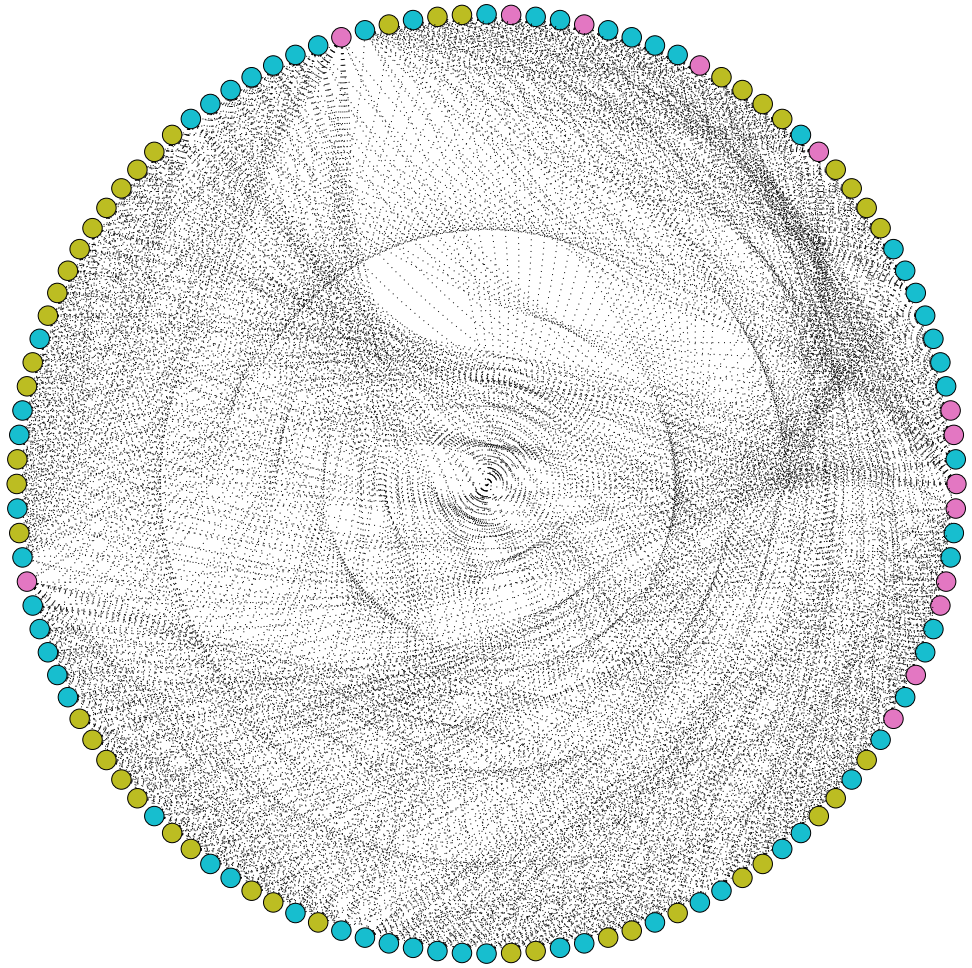
Decalin





Bicyclopentyl





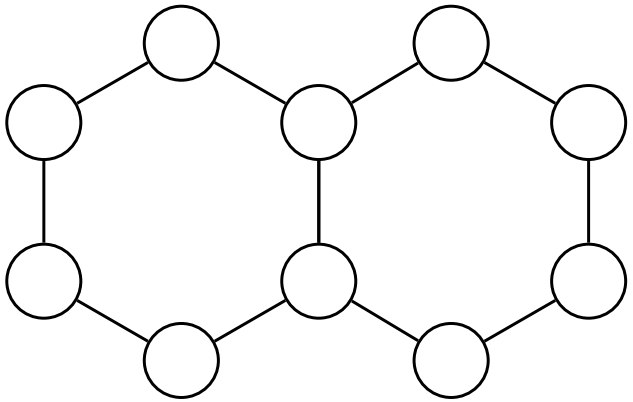
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Loopy Weisfeiler-Leman

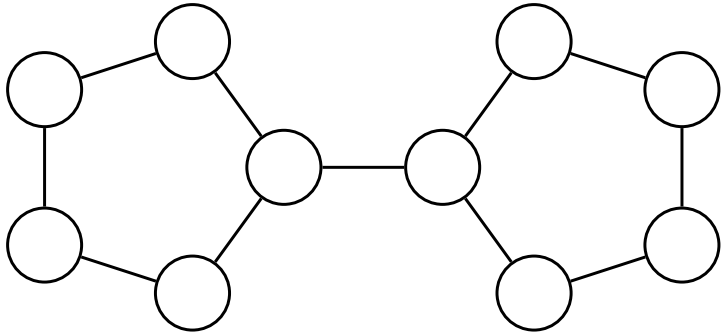
Loopy Weisfeiler Leman

Our approach to account for loops

Decalin



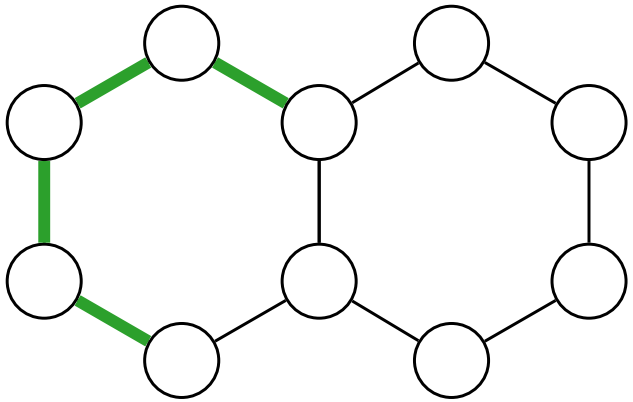
Bicyclopentyl



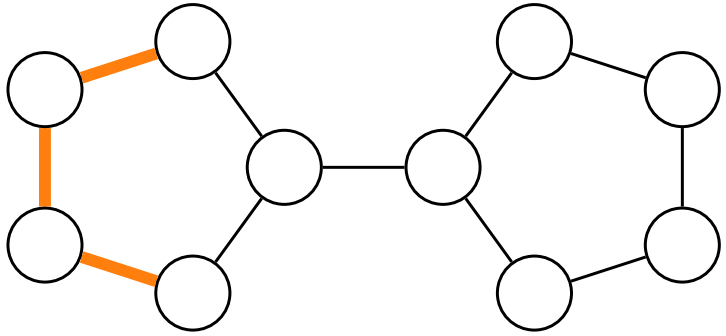
Loopy Weisfeiler Leman

Our approach to account for loops

Decalin



Bicyclopentyl



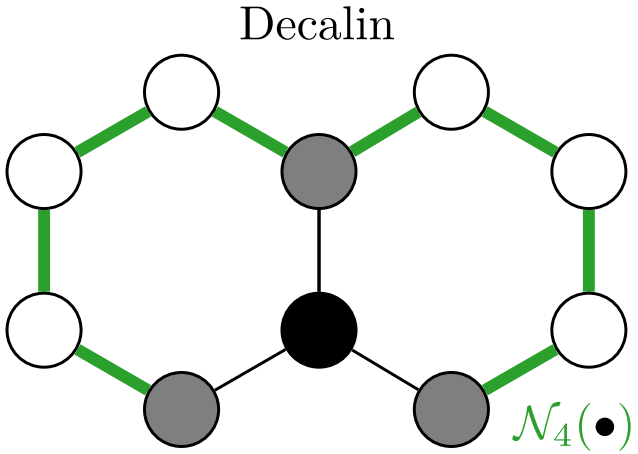
Definition (Simple Path)**.** *Given a graph G, a* simple path *of length r is a collection $\mathbf{p} = \{p_i\}_{i=1}^{r+1}$ of $r+1$ nodes such that consecutive nodes are adjacent, i.e.,*

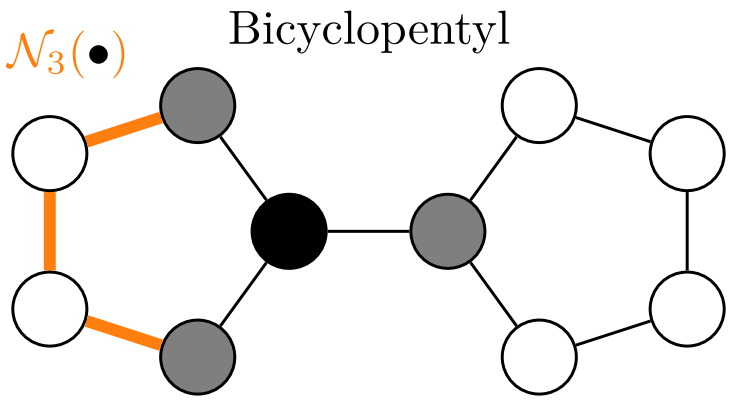
$$\{p_i, p_{i+1}\} \in E(G),\ \forall i \in \{1, \ldots, r\},$$

and there are no repeated nodes, i.e., $i \neq j \implies p_i \neq p_j$.

Loopy Weisfeiler Leman

Our approach to account for loops





Definition (Simple Path)**.** *Given a graph G, a* simple path *of length r is a collection $\mathbf{p} = \{p_i\}_{i=1}^{r+1}$ of $r+1$ nodes such that consecutive nodes are adjacent, i.e.,*

$$\{p_i, p_{i+1}\} \in E(G),\ \forall i \in \{1, \dots, r\},$$

and there are no repeated nodes, i.e., $i \neq j \implies p_i \neq p_j$.

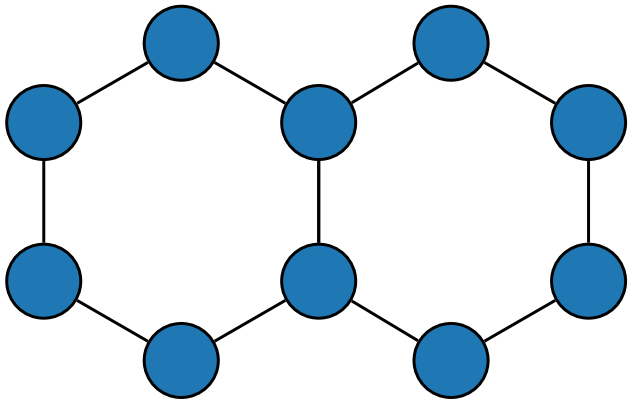
Definition (r-Neighborhood)**.** *Given a graph G and an integer $r \geq 1$, we define the r-*neighborhood *$\mathcal{N}_r(v)$ of $v \in V(G)$ as the set of all simple paths of length r between distinct direct neighbors of v which do not contain v, i.e.,*

$$\mathcal{N}_r(v) \coloneqq \{\mathbf{p} \mid \mathbf{p}\ r\text{-path},\ p_1, p_{r+1} \in \mathcal{N}(v), v \notin \mathbf{p}\}.$$

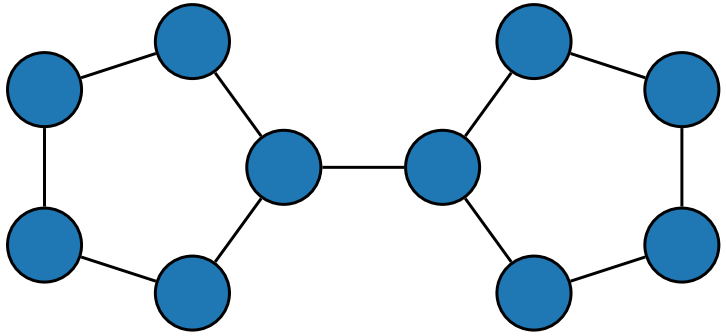
Loopy Weisfeiler Leman

Our approach to account for loops

Decalin



Bicyclopentyl

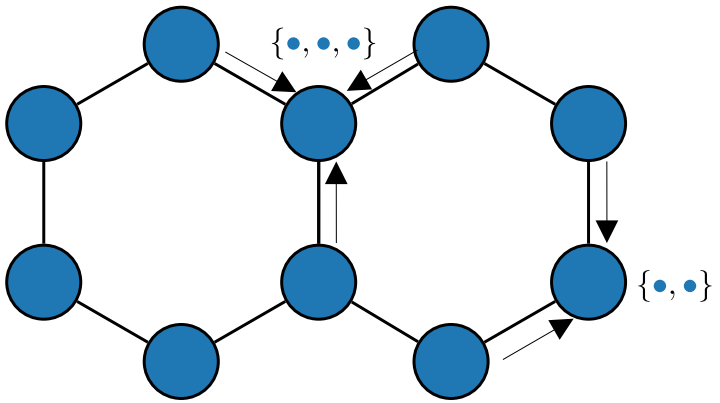


1. Assign to each node the same label ●.

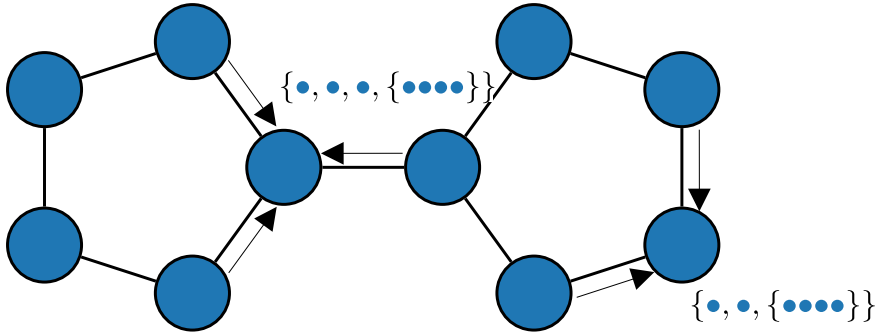
Loopy Weisfeiler Leman

Our approach to account for loops

Decalin



Bicyclopentyl

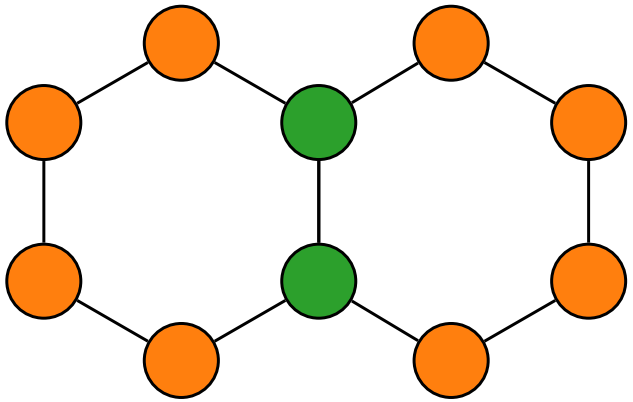


1. Assign to each node the same label •.
2. Collect all labels from direct neighbors and r-neighbors.

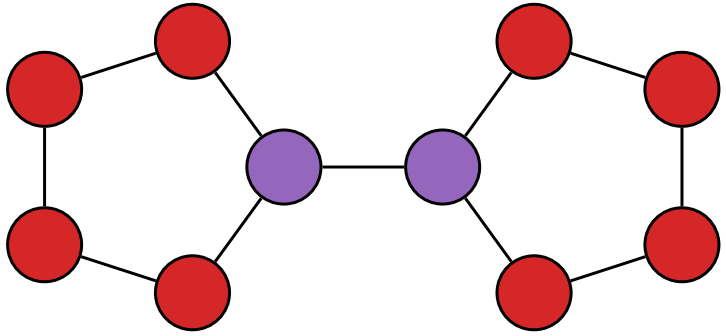
Loopy Weisfeiler Leman

Our approach to account for loops

Decalin



Bicyclopentyl



1. Assign to each node the same label ●.
2. Collect all labels from direct neighbors and r-neighbors.
3. Compress to get a new label

$\{\bullet, \bullet\} \mapsto \bullet$

$\{\bullet, \bullet, \bullet\} \mapsto \bullet$

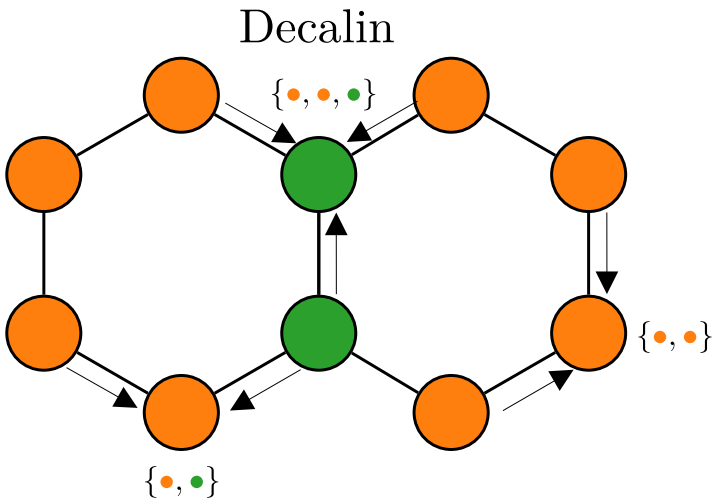
$\{\bullet, \bullet, \{\bullet\bullet\bullet\}\} \mapsto \bullet$

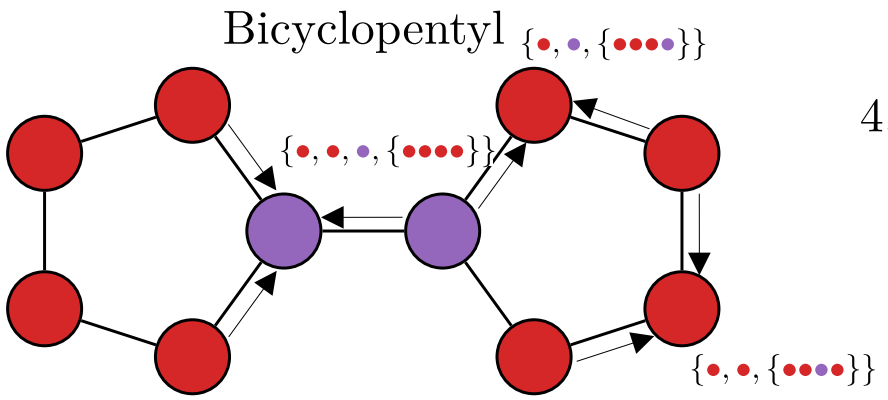
$\{\bullet, \bullet, \bullet, \{\bullet\bullet\bullet\}\} \mapsto \bullet$

The colours are already different, so we can stop the algorithm.

Loopy Weisfeiler Leman

Our approach to account for loops





1. Assign to each node the same label •.
2. Collect all labels from direct neighbors and r-neighbors.
3. Compress to get a new label

$$\{\bullet, \bullet\} \mapsto \bullet$$
$$\{\bullet, \bullet, \bullet\} \mapsto \bullet$$
$$\{\bullet, \bullet, \{\bullet\bullet\bullet\}\} \mapsto \bullet$$
$$\{\bullet, \bullet, \bullet, \{\bullet\bullet\bullet\}\} \mapsto \bullet$$

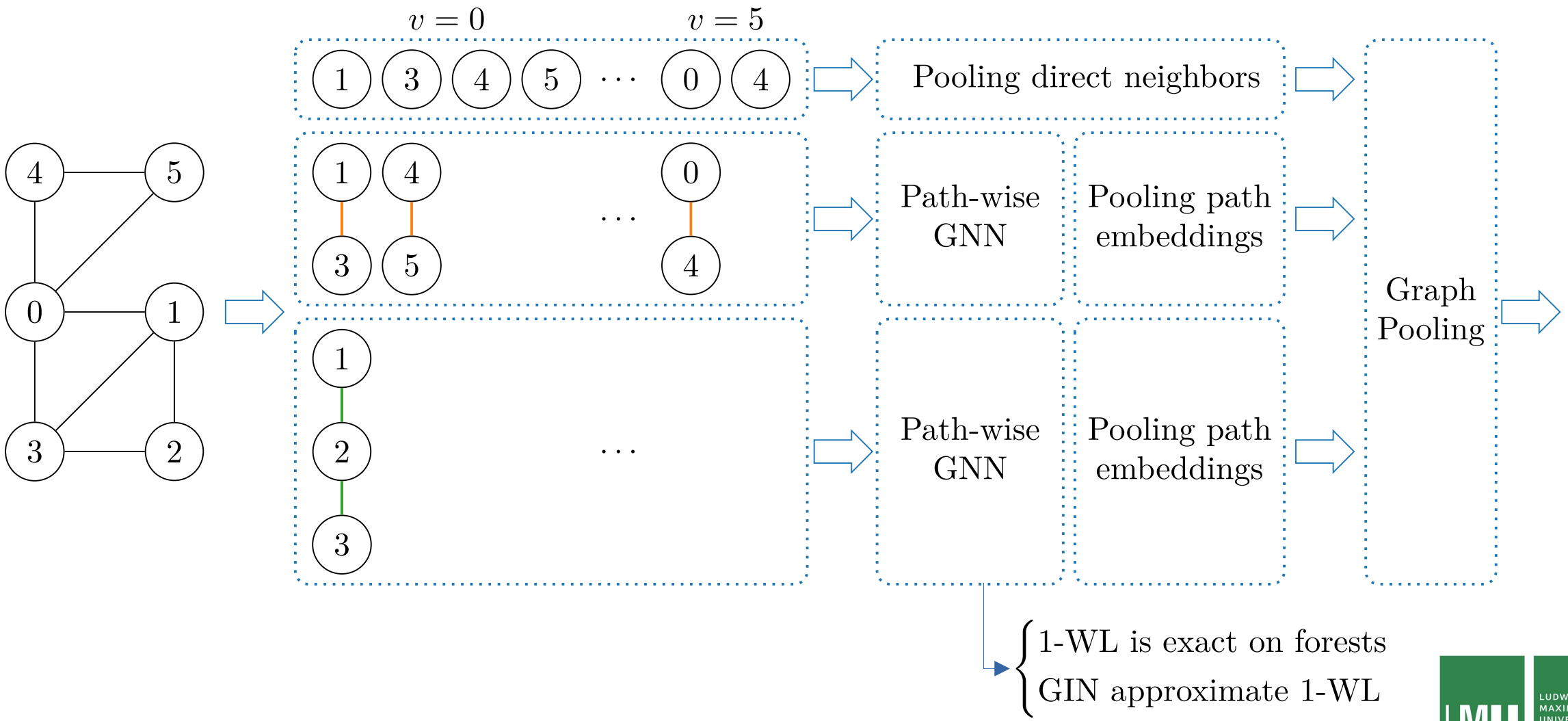
4. Repeat until there is no change in the partition of nodes with same label.

The colours are already different, so we can stop the algorithm.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

GNN emulating Loopy Weisfeiler-Leman

Loopy Graph Neural Network

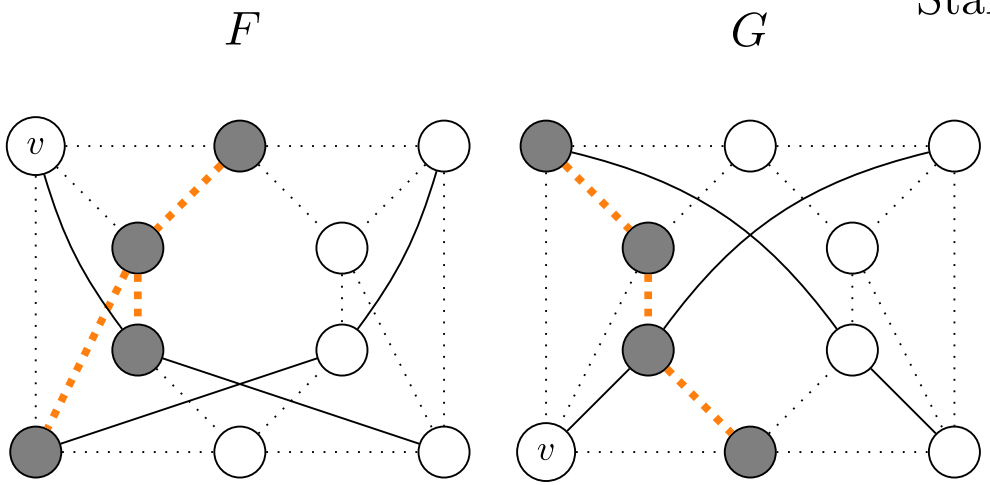


LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

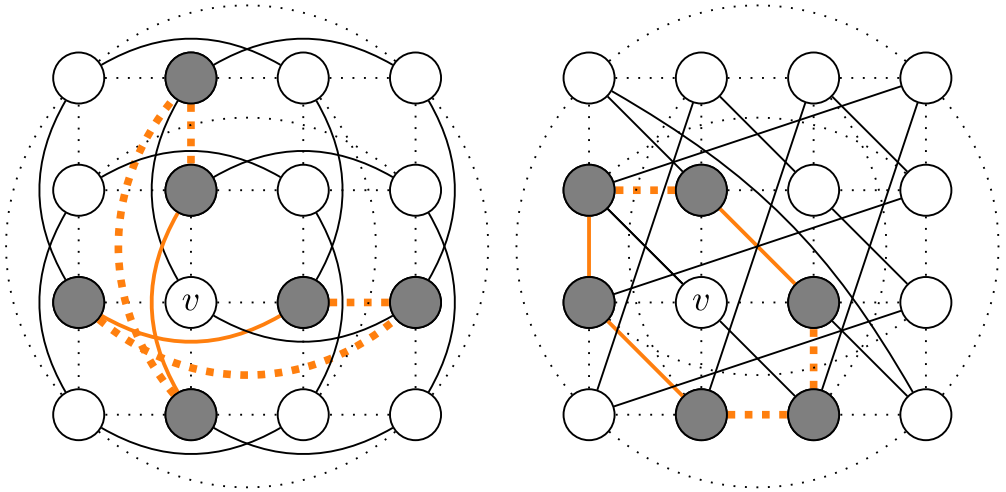
Expressivity of Loopy Weisfeiler-Leman

Isomorphism Expressivity

Standard Expressivity Measure



1-WL equivalent



4×4 rook graph Shrikhande graph

3-WL equivalent

$\rightarrow$ Comparison with k-WL hierarchy.

Proposition (Hierarchy of r-ℓWL)**.** *Let $r > 0$.*

i) r-ℓ WL is strictly more powerful than 1-WL;

i) r-ℓ WL is strictly more powerful than $(r-1)$-ℓ WL;

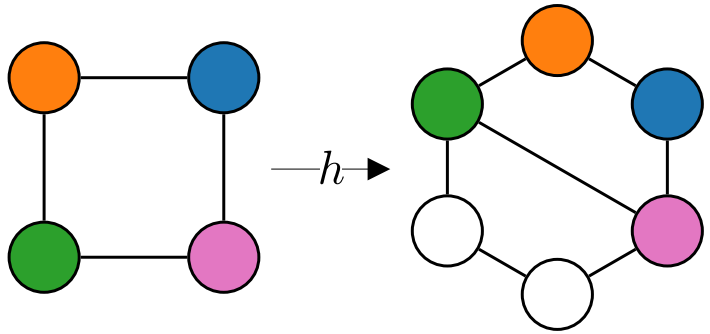
Advantages:

- Easy to analyze.

Disadvantages:

- Only qualitative measure.

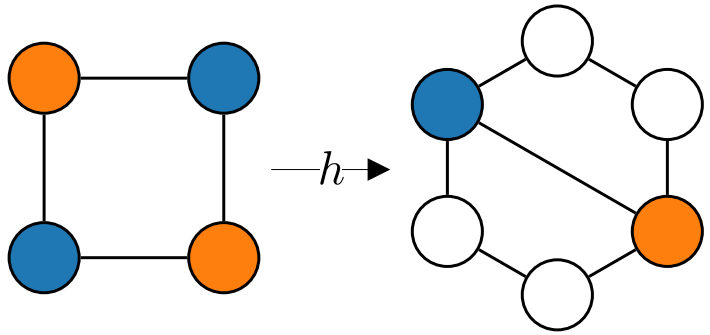
Subgraph Expressivity



Definition (Subgraph Isomorphism)**.** *A* subgraph isomorphism *is an injective homomorphism. The set of subgraph isomorphisms from* F *to* G *is denoted by* $\mathrm{Sub}(F, G)$*, and its cardinality by* $\mathrm{sub}(F, G) = |\mathrm{Sub}(F, G)|$.

Homomorphism Expressivity

Quantitative Expressivity Measure: Counting Patterns



Definition (Homomorphism)**.** *Let* F *and* G *be two graphs. A* homomorphism *from* F *to* G *is a map* $h : V(F) \to V(G)$ *such that*

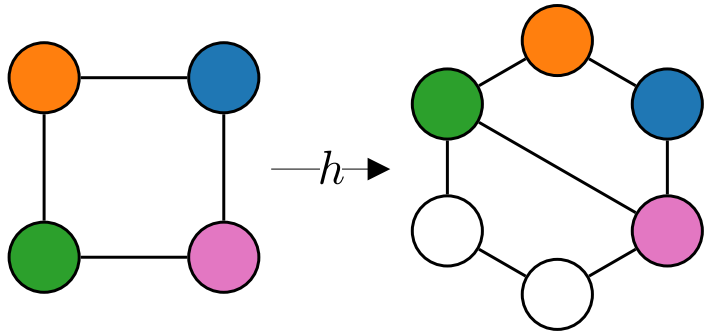
$$\{u, v\} \in E(F) \implies \{h(u), h(v)\} \in E(G).$$

The set of homomorphisms from F *to* G *is denoted by* $\mathrm{Hom}(F, G)$*, and its cardinality by* $\mathrm{hom}(F, G) := |\mathrm{Hom}(F, G)|$.

→ Counting homomorphism is a complete isomorphic measure

$$G \cong H \iff \mathrm{hom}(F, G) = \mathrm{hom}(F, H)\ \forall F$$

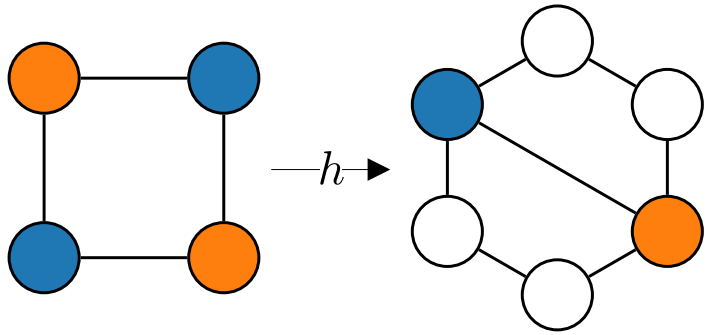
Subgraph Expressivity



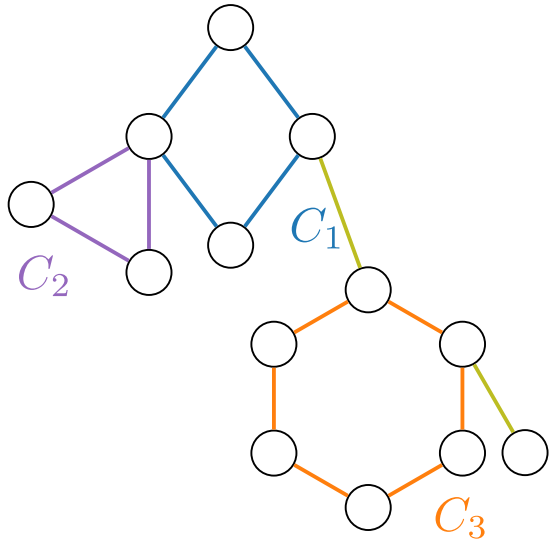
Proposition (Subgraph Counting Power of r-ℓWL)**.** *For any $r \geq 1$, r-ℓWL can subgraph-count all cycles with at most $r+2$ nodes.*

Homomorphism Expressivity

Quantitative Expressivity Measure: Counting Patterns



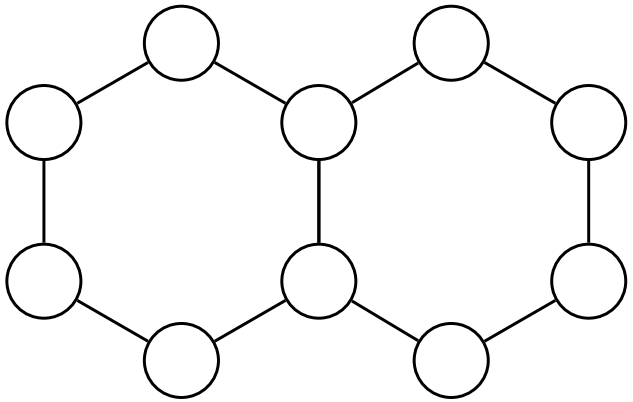
Definition (Cactus Graph)**.** *A* cactus graph *is a graph where every edge lies on at most one simple cycle. For $r \geq 2$, an* r-cactus *graph is a cactus where every simple cycle has at most r vertices. We denote by $\mathcal{M}$ the set of all cactus graphs, and by $\mathcal{M}^r$ the set comprising all q-cactus graphs for $q \leq r$.*



Proposition (Homomorphism Counting Power of r-ℓWL)**.** *Let $r \geq 0$. Then, r-ℓWL can homomorphism-count $\mathcal{M}^{r+2}$.*

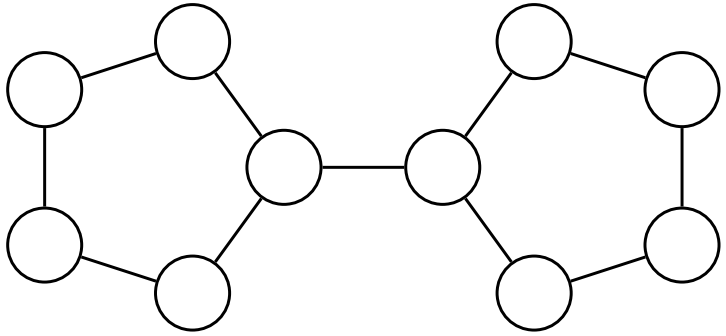
Subgraph Expressivity
Homomorphism Expressivity

Decalin

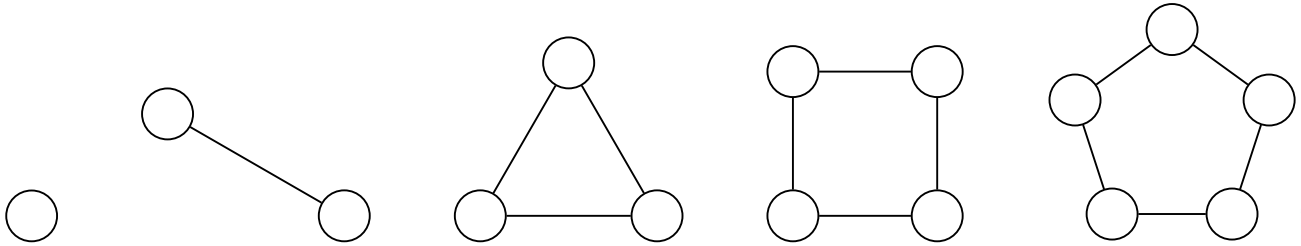


10	11	0	0	0
10	11	0	78	0

Bicyclopentyl

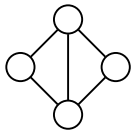
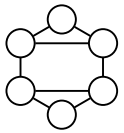
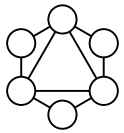
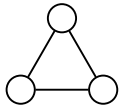
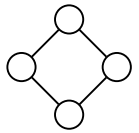
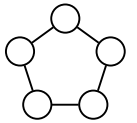
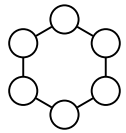
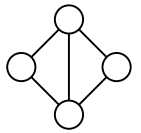
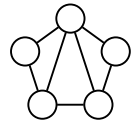


10	11	0	0	2
10	11	0	78	20



Patterns

Subgraph Expressivity
Homomorphism Expressivity

Model	$\hom(F,G)$			$\mathrm{sub}(F,G)$					
									
MPNN	0.300	0.233	0.254	0.358	0.208	0.188	0.146	0.261	0.205
Subgraph GNN	0.011	0.015	0.012	0.010	0.020	0.024	0.046	0.007	0.027
Local 2-GNN	0.008	0.008	0.010	0.008	0.011	0.017	0.034	0.007	0.016
Local 2-FGNN	0.003	**0.005**	**0.004**	0.003	0.004	0.010	0.020	0.003	0.010
r-ℓGIN	**0.001** (r=2)	0.006 (r=3)	0.009 (r=3)	**0.0005** (r=1)	**0.0005** (r=2)	**0.0003** (r=3)	**0.0003** (r=4)	**0.001** (r=2)	**0.0004** (r=3)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Experiments on Molecular Datasets

Experiments on ZINC

Model	ZINC12K	ZINC250K
GIN	0.163 ± 0.004	0.088 ± 0.002
GCN	0.321 ± 0.009	-
GAT	0.384 ± 0.007	-
GSN	0.115 ± 0.012	-
CIN	0.079 ± 0.006	0.022 ± 0.002
NestedGNN	0.111 ± 0.003	0.029 ± 0.001
SUN	0.083 ± 0.003	-
GNNAK	0.080 ± 0.001	-
I2-GNN	0.083 ± 0.001	0.023 ± 0.001
DRFWL	0.077 ± 0.002	0.025 ± 0.003
SignNet	0.084 ± 0.004	0.024 ± 0.003
HIMP	0.151 ± 0.006	0.036 ± 0.002
PathNN	0.090 ± 0.004	-
5-ℓGIN	0.072 ± 0.002	0.022 ± 0.001

Experiments on QM9

	Model							
Target	1-GNN	1-2-3-GNN	DTNN	DeepLRP	NestedGNN	I2-GNN	DRFFWL	5-ℓGIN
μ	0.493	0.476	0.244	0.364	0.428	0.428	0.346	0.350 ± 0.011
α	0.78	0.27	0.95	0.298	0.290	0.230	0.222	0.217 ± 0.025
$\varepsilon_{\text{homo}}$	0.00321	0.00337	0.00388	0.00254	0.00265	0.00261	0.00226	0.00205 ± 0.00005
$\varepsilon_{\text{lumo}}$	0.00355	0.00351	0.00512	0.00277	0.00297	0.00267	0.00225	0.00216 ± 0.00004
$\Delta(\varepsilon)$	0.0049	0.0048	0.0112	0.00353	0.0038	0.0038	0.00324	0.00321 ± 0.00014
R^2	34.1	22.9	17.0	19.3	20.5	18.64	15.04	13.21 ± 0.19
ZVPE	0.00124	0.00019	0.00172	0.00055	0.0002	0.00014	0.00017	0.000127 ± 0.000003
U_0	2.32	0.0427	2.43	0.413	0.295	0.211	0.156	0.0418 ± 0.0520
U	2.08	0.111	2.43	0.413	0.361	0.206	0.153	0.023 ± 0.023
H	2.23	0.0419	2.43	0.413	0.305	0.269	0.145	0.0352 ± 0.0304
G	1.94	0.0469	2.43	0.413	0.489	0.261	0.156	0.0118 ± 0.0015
C_v	0.27	0.0944	2.43	0.129	0.174	0.0730	0.0901	0.0702 ± 0.0024

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Thank you!