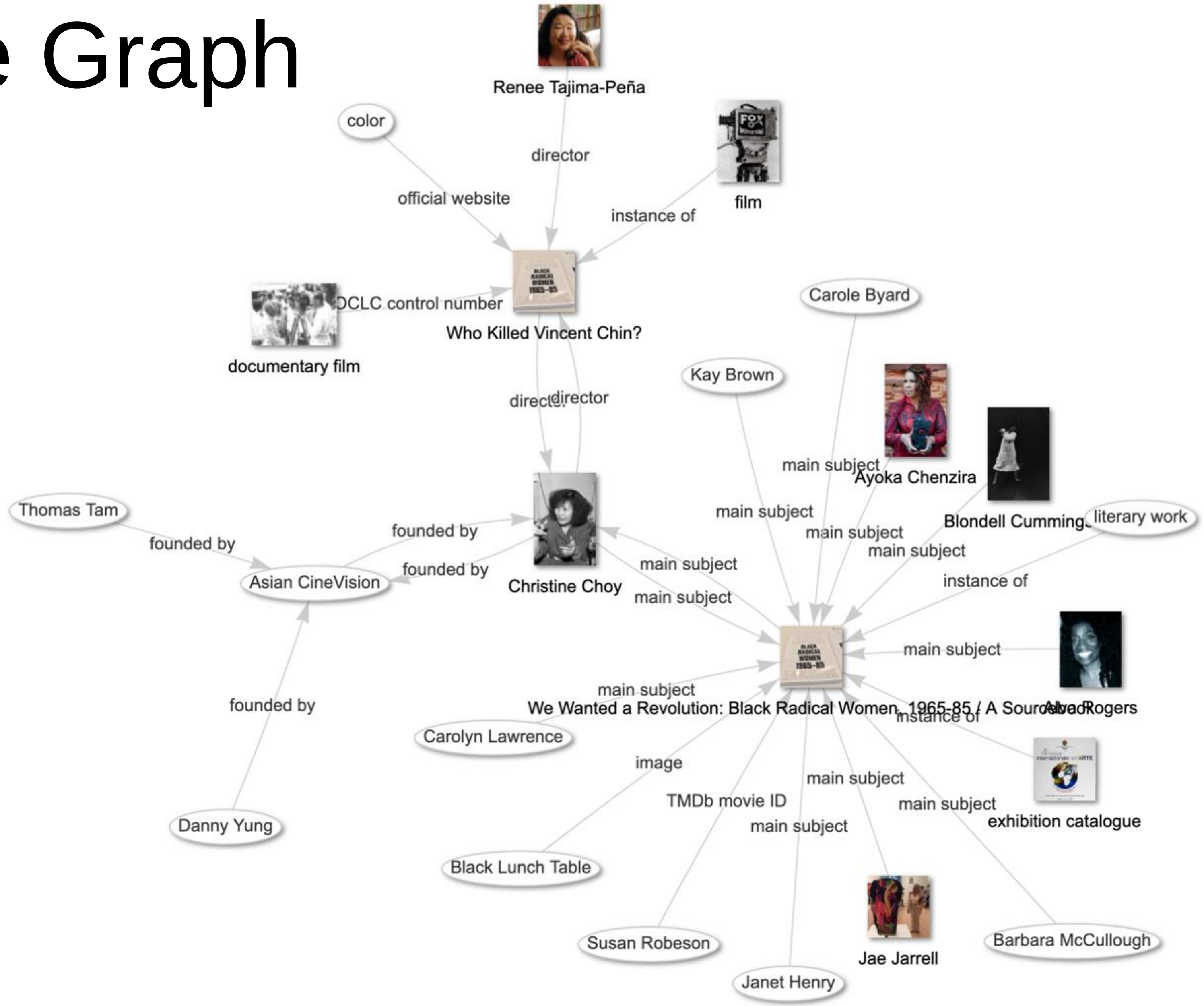
State of Graph Neural Network:

Message Passing, High-Order Modeling, Biconnectivity

2024-08-30

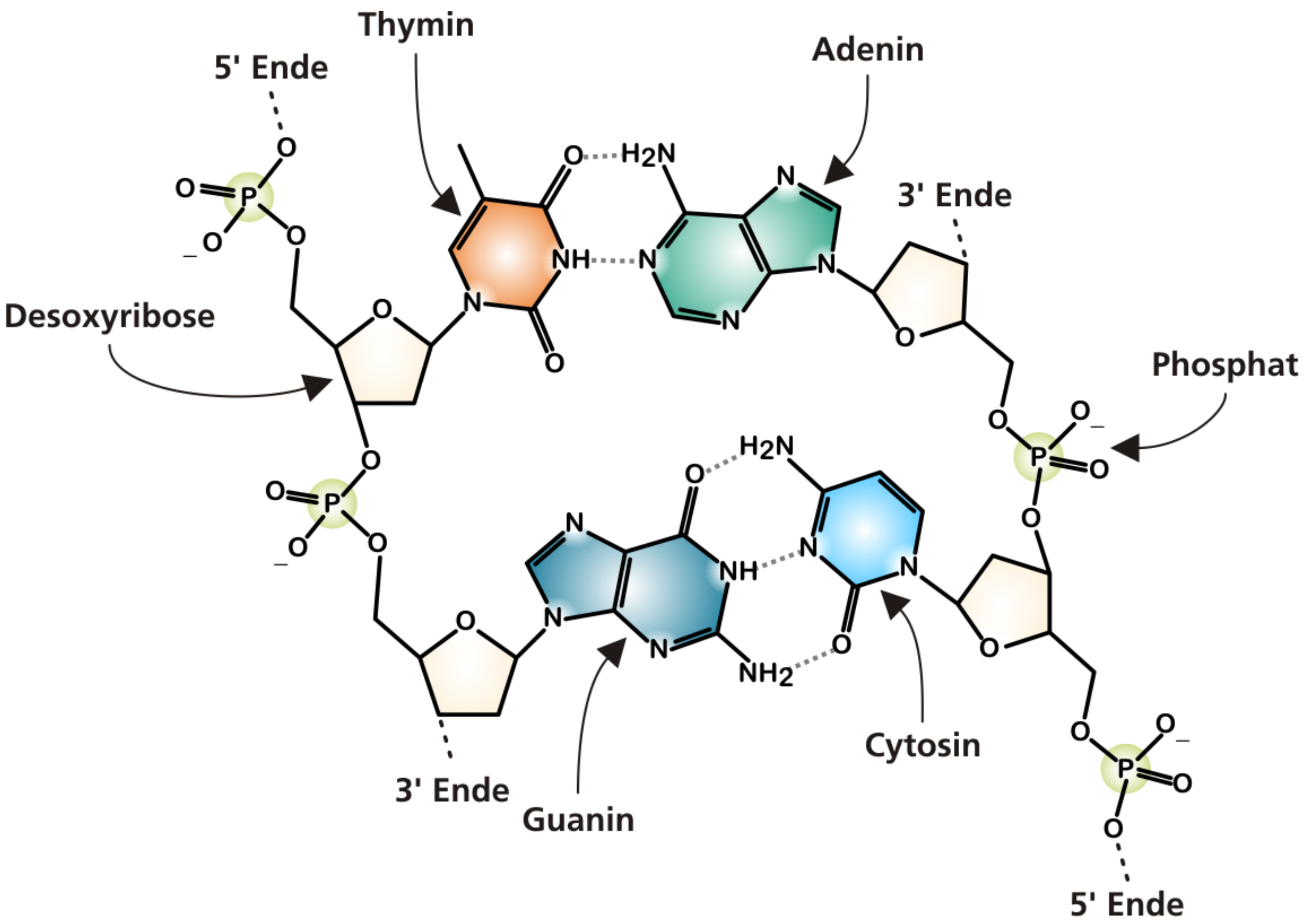
Li Peng-Hsuan 李朋軒

Knowledge Graph

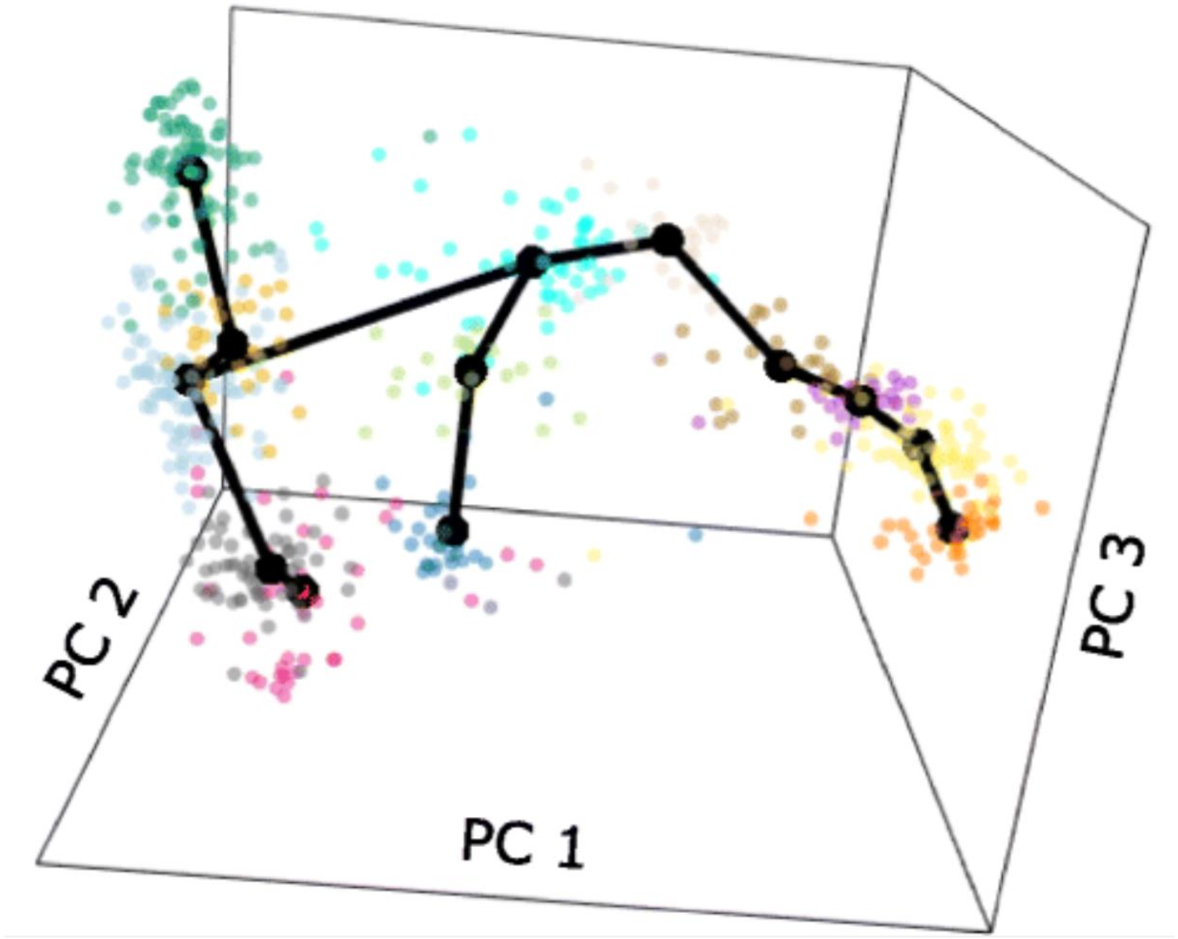


Social Network

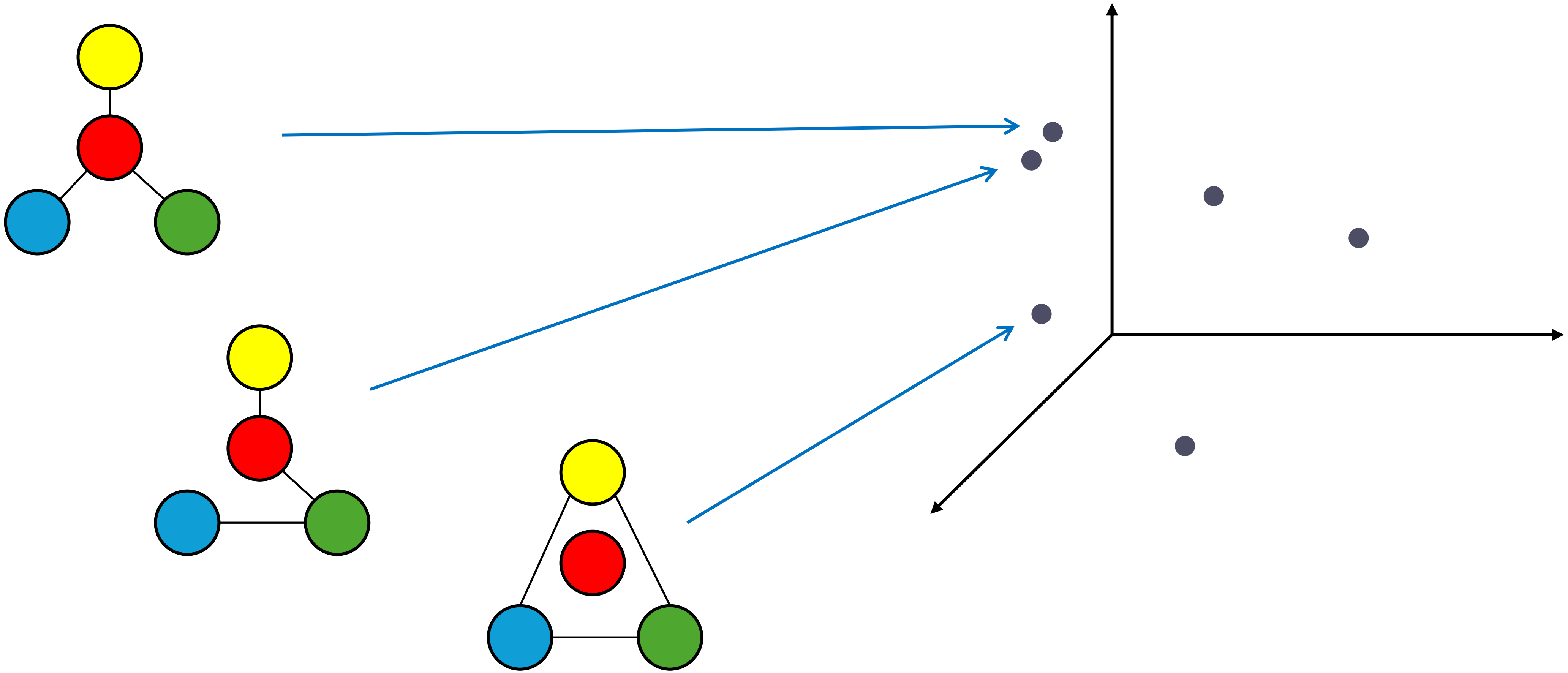
Molecular Structure



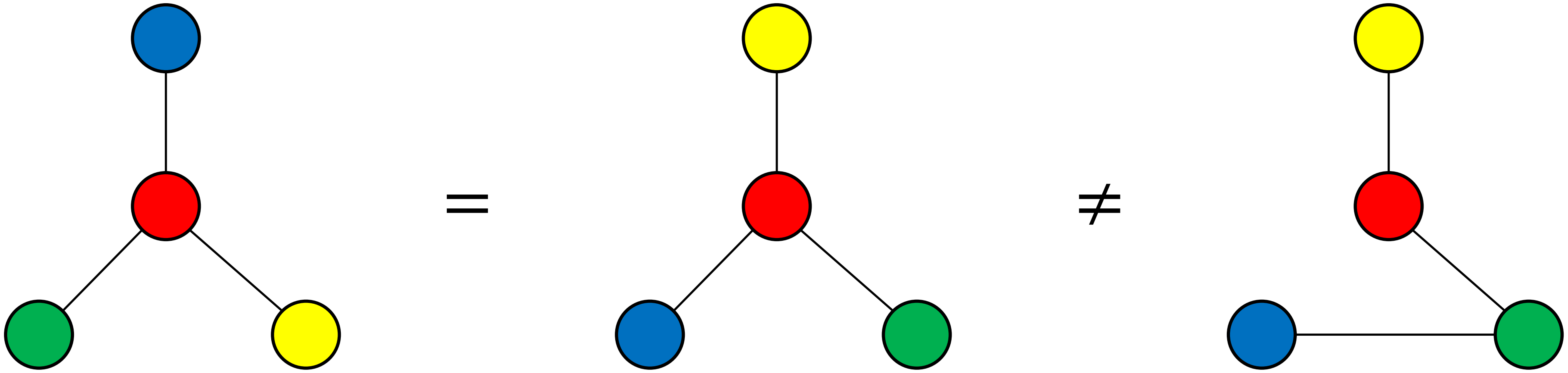
Single Cell Trajectory Tree



Graph Embedding



Graph Isomorphism



Agenda

(~2018) Message passing

→ Graph isomorphism, Weisfeiler-Leman, MPNN

(~2022) High-order modeling

→ Node features, k-WL (k-GNN, k-IGN), subgraph GNN

(~2024) Biconnectivity

→ Biconnectivity, GD-WL, Graphormer-GD

Agenda

(~2018) Message passing

→ Graph isomorphism, Weisfeiler-Leman, MPNN

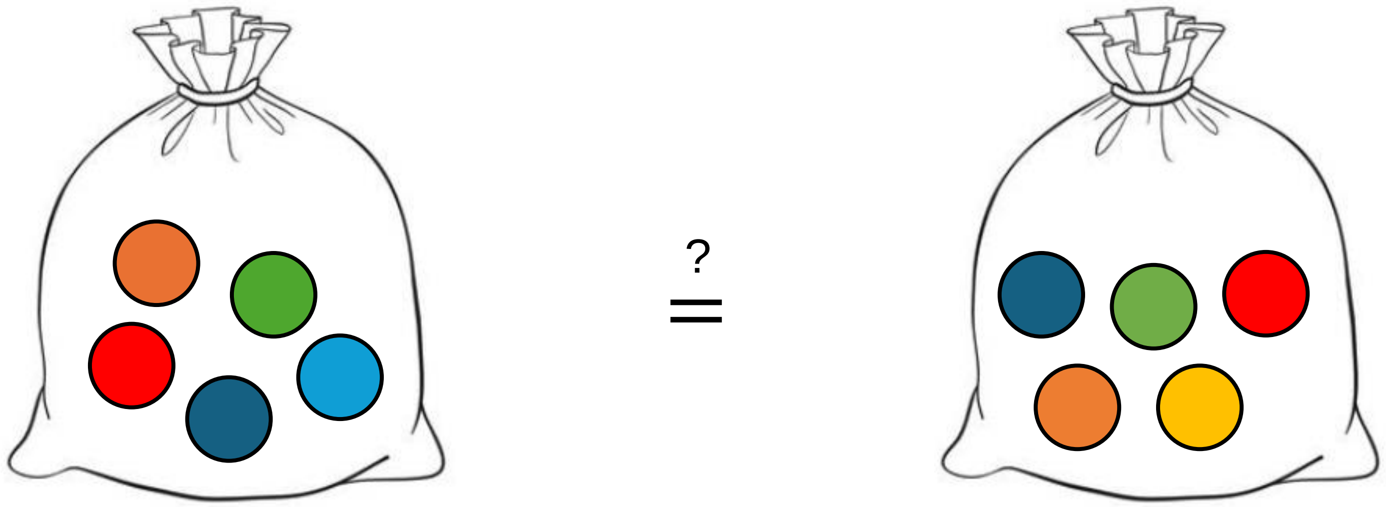
(~2022) High-order modeling

→ Node features, k-WL (k-GNN, k-IGN), subgraph GNN

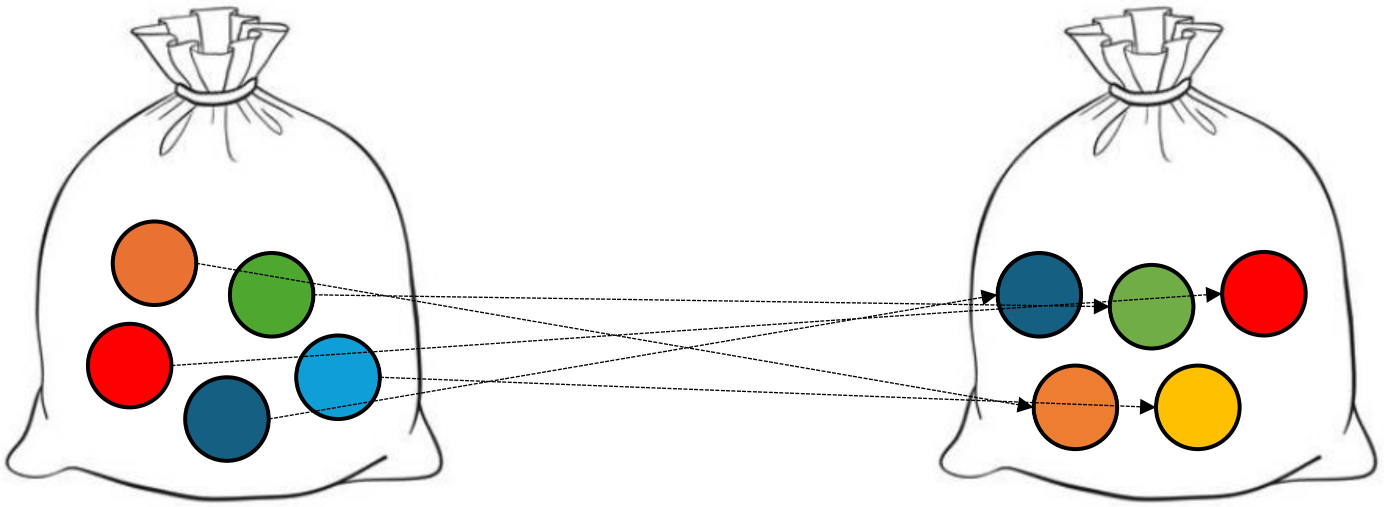
(~2024) Biconnectivity

→ Biconnectivity, GD-WL, Graphormer-GD

Comparing Sets

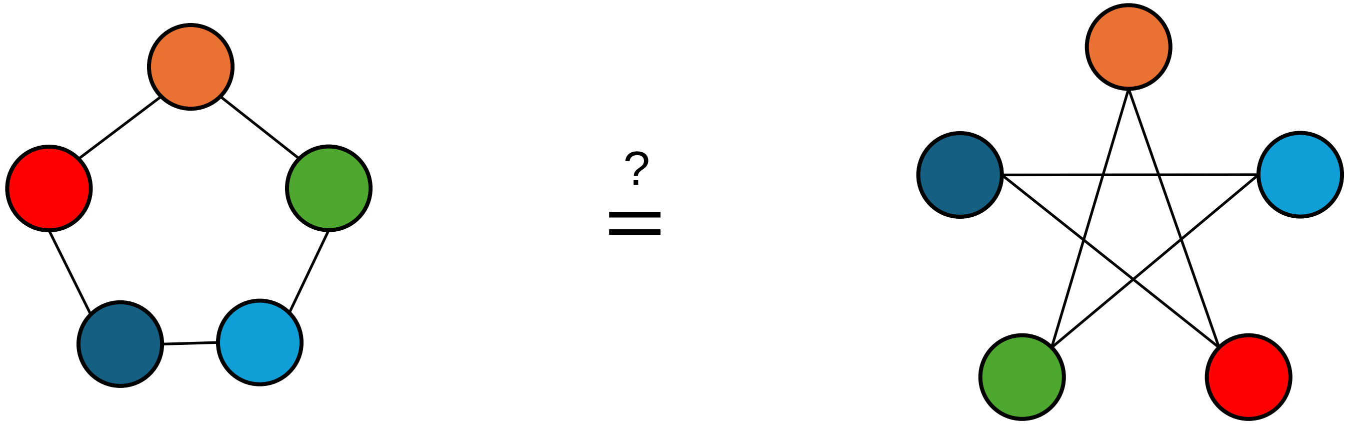


The Same Set

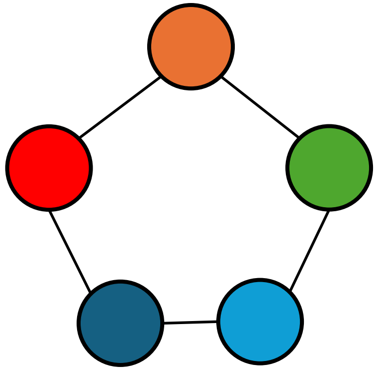


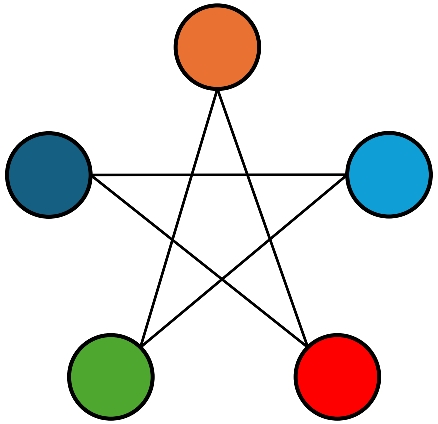
“=”: a bijective mapping s.t. member labels are preserved

Comparing Graphs



Graph Isomorphism





“=”: a bijective mapping s.t. nodes (labels), edges (labels) are preserved

Graph Isomorphism Test

- NP-completeness not known

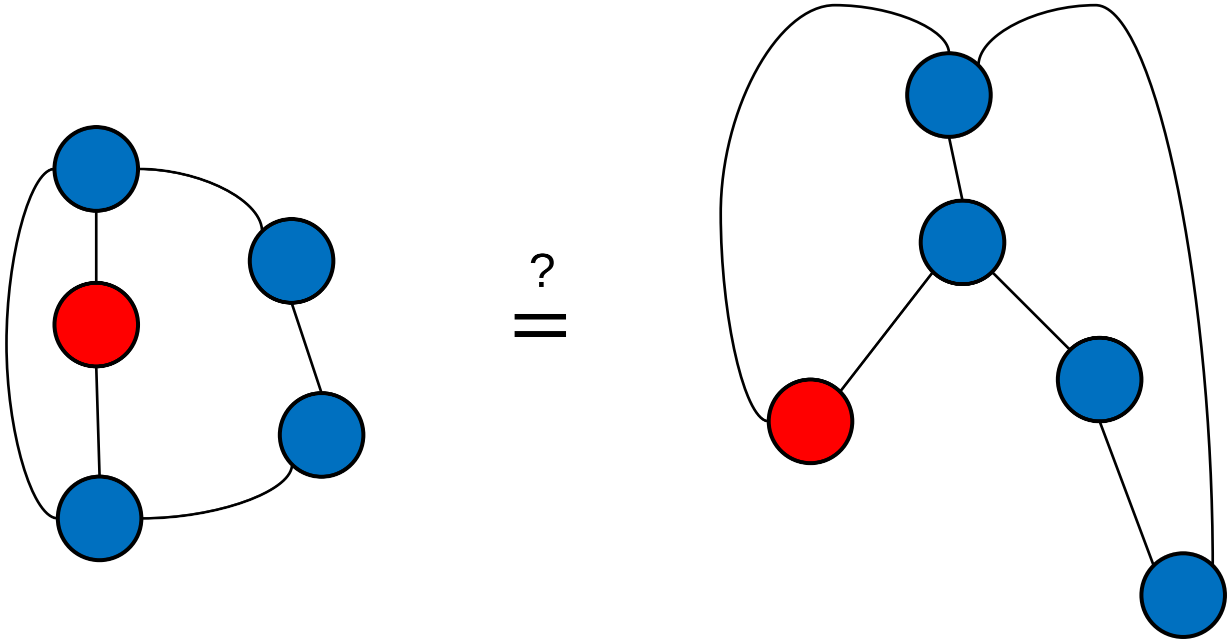
- The current algorithm with the best claim:

  → Quasi-polynomial time $\exp\left((\log n)^{O(1)}\right)$

Graph Isomorphism in Quasipolynomial Time.
https://doi.org/10.48550/arXiv.1512.03547

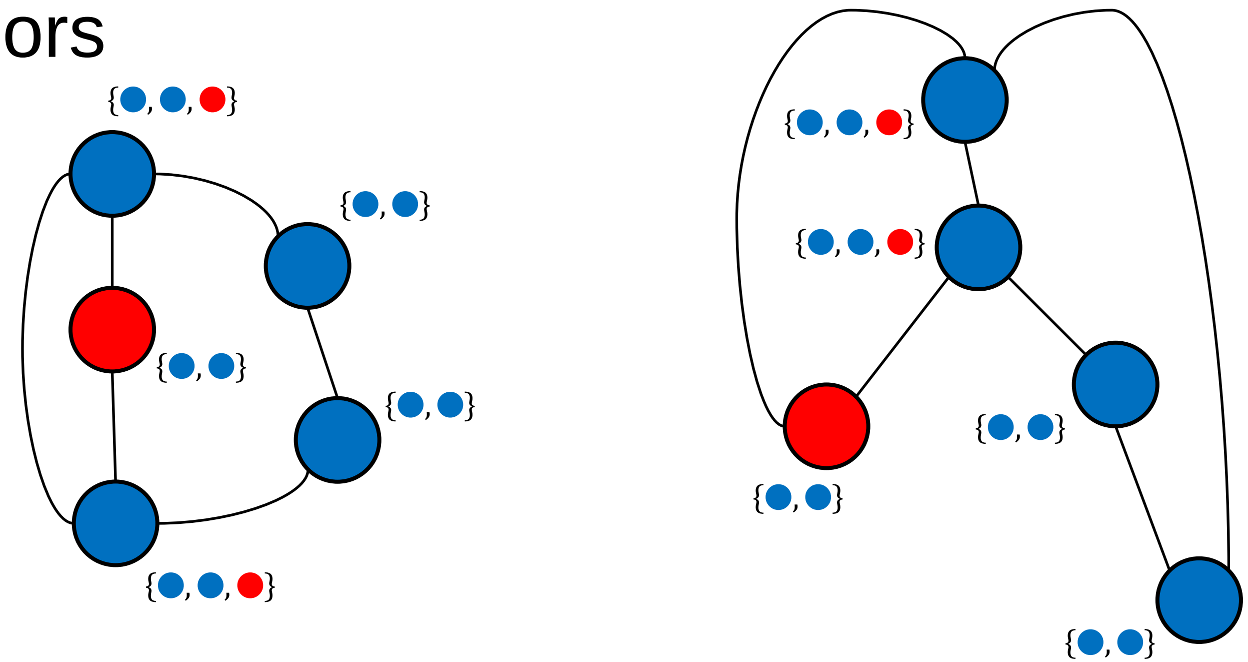
Weisfeiler-Leman

Original



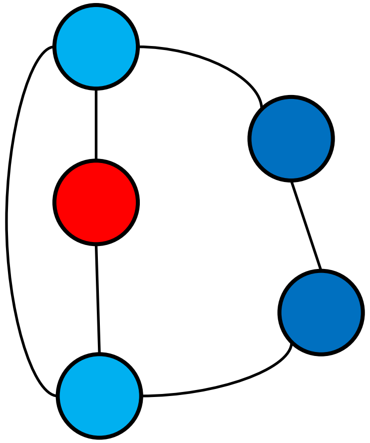
Weisfeiler-Leman

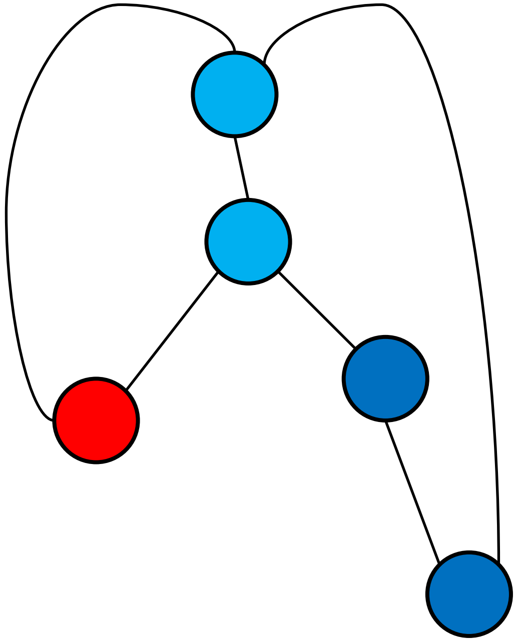
Collect neighbors



Weisfeiler-Leman

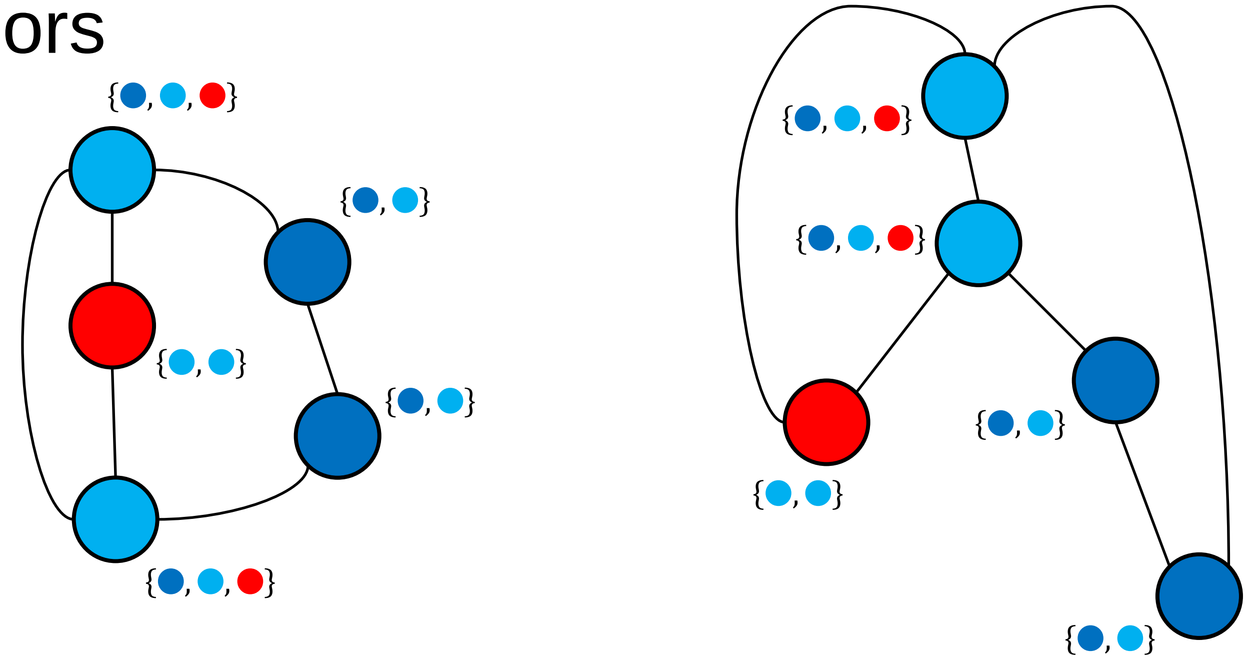
Refine colors





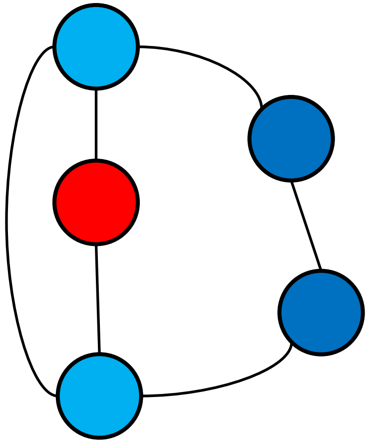
Weisfeiler-Leman

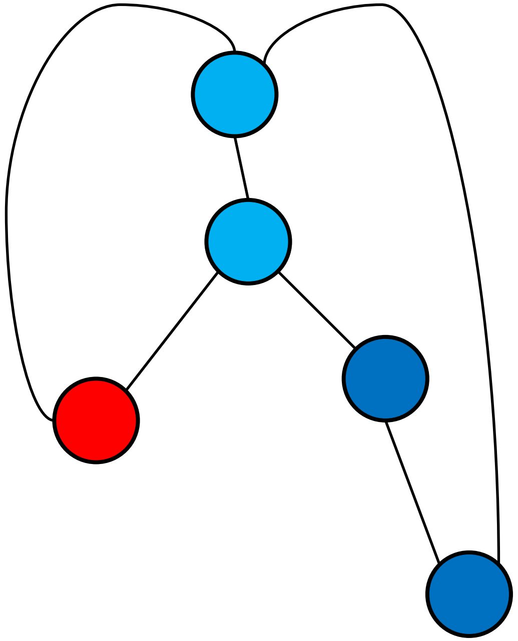
Collect neighbors



Weisfeiler-Leman

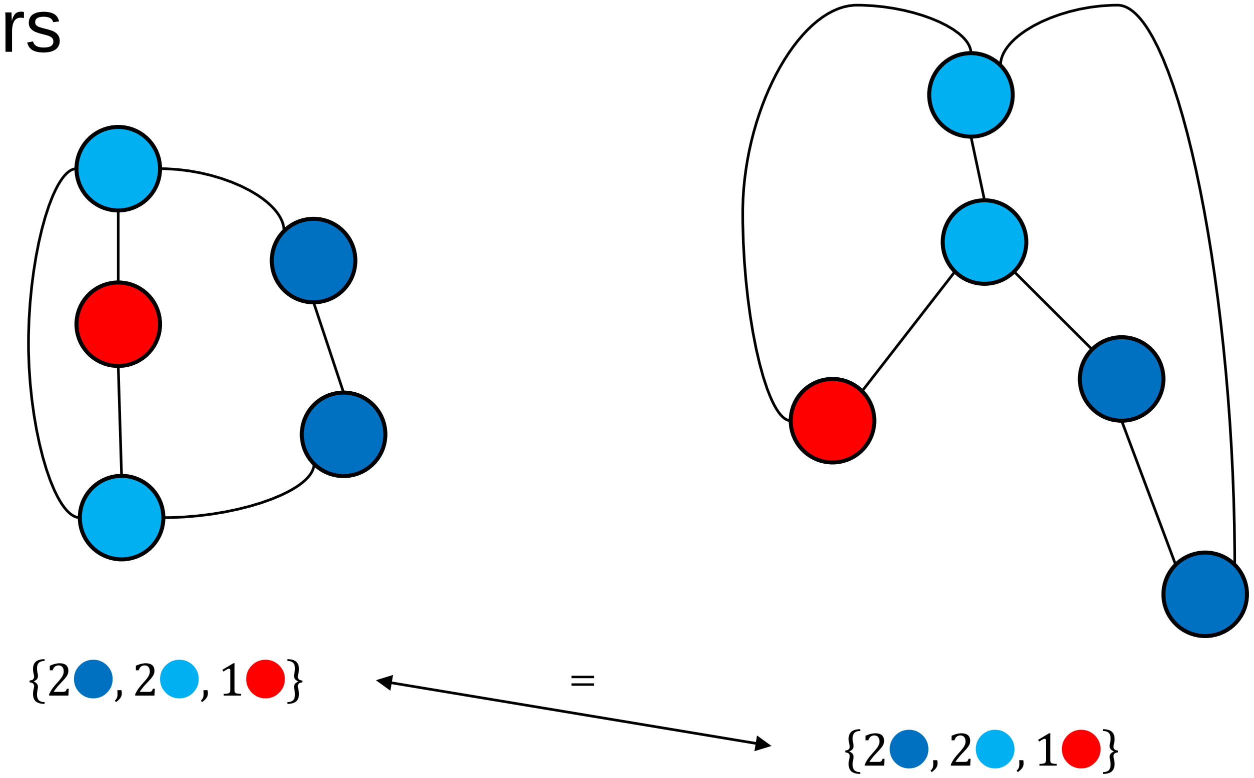
Refine colors → converged





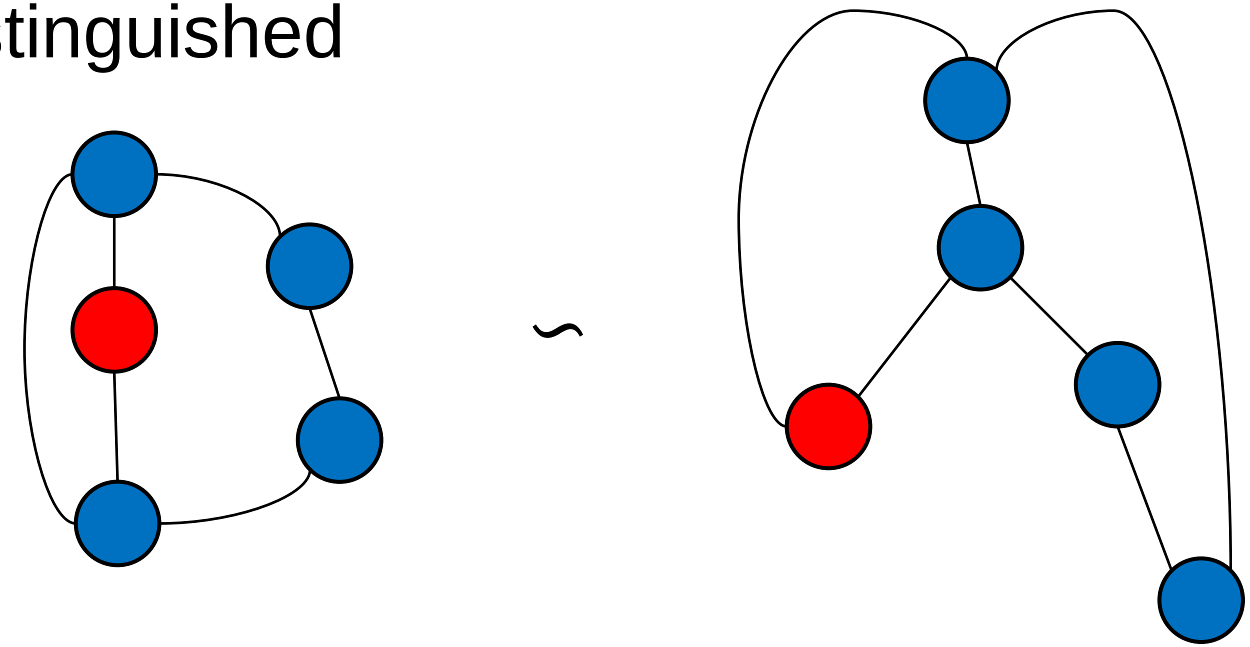
Weisfeiler-Leman

Compare colors



Weisfeiler-Leman

Result: not distinguished



Weisfeiler-Leman

- Very fast

  $O(n + e)$

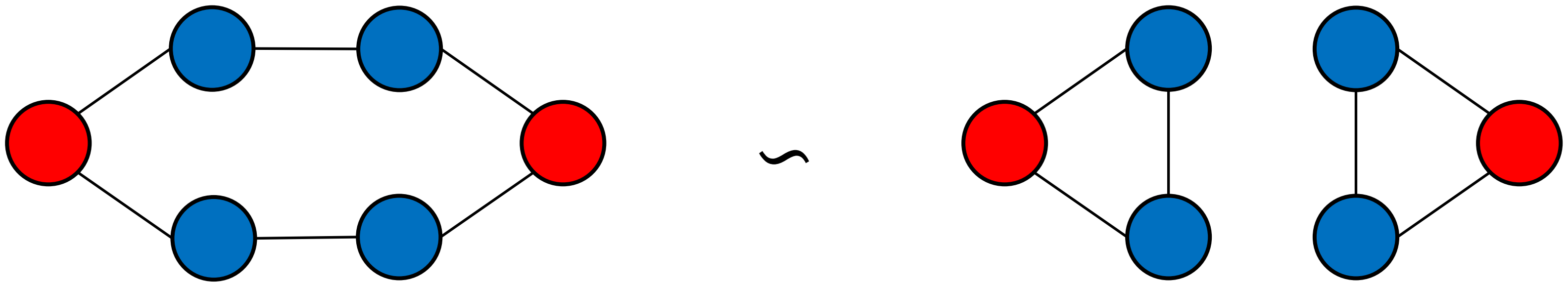
- Sound

  Any isomorphic graphs ⇒ not distinguished

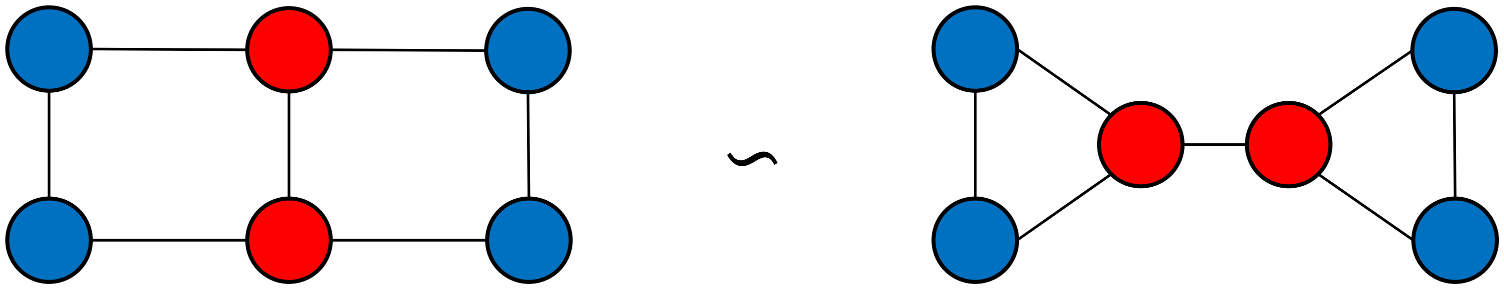
- Incomplete

  [fail to accomplish] any non-isomorphic graphs ⇒ distinguished

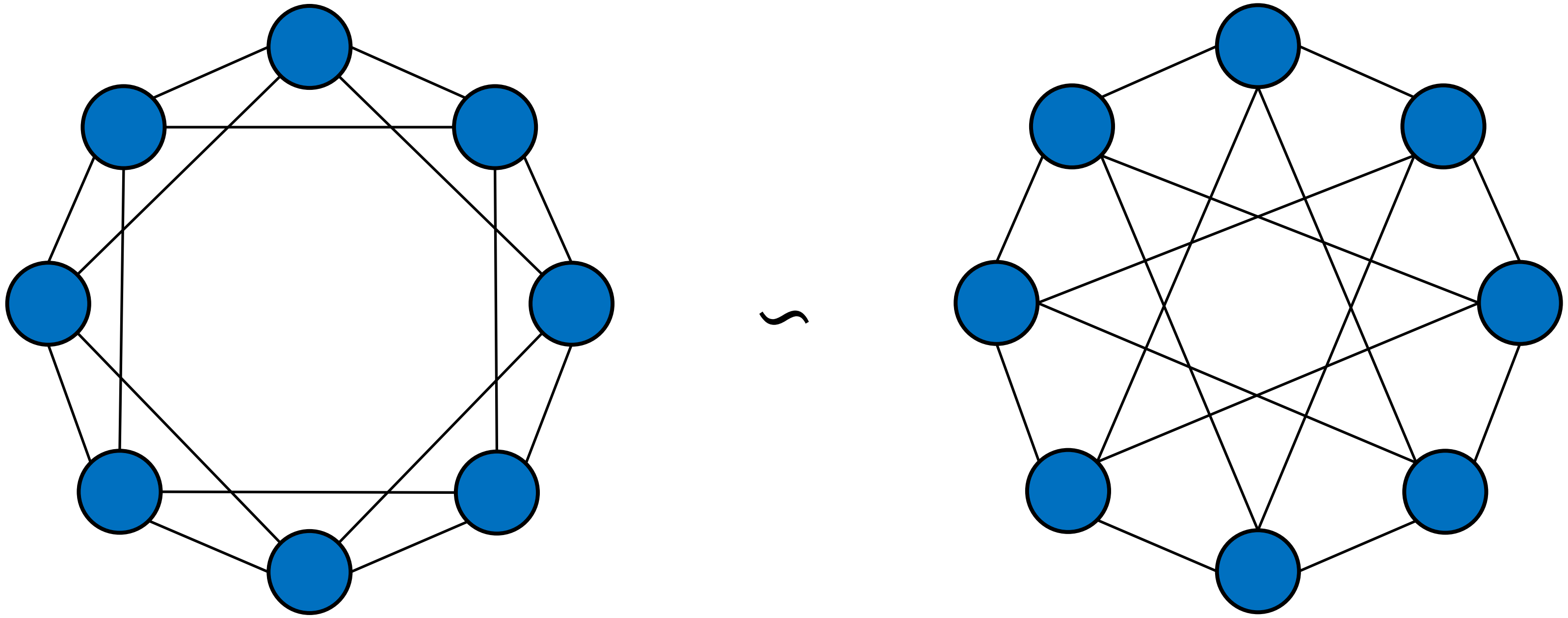
Weisfeiler-Leman: Limitations



Weisfeiler-Leman: Limitations



Weisfeiler-Leman: Limitations



MPNN

- Message-Passing Neural Network (MPNN)

  → The vanilla GNN

  → Encode each node and its neighbors

  → Iterate (multi-layers)

MPNN: Graph Isomorphism

- Expressive power of MPNN

  → MPNNs are at most as powerful as Weisfeiler-Leman

  → GCN, GraphSAGE, GAT are less powerful than Weisfeiler-Leman

  → GIN is as powerful as Weisfeiler-Leman

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1 + \epsilon^{(k)}\right) \cdot h_v^{(k-1)} + \sum\nolimits_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right)$$

How Powerful are Graph Neural Networks?
https://doi.org/10.48550/arXiv.1810.00826

Agenda

(~2018) Message passing

→ Graph isomorphism, Weisfeiler-Leman, MPNN

(~2022) High-order modeling

→ Node features, k-WL (k-GNN, k-IGN), subgraph GNN

(~2024) Biconnectivity

→ Biconnectivity, GD-WL, Graphormer-GD

Node Features

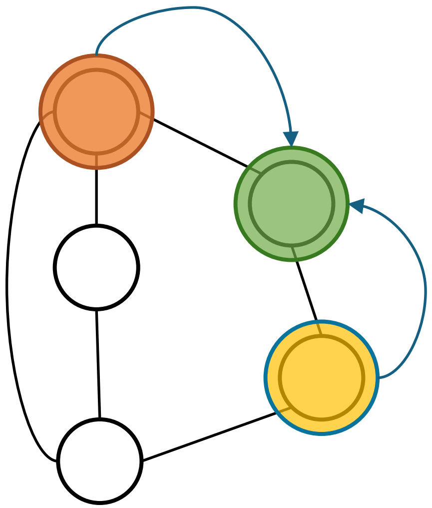
- Add domain specific features
- Add substructure (triangle, clique, cycle) counting features
- Add node IDs

  → Arbitrary fixed id—modeling one big graph

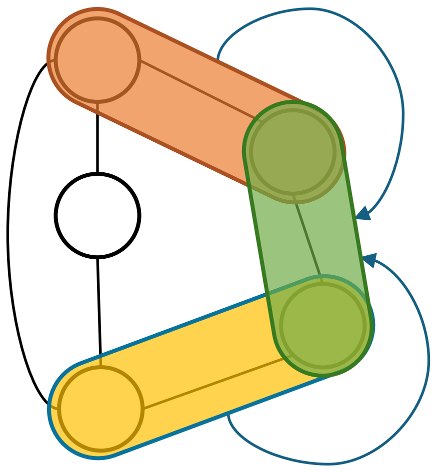
  → Random id—generalizability to unseen graphs

Identity-aware Graph Neural Networks.
https://doi.org/10.1609/aaai.v35i12.17283

High-Order Weisfeiler-Leman



Weisfeiler-Leman (MPNN)



k-WL (k-GNN)

k-WL (k-GNN)

- Just like Weisfeiler-Leman (MPNN), but

  “Nodes” → k-node subgraphs

  “Neighbors” → subgraphs with a (k-1)-node intersection

Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks.
https://doi.org/10.1609/aaai.v33i01.33014602

Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings.
https://doi.org/10.48550/arXiv.1904.01543

Provably Powerful Graph Networks.
https://doi.org/10.48550/arXiv.1905.11136

k-WL (k-GNN): Graph Isomorphism

- Expressive power

  → 1-WL (Weisfeiler-Leman) = 2-WL < 3-WL < 4-WL < …

  → 3-WL = 2-FWL = 3-IGN

- Complexity

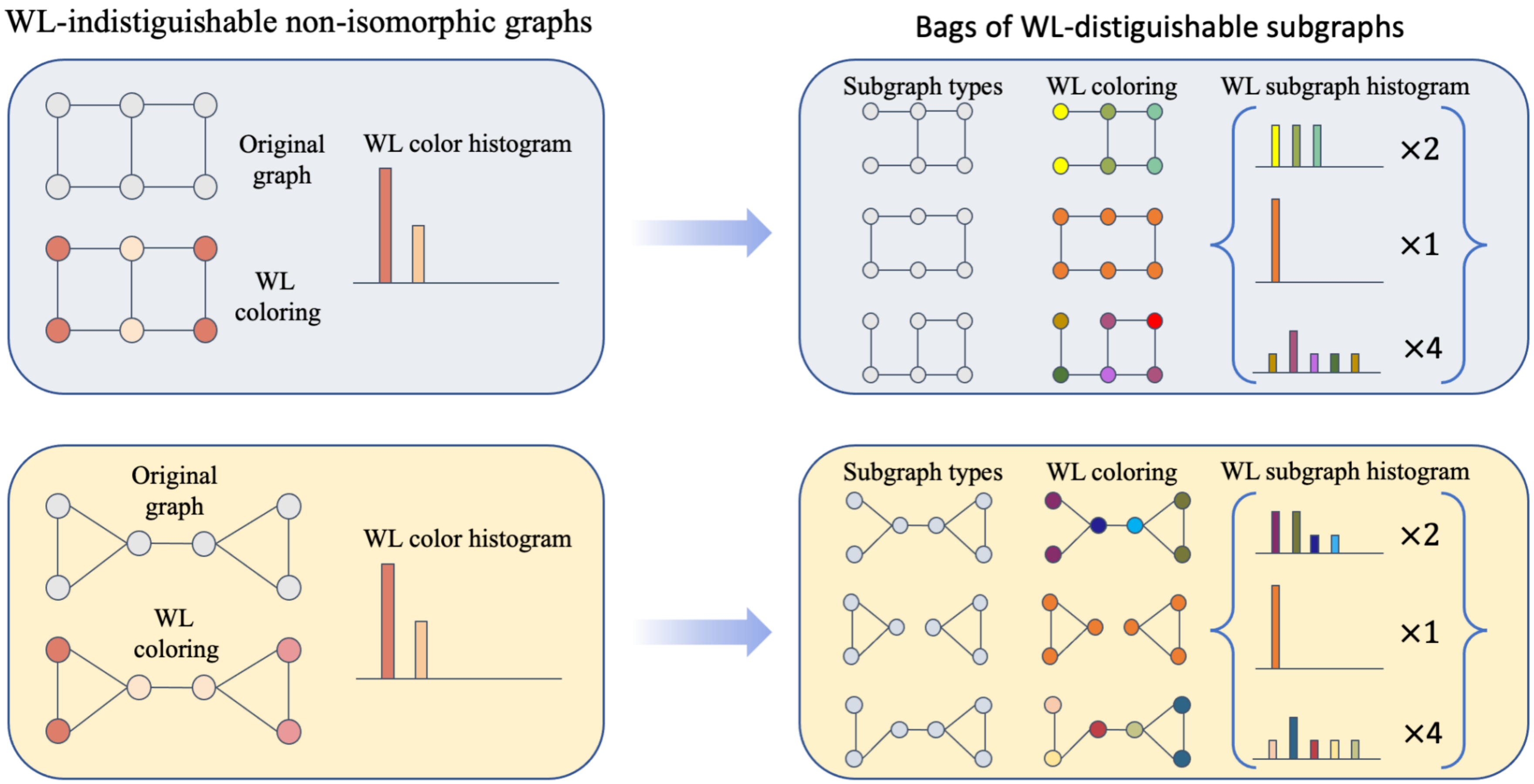
  → $O(n^k)$

Invariant and Equivariant Graph Networks.
https://doi.org/10.48550/arXiv.1812.09902

On the Universality of Invariant Networks.
https://doi.org/10.48550/arXiv.1901.09342

Universal Invariant and Equivariant Graph Neural Networks.
https://doi.org/10.48550/arXiv.1905.04943

Subgraph GNN



ESAN (DSS-GNN)

Equivariant Subgraph Aggregation Networks.
https://doi.org/10.48550/arXiv.2110.02910

Subgraph GNN: Graph Isomorphism

- Expressive power

  → Subgraph GNN (SUN) = 3-IGN = 3-WL

- Empirical performance

  → Subgraph GNN (SUN ≥ GNN-AK^{+} ≥ DSS-GNN) > k-WL / k-IGN

Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries.
https://doi.org/10.48550/arXiv.2206.11140

Agenda

(~2018) Message passing

→ Graph isomorphism, Weisfeiler-Leman, MPNN

(~2022) High-order modeling

→ Node features, k-WL (k-GNN, k-IGN), subgraph GNN

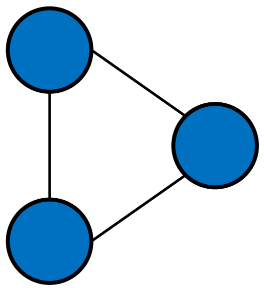
(~2024) Biconnectivity

→ Biconnectivity, GD-WL, Graphormer-GD

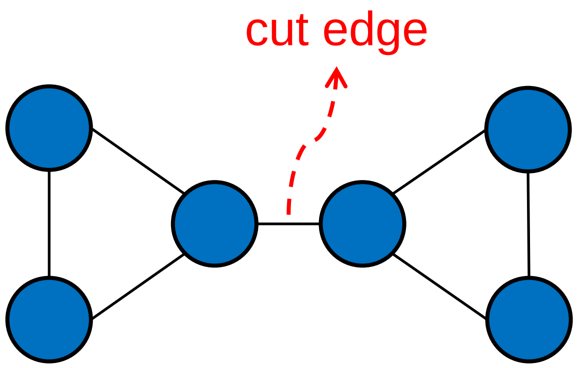
Biconnectivity

- Edge-biconnected graph

  → Remove any edge and the graph is still connected



edge-biconnected

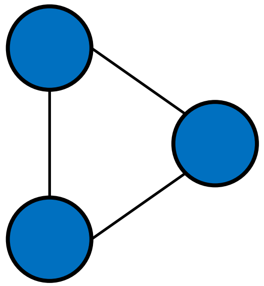


not edge-biconnected

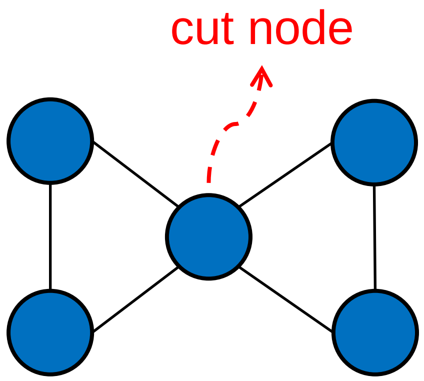
Biconnectivity

- Node-biconnected graph

  → Remove any node and the graph is still connected



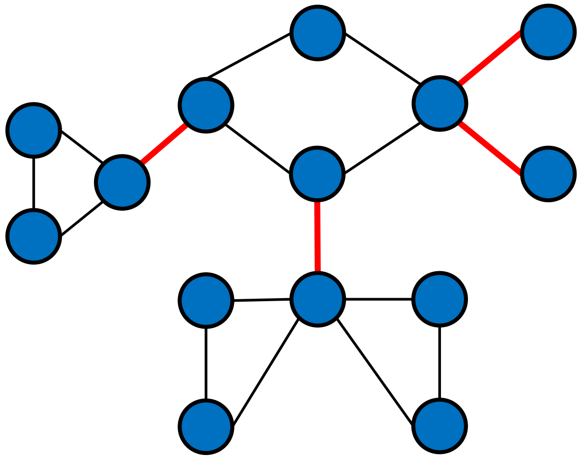
node-biconnected



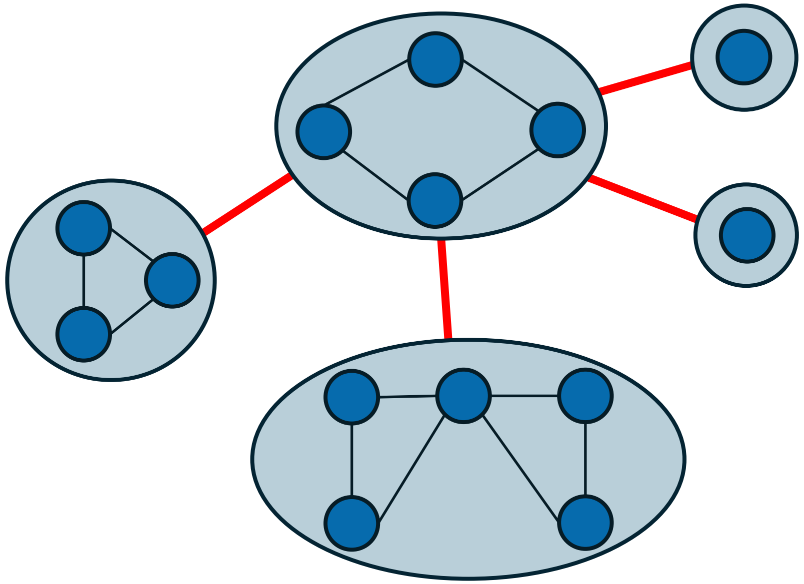
not node-biconnected

Biconnectivity

- A connected graph → a tree of biconnected components



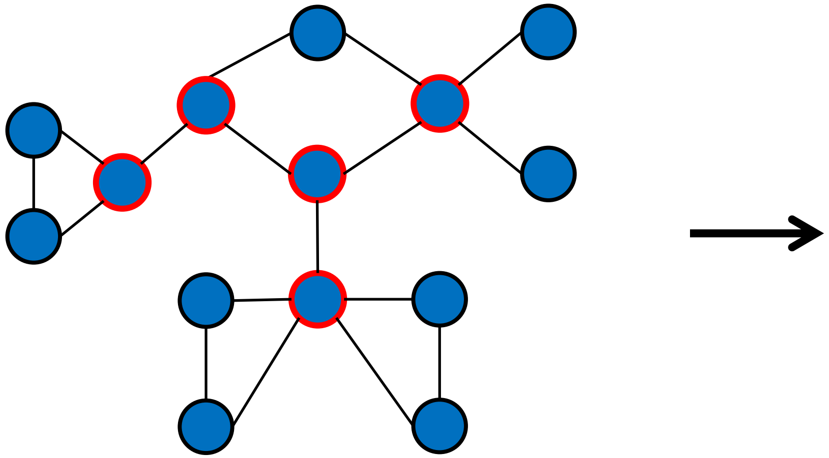
connected graph



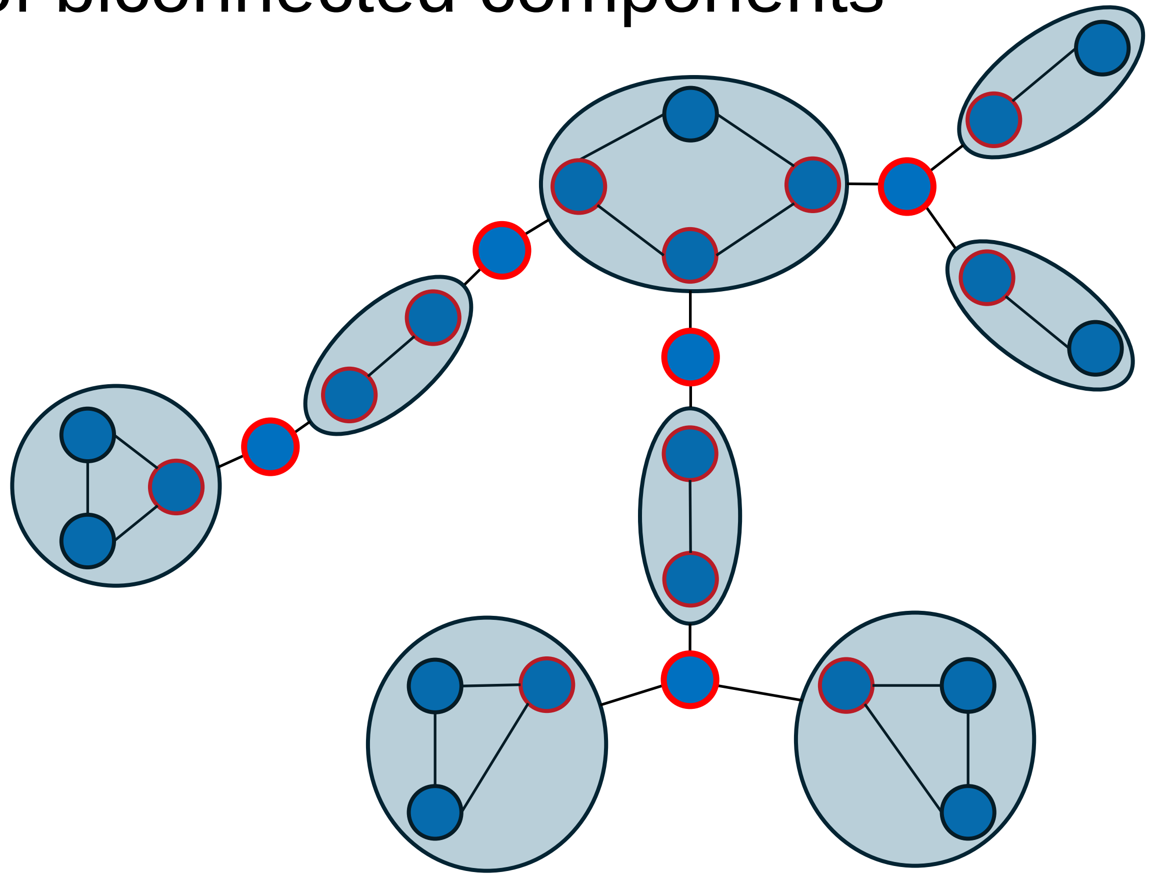
cut-edge tree

Biconnectivity

- A connected graph → a tree of biconnected components



connected graph



cut-node tree

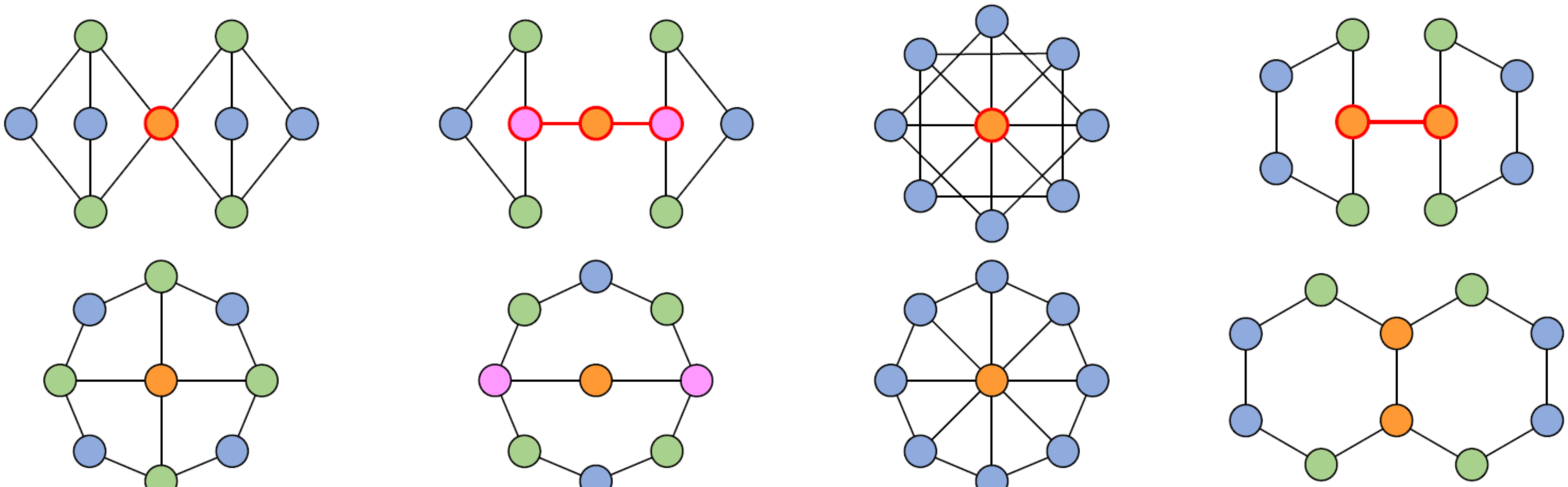
Biconnectivity: Importance

- Real world implications
  - Cut edges → molecule bonds for chemical reactions
  - Cut nodes → key persons linking social groups
  - etc.
- Computational complexity
  - Can be solved in $O(n + e)$ algorithmically → should be solved by GNN

Biconnectivity Problems

- Identify cut edge / cut node
- Distinguish graphs with different cut-edge trees / cut-node trees

Biconnectivity: Contrasting Examples



(a) (b) (c) (d)

Rethinking the Expressive Power of GNNs via Graph Biconnectivity.
https://doi.org/10.48550/arXiv.2301.09505

Biconnectivity: GNNs

- Methods that fail biconnectivity problems

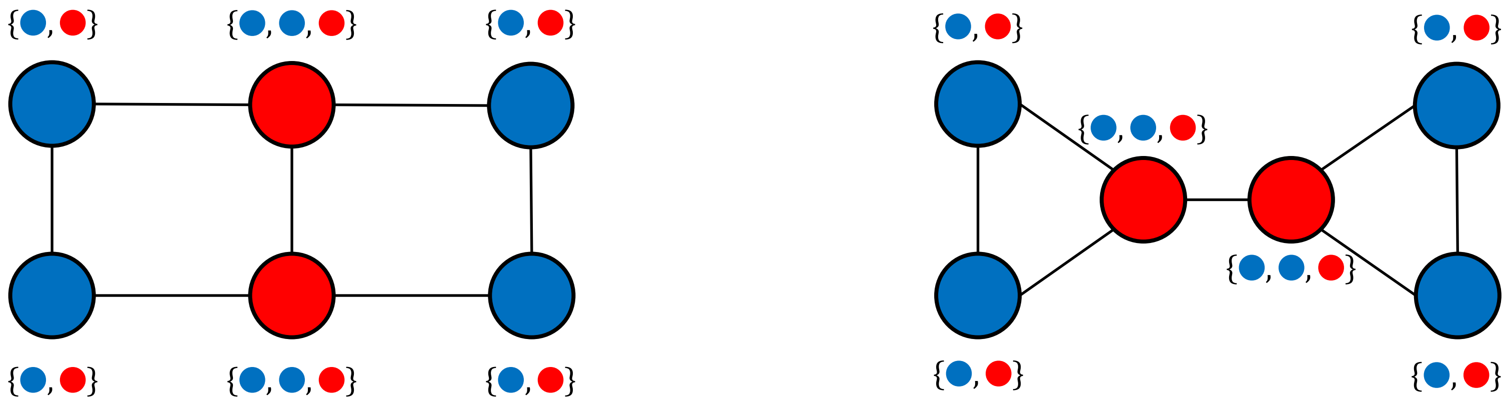
  → 1-WL (MPNN), substructure counting, GraphSNN, GNN-AK, …

- Methods that solves biconnectivity problems

  → RD-WL, ESAN (DSS-GNN), 3-WL, 3-IGN

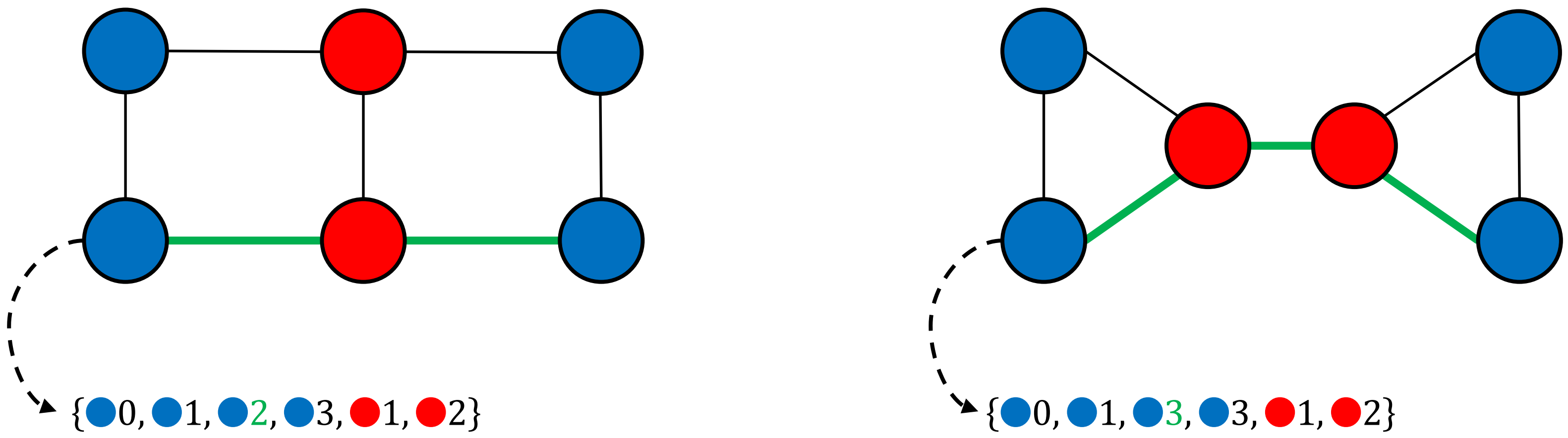
1-WL

For each node: collect <color> from neighbors



Distance-WL

For each node: collect <color, distance> from all nodes



GD-WL

- Generalized Distance (GD)

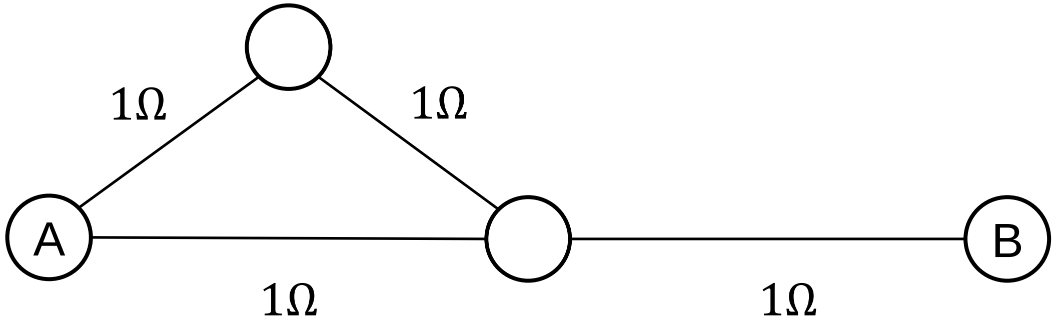
  → Referring to any valid metric for node distance

- Shortest-Path Distance (SPD)

  → SPD-WL solves cut edge and cut-edge tree problems

  → SPD-WL fails cut node and cut-node tree problems

Resistance Distance (RD)



$$\mathrm{RD}(\mathrm{A},\mathrm{B}) = \mathrm{resistance}(\mathrm{A},\mathrm{B}) = \frac{5}{3}\ (\Omega)$$

GD-WL

- Generalized Distance (GD)

  → Referring to any valid metric for node distance

- Shortest-Path Distance (SPD)

  → SPD-WL only solves cut edge and cut-edge tree problems

- Resistance Distance (RD)

  → RD-WL solves all biconnectivity problems

GD-WL: GNN

- To model GD-WL, a GNN must be

invariant to node permutation	→ for the nature of graph data
expressive at node level	→ for encoding node features
modeling global node-node distance	→ for GD
iterating	→ for WL

GD-WL: GNN

- To model GD-WL, a GNN must be

invariant to node permutation	→ attention, (node-level) MLP
expressive at node level	→ (node-level) MLP
modeling global node-node distance	→ relative distance
iterating	→ stacking layers

Transformer

GD-Transformer (Graphormer-GD)

$$\mathbf{A}^h(\mathbf{X}^{(l)}) = \phi_1^{l,h}(\mathbf{D}) \odot \mathrm{softmax}\left(\mathbf{X}^{(l)}\mathbf{W}_Q^{l,h}(\mathbf{X}^{(l)}\mathbf{W}_K^{l,h})^\top + \phi_2^{l,h}(\mathbf{D})\right)$$

$$\hat{\mathbf{X}}^{(l)} = \mathbf{X}^{(l)} + \sum_{h=1}^{H} \mathbf{A}^h(\mathbf{X}^{(l)})\mathbf{X}^{(l)}\mathbf{W}_V^{l,h}\mathbf{W}_O^{l,h}$$

$$\mathbf{X}^{(l+1)} = \hat{\mathbf{X}}^{(l)} + \mathrm{GELU}(\hat{\mathbf{X}}^{(l)}\mathbf{W}_1^l)\mathbf{W}_2^l$$

Rethinking the Expressive Power of GNNs via Graph Biconnectivity.
https://doi.org/10.48550/arXiv.2301.09505

Agenda

(~2018) Message passing

→ Graph isomorphism, Weisfeiler-Leman, MPNN

(~2022) High-order modeling

→ Node features, k-WL (k-GNN, k-IGN), subgraph GNN

(~2024) Biconnectivity

→ Biconnectivity, GD-WL, Graphormer-GD

		1-WL	GD-WL	Subgraph GNN	3-WL
Model		MPNN	Transformer + GD	Node deletion + GNNs	High-order GNN
Complexity		$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^3)$
Theoretical expressiveness	1-WL	O	O	O	O
	Biconnectivity	X	O	O	O
	3-WL	X	?	O	O
Empirical performance		Bad	Good	Good	Bad
Model variants		GCN, GAT, GraphSAGE, GIN[1]	GD-Graphormer[2]	SUN[3], GNN-AK[+4], ESAN (DSS-GNN)[5]	k-GNN[6], δ-k-LWL[+7], k-IGN[8]

[1] GIN

How Powerful are Graph Neural Networks? https://doi.org/10.48550/arXiv.1810.00826

[2] GD-Graphormer

Rethinking the Expressive Power of GNNs via Graph Biconnectivity. https://doi.org/10.48550/arXiv.2301.09505

[3] SUN

Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries. https://doi.org/10.48550/arXiv.2206.11140

[4] GNN-AK$^+$

From Stars to Subgraphs: Uplifting Any GNN with Local Structure Awareness. https://doi.org/10.48550/arXiv.2110.03753

[5] ESAN (DSS-GNN)

Equivariant Subgraph Aggregation Networks. https://doi.org/10.48550/arXiv.2110.02910

[6] k-GNN

Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. https://doi.org/10.1609/aaai.v33i01.33014602

[7] δ-k-LWL$^+$

Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. https://doi.org/10.48550/arXiv.1904.01543

[8] k-IGN

Universal Invariant and Equivariant Graph Neural Networks. https://doi.org/10.48550/arXiv.1905.04943