Rethinking the Expressive Power of GNNs via Graph Biconnectivity (ICLR 2023 Outstanding Paper)

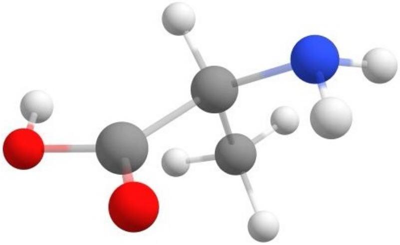
Bohang Zhang, Shengjie Luo, Liwei Wang, Di He

Peking University

June 12, 2023

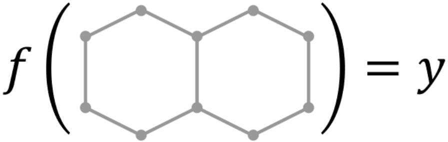
Introduction

- Graph neural networks (GNNs) have become the dominant approach for learning graph-structured data.

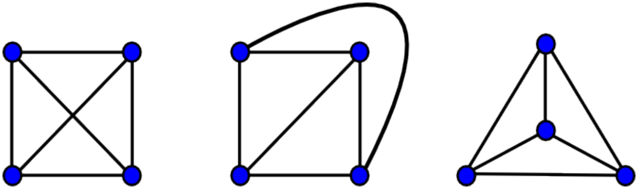


The Expressive Power of GNNs

- Are GNNs able to learn a general function on graphs?

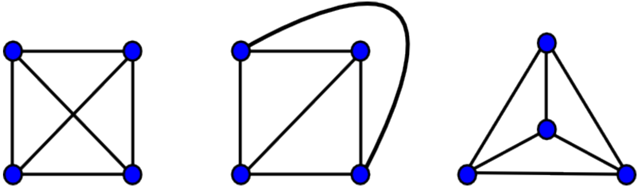


- A highly related condition: GNN should be able to distinguish topologically different graphs.



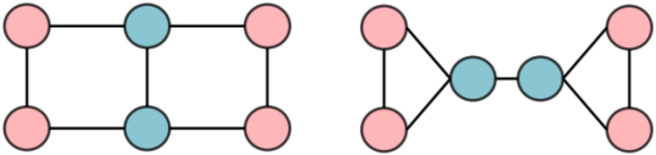
Graph isomorphism

- Graph isomorphism problem: Given two graphs $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$, determine if there is a bijective mapping $f : \mathcal{V}_G \to \mathcal{V}_H$, such that $\{u, v\} \in \mathcal{E}_G$ iff $\{f(u), f(v)\} \in \mathcal{E}_H$.
- Seminal work: Morris et al. [2019], Xu et al. [2019] first linked GNN expressivity to an important algorithm called Weisfeiler-Lehman test [Weisfeiler and Leman, 1968].



MPNNs are at Most as Expressive as 1-WL

- Whenever 1-WL fails to distinguish two non-isomorphic graphs, MPNNs also fail.
- Failure cases:



- It is a central problem to study how to design more expressive GNNs beyond the 1-WL test.

Related Works

- Higher-order GNNs [Morris et al., 2019, 2020, Maron et al., 2019, Geerts and Reutter, 2022]
- Substructure-based GNNs [Bouritsas et al., 2022, Barceló et al., 2021, Bodnar et al., 2021b,a]
- Subgraph GNNs [Cotta et al., 2021, Zhang and Li, 2021, You et al., 2021, Bevilacqua et al., 2022, Zhao et al., 2022, Qian et al., 2022, Frasca et al., 2022, Huang et al., 2023]

However, these methods suffer from at least one of the following drawbacks:

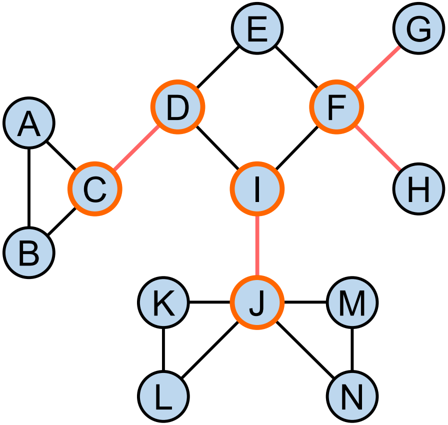
- High computation/memory cost
- Unclear what power they can systematically and provably gain.
- Expressiveness justified by toy examples

Fundamental Questions

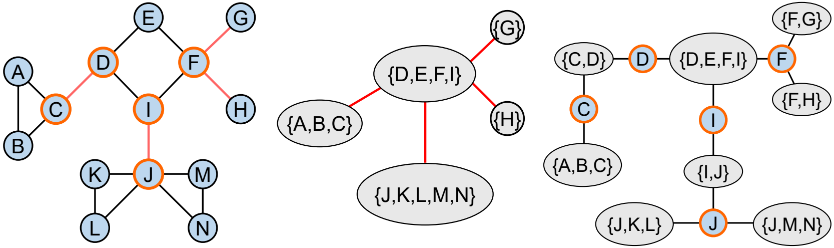
- Can we develop a class of principled and convincing metrics beyond the WL hierarchy that can
  - formally measure the expressive power of different GNN families
  - guide the design of provably better GNN architectures

Graph Biconnectivity

- A central property in graph theory
- Key concepts:
  - cut vertex
  - cut edge
  - biconnected components
  - block cut tree



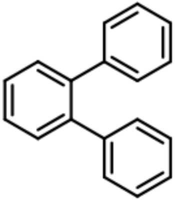
Concepts related to Biconnectivity



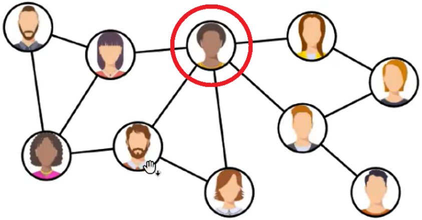
- Cut vertices/edges can be regarded as "hubs" in a graph that link different subgraphs into a whole.
- The link between cut vertices/edges and biconnected components forms exactly a *tree* structure, called the Block Cut-vertex Tree and Block Cut-edge Tree, respectively.

Biconnectivity is Important for Both Theory and Practice

- From a practical perspective:
  - Chemical reactions are highly related to edge-biconnectivity of molecule graphs.
  - Social networks are related to vertex-biconnectivity.

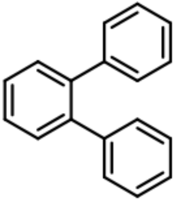


1,2-diphenylbenzene

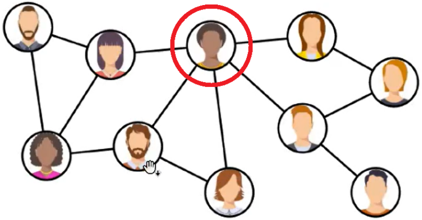


Biconnectivity is Important for Both Theory and Practice

- From a practical perspective:
  - Chemical reactions are highly related to edge-biconnectivity of molecule graphs.
  - Social networks are related to vertex-biconnectivity.
- From a theoretical perspective:
  - Network flow and spanning tree.
  - Planar graph isomorphism [Hopcroft and Tarjan, 1972].



1,2-diphenylbenzene



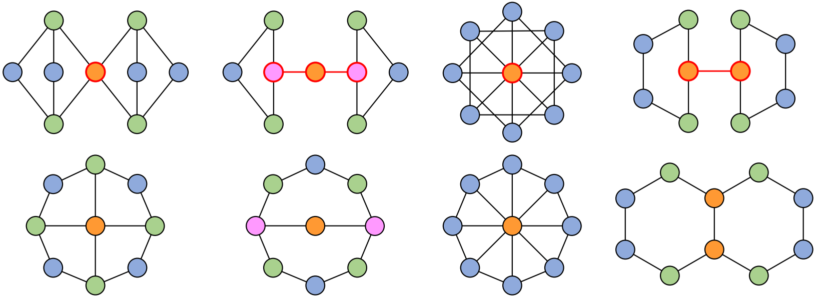
Biconnectivity Can be Efficiently Computed!

- Linear-time algorithm exists for all biconnectivity problems by using Depth-first Search [Tarjan, 1972].
  - Identifying all cut vertices/edges;
  - Finding all biconnected components;
  - Building block cut trees.
- Remark: the complexity is the same as an MPNN!

Problem Formulation

- Three types of biconnectivity problems (with increasing difficulties):
  - **Distinguish whether a graph is vertex/edge-biconnected**:
    for any graphs G, H where G is vertex/edge-biconnected but H is not, their graph representations are different.
  - **Identify cut vertices**:
    for any graphs G, H and nodes $u \in \mathcal{V}_G, v \in \mathcal{V}_H$ where u is a cut vertex but v is not, their node features are different.
    Identify cut edges:
    for any $\{u, v\} \in \mathcal{E}_G$ and $\{w, x\} \in \mathcal{E}_H$ where $\{u, v\}$ is a cut edge but $\{w, x\}$ is not, their edge features are different.
  - **Distinguish block cut-vertex/edge trees**:
    for any graphs G, H satisfying $\mathrm{BCVTree}(G) \not\simeq \mathrm{BCVTree}(H)$ (or $\mathrm{BCETree}(G) \not\simeq \mathrm{BCETree}(H)$), their graph representations are different.

Can 1-WL Solve Biconnectivity Problems?



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

- The answer is no. They cannot even solve the easiest problem: to distinguish whether a graph is vertex/edge-biconnected!

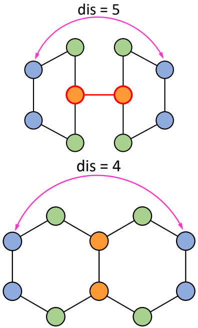
How about Advanced GNN Architectures?

- We investigate three types of popular GNNs in prior works:
  - Substructure-based GNNs [Bouritsas et al., 2022];
  - Simplicial/Cullular GNNs [Bodnar et al., 2021b,a];
  - Overlap Subgraph GNN [Wijesinghe and Wang, 2022];
- Unfortunately, still, none of these GNNs can solve even the easiest biconnectivity task.
- The only exception is a special subgraph GNN called ESAN [Bevilacqua et al., 2022].

Our Motivation

- Problem: **Can we design a principled and efficient GNN framework with provable expressiveness for biconnectivity?**

- Let us restart from the classic 1-WL. Why cannot it encode biconnectivity?



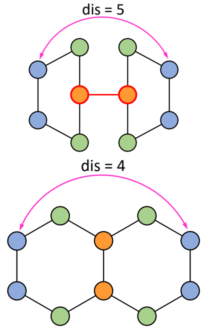
Our Motivation

- Problem: **Can we design a principled and efficient GNN framework with provable expressiveness for biconnectivity?**

- Let us restart from the classic 1-WL. Why cannot it encode biconnectivity?

- We argue that a major weakness is that it is agnostic to *distance information* between nodes, since each node can only "see" its *neighbors* in aggregation.

- Idea: incorporating distance into the aggregation procedure!



Our Approach: GD-WL

Algorithm 1: The Genealized Distance Weisfeiler-Lehman Algorithm

Input : Graph $G = (\mathcal{V}, \mathcal{E})$, distance metric $d_G : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$

Output: Color mapping $\chi_G : \mathcal{V} \to \mathcal{C}$

Initialize: $\chi_G^0(v) := c_0$ for all $v \in \mathcal{V}$ where $c_0 \in \mathcal{C}$ is a fixed color

for $t \leftarrow 1$ **to** T **do**

  for each $v \in \mathcal{V}$ **do**

    $\chi_G^t(v) := \operatorname{hash}\left(\{\!\{(d_G(v,u), \chi_G^{t-1}(u)) : u \in \mathcal{V}\}\!\}\right)$

Return: χ_G^T

Special Case: SPD-WL

- When choosing the *shortest path distance* $d_G = \mathrm{dis}_G$, we obtain SPD-WL.
- It can be equivalently written as

$$\chi_G^{t+1}(v) = \mathrm{hash}\left(\chi_G^t(v), \{\{\chi_G^t(u) : u \in \mathcal{N}_G(v)\}\}, \{\{\chi_G^t(u) : \mathrm{dis}_G(v,u) = 2\}\}, \right.$$
$$\left. \cdots, \{\{\chi_G^t(u) : \mathrm{dis}_G(v,u) = n-1\}\}, \{\{\chi_G^t(u) : \mathrm{dis}_G(v,u) = \infty\}\}\right).$$

- It is strictly more powerful than 1-WL since it additionally aggregates the k-hop neighbors for all $k > 1$.

Special Case: SPD-WL

- SPD-WL is fully expressive for edge-biconnectivity.

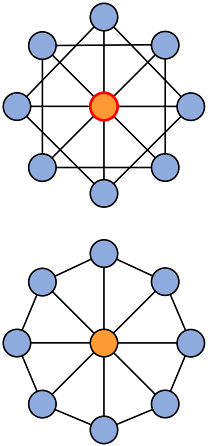
Theorem

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be two graphs, and let χ_G and χ_H be the corresponding SPD-WL color mapping. Then the following holds:

- For any two edges $\{w_1, w_2\} \in \mathcal{E}_G$ and $\{x_1, x_2\} \in \mathcal{E}_H$, if $\{\!\{\chi_G(w_1), \chi_G(w_2)\}\!\} = \{\!\{\chi_H(x_1), \chi_H(x_2)\}\!\}$, then $\{w_1, w_2\}$ is a cut edge if and only if $\{x_1, x_2\}$ is a cut edge.
- If $\{\!\{\chi_G(w) : w \in \mathcal{V}_G\}\!\} = \{\!\{\chi_H(w) : w \in \mathcal{V}_H\}\!\}$, then $\mathrm{BCETree}(G) \simeq \mathrm{BCETree}(H)$.

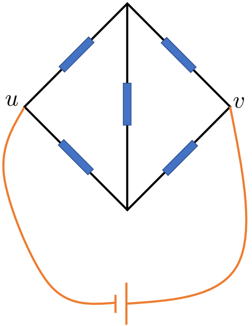
Discussions

- However, SPD-WL cannot distinguish vertex-biconnectivity (see the right figure).



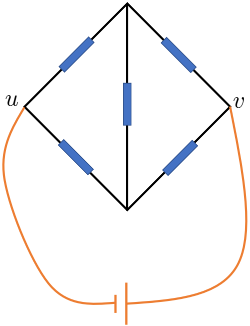
Another Special Case: RD-WL

- Due to the generality of GD-WL, we can use arbitrary distance metrics.
- Another basic metric in graph theory is the *Resistance Distance* (RD).
  - $\mathrm{dis}^{\mathsf{R}}_G(u, v)$: the effective resistance between u and v when treating G as an electrical network where each edge corresponds to a resistance of one ohm.



Another Special Case: RD-WL

- Due to the generality of GD-WL, we can use arbitrary distance metrics.
- Another basic metric in graph theory is the *Resistance Distance* (RD).
    - $\mathrm{dis}_G^{\mathsf{R}}(u, v)$: the effective resistance between u and v when treating G as an electrical network where each edge corresponds to a resistance of one ohm.
- Properties of RD:
    - Valid *metric*: non-negative, semidefinite, symmetric, and satisfies the triangular inequality.
    - Similar to SPD, $0 \le \mathrm{dis}_G^{\mathsf{R}}(u, v) \le n - 1$, and $\mathrm{dis}_G^{\mathsf{R}}(u, v) = \mathrm{dis}_G(u, v)$ if G is a tree.
    - RD is highly related to the graph Laplacian and can be efficiently calculated.



Another Special Case: RD-WL

Theorem

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be two graphs, and let χ_G and χ_H be the corresponding RD-WL color mapping. Then the following holds:

- For any two nodes $w \in \mathcal{V}_G$ and $x \in \mathcal{V}_H$, if $\chi_G(w) = \chi_H(x)$, then w is a cut vertex if and only if x is a cut vertex.
- If $\{\!\{\chi_G(w) : w \in \mathcal{V}_G\}\!\} = \{\!\{\chi_H(w) : w \in \mathcal{V}_H\}\!\}$, then $\mathrm{BCVTree}(G) \simeq \mathrm{BCVTree}(H)$.

- Therefore, RD-WL is fully expressive for vertex-biconnectivity.

Another Special Case: RD-WL

Theorem

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be two graphs, and let χ_G and χ_H be the corresponding RD-WL color mapping. Then the following holds:

- For any two nodes $w \in \mathcal{V}_G$ and $x \in \mathcal{V}_H$, if $\chi_G(w) = \chi_H(x)$, then w is a cut vertex if and only if x is a cut vertex.
- If $\{\!\{\chi_G(w) : w \in \mathcal{V}_G\}\!\} = \{\!\{\chi_H(w) : w \in \mathcal{V}_H\}\!\}$, then $\mathrm{BCVTree}(G) \simeq \mathrm{BCVTree}(H)$.

- Therefore, RD-WL is fully expressive for vertex-biconnectivity.

Corollary

When using both SPD and RD (i.e., by setting $d_G(u, v) := (\mathrm{dis}_G(u, v), \mathrm{dis}^{\mathsf{R}}_G(u, v))$), the corresponding GD-WL is fully expressive for both vertex-biconnectivity and edge-biconnectivity.

Practical Implementation

- GD-WL enjoys great simplicity and full parallelizability.
- **Graphormer-GD**: (A Transformer-like architecture)

$$\mathbf{Y}^h = \left[\phi_1^h(\mathbf{D}) \odot \mathrm{softmax}\left(\mathbf{X}\mathbf{W}_Q^h(\mathbf{X}\mathbf{W}_K^h)^\top + \phi_2^h(\mathbf{D})\right)\right] \mathbf{X}\mathbf{W}_V^h$$

- Computational cost: $O(n^2)$.

Theorem

When choosing proper functions ϕ_1^h and ϕ_2^h and using a sufficiently large number of heads and layers, Graphormer-GD is as powerful as GD-WL.

Detecting Cut Vertices/Edges

Accuracy on cut vertex (articulation point) and cut edge (bridge) detection tasks.

Model	Cut Vertex Detection	Cut Edge Detection
GCN [Kipf and Welling, 2017]	51.5%±1.3%	62.4%±1.8%
GAT [Veličković et al., 2018]	52.0%±1.3%	62.8%±1.9%
GIN [Xu et al., 2019]	53.9%±1.7%	63.1%±2.2%
GSN [Bouritsas et al., 2022]	60.1%±1.9%	70.7%±2.1%
Graphormer [Ying et al., 2021]	76.4%±2.8%	84.5%±3.3%
Graphormer-GD (ours)	100%	100%
- w/o. Resistance Distance	83.3%±2.7%	100%

- GD-WL achieves 100% accuracy on both tasks, which is consistent to our theory. In contrast, prior GNNs fails on both tasks.

ZINC Dataset

Method	Model	Time (s)	Params	Test MAE ZINC-Subset	Test MAE ZINC-Full
MPNNs	GIN [Xu et al., 2019]	8.05	509,549	0.526±0.051	0.088±0.002
	GraphSAGE [Hamilton et al., 2017]	6.02	505,341	0.398±0.002	0.126±0.003
	GAT [Veličković et al., 2018]	8.28	531,345	0.384±0.007	0.111±0.002
	GCN [Kipf and Welling, 2017]	5.85	505,079	0.367±0.011	0.113±0.002
Higher-order GNNs	RingGNN [Chen et al., 2019]	178.03	527,283	0.353±0.019	-
	3WLGNN [Maron et al., 2019]	179.35	507,603	0.303±0.068	-
Substructure-based GNNs	GSN [Bouritsas et al., 2022]	-	∼500k	0.101±0.010	-
	CIN-Small [Bodnar et al., 2021a]	-	∼100k	0.094±0.004	0.044±0.003
Subgraph GNNs	NGNN [Zhang and Li, 2021]	-	∼500k	0.111±0.003	0.029±0.001
	DSS-GNN [Bevilacqua et al., 2022]	-	445,709	0.097±0.006	-
	GNN-AK [Zhao et al., 2022]	-	∼500k	0.105±0.010	-
	GNN-AK+ [Zhao et al., 2022]	-	∼500k	0.091±0.011	-
	SUN [Frasca et al., 2022]	15.04	526,489	0.083±0.003	-
Graph Transformers	GT [Dwivedi and Bresson, 2021]	-	588,929	0.226±0.014	-
	SAN [Kreuzer et al., 2021]	-	508,577	0.139±0.006	-
	Graphormer [Ying et al., 2021]	12.26	489,321	0.122±0.006	0.052±0.005
GD-WL	Graphormer-GD (ours)	12.52	502,793	**0.081**±**0.009**	**0.025**±**0.004**

Thank You!

References I

Pablo Barceló, Floris Geerts, Juan Reutter, and Maksimilian Ryschkov. Graph neural networks with local graph parameters. In *Advances in Neural Information Processing Systems*, volume 34, pages 25280–25293, 2021.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montufar, and Michael M. Bronstein. Weisfeiler and lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pages 1026–1037. PMLR, 2021b.

References II

Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *Advances in neural information processing systems*, 32, 2019.

Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. Reconstruction for powerful graph representations. In *Advances in Neural Information Processing Systems*, volume 34, pages 1713–1726, 2021.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

References III

Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. Understanding and extending subgraph gnns by rethinking their symmetries. *ArXiv*, abs/2206.11140, 2022.

Floris Geerts and Juan L Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 1025–1035, 2017.

John E Hopcroft and Robert Endre Tarjan. Isomorphism of planar graphs. In *Complexity of computer computations*, pages 131–152. Springer, 1972.

Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. Boosting the cycle counting power of graph neural networks with i\$^2\$-GNNs. In *The Eleventh International Conference on Learning Representations*, 2023.

References IV

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in neural information processing systems*, volume 32, pages 2156–2167, 2019.

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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

References V

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: towards scalable higher-order graph embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 21824–21840, 2020.

Chendi Qian, Gaurav Rattan, Floris Geerts, Christopher Morris, and Mathias Niepert. Ordered subgraph aggregation networks. *arXiv preprint arXiv:2206.11168*, 2022.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1 (2):146–160, 1972.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

References VI

Asiri Wijesinghe and Qing Wang. A new perspective on" how graph neural networks go beyond weisfeiler-lehman?". In *International Conference on Learning Representations*, 2022.

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

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10737–10745, 2021.

Muhan Zhang and Pan Li. Nested graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 15734–15747, 2021.

References VII

Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any gnn with local structure awareness. In *International Conference on Learning Representations*, 2022.