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

Bohang Zhang, Shengjie Luo, Liwei Wang, Di He

Peking University

2021.3.31

## Index

(1) Introduction
(2) Biconnectivity
(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

44 Generalized Distance Weisfeiler-Lehman Test
(5) Conclusion

## Index

(1) Introduction

2 Biconnectivity
(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

44 Generalized Distance Weisfeiler-Lehman Test
(5) Conclusion

## Introduction

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



## Introduction

- Message-passing neural networks (MPNNs) [Gilmer et al., 2017, Kipf and Welling, 2017, Hamilton et al., 2017, Veličković et al., 2018]:
- Maintain a node feature $h(v)$ for each node $v$;
- Update:

$$
h^{(l)}(v)=\operatorname{UPDATE}^{(l)}\left(h^{(l-1)}(v), \operatorname{AGGR}^{(l)}\left(\left\{\left\{h^{(l-1)}(u): u \in \mathcal{N}_{G}(v)\right\}\right\}\right)\right)
$$

- Graph representation is obtained by pooling all node representations.


MPNN Update

## Introduction

- MPNNs:
- Maintain a node feature $h(v)$ for each node $v$;
- Update:

$$
h^{(l)}(v)=\operatorname{UPDATE}^{(l)}\left(h^{(l-1)}(v), \operatorname{AGGR}^{(l)}\left(\left\{\left\{h^{(l-1)}(u): u \in \mathcal{N}_{G}(v)\right\}\right\}\right)\right)
$$

- Graph representation is obtained by pooling all node representations.
- Examples:
- GCN [Kipf and Welling, 2017]:

$$
\boldsymbol{h}_{v}^{(l)}=\operatorname{ReLU}\left(\boldsymbol{W}\left(\frac{1}{\mathcal{N}_{G}(v)+1} \sum_{u \in \mathcal{N}_{G}(v) \cup v} \boldsymbol{h}_{u}^{(l-1)}\right)+\boldsymbol{b}\right)
$$

- GIN [Xu et al., 2019]:

$$
\boldsymbol{h}_{v}^{(l)}=\operatorname{MLP}\left((1+\epsilon) \boldsymbol{h}_{v}^{(l-1)}+\sum_{u \in \mathcal{N}_{G}(v)} \boldsymbol{h}_{u}^{(l-1)}\right)
$$

## 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 $G=\left(\mathcal{V}_{G}, \mathcal{E}_{G}\right)$ and $H=\left(\mathcal{V}_{H}, \mathcal{E}_{H}\right)$, determine if there is a bijective mapping $f: \mathcal{V}_{G} \rightarrow \mathcal{V}_{H}$, such that $\{u, v\} \in \mathcal{E}_{G}$ iff $\{f(u), f(v)\} \in \mathcal{E}_{H}$.
- Hardness: no polynomial algorithm has been found.
- Therefore, to study the expressive power of GNNs, it is important to characterize what graphs GNNs cannot distinguish.
- 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].



## The Classic Weisfeiler-Lehman Test

- Given a graph $G=(\mathcal{V}, \mathcal{E}), 1$-WL computes a color mapping $\chi_{G}: \mathcal{V}_{G} \rightarrow \mathcal{C}$ by iteratively refining each node color using its neighboring node colors.


## Algorithm 1: The 1-dimensional Weisfeiler-Lehman Algorithm

1 Initialize: $\chi_{G}^{0}(v):=c$ for all $v \in \mathcal{V}(c \in \mathcal{C}$ is a fixed color $)$
2 for $t \leftarrow 1$ to $T$ do
$3 \quad$ for each $v \in \mathcal{V}$ do
4

$$
\left\lfloor\chi_{G}^{t}(v):=\operatorname{hash}\left(\chi_{G}^{t-1}(v),\left\{\left\{\chi_{G}^{t-1}(u): u \in \mathcal{N}_{G}(v)\right\}\right\}\right)\right.
$$

5 Return: $\chi_{G}^{T}$

- If $\left\{\left\{\chi_{G}(v): v \in \mathcal{V}_{G}\right\}\right\} \neq\left\{\left\{\chi_{H}(v): v \in \mathcal{V}_{H}\right\}\right\}$, then $G$ is not isomorphic to $H$ !


Example of 1-WL (Color refinement) iterations.

## MPNNs are at Most as Expressive as $1-W L$

- 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 $1-\mathrm{WL}$ test.


## Higher-order GNNs

- Leveraging higher-order WL variants to design provably more powerful GNNs [Morris et al., 2019, 2020, Maron et al., 2019, Geerts and Reutter, 2022].



## Higher-order GNNs

- Leveraging higher-order WL variants to design provably more powerful GNNs [Morris et al., 2019, 2020, Maron et al., 2019, Geerts and Reutter, 2022].

- Severe computation/memory costs
- Coarse bound between 1-WL and 3-WL [Morris et al., 2022]
- Unclear about necessity for real-world tasks


## Higher-order GNNs

- Leveraging higher-order WL variants to design provably more powerful GNNs [Morris et al., 2019, 2020, Maron et al., 2019, Geerts and Reutter, 2022].

- Severe computation/memory costs
- Coarse bound between 1-WL and 3-WL [Morris et al., 2022]
- Unclear about necessity for real-world tasks
- Overall, the WL hierarchy is too abstract to guide designing practical GNNs!


## Other Related Works on Expressive GNNs

- Other works still keeps the message-passing framework for efficiency.


## Other Related Works on Expressive GNNs

- Other works still keeps the message-passing framework for efficiency.
- Substructure-based GNNs [Bouritsas et al., 2022, Barceló et al., 2021, Bodnar et al., 2021b,a]:
- Based on heuristics and requiring specific domain knowledge.


## Other Related Works on Expressive GNNs

- Other works still keeps the message-passing framework for efficiency.
- Substructure-based GNNs [Bouritsas et al., 2022, Barceló et al., 2021, Bodnar et al., 2021b,a]:
- Based on heuristics and requiring specific domain knowledge.
- 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]:
- Unclear what power they can systematically and provably gain.
- Expressiveness justified by toy examples
- Unclear of the expressivity relation of different design paradigms



## Topics Involved in This Talk

- 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


## Index

(1) Introduction

## (2) Biconnectivity

(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

4 Generalized Distance Weisfeiler-Lehman Test
(5) Conclusion

## 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.


1,2-diphenylbenzene

- From a theoretical perspective:
- Network flow and spanning tree.
- Planar graph isomorphism [Hopcroft and Tarjan, 1972].



## 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!


## Index

## (1) Introduction

(2) Biconnectivity
(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

44 Generalized Distance Weisfeiler-Lehman Test
(5) Conclusion

## Problem Formulation

- Most common GNN architectures can be cast into corresponding color refinement (CR) algorithms.
- A CR algorithm takes a graph $G$ as input and outputs a color mapping $\chi_{G}: \mathcal{V}_{G} \rightarrow \mathcal{C}$ where $\mathcal{C}$ is called the color set.
- Several concepts in a CR algorithm:
- Node feature: $\chi_{G}(u)$ for $u \in \mathcal{V}$
- Edge feature: $\left\{\left\{\chi_{G}(u), \chi_{G}(v)\right\}\right\}$ for $\{u, v\} \in \mathcal{E}$
- Graph representation: $\left\{\left\{\chi_{G}(u): u \in \mathcal{V}_{G}\right\}\right\}$


## Problem Formulation

- Three types of biconnectivity problems (with increasing difficulties):


## 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.


## 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.


## 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 BCVTree $(G) \not \approx \operatorname{BCVTree}(H)$ (or $\operatorname{BCETree}(G) \nsucceq \operatorname{BCETree}(H)$ ), 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.




## Subgraph GNNs

- In the last year, subgraph GNNs has emerged as a new trend for designing expressive GNNs.
- Idea: Graphs indistinguishable by MPNNs can be easily distinguished via subgraphs.











## Subgraph GNNs










- Key question:
- How can we transform a graph into subgraphs?
- How can we design equivariant GNNs to process a collection of subgraphs?


## Subgraph Generation Policies

- Commonly-used policies:
- Note deletion [Cotta et al., 2021];
- $k$-hop ego network [Zhang and Li, 2021, You et al., 2021, Zhao et al., 2022, Bevilacqua et al., 2022];


The original graph


Node deletion


2-hop ego network

## Subgraph Generation Policies

- Commonly-used policies:
- Note deletion [Cotta et al., 2021];
- $k$-hop ego network [Zhang and Li, 2021, You et al., 2021, Zhao et al., 2022, Bevilacqua et al., 2022];
- Feature initialization:
- Constant;
- Node marking [Qian et al., 2022];
- Distance encoding [Zhang and Li, 2021, Zhao et al., 2022].


The original graph


Node deletion


2-hop ego network


Constant


Node marking


Distance Encoding

## Equivariant Subgraph Aggregation Network (ESAN)

- The most straightforward approach: separately performing message-passing for each subgraph.

$$
\chi_{G_{i}}^{t+1}(v)=\operatorname{hash}\left(\chi_{G_{i}}^{t}(v),\left\{\left\{\chi_{G_{i}}^{t}(u): u \in \mathcal{N}_{G_{i}}(v)\right\}\right\}\right)
$$

- Recently, Bevilacqua et al. [2022] proposed an advanced approach, called DSS-WL:

$$
\begin{gathered}
\chi_{G_{i}}^{t+1}(v)=\operatorname{hash}\left(\chi_{G_{i}}^{t}(v),\left\{\left\{\chi_{G_{i}}^{t}(u): u \in \mathcal{N}_{G_{i}}(v)\right\}\right\},\right. \\
\left.\chi_{G}^{t}(v),\left\{\left\{\chi_{G}^{t}(u): u \in \mathcal{N}_{G}(v)\right\}\right\}\right) \\
\chi_{G}^{t+1}(v)=\operatorname{hash}\left(\left\{\left\{\chi_{G_{i}}^{t+1}(v): i \in[m]\right\}\right\}\right)
\end{gathered}
$$

- DSS-WL adds cross-graph aggregations.


## The Expressiveness of ESAN

- It is straightforward to see that DSS-WL is strict more powerful than 1-WL.
- However, an in-depth understanding of what additional power DSS-WL gains over 1-WL is still limited, and a theoretical justification of cross-graph aggregation is still lacking.


## Our Result: DSS-WL is Provably Expressive for Both Types of Biconnectivity Problems

## Theorem

Let $G=\left(\mathcal{V}_{G}, \mathcal{E}_{G}\right)$ and $H=\left(\mathcal{V}_{H}, \mathcal{E}_{H}\right)$ be two graphs, and let $\chi_{G}$ and $\chi_{H}$ be the corresponding DSS-WL stable color mapping with node marking policy. 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.
- For any two edges $\left\{w_{1}, w_{2}\right\} \in \mathcal{E}_{G}$ and $\left\{x_{1}, x_{2}\right\} \in \mathcal{E}_{H}$, if $\left\{\left\{\chi_{G}\left(w_{1}\right), \chi_{G}\left(w_{2}\right)\right\}\right\}=\left\{\left\{\chi_{H}\left(x_{1}\right), \chi_{H}\left(x_{2}\right)\right\}\right\}$, then $\left\{w_{1}, w_{2}\right\}$ is a cut edge if and only if $\left\{x_{1}, x_{2}\right\}$ is a cut edge.
- The proof is highly technical but insightful.


## How can DSS-WL distinguish biconnectivity?

- Our proof discovers a crucial advantage of DSS-WL: it implicitly encodes distance information!


## Theorem

Let $w$ and $x$ be two nodes in connected graph $G$ with the same DSS-WL color, i.e. $\chi_{G}(w)=\chi_{G}(x)$. Then for any color $c \in \mathcal{C}$,

$$
\left\{\left\{\operatorname{dis}_{G}(w, v): v \in \chi_{G}^{-1}(c)\right\}\right\}=\left\{\left\{\operatorname{dis}_{G}(x, v): v \in \chi_{G}^{-1}(c)\right\}\right\} .
$$

- We will show distance information plays a vital role in distinguishing biconnectivity when combining with color refinement algorithms.


## Discussions

- Our analysis provides a novel understanding and a strong justification for the success of DSS-WL in two aspects: distance and biconnectivity. Both are fundamental structural properties of graphs but are lacking in 1-WL.


## Discussions

- Our analysis provides a novel understanding and a strong justification for the success of DSS-WL in two aspects: distance and biconnectivity. Both are fundamental structural properties of graphs but are lacking in 1-WL.
- How about other graph generation policies?
- In contrast, the ego-network policy $\pi_{\mathrm{EGO}(k)}$ cannot distinguish cut vertices.
- Implication: the ego-network policy with node marking is strictly more expressive than without marking.


## Discussions

- Our analysis provides a novel understanding and a strong justification for the success of DSS-WL in two aspects: distance and biconnectivity. Both are fundamental structural properties of graphs but are lacking in 1-WL.
- How about other graph generation policies?
- In contrast, the ego-network policy $\pi_{\mathrm{EGO}(k)}$ cannot distinguish cut vertices.
- Implication: the ego-network policy with node marking is strictly more expressive than without marking.
- How about vanilla subgraph GNN without cross-graph aggregation?
- We prove that vanilla subgraph GNN cannot identify cut vertices when the color of each node is defined as its associated subgraph representation.
- This theoretically reveals the importance of cross-graph aggregation and justifies the design of DSS-WL.


## Is it done?

- DSS-WL is quite sophisticated, which requires a delicate design of graph generation policies and cross-graph aggregations. Can we develop a simpler approach?
- DSS-WL requires $O\left(n^{2}\right)$ memory and $O(n m)$ computational cost for a graph of $n$ nodes and $m$ edges. Can we develop a more efficient approach?


## Index

(2) Biconnectivity
(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

4 Generalized Distance Weisfeiler-Lehman Test
(5) Conclusion

## Our Motivation

- Problem: Can we design a principled and efficient GNN framework with provable expressiveness for biconnectivity?
- Let us restart from the classic $1-W L$. 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 $1-W L$. 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 2: The Genealized Distance Weisfeiler-Lehman Algorithm
Input : Graph $G=(\mathcal{V}, \mathcal{E})$, distance metric $d_{G}: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_{+}$
Output: Color mapping $\chi_{G}: \mathcal{V} \rightarrow \mathcal{C}$
1 Initialize: $\chi_{G}^{0}(v):=c_{0}$ for all $v \in \mathcal{V}$ where $c_{0} \in \mathcal{C}$ is a fixed color
2 for $t \leftarrow 1$ to $T$ do
for each $v \in \mathcal{V}$ do
$\left\lfloor\chi_{G}^{t}(v):=\operatorname{hash}\left(\left\{\left\{\left(d_{G}(v, u), \chi_{G}^{t-1}(u)\right): u \in \mathcal{V}\right\}\right\}\right)\right.$
5 Return: $\chi_{G}^{T}$

## Special Case: SPD-WL

- When choosing the shortest path distance $d_{G}=\operatorname{dis}_{G}$, we obtain SPD-WL.
- It can be equivalently written as
$\chi_{G}^{t+1}(v)=\operatorname{hash}\left(\chi_{G}^{t}(v),\left\{\left\{\chi_{G}^{t}(u): u \in \mathcal{N}_{G}(v)\right\}\right\},\left\{\left\{\chi_{G}^{t}(u): \operatorname{dis}_{G}(v, u)=2\right\}\right\}\right.$,

$$
\left.\cdots,\left\{\left\{\chi_{G}^{t}(u): \operatorname{dis}_{G}(v, u)=n-1\right\}\right\},\left\{\left\{\chi_{G}^{t}(u): \operatorname{dis}_{G}(v, u)=\infty\right\}\right\}\right) .
$$

- 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 $G=\left(\mathcal{V}_{G}, \mathcal{E}_{G}\right)$ and $H=\left(\mathcal{V}_{H}, \mathcal{E}_{H}\right)$ be two graphs, and let $\chi_{G}$ and $\chi_{H}$ be the corresponding SPD-WL color mapping. Then the following holds:

- For any two edges $\left\{w_{1}, w_{2}\right\} \in \mathcal{E}_{G}$ and $\left\{x_{1}, x_{2}\right\} \in \mathcal{E}_{H}$, if $\left\{\left\{\chi_{G}\left(w_{1}\right), \chi_{G}\left(w_{2}\right)\right\}\right\}=\left\{\left\{\chi_{H}\left(x_{1}\right), \chi_{H}\left(x_{2}\right)\right\}\right\}$, then $\left\{w_{1}, w_{2}\right\}$ is a cut edge if and only if $\left\{x_{1}, x_{2}\right\}$ is a cut edge.
- If $\left\{\left\{\chi_{G}(w): w \in \mathcal{V}_{G}\right\}\right\}=\left\{\left\{\chi_{H}(w): w \in \mathcal{V}_{H}\right\}\right\}$, then $\operatorname{BCETree}(G) \simeq \operatorname{BCETree}(H)$.


## Discussions

- The result is highly non-trivial. It combines three seemingly unrelated concepts (i.e., SPD, biconnectivity, and the WL test) into a unified conclusion.
- Distinguishing non-isomorphic graphs with different block cut-edge trees can be much easily solved than the general case [Cai et al., 1992, Babai, 2016].


## Discussions

- The result is highly non-trivial. It combines three seemingly unrelated concepts (i.e., SPD, biconnectivity, and the WL test) into a unified conclusion.
- Distinguishing non-isomorphic graphs with different block cut-edge trees can be much easily solved than the general
 case [Cai et al., 1992, Babai, 2016].
- 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).
- $\operatorname{dis}_{G}^{\mathrm{R}}(u, v)$ : the effective resistance between $u$ and $v$ when treating $G$ as an electrical network where each edge corresponds to a resistance of one ohm.



## 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).
- $\operatorname{dis}_{G}^{\mathrm{R}}(u, v)$ : 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, $0 \leq \operatorname{dis}_{G}^{\mathrm{R}}(u, v) \leq n-1$, and $\operatorname{dis}_{G}^{\mathrm{R}}(u, v)=\operatorname{dis}_{G}(u, v)$ 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 $G=\left(\mathcal{V}_{G}, \mathcal{E}_{G}\right)$ and $H=\left(\mathcal{V}_{H}, \mathcal{E}_{H}\right)$ be two graphs, and let $\chi_{G}$ and $\chi_{H}$ 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 $\left\{\left\{\chi_{G}(w): w \in \mathcal{V}_{G}\right\}\right\}=\left\{\left\{\chi_{H}(w): w \in \mathcal{V}_{H}\right\}\right\}$, then $\operatorname{BCVTree}(G) \simeq \operatorname{BCVTree}(H)$.
- Therefore, RD-WL is fully expressive for vertex-biconnectivity.


## Another Special Case: RD-WL

## Theorem

Let $G=\left(\mathcal{V}_{G}, \mathcal{E}_{G}\right)$ and $H=\left(\mathcal{V}_{H}, \mathcal{E}_{H}\right)$ be two graphs, and let $\chi_{G}$ and $\chi_{H}$ 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 $\left\{\left\{\chi_{G}(w): w \in \mathcal{V}_{G}\right\}\right\}=\left\{\left\{\chi_{H}(w): w \in \mathcal{V}_{H}\right\}\right\}$, then $\operatorname{BCVTree}(G) \simeq \operatorname{BCVTree}(H)$.
- Therefore, RD-WL is fully expressive for vertex-biconnectivity.


## Corollary

When using both SPD and RD (i.e., by setting $\left.d_{G}(u, v):=\left(\operatorname{dis}_{G}(u, v), \operatorname{dis}_{G}^{\mathrm{R}}(u, v)\right)\right)$, 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)

$$
\mathbf{Y}^{h}=\left[\phi_{1}^{h}(\mathbf{D}) \odot \operatorname{softmax}\left(\mathbf{X} \mathbf{W}_{Q}^{h}\left(\mathbf{X} \mathbf{W}_{K}^{h}\right)^{\top}+\phi_{2}^{h}(\mathbf{D})\right)\right] \mathbf{X} \mathbf{W}_{V}^{h}
$$

- Conputational cost: $O\left(n^{2}\right)$.


## Theorem

When choosing proper functions $\phi_{1}^{h}$ and $\phi_{2}^{h}$ and using a sufficiently large number of heads and layers, Graphormer-GD is as powerful as GD-WL.

## Upper Bound of GD-WL

- The upper bound of the expressiveness of GD-WL is 2-FWL.


## Theorem

The 2-FWL algorithm is more powerful than both SPD-WL and RD-WL.

## Corollary

The 2-FWL is fully expressive for both vertex-biconnectivity and edge-biconnectivity.

## Detecting Cut Vertices/Edges

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

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

- 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 | ZINC-Full |
| MPNNs | GIN [ Xu et al., 2019] | 8.05 | 509,549 | $0.526 \pm 0.051$ | 0.088 $\pm 0.002$ |
|  | GraphSAGE [Hamilton et al., 2017] | 6.02 | 505,341 | $0.398 \pm 0.002$ | $0.126 \pm 0.003$ |
|  | GAT [Veličković et al., 2018] | 8.28 | 531,345 | $0.384 \pm 0.007$ | $0.111 \pm 0.002$ |
|  | GCN [Kipf and Welling, 2017] | 5.85 | 505,079 | $0.367 \pm 0.011$ | $0.113 \pm 0.002$ |
| Higher-order | RingGNN [Chen et al., 2019] | 178.03 | 527,283 | $0.353 \pm 0.019$ |  |
| GNNs | 3WLGNN [Maron et al., 2019] | 179.35 | 507,603 | $0.303 \pm 0.068$ | - |
| Substructurebased GNNs | GSN [Bouritsas et al., 2022] | - | ~500k | $0.101 \pm 0.010$ | - |
|  | CIN-Small [Bodnar et al., 2021a] | - | $\sim 100 \mathrm{k}$ | $0.094 \pm 0.004$ | $0.044 \pm 0.003$ |
| Subgraph GNNs | NGNN [Zhang and Li, 2021] | - | ~500k | $0.111 \pm 0.003$ | $0.029 \pm 0.001$ |
|  | DSS-GNN [Bevilacqua et al., 2022] |  | 445,709 | $0.097 \pm 0.006$ | - |
|  | GNN-AK [Zhao et al., 2022] | - | ~500k | $0.105 \pm 0.010$ | - |
|  | GNN-AK+ [Zhao et al., 2022] | - | ~500k | $0.091 \pm 0.011$ | - |
|  | SUN [Frasca et al., 2022] | 15.04 | 526,489 | $0.083 \pm 0.003$ | - |
| Graph Transformers | GT [Dwivedi and Bresson, 2021] | - | 588,929 | $0.226 \pm 0.014$ | - |
|  | SAN [Kreuzer et al., 2021] | - | 508,577 | $0.139 \pm 0.006$ | - |
|  | Graphormer [Ying et al., 2021] | 12.26 | 489,321 | $0.122 \pm 0.006$ | $0.052 \pm 0.005$ |
| GD-WL | Graphormer-GD (ours) | 12.52 | 502,793 | $\mathbf{0 . 0 8 1} \pm 0.009$ | $\mathbf{0 . 0 2 5} \pm 0.004$ |

## Index

(1) Introduction
(2) Biconnectivity
(3) Investigating Known GNNs Architectures via Graph Biconnectivity

- Problem Formulation
- Failure Examples
- Pravable Expressiveness of ESAN and DSS-WL

4 Generalized Distance Weisfeiler-Lehman Test

## (5) Conclusion

## Take aways

- Graph biconnectivity is a central property.
- Most prior GNNs are not expressive for biconnectivity.
- There are deep relations between distance and biconnectivity.


## Open Directions

- More efficient architectures?
- A deeper understanding of GD-WL (e.g., its spectral properties)
- Encoding other distance metrics?
- Beyond biconnectivity: higher-order connectivity metrics


## References I

László Babai. Graph isomorphism in quasipolynomial time. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing, pages 684-697, 2016.

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.

## References II

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.
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.
Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. Combinatorica, 12(4):389-410, 1992.

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.

## References III

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. AAAI Workshop on Deep Learning on Graphs: Methods and Applications, 2021.
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.
Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In International conference on machine learning, pages 1263-1272. PMLR, 2017.
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.

## References IV

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.
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.

## References V

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.
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.
Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh.
Speqnets: Sparsity-aware permutation-equivariant graph networks. In International Conference on Machine Learning, pages 16017-16042. PMLR, 2022.

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

## References VI

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.
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.

## References VII

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.
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.

## Thank You!

