Coloring Graphs and Beyond

Joint work with Soheil Anbouhi

Abdullah Naeem Malik

Department of Mathematics
College of Arts and Sciences
Florida State University

April 15, 2024

Weisfeiler-Lehman Test[WL68]

Weisfeiler-Lehman Test[WL68]

Definition

A **coloring** of a (undirected) graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$.

Weisfeiler-Lehman Test[WL68]

Definition

A **coloring** of a (undirected) graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$.

Definition

A (perfect) **hashing** is any injective function.

Weisfeiler-Lehman Test[WL68]

Definition

A **coloring** of a (undirected) graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$.

Definition

A (perfect) **hashing** is any injective function.

Basic idea

start with $c(v) = c^{(0)}(v)$, and
$c^{(t+1)}(v) = \text{hash}\left(c^{(t)}(v), \{\{c^{(t)}(w) : w \in N(v)\}\}\right)$[WL68]

Weisfeiler-Lehman Test[WL68]

Definition

A **coloring** of a (undirected) graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$.

Definition

A (perfect) **hashing** is any injective function.

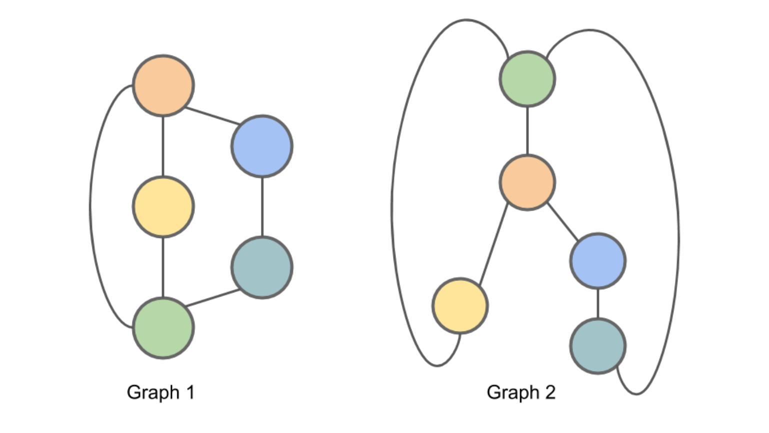
Basic idea

start with $c(v) = c^{(0)}(v)$, and
$c^{(t+1)}(v) = \text{hash}\left(c^{(t)}(v), \{\!\{c^{(t)}(w) : w \in N(v)\}\!\}\right)$[WL68]

Weisfeiler-Lehman Test

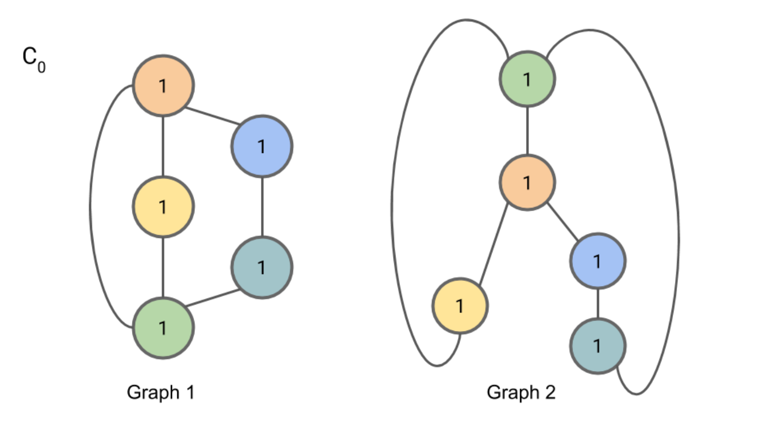
If two graphs have different colorings, then the graphs are not isomorphic. Test is inconclusive if coloring of graphs is the same[MBHSL19]

Weisfeiler-Lehman Test[WL68]



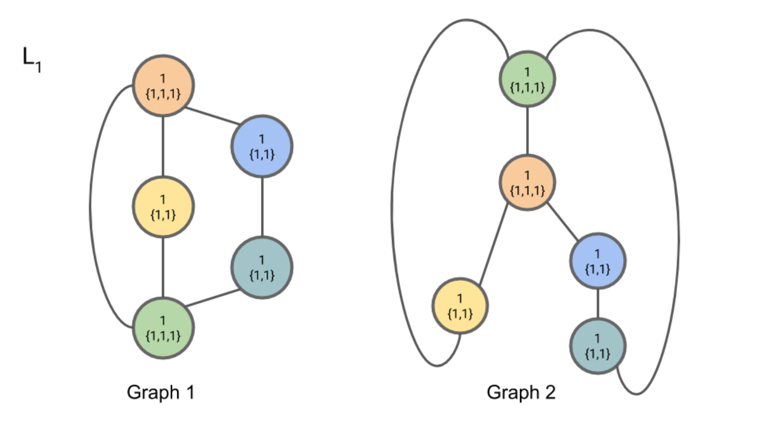
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



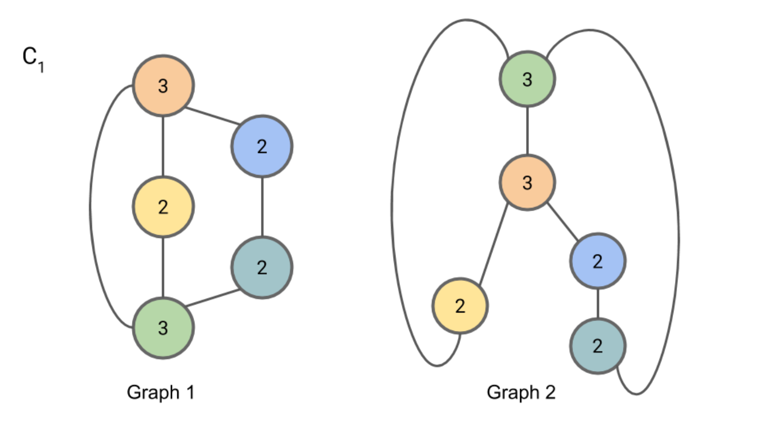
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



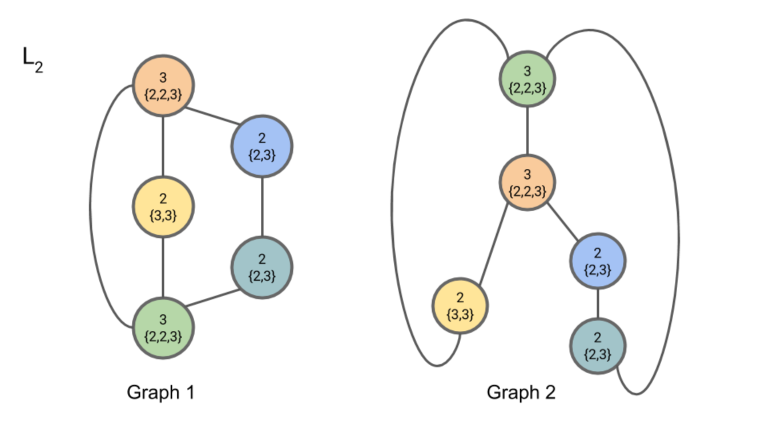
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



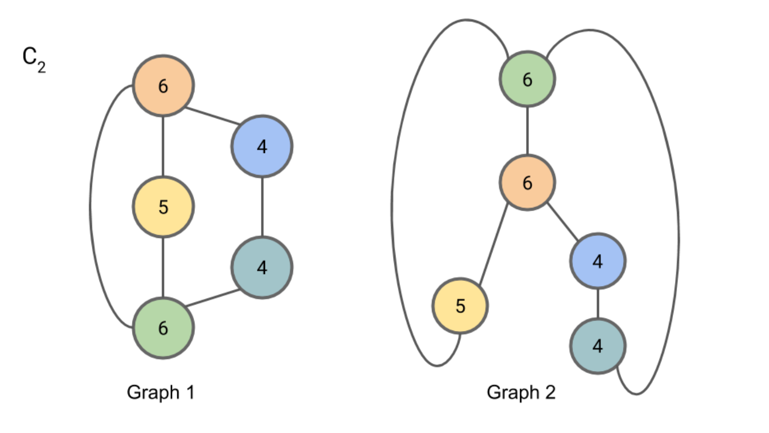
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



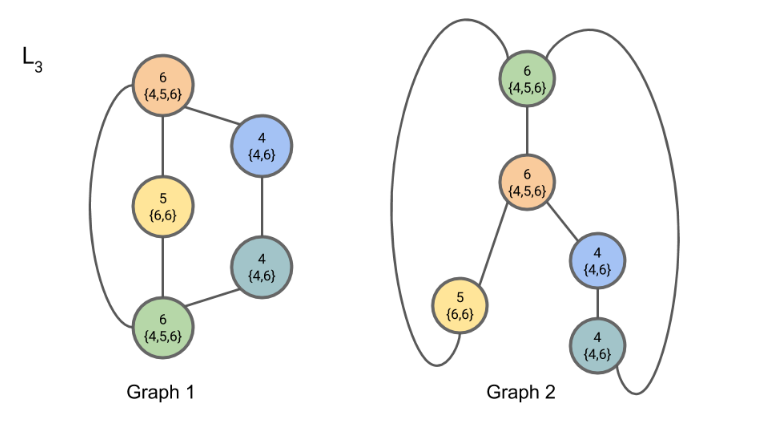
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



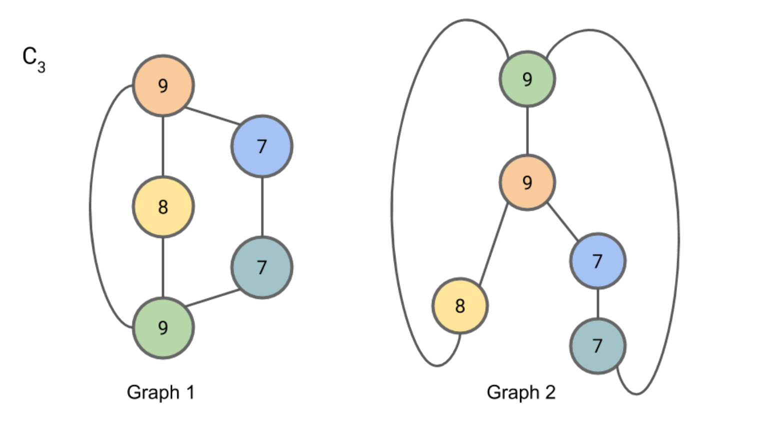
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Lehman Test[WL68]



Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm[WL68]

Recall..

A **coloring** of a graph $G = (V, E)$ at iteration t is a function $c^{(t)} : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Weisfeiler-Leman Algorithm[WL68]

Recall..

A **coloring** of a graph $G = (V, E)$ at iteration t is a function $c^{(t)} : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Algorithm Weisfeiler-Leman (WL) or Naive vertex refinement[WL68]

Input: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
$c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$
while $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**
  $c^{(t+1)}(v) \longleftarrow \mathrm{hash}\left(c^{(t)}(v), \{\!\{c^{(t)}(w) : w \in N(v)\}\!\}\right)$
Output: $c^{(T)}(v)\ \forall v \in V$

Weisfeiler-Leman Algorithm[WL68]

Recall..

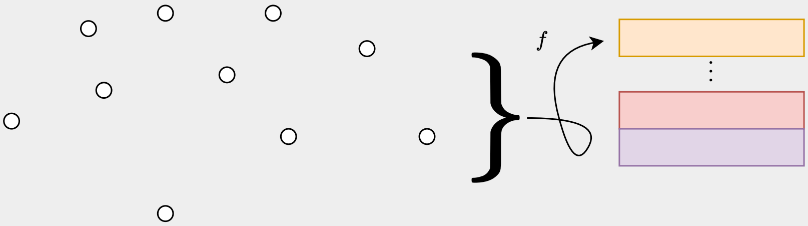
A **coloring** of a graph $G = (V, E)$ at iteration t is a function $c^{(t)} : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Algorithm Weisfeiler-Leman (WL) or Naive vertex refinement[WL68]

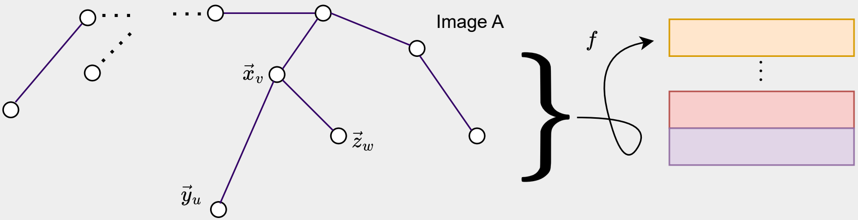
Input: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
$c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$
while $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**
    $c^{(t+1)}(v) \longleftarrow \text{hash}\left(c^{(t)}(v), \{\!\{c^{(t)}(w) : w \in N(v)\}\!\}\right)$
Output: $c^{(T)}(v)\ \forall v \in V$

Secretly message passing!

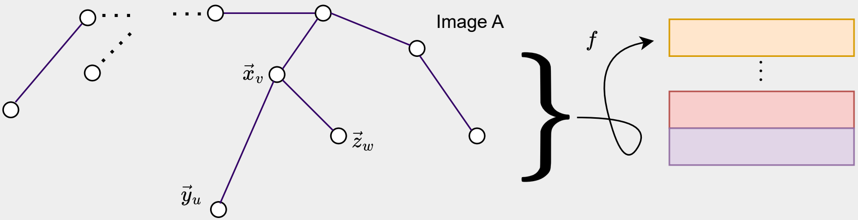
Neural Networks



Message passing in Graph Neural Networks

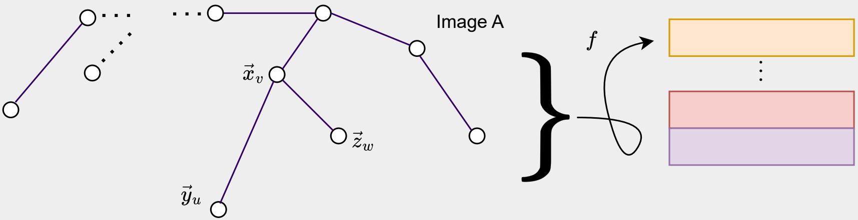


Message passing in Graph Neural Networks



$$x_v^{(k+1)} = \mathrm{COMBINE}\left(x_v^{(k)}, \mathrm{AGGREGATE}^{(k+1)}\left(\left\{x_u^{(k)} : u \in N(v)\right\}\right)\right)$$

Message passing in Graph Neural Networks



$$x_v^{(k+1)} = \mathrm{COMBINE}\left(x_v^{(k)}, \mathrm{AGGREGATE}^{(k+1)}\left(\left\{x_u^{(k)} : u \in N(v)\right\}\right)\right)$$

$$F^{out} = \psi \circ ((AW + I)\, F).$$

$$\psi : \mathbb{R} \longrightarrow \mathbb{R}$$

Examples of Message Passing

AGGREGATE	COMBINE	Ref
$\mathrm{MAX}\left(\left\{\sigma\left(W_1.x_u^{(k)}\right)\right\}, u \in N(v)\right)$	$W_2.\left[x_v^{(k)}, a_v^{(k+1)}\right]$	GraphSAGE[HYL17]
$W_1.\mathrm{MEAN}\left(x_u^{(k)}, u \in N(v) \cup \{v\}\right)$	$\sigma\left(\left\{W_2.a_v^{(k+1)}\right\}\right)$	GCN[KW08]

Message passing in Graph Neural Networks

How powerful are graph neural networks?[MBHSL19, XHLJ18]

Theorem

Let G_1 and G_2 be any two non-isomorphic graphs. If a graph neural network $f : \mathcal{G} \longrightarrow \mathbb{R}^d$ maps G_1 and G_2 to different embeddings, the Weisfeiler-Leman graph isomorphism test also decides G_1 and G_2 are not isomorphic. Converse holds if COMBINE and AGGREGATE are injective[XHLJ18].

Message passing in Graph Neural Networks

How powerful are graph neural networks?[MBHSL19, XHLJ18]

Theorem

Let G_1 and G_2 be any two non-isomorphic graphs. If a graph neural network $f : \mathcal{G} \longrightarrow \mathbb{R}^d$ maps G_1 and G_2 to different embeddings, the Weisfeiler-Leman graph isomorphism test also decides G_1 and G_2 are not isomorphic. Converse holds if COMBINE and AGGREGATE are injective[XHLJ18].

Proof sketch

Compare

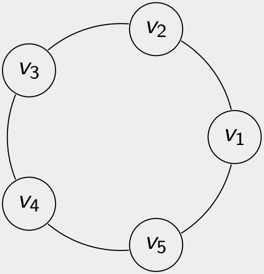
$x_v^{(k+1)} = \mathrm{COMBINE}\Big(x_v^{(k)}, \mathrm{AGGREGATE}^{(k+1)}\Big(\Big\{x_u^{(k)} : u \in N(v)\Big\}\Big)\Big)$

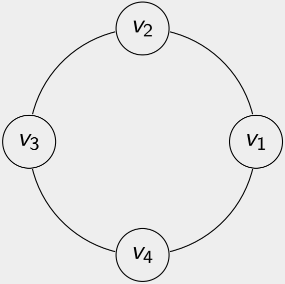
with

$c^{(t+1)}(v) = \mathrm{hash}\big(c^{(t)}(v), \{\!\{c^{(t)}(w) : w \in N(v)\}\!\}\big)$

Limits of WL Test

WL Test fails to distinguish the following graphs:





k-Weisfeiler-Leman Algorithm

Algorithm k-Weisfeiler-Leman (k-WL)

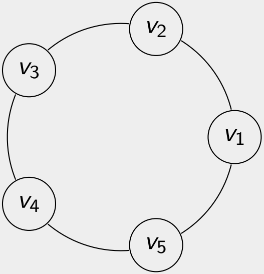
Input: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
$c\left(\overrightarrow{v}\right) = c^{(0)}\left(\overrightarrow{v}\right) \longleftarrow hash\left(x_{\overrightarrow{v}}\right)$
while $c^{(t)}\left(\overrightarrow{v}\right) = c^{(t+1)}\left(\overrightarrow{v}\right) \forall \overrightarrow{v} \in V^k$ **do**
    $c_i^{(t+1)}\left(\overrightarrow{v}\right) \longleftarrow \{\!\{c^{(t)}\left(\overrightarrow{w}\right) : w \in N_i\left(\overrightarrow{v}\right)\}\!\} \; \forall \overrightarrow{v} \in V^k$
    $c^{(t+1)}\left(\overrightarrow{v}\right) \longleftarrow \text{hash}\left(c^{(t)}\left(\overrightarrow{v}\right), c_1^{(t+1)}\left(\overrightarrow{v}\right), \ldots, c_k^{(t+1)}\left(\overrightarrow{v}\right)\right) \; \forall \overrightarrow{v} \in V^k$
Output: $c^{(T)}\left(\overrightarrow{v}\right) \; \forall \overrightarrow{v} \in V^k$

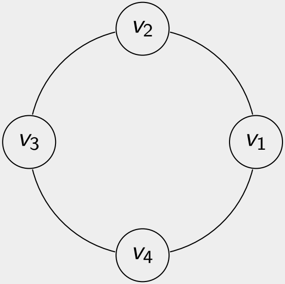
where $\text{hash}(x_{\overrightarrow{v}}) = \text{hash}(x_{\overrightarrow{w}})$ iff **(a)** $x_{v_i} = x_{w_i}$ and **(b)** $(v_i, v_j) \in E$ iff $(w_i, w_j) \in E$.
N.B.: $N_i\left(\overrightarrow{v}\right) = \{(v_1, \ldots, v_{i-1}, u, v_{i+1}, \ldots, v_k) : u \in V\}$.

Weisfeiler-Leman Hierarchies

Weisfeiler-Leman Hierarchies

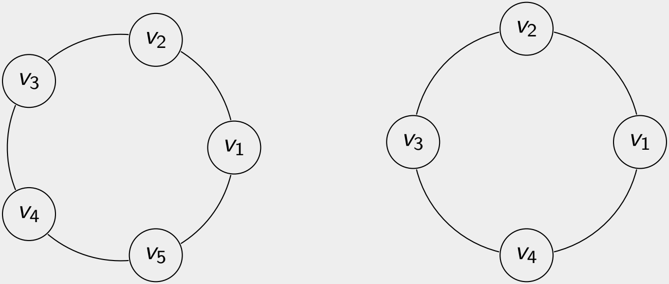
WL Test fails to distinguish the following graphs, whereas 5-WL does not:





Weisfeiler-Leman Hierarchies

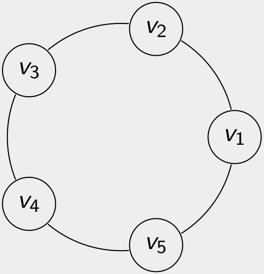
WL Test fails to distinguish the following graphs, whereas 5-WL does not:

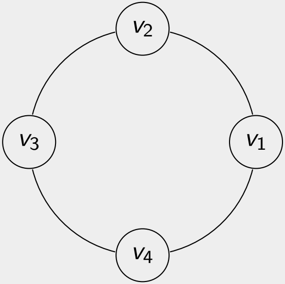


k-WL is strictly weaker than $(k+1)$-WL[HV21]

Weisfeiler-Leman Hierarchies

WL Test fails to distinguish the following graphs, whereas 5-WL does not:





k-WL is strictly weaker than $(k+1)$-WL[HV21]

However.. for every k, there is an infinite family of graphs for which the k-WL test fails[CFI92]

Message Passing on Simplicial Complexes

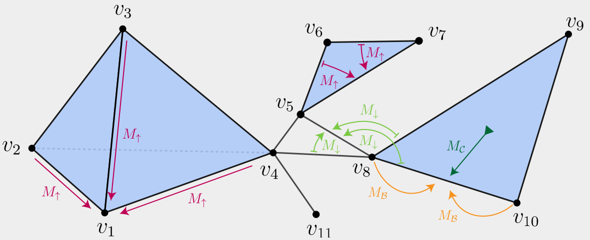


Image from [BFW+21]

Simplicial WL [BFW+21]: $c^{t+1}(\sigma) =$hash of coloring at t of σ and..

Message Passing on Simplicial Complexes

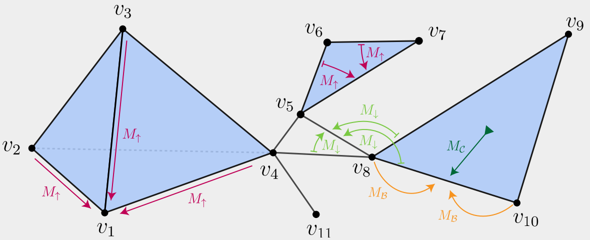


Image from [BFW$^+$21]

Simplicial WL [BFW$^+$21]: $c^{t+1}(\sigma) =$hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

Message Passing on Simplicial Complexes

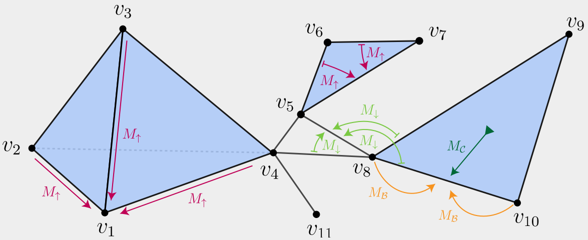


Image from [BFW+21]

Simplicial WL [BFW+21]: $c^{t+1}(\sigma) =$hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

SWL is strictly stronger than 3-WL[BFW+21]

Message Passing on Simplicial Complexes

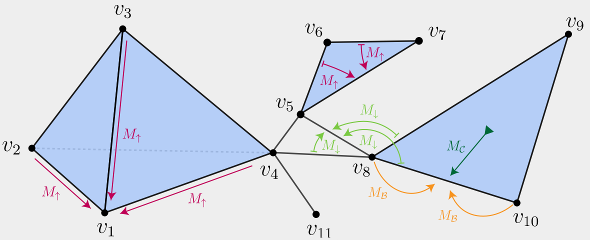


Image from [BFW$^+$21]

Simplicial WL [BFW$^+$21]: $c^{t+1}(\sigma)$ =hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

SWL is strictly stronger than 3-WL[BFW$^+$21]

Lemma

For each $k \leq n$ and $a : V(G)^k \longrightarrow \mathbb{N}$ and $c : \mathcal{K} \longrightarrow \mathbb{N}$, where dim$\mathcal{K} = n$, SWL is strictly stronger than k-WL.

Message Passing on Simplicial Complexes

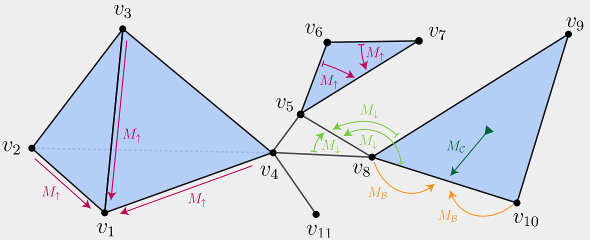


Image from [BFW+21]

Simplicial WL [BFW+21]: $c^{t+1}(\sigma)$ =hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

SWL is strictly stronger than 3-WL[BFW+21]

Lemma

For each $k \leq n$ and $a : V(G)^k \longrightarrow \mathbb{N}$ and $c : \mathcal{K} \longrightarrow \mathbb{N}$, where dim$\mathcal{K} = n$, SWL is (almost) strictly stronger than k-WL.

Can we get better directions?

Can we get better directions?

Algorithm Directed Weisfeiler-Leman (DWL)[MG21]

1: **Input**: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
2: $c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$
3: **while** $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**
4: $c^{(t+1)}(v) \longleftarrow \text{hash}\left(\begin{array}{c} c^{(t)}(v), \{\!\{c^{(t)}(w) : w \in N_{\text{in}}(v)\}\!\}, \\ \{\!\{c^{(t)}(u) : w \in N_{\text{out}}(u)\}\!\} \end{array}\right)$
5: **Output:** $c^{(T)}(v)\ \forall v \in V$

DWL is strictly stronger than WL[BFW$^+$21]

Yes, we can

$$G_1 \rightrightarrows G_0 \qquad \textbf{Directed Graph}$$

Yes, we can

$$\begin{tikzcd} G_1 \arrow[r, bend left, shift left] \arrow[r, bend left] & G_0 \arrow[l, bend left] \end{tikzcd}$$

Directed Graph

Yes, we can

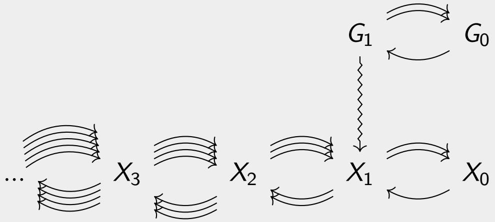
$$\begin{tikzcd}
G_1 \arrow[r, shift left=2] \arrow[r, shift left=4] & G_0 \arrow[l, shift left=2]
\end{tikzcd}$$

Directed Graph

$$\begin{tikzcd}
\cdots \arrow[r, shift left=2] \arrow[r, shift left=4] \arrow[r, shift left=6] \arrow[r, shift left=8] \arrow[r, shift left=10] & X_3 \arrow[l, shift left=2] \arrow[l, shift left=4] \arrow[l, shift left=6] \arrow[l, shift left=8] \arrow[r, shift left=2] \arrow[r, shift left=4] \arrow[r, shift left=6] \arrow[r, shift left=8] & X_2 \arrow[l, shift left=2] \arrow[l, shift left=4] \arrow[l, shift left=6] \arrow[r, shift left=2] \arrow[r, shift left=4] \arrow[r, shift left=6] & X_1 \arrow[l, shift left=2] \arrow[l, shift left=4] \arrow[r, shift left=2] \arrow[r, shift left=4] & X_0 \arrow[l, shift left=2]
\end{tikzcd}$$

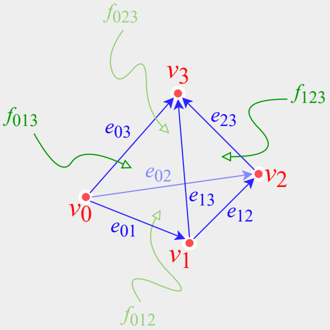
Simplicial Set

Yes, we can



Directed Graph

Simplicial Set



$\nu_0, \nu_1, \nu_2, \nu_3 \in X_0$

$e_{01}, e_{02}, e_{03}, e_{12}, e_{13}, e_{23} \in X_1$

$f_{012}, f_{013}, f_{023}, f_{123} \in X_2$

$g_{0123} \in X_3$

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.
- Update colors according to Simplicial Set Weisfeiler-Lehman Test (SSWL)

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.
- Update colors according to Simplicial Set Weisfeiler-Lehman Test (SSWL)
- If coloring distribution of the two simplicial sets is different, then the two simplicial sets are not isomorphic

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.
- Update colors according to Simplicial Set Weisfeiler-Lehman Test (SSWL)
- If coloring distribution of the two simplicial sets is different, then the two simplicial sets are not isomorphic

Update rules for SSWL

$c^{t+1}(\sigma) =$hash of coloring at t of σ and coloring of its

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.
- Update colors according to Simplicial Set Weisfeiler-Lehman Test (SSWL)
- If coloring distribution of the two simplicial sets is different, then the two simplicial sets are not isomorphic

Update rules for SSWL

$c^{t+1}(\sigma) =$hash of coloring at t of σ and coloring of its boundaries, coboundaries, upper-neighbors and lower-neighbors whenever $\sigma \in X_k$

Simplicial Set WL

- Assign all simplices of every dimension of two simplicial sets the same initial color.
- Update colors according to Simplicial Set Weisfeiler-Lehman Test (SSWL)
- If coloring distribution of the two simplicial sets is different, then the two simplicial sets are not isomorphic

Update rules for SSWL

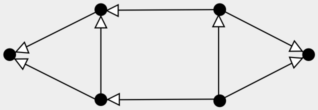
$c^{t+1}(\sigma) =$hash of coloring at t of σ and coloring of its i-th boundaries, i-th coboundaries, i-th upper-neighbors and i-th lower-neighbors for $0 \leq i \leq k$ whenever $\sigma \in X_k$

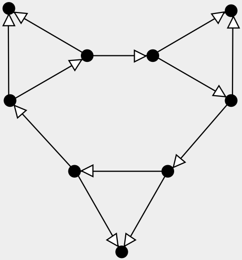
Simplicial Set Weisfeiler-Leman Algorithm

SSWL is strictly stronger than SWL[BFW+21]

Simplicial Set Weisfeiler-Leman Algorithm

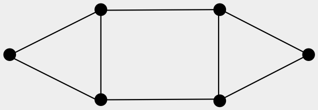
SSWL is strictly stronger than SWL[BFW+21]

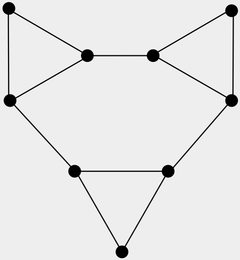




Simplicial Set Weisfeiler-Leman Algorithm

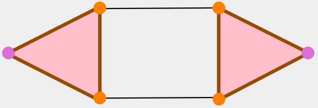
SSWL is strictly stronger than SWL[BFW$^+$21]

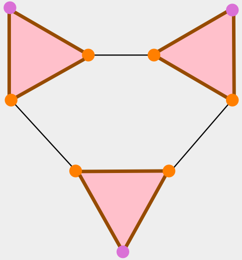




Simplicial Set Weisfeiler-Leman Algorithm

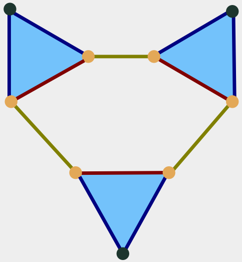
SSWL is strictly stronger than SWL[BFW^{+}21]





Simplicial Set Weisfeiler-Leman Algorithm

SSWL is strictly stronger than SWL[BFW$^+$21]



Message Passing with Higher Dimensional Data

Summary:

$$\begin{array}{ccc} DWL & \sqsubset & WL \\ & & \sqcup \\ & & 3\text{-WL} \\ & & \sqcup \\ & & SWL \end{array}$$

Message Passing with Higher Dimensional Data

Summary:

$$\begin{array}{ccc} DWL & \sqsubset & WL \\ \sqcup & & \sqcup \\ ? & & k\text{-WL} \\ \sqcup & & \sqcup \\ SSWL & \sqsubset & SWL \end{array}$$

References I

- 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*, International Conference on Machine Learning, PMLR, 2021, pp. 1026–1037.
- Jin-Yi Cai, Martin Fürer, and Neil Immerman, *An optimal lower bound on the number of variables for graph identification*, Combinatorica **12** (1992), no. 4, 389–410.
- Glenn Fowler, Robert Haralick, F Gail Gray, Charles Feustel, and Charles Grinstead, *Efficient graph automorphism by vertex partitioning*, Artificial Intelligence **21** (1983), no. 1-2, 245–269.
- Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec, *Open graph benchmark: Datasets for machine learning on graphs*, arXiv preprint arXiv:2005.00687 (2020).

References II

- Ningyuan Teresa Huang and Soledad Villar, *A short tutorial on the weisfeiler-lehman test and its variants*, ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 8533–8537.
- Will Hamilton, Zhitao Ying, and Jure Leskovec, *Inductive representation learning on large graphs*, Advances in neural information processing systems **30** (2017).
- Thomas N Kipf and Max Welling, *Semi-supervised classification with graph convolutional networks*, IEEE Transactions on Neural Networks **20** (2008), no. 1, 61–80.
- Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman, *Provably powerful graph networks*, Advances in neural information processing systems **32** (2019).

References III

Martin Mladenov Pascal Schweitzer Martin Grohe, Kristian Kersting, *Color refinement and its applications*, An Introduction to Lifted Probabilistic Inference (Sriraam Natarajan Davide Poole Guy Van den Broeck, Kristian Kersting, ed.), MIT Press, 2021.

Boris Weisfeiler and Andrei Leman, *The reduction of a graph to canonical form and the algebra which appears therein*, nti, Series **2** (1968), no. 9, 12–16.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, *How powerful are graph neural networks?*, arXiv preprint arXiv:1810.00826 (2018).

Thank you!