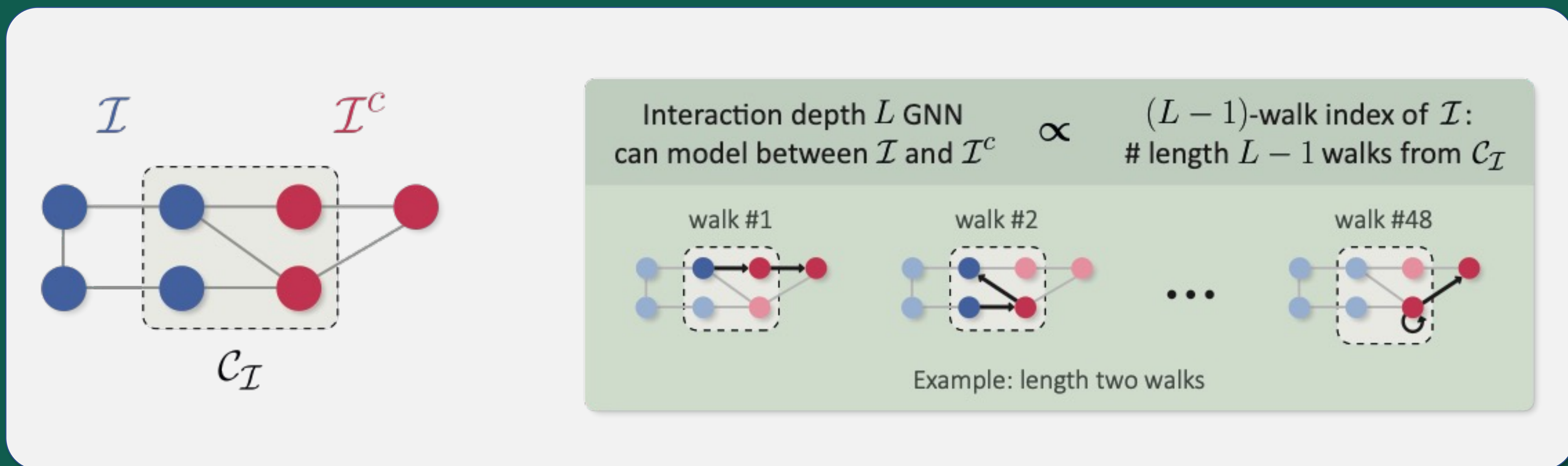
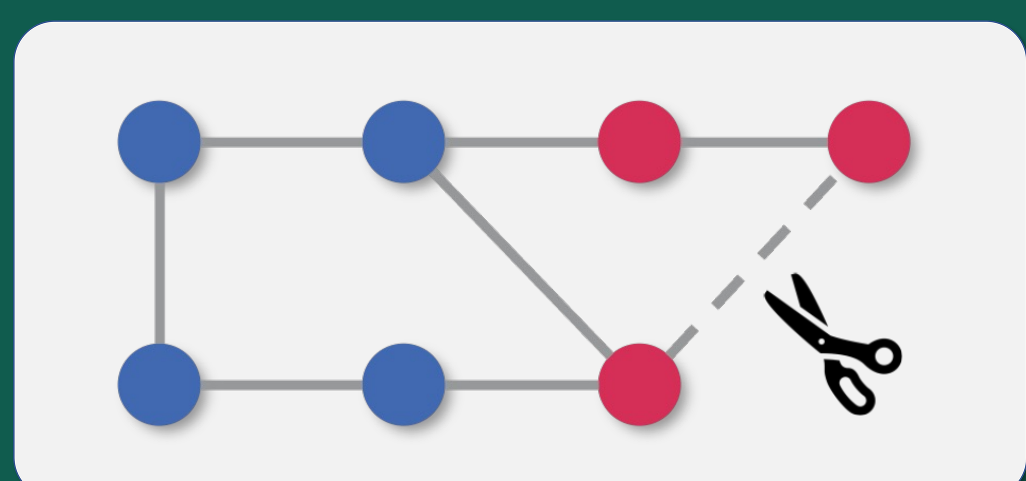
We **quantify the ability** of **Graph Neural Networks** to **model interactions** between vertices



Our theory leads to a **simple & efficient edge sparsification** algorithm that **outperforms** alternative methods



On the Ability of Graph Neural Networks to Model Interactions Between Vertices

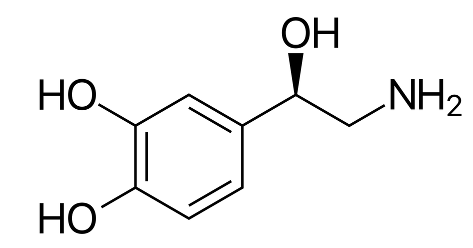
Noam Razin, Tom Verbin, Nadav Cohen
Tel Aviv University

link to paper

1) Expressivity in Graph Neural Networks (GNNs)

GNNs are purposed for **modeling interactions between vertices**

Molecular Data – Graph Prediction



Social Networks – Vertex Prediction

Fundamental Question: ***expressivity*** — which functions can GNNs realize?

Existing Analyses of Expressivity: mostly focus on

Distinguishing non-isomorphic graphs *(e.g. Xu et al. 2019, Morris et al. 2019)*

Computability of graph properties *(e.g. Chen et al. 2020, Garg et al. 2020)*

Limitations of Existing Analyses

(1) Often treat asymptotic regimes of unbounded width or depth

(2) No formalization for ability of GNNs to model interactions

Q: how do graph structure and GNN size affect modeled interactions?

2) Formalizing Strength of Interaction via Separation Rank

Separation Rank: measure of interaction modeled between input variables

For $f : (\mathbb{R}^D)^N \to \mathbb{R}$ and $\mathcal{I} \subseteq \{1, \ldots, N\}$:

$$\mathrm{sep}(f; \mathcal{I}) := \min R \text{ s.t. } f(x_1, \ldots, x_N) = \sum_{r=1}^{R} g_r(\{x_u\}_{u \in \mathcal{I}}) \cdot \bar{g}_r(\{x_v\}_{v \in \mathcal{I}^c})$$

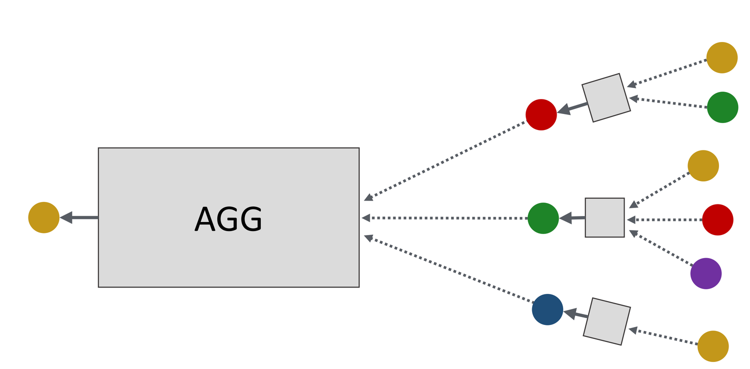
Usages:

(1) Entanglement in physics (2) Analyses of various NN architectures

(e.g. Cohen & Shashua 2017, Levine et al. 2018;2020, R et al. 2022)

3) Analyzed GNN Architecture

Vast majority of GNNs follow message-passing paradigm



$\mathcal{V} := \{1, \ldots, |\mathcal{V}|\}$ — vertices of input graph

$x_1, \ldots, x_{|\mathcal{V}|}$ — input vertex features

$h^{(l,u)}$ — hidden embedding of $u \in \mathcal{V}$ at layer l

$h^{(l,u)} = \mathrm{AGG}(\{ W^{(l)} h^{(l-1,v)} : v \in \mathrm{neigh}(u) \})$

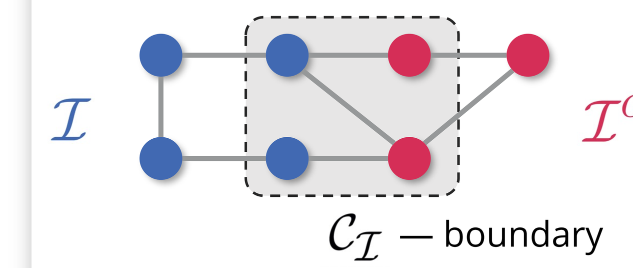
Prior work: studied interactions modeled by other NNs w/ poly non-linearity

(e.g. Cohen & Shashua 2017, Levine et al. 2018;2020, R et al. 2022)

Our theory: message-passing GNNs w/ product aggregation

4) Theory: Quantifying Ability of GNNs to Model Interactions

Walk Index (WI)



$\mathrm{WI}_{L-1}(\mathcal{I}) :=$ # length $L-1$ walks from $\mathcal{C}_\mathcal{I}$

$\mathrm{WI}_{L-1,t}(\mathcal{I}) :=$ # length $L-1$ walks from $\mathcal{C}_\mathcal{I}$ to $t \in \mathcal{V}$

Theorem

For depth L GNN of width D_h, subset of vertices $\mathcal{I} \subseteq \mathcal{V}$, and target $t \in \mathcal{V}$

Graph Prediction	Vertex Prediction
$\mathrm{sep}(GNN; \mathcal{I}) = D_h^{\mathcal{O}(\mathrm{WI}_{L-1}(\mathcal{I}))}$	$\mathrm{sep}(GNN^{(t)}; \mathcal{I}) = D_h^{\mathcal{O}(\mathrm{WI}_{L-1,t}(\mathcal{I}))}$

* Nearly matching lower bounds

Interaction GNNs model across partition is determined by walk index

Experiment: implications of theory apply to various GNNs (e.g. GCN & GIN)

5) Application: Expressivity Preserving Edge Sparsification

Edge Sparsification: remove edges to reduce compute/memory costs

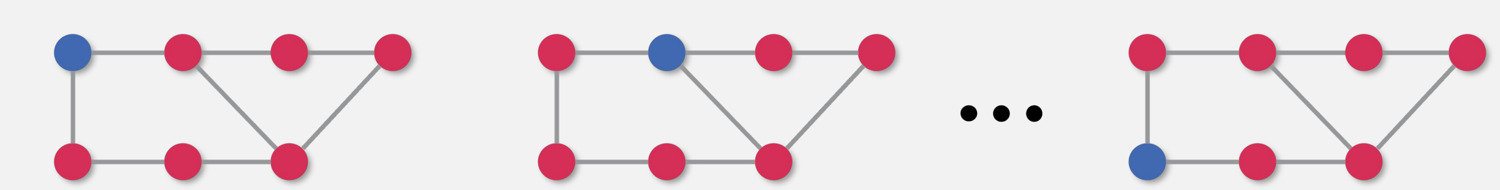
Theory: walk index of $\mathcal{I} \subseteq \mathcal{V}$ key for modeling interaction between $\mathcal{I}$ & $\mathcal{I}^c$

(L – 1)-Walk Index Sparsification (WIS)

Idea: greedily prune edge whose removal harms interactions the least

Algorithm: until desired # edges are removed:

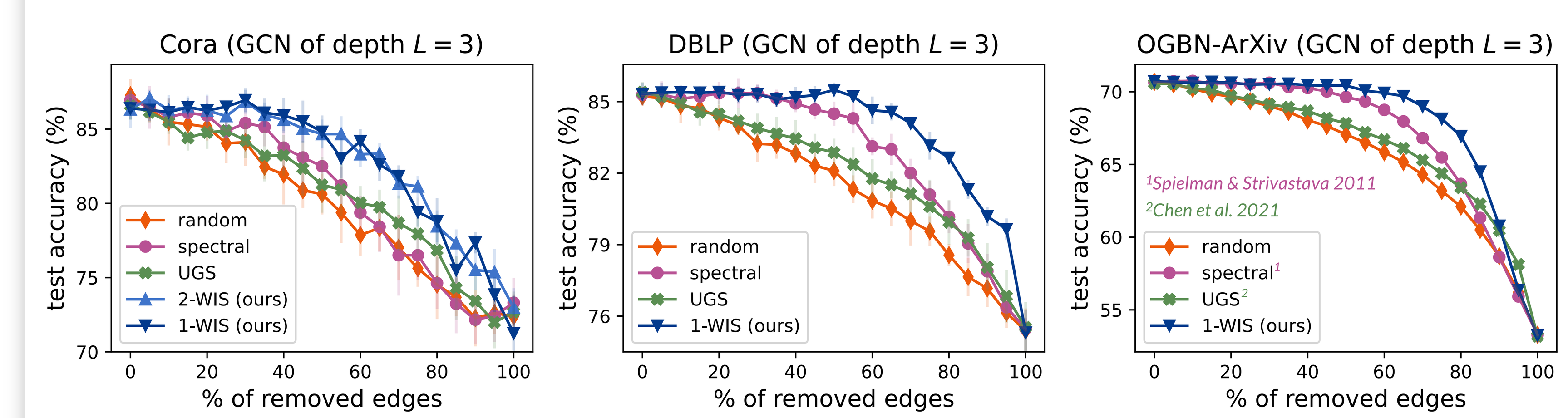
- Per edge, compute tuple holding what the *(L – 1)*-walk indices of $\{1\}, \ldots, \{|\mathcal{V}|\}$ will be if the edge is removed



- Remove edge w/ maximal walk index tuple (by some order over tuples)

1-WIS: particularly simple & efficient implementation

Experiment: comparison of edge sparsification algorithms



WIS outperforms existing methods while being simple & efficient