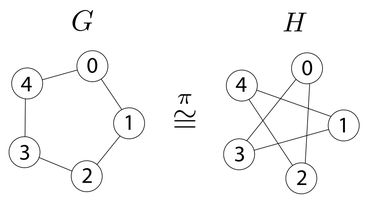
Canonical Tree Cover Neural Networks for Expressive and Invariant Graph Learning

Michael Ito, Danai Koutra, and Jenna Wiens
University of Michigan Computer Science and Engineering,
Correspondence to: mbito@umich.edu

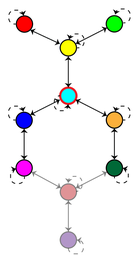
Link to Paper

In graph learning, capturing a graph's **natural symmetries** (isomorphism invariance) is essential for **learning** and **generalization.**

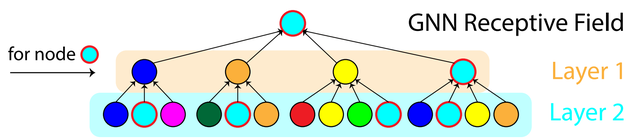
Graph Isomorphism



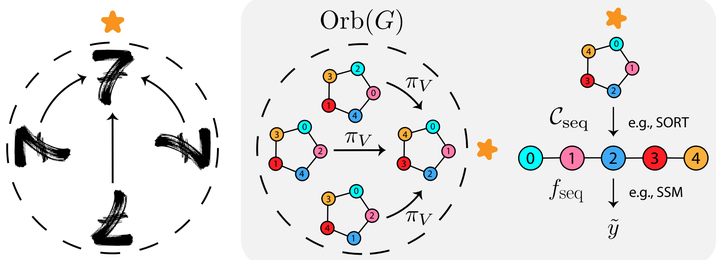
Message-passing GNNs (MPNNs) are naturally **architecturally invariant** on graphs, but are fundamentally limited in **expressive power**, **oversmooth**, and **oversquash**.



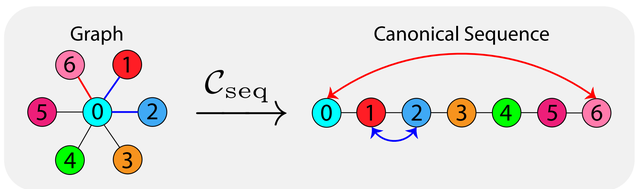
$$\mathbf{h}^{(l+1)}(i) = \text{Aggregate}(\{\mathbf{h}^{(l)}(j) : j \in \mathcal{N}(i)\})$$



Canonicalization offers a powerful alternative by mapping each sample to a **unique invariant representative**. **Canonical GNNs** map each graph to a unique invariant sequence.



- ✦ However, canonical sequences **distort graph distances**
- ✦ Sequences provide a **poor coordinate system** for graphs



Our Proposed Approach

Canonical Tree Cover Neural Networks (CTNNs)

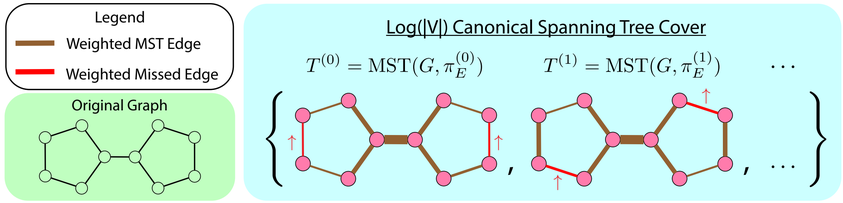
CTNNs leverage a **cover of canonical spanning trees** vs. a single sequence by extracting **MSTs** and using **evolving edge weights** derived from canonical node labels.

Intuitions:

1. Trees better **preserve distances** on sparse graphs
2. Multiple trees **increase expressivity**

Edge Initialization and Updates

$$\pi_E^{(0)}(e) = -\big(\pi_V(e_u) + \pi_V(e_v)\big) \quad \pi_E^{(k+1)}(e) = \pi_E^{(k)}(e) + \tau \mathbf{1}[e \in T^{(k)}]$$



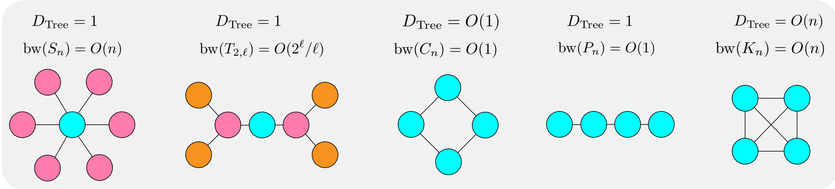
Invariance, Distortion, and Expressivity

Proposition (Probabilistic invariance)**.** A randomized graph representation $X(G)$ is *probabilistically invariant* if its distribution is unchanged under any node relabeling, i.e., $X(G) \overset{d}{=} X(g \cdot G)$ for every permutation $g \in \mathbb{S}_n$. The random output $f_{\text{CTNN}}(G)$ is probabilistically invariant in this sense:

$$f_{\text{CTNN}}(G) \overset{d}{=} f_{\text{CTNN}}(g \cdot G) \quad \text{for all } g \in \mathbb{S}_n.$$

Then, $\Phi(G) \coloneqq \mathbb{E}\big[f_{\text{CTNN}}(G)\big]$ satisfies $\Phi(G) = \Phi(g \cdot G)$ for all $g \in \mathbb{S}_n$.

1. Uniform spanning trees have expected distortion upper bounded by **random walk hitting times.**
2. CTNNs are **more expressive** than traditional MPNNs.



Experimental Results

Design space of Canonical GNNs

Approach	Domain Driven	Node Labeler	Canonicalizer
Fingerprint	Yes	NA	Handcrafted chemical descriptors
SMILES	Yes	Atom ranks	Canonical SMILES algorithm
Primary Seq.	Yes	NA	Identity
DGCNN	No	MPNN	Differentiable sort (SortPooling)
RCM	No	Degree	Reverse Cuthill–McKee ordering
CTNN	No	Degree	Minimum Spanning Tree

Molecule and Protein Benchmarks Median (min, max)

	MoleculeNet	ProteinShake
	PCBA	**GO BIO**
# Graphs	440K	22K
Avg. $\lvert V\rvert$	26.0	254.5
Avg. $\lvert E\rvert$	28.1	698.5
Metric	AUC ↑	ACC ↑
GCN	78.5 (78.0, 79.3)	59.2 (57.9, 69.7)
SMILES	80.4 (80.1, 80.7)	—
Primary Sequence	—	74.3 (69.2, 79.5)
DGCNN	84.9 (84.0, 85.3)	62.0 (59.7, 68.9)
RCM	84.6 (84.5, 84.9)	68.4 (66.7, 69.3)
CTNN (ours)	**87.4 (87.0, 87.5)**	**82.0 (81.2, 83.2)**

Distortion Analysis of CTNNs and Canonical GNNs

Canonical GNNs **incur large stretch**, especially on large protein graphs. CTNN significantly **reduces stretch** across all graphs.

Table 1: Mean ± s.d. of empirical max stretch

	Max Stretch ↓	
	PCBA	**GO MOL**
SMILES	18.82 ± 7.15	—
Primary Sequence	—	165.72 ± 44.51
DGCNN	19.04 ± 5.35	192.56 ± 15.54
RCM	3.76 ± 0.97	33.76 ± 9.11
CTNN (ours)	**2.29 ± 0.27**	**17.56 ± 4.56**

By leveraging canonical tree covers, **CTNNs overcome limitations** of canonical sequence GNNs and offer an **invariant** and **expressive** framework for learning on sparse graphs.

Funding: DE-SC0023112