GraphReach: Position-Aware Graph Neural Network using Reachability Estimations

Sunil Nishad[a,1], Shubhangi Agarwal[a,2], Arnab Bhattacharya[a,3], Sayan Ranu[b,4]

[a]Indian Institute of Technology Kanpur, India , [b]Indian Institute of Technology Delhi, India

Objectives

An *inductive* GNN model that outputs node embeddings that are:

- **Holistic**: Maximum absorption of graph knowledge
- **Meaningful**: Real-world semantics are attended to
- **Robust**: Small alteration to graph does not affect embeddings

Motivation

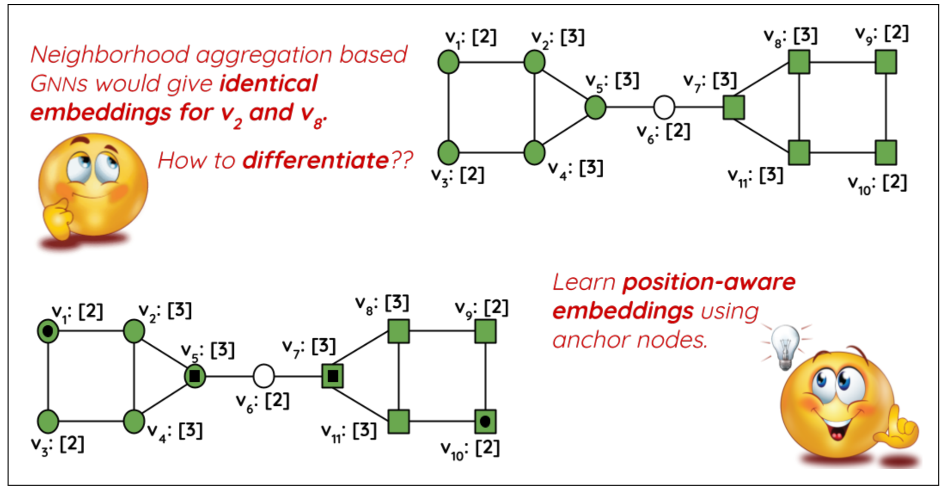

Figure: Traditional vs. position-aware GNNs.

- Most GNNs fail to distinguish nodes with similar neighbourhoods
- Traditional GNNs rely on neighborhood information or are *transductive*
- May be better in exploiting neighborhood feature distribution
- P-GNN [1] encapsulates position/ location of a node using shortest paths

Contributions

GraphReach: A node embedding learning model based on

- ***Reachability estimations*** between anchors and nodes, to aid incorporation of all paths, and
- ***Strategic anchor selection*** to cover all nodes and maximize differentiability of output embeddings, and is, thus,
- ***Inductive***, i.e., can cater to previously unseen nodes and edges, and
- ***Resilient*** to adversarial attacks.

GraphReach

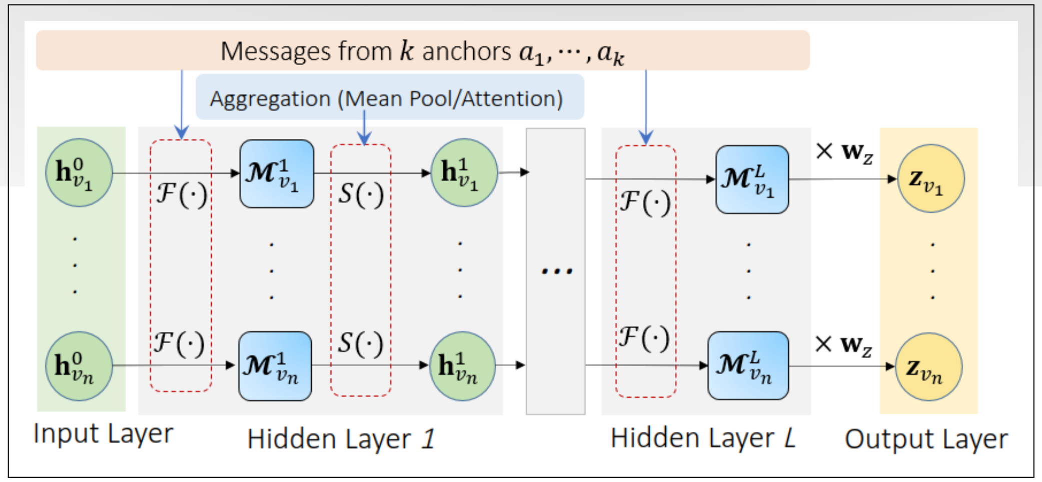

Figure: GraphReach architecture

Anchor Selection and Reachability Estimation

- Conduct multiple **random walks** from each node v.
- Form **bipartite graphs** from these random walks.
- Compute **marginal reachability** of vertices.
- Select k vertices with highest scores as **anchors** $\mathcal{A} = (a_1, \ldots, a_k)$.
- **Reachability estimation** $s(v, a_i)$ is *average* #times node v is visited across all walks from anchor a_i.

Message Computation and Aggregation

- **Message Computation function**: Combine node attributes $\left(h^l_v\right)$ and position information.

$$\mathcal{F}\left(v, a, \mathbf{h}^l_v, \mathbf{h}^l_a\right) = \left(\left(s(v,a) \times \mathbf{h}^l_v\right) \,\|\, \left(s(a,v) \times \mathbf{h}^l_a\right)\right) \quad (1)$$

  – Reachability estimations used in both directions to address asymmetricity.

- **Message Matrix** $\mathcal{M}^l_v$ for node v comprises linearly transformed messages from *all* anchors.

$$\mathcal{M}^l_v = \left(\bigoplus_{a \in \mathcal{A}} \mathcal{F}\left(v, a, \mathbf{h}^l_v, \mathbf{h}^l_a\right)\right) \cdot \mathbf{W}^l_{\mathcal{M}} \quad (2)$$

- **Aggregator**: *Mean-pooling*.

$$\mathcal{S}^M(\mathcal{M}^l_v) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{M}^l_v[i] \quad (3)$$

Output Layer : Transform $\mathcal{M}^L_v$ to output k-dimensional node embeddings.

$$\mathbf{z}_v \leftarrow \sigma(\mathcal{M}^L_v . \mathbf{W}_Z) \;\forall v \quad (4)$$

Experimental Results

- GraphReach is **better** than other architectures in real-world datasets

Models	Email	Protein
GNN*	0.545 ± 0.012	0.528 ± 0.011
P-GNN	0.640 ± 0.029	0.631 ± 0.175
GraphReach	0.949 ± 0.009	0.904 ± 0.003

(a) Pairwise Node Classification (PNC)

Models	Communities	PPI
GNN*	0.692 ± 0.049	0.803 ± 0.005
P-GNN	0.985 ± 0.008	0.808 ± 0.003
GraphReach	0.991 ± 0.003	0.810 ± 0.002

(b) Link Prediction (LP)

Table: ROC AUC (GNN*: Best accuracy obtained among GCN [2], GraphSAGE [3], GIN [4] and GAT [5])

- GraphReach is more robust against graph modifications (**adversarial attacks**)

Task	P-GNN			GraphReach		
	Bf	Af	Δ	Bf	Af	Δ
PNC	0.92	0.82	−0.10	1.00	0.98	−0.02
LP	1.00	0.89	−0.11	1.00	1.00	−0.00

Bf: ROC AUC before collusion
Af: ROC AUC after collusion
Δ: Change in accuracy due to collusion

Table: Robustness to adversarial attacks. (Dataset: Communities)

- GraphReach utilizes **structural information** the best in absence of node features

Task	Dataset	GNN*		P-GNN		GraphReach	
		S+T	S	S+T	S	S+T	S
NC	CoRA	**0.92**	0.52	0.73	0.50	0.84	**0.86**
NC	CiteSeer	**0.82**	0.52	0.73	0.55	0.75	**0.71**

NC: Node Classification (Task)
S: Input is only the graph structure
S+T: Input consists of the graph structure with node attributes

Table: ROC AUC (Traditional GNNs vs Position-aware GNNs).

- **Ablation:** Mean pooling is simpler but almost as good as attention aggregators.
- **Parameters:** Small number of random walks of short/medium lengths were enough.

References

[1] P-GNN: *ICML 2019*; [2] GCN: *ICLR, 2017*; [3] GraphSAGE: *NIPS, 2017*; [4] GIN: *ICLR, 2019*; [5] GAT: *ICLR, 2018*.

For more details

Scan the QR codes.

⟵ **arXiv paper** | **github code** ⟶

arXiv

GitHub

If you have any questions, you can reach out to us!

[1]snishad@cse.iitk.ac.in, [2]sagarwal@cse.iitk.ac.in, [3]arnabb@cse.iitk.ac.in, [4]sayanranu@cse.iitd.ac.in