KAIST CMU | ICML International Conference On Machine Learning

Towards Deep Attention in GNNs:

Problems and Remedies



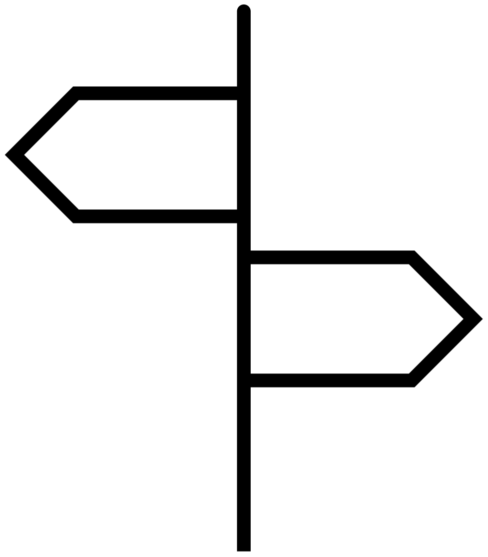

Soo Yong Lee
KAIST

Fanchen Bu
KAIST

Jaemin Yoo
CMU

Kijung Shin
KAIST

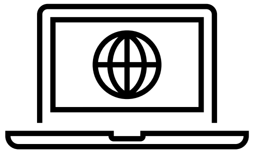
Contents



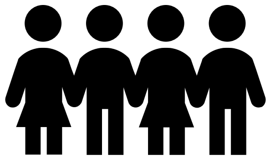
Graphs

- **What are Graphs?**
  - Graphs are relational data
  - Consists of nodes and edges

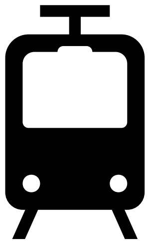
- **Graphs are everywhere!**
  - Can represent a wide range of real-world networks



Web Networks
Node = Webpage
Edge = Hyperlinks



Social Networks
Node = User
Edge = Follow



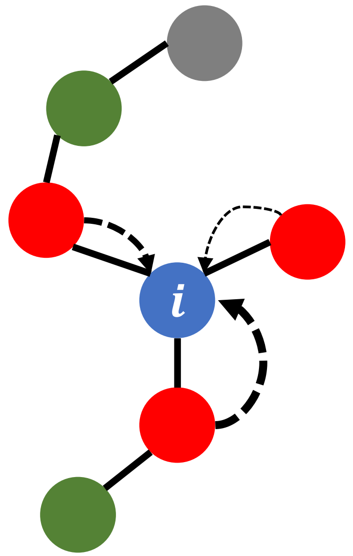
Transportation Networks
Node = Region
Edge = Road Connection

Co-Purchase Networks
Node = Product
Edge = Often Co-Purchased

Graph Neural Networks (GNNs)

- **Graph Neural Networks** (GNNs)
  - Can solve various graph-related tasks
  - Learn graph representation
- **To enhance its expressiveness**:
  - Graph Attention
    - Learns the weight for feature propagation
  - Deep GNN
    - Increases receptive fields
    - Stacks non-linearity

1-Layer Graph Attention:
Receptive Field of Node i

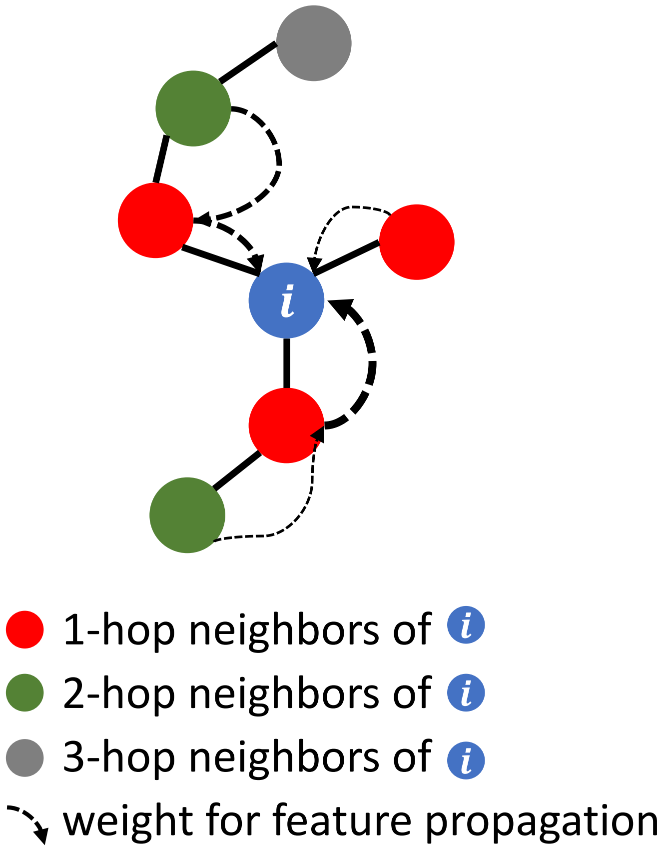


1-hop neighbors of i
2-hop neighbors of i
3-hop neighbors of i
weight for feature propagation

Graph Neural Networks (GNNs)

- **Graph Neural Networks** (GNNs)
  - Can solve various graph-related tasks
  - Learn graph representation
- **To enhance its expressiveness**:
  - Graph Attention
    - Learns the weight for feature propagation
  - Deep GNN
    - Increases receptive fields
    - Stacks non-linearity

2-Layer Graph Attention: Receptive Field of Node i



Goal of the Present Study

?

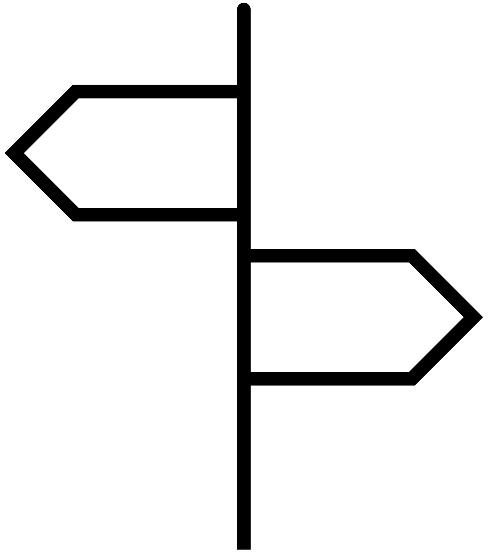
Question of Interest

Can existing graph attention **remain expressive** over deep layers?

How to design an **expressive deep graph attention**?

Can it solve **node classification** problem?

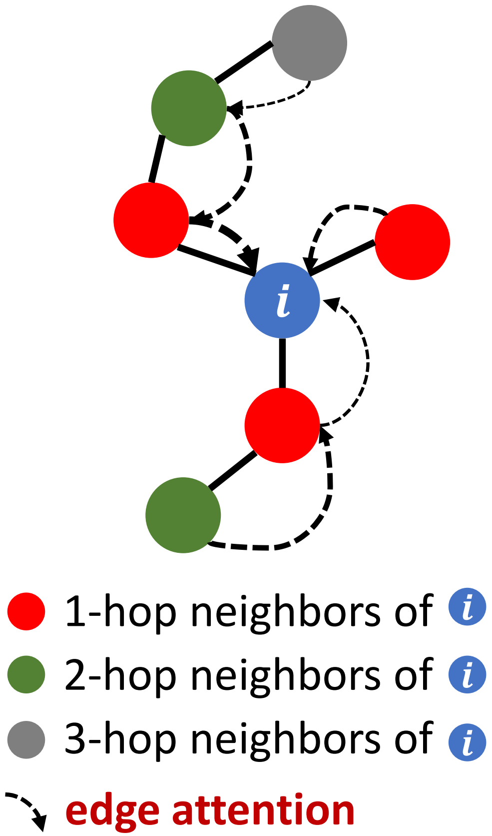
Contents



Graph Attention for GNNs

- **Edge Attention $A^{(k)}$**
  - Intuition: learns importance *within* each hop
  - Models: GAT[1], FAGCN[2]
- **Hop Attention $\Gamma^{(k)}$**
  - Intuition: learns importance *of* each hop
  - Models: GPRGNN[3], DAGNN[4]

Illustration of Hop Attention

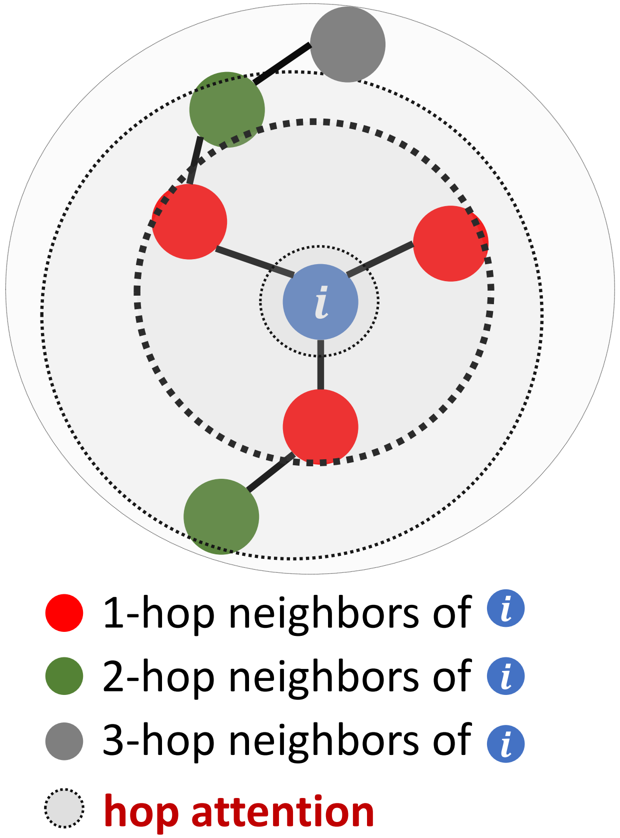


[1] Velickovic,P., Cucurull,G., Casanova,A., Romero,A., Lio,P., and Bengio,Y. Graph attention networks. In ICLR, 2018.
[2] Bo,D., Wang,X., Shi,C., and Shen,H. Beyond low frequency information in graph convolutional networks. In AAAI, 2021.
[3] Liu,M., Gao,H., and Ji,S. Towards deeper graph neural networks. In KDD, 2020.
[4] Chien,E., Peng,J., Li,P., and Milenkovic,O. Adaptive universal generalized pagerank graph neural network. In ICLR, 2021.

Graph Attention for GNNs

- **Edge Attention $A^{(k)}$**
  - Intuition: learns importance *within* each hop
  - Models: GAT[1], FAGCN[2]
- **Hop Attention $\Gamma^{(k)}$**
  - Intuition: learns importance *of* each hop
  - Models: GPRGNN[3], DAGNN[4]

Illustration of Hop Attention



[1] Velickovic,P., Cucurull,G., Casanova,A., Romero,A., Lio,P., and Bengio,Y. Graph attention networks. In ICLR, 2018.
[2] Bo,D., Wang,X., Shi,C., and Shen,H. Beyond low frequency information in graph convolutional networks. In AAAI, 2021.
[3] Liu,M., Gao,H., and Ji,S. Towards deeper graph neural networks. In KDD, 2020.
[4] Chien,E., Peng,J., Li,P., and Milenkovic,O. Adaptive universal generalized pagerank graph neural network. In ICLR, 2021.

Theoretical Limitations to Deep Graph Attention

- **All Graph Attention Models Suffer From Two Problems**
  - P1: Vulnerability of Node Feature Over-Smoothing
    - (Informal) The **attention coefficients become identical** for over-smoothed node features
  - P2: Smooth Cumulative Attention
    - (Informal) **Cumulative attention vectors become identical** for all nodes at very deep layer
- ***Both problems are critically contrary to the goal of attention***

Contents

AERO-GNN : Overview

- We propose <u>A</u>ttentive D<u>e</u>ep P<u>ro</u>pagation GNN (**AERO-GNN**)

- **Model Overview**
  - At every propagation layer k, AERO-GNN learns $\boldsymbol{A}^{(\boldsymbol{k})}$ and $\boldsymbol{\Gamma}^{(\boldsymbol{k})}$

$$H^{(k)} = \begin{cases} \mathrm{MLP}(X), & \text{if } k = 0, \\ \underline{\mathcal{A}^{(k)}} H^{(k-1)}, & \text{if } 1 \leq k \leq k_{max}, \end{cases}$$

$$Z^{(k)} = \sum_{l=0}^{k} \underline{\Gamma^{(l)}} H^{(l)}, \forall 1 \leq k \leq k_{max},$$

$$Z^{*} = \sigma(Z^{(k_{max})}) W^{*},$$

AERO-GNN : Attention Functions

- **Design Question :**
  - How do we design an expressive deep graph attention?

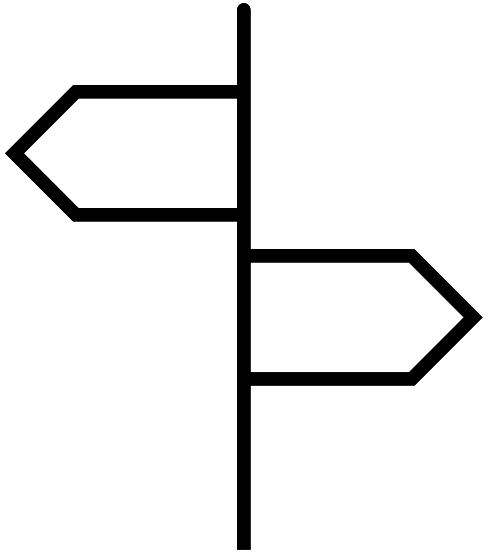
- **Key Properties :**
  - Key 1. Stacking **non-linearity**
  - Key 2. Learn both $\boldsymbol{A}^{(\boldsymbol{k})}$ and $\boldsymbol{\Gamma}^{(\boldsymbol{k})}$ (edge and hop attention)
  - Key 3. Use features from the previous layers $\boldsymbol{Z}$
  - Key 4. Use **negative** attention
  - Key 5. Have **node-adaptive** hop attention $\boldsymbol{\Gamma}^{(\boldsymbol{k})}$

AERO-GNN : Attention Functions

- **Bottom Line :**
  - ***Attention functions of AERO-GNN is flexible and expressive!***
  - ***They allow AERO-GNN to mitigate problems of deep graph attention.***
    - Vulnerability to Over-Smoothing & Smooth Cumulative Attention

Properties of Attention Functions					
	Stacking Non-Linearity	Edge & Hop	Z as Input	Negative Attention	Node-Adaptive
GATv2	O	X	X	X	X
FAGCN	O	X	O	O	X
GPRGNN	X	X	X	O	X
DAGNN	X	X	X	X	O
AERO-GNN	O	O	O	O	O

Contents



Performance (Mean ± Std, 100 trials)

- **AERO-GNN achieves the best overall performance (See high A.R.)!**

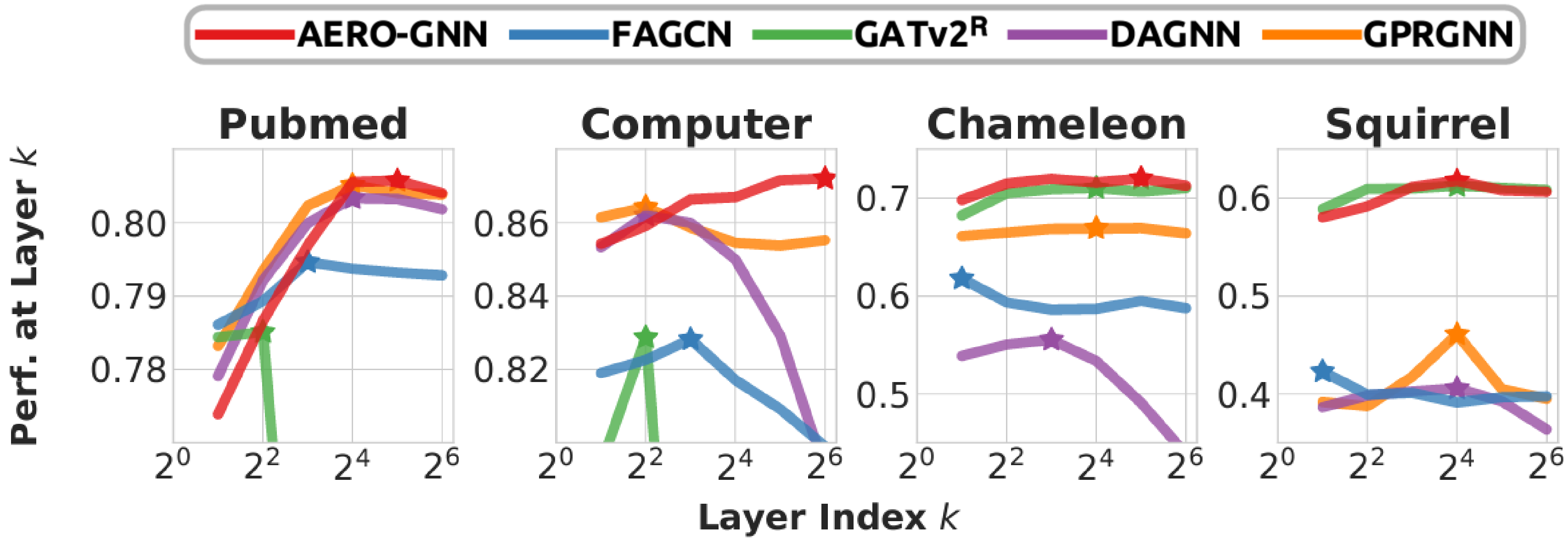
Table 3: Node Classification Performance on Real-World Graphs

Dataset	Chameleon	Squirrel	Actor	Texas	Cornell	Wisconsin	Computer	Photo	Wiki-CS	Pubmed	Citeseer	Cora	A.R.
Homophily	0.04	0.03	0.01	0.00	0.02	0.05	0.70	0.77	0.57	0.66	0.63	0.77	
GCN	67.97 ± 2.5	53.33 ± 1.3	30.57 ± 0.7	65.65 ± 4.8	58.41 ± 3.3	62.02 ± 5.9	81.27 ± 1.4	90.24 ± 1.3	79.08 ± 0.5	79.54 ± 0.4	72.50 ± 0.5	83.15 ± 0.5	9.1
APPNP	53.04 ± 2.2	40.37 ± 1.5	35.49 ± 1.0	79.89 ± 4.2	80.16 ± 5.9	84.24 ± 4.6	81.27 ± 1.4	91.12 ± 1.2	79.05 ± 0.5	79.90 ± 0.3	73.06 ± 0.3	83.60 ± 1.3	7.8
GCN-II	60.38 ± 1.9	48.76 ± 2.4	35.77 ± 1.0	78.59 ± 6.6	78.84 ± 6.6	83.20 ± 4.7	84.24 ± 1.2	91.81 ± 0.9	79.28 ± 0.6	80.14 ± 0.6	**73.20 ± 0.8**	**85.33 ± 0.5**	5.5
A-DGN	69.63 ± 2.0	57.77 ± 1.9	36.41 ± 1.0	82.22 ± 4.8	**83.14 ± 6.7**	**85.84 ± 4.0**	83.70 ± 1.5	90.53 ± 1.3	79.11 ± 0.6	78.68 ± 0.6	70.16 ± 0.9	79.84 ± 0.9	6.4
GAT	68.01 ± 2.5	54.49 ± 1.7	30.36 ± 0.9	60.46 ± 6.2	58.22 ± 3.7	63.59 ± 6.1	84.46 ± 1.3	89.88 ± 1.1	79.44 ± 0.5	78.94 ± 0.4	71.89 ± 0.6	83.78 ± 0.5	8.5
GATv2	69.06 ± 2.2	57.67 ± 2.4	30.27 ± 0.8	60.32 ± 7.0	58.35 ± 3.8	61.94 ± 4.7	84.19 ± 1.2	89.87 ± 1.2	79.64 ± 0.5	79.12 ± 0.3	71.15 ± 1.2	83.88 ± 0.6	8.9
GATv2^R	**70.88 ± 1.9**	**61.23 ± 1.5**	33.73 ± 0.9	60.68 ± 6.6	57.32 ± 4.5	60.61 ± 5.1	81.73 ± 2.2	88.71 ± 1.7	79.75 ± 0.6	78.28 ± 0.4	71.00 ± 0.8	82.42 ± 0.6	9.3
GT	69.34 ± 1.2	55.04 ± 1.9	36.29 ± 1.0	**84.08 ± 5.6**	80.00 ± 4.9	**84.80 ± 4.3**	84.38 ± 1.3	91.28 ± 1.1	**79.93 ± 0.5**	79.04 ± 0.5	70.16 ± 0.8	82.09 ± 0.7	5.6
FAGCN	60.98 ± 2.3	42.20 ± 1.8	35.67 ± 0.9	77.00 ± 7.7	78.32 ± 6.3	82.41 ± 3.8	82.79 ± 2.7	91.99 ± 1.0	79.27 ± 0.6	79.19 ± 0.4	71.55 ± 0.8	83.88 ± 0.5	7.5
DMP	63.79 ± 4.1	34.19 ± 7.6	28.30 ± 2.7	66.08 ± 7.0	56.41 ± 5.5	62.73 ± 4.5	70.58 ± 11.3	82.63 ± 4.1	56.41 ± 7.8	70.07 ± 4.1	59.12 ± 4.4	75.05 ± 3.8	12.8
MixHop	60.30 ± 2.1	41.05 ± 2.0	**36.48 ± 1.2**	77.73 ± 7.3	75.95 ± 5.7	82.12 ± 4.5	79.52 ± 2.1	89.45 ± 1.5	78.59 ± 0.7	80.10 ± 0.4	71.42 ± 0.9	81.61 ± 0.8	9.3
GPRGNN	66.92 ± 1.7	46.32 ± 1.5	35.58 ± 0.9	81.51 ± 6.6	76.86 ± 7.1	84.06 ± 5.2	85.82 ± 0.9	**92.41 ± 0.7**	79.67 ± 0.5	80.28 ± 0.4	71.59 ± 0.8	84.20 ± 0.5	**5.2**
DAGNN	54.99 ± 2.0	40.03 ± 1.4	33.69 ± 1.0	61.35 ± 6.1	63.89 ± 7.0	62.27 ± 4.2	**85.83 ± 0.8**	92.30 ± 0.7	79.31 ± 0.6	**80.44 ± 0.5**	73.16 ± 0.6	**84.43 ± 0.5**	7.2
AERO-GNN	**71.58 ± 2.4**	**61.76 ± 2.4**	**36.57 ± 1.1**	**84.35 ± 5.2**	**81.24 ± 6.8**	**84.80 ± 3.3**	**86.69 ± 1.4**	**92.50 ± 0.7**	**79.95 ± 0.5**	**80.59 ± 0.5**	**73.20 ± 0.6**	83.90 ± 0.5	**1.4**

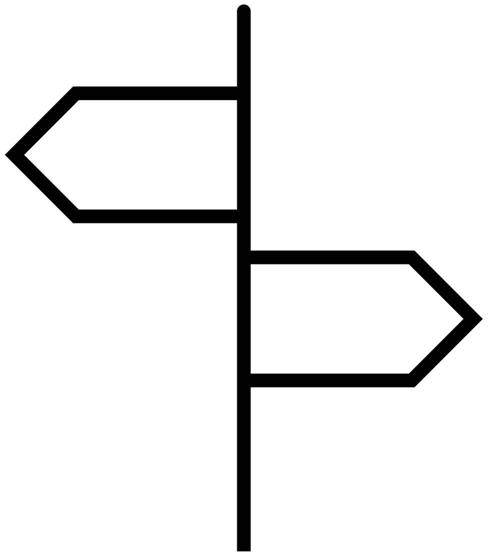
- In each column, ■ indicates ranking the first, and ■ indicates ranking the second. A.R. denotes average ranking.

Performance Over Layers

- **AERO-GNN has**
  - Highest best performance across model depth (see ★ in the Figure)
  - Better performance over layers k (see **trend** in the Figure)



Contents



Summary

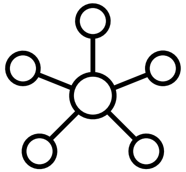
Problem

Two Limitations to
Deep Graph Attention

Solution: AERO-GNN

Theoretically and Empirically
Mitigates the Problems

Implications for Graph Learning



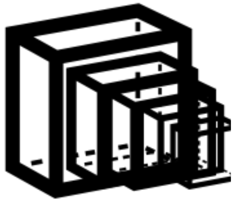
Attention-Based GNNs

A larger focus has been placed on designing **a more expressive layer**

- with new designs

- with new loss terms

- with more features



Deep GNNs

Making deeper GNNs have been an important **setback to GNN research**

- over-smoothing

- over-squashing

- over-correlation

We Bridge the Two

The two are complementary

Thank You