Graph Pointer Neural Networks

Tianmeng Yang[1,2*†], Yujing Wang[1,2†], Zhihan Yue[1], Yaming Yang[2], Yunhai Tong[1], Jing Bai[2]

[1]School of Electronics Engineering and Computer Science, Peking University
[2]Microsoft Research Asia
{youngtimmy, zhihan.yue, yhtong}@pku.edu.cn, {yujwang, yayaming, jbai}@microsoft.com

Code : None

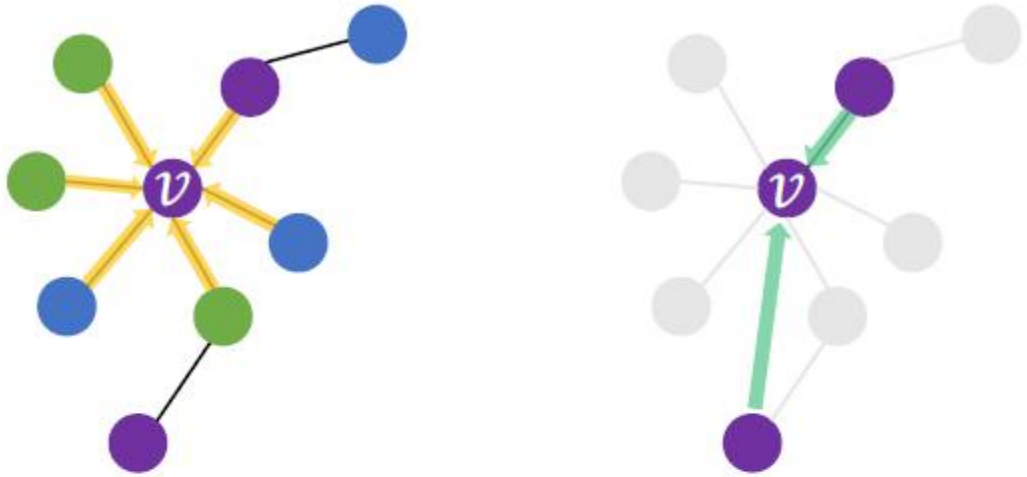
AAAI_2022

1.Introduction

2.Method

3.Experiments

Introduction



(a) Local aggregation. (b) Non-local aggregation.

Figure 1: Visual illustration of the local aggregation of existing GNNs and non-local aggregation in GPNN in one propagation step. The colors of nodes represent their labels, while grey means ignored. Traditional GNNs aggregate all nodes in the local neighborhood including noises, while GPNN selectively filters the irrelevant nodes and captures non-local features.

Method

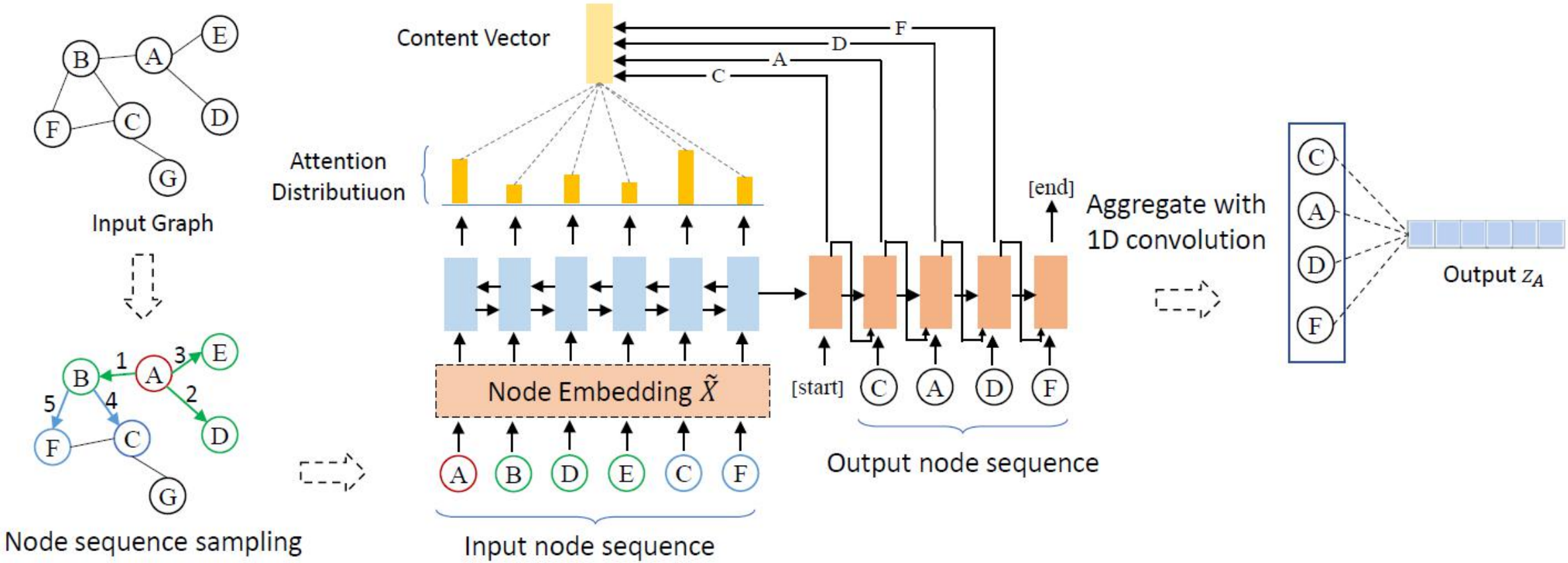


Figure 2: An illustration of the graph pointer generator layer. With the central node A and a sampling depth k=2, the neighbors within two hops are assembled after the node sequence sampling. The pointer network then selects the most relevant nodes to A, followed by a 1D-convolution layer to extract high-level and non-local features at the end.

Method

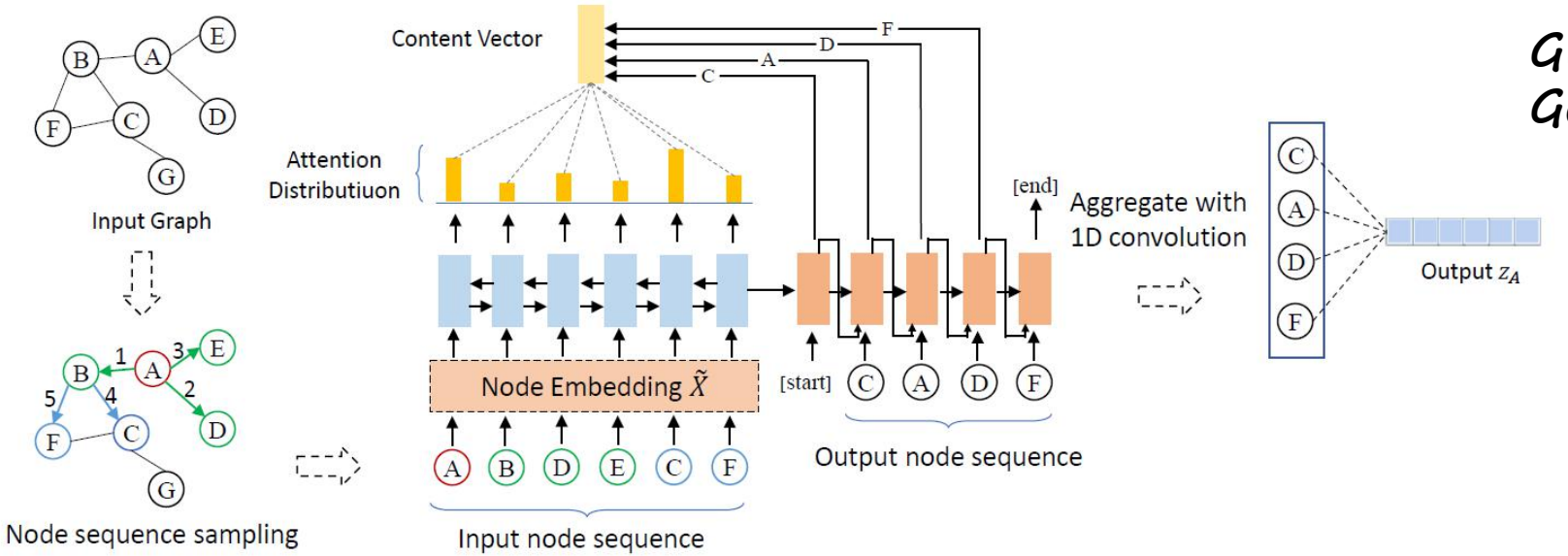


Figure 2: An illustration of the graph pointer generator layer. With the central node A and a sampling depth k=2, the neighbors within two hops are assembled after the node sequence sampling. The pointer network then selects the most relevant nodes to A, followed by a 1D-convolution layer to extract high-level and non-local features at the end.

Multi-hop node sequence sampling

Graph Pointer Generator

Node embedding

$$\hat{X} = GCN(X) \qquad \in \mathbb{R}^{N \times d} \tag{9}$$

Pointer

$$e_i = tanh(W[e_{i-1}, \hat{x}_i]) \tag{10}$$

e_0 is initialed to 0

entire sequence of L nodes

$E = \{e_1, e_2, \ldots, e_L\}$

$$d_i = tanh(W[d_{i-1}, \hat{x}_{c_{i-1}}]) \tag{11}$$

d_0 is the output hidden state e_L

where d_0 is the output hidden state e_L from the encoder, c_{i-1} is the index of selected node at time step $i-1$, c_0 is a signal of $[start]$.

$$u_j^i = v^T tanh(W_1 e_j + W_2 d_i) \qquad j \in (1, 2, \ldots, L) \tag{12}$$

$$p(c_i | c_1, c_2, \ldots, c_{i-1}, s) = softmax(u_i) \tag{13}$$

$$o = \{\hat{x}_{c_1}, \hat{x}_{c_2}, \ldots, \hat{x}_{c_m}\} \qquad \in \mathbb{R}^{m \times d} \tag{14}$$

Method

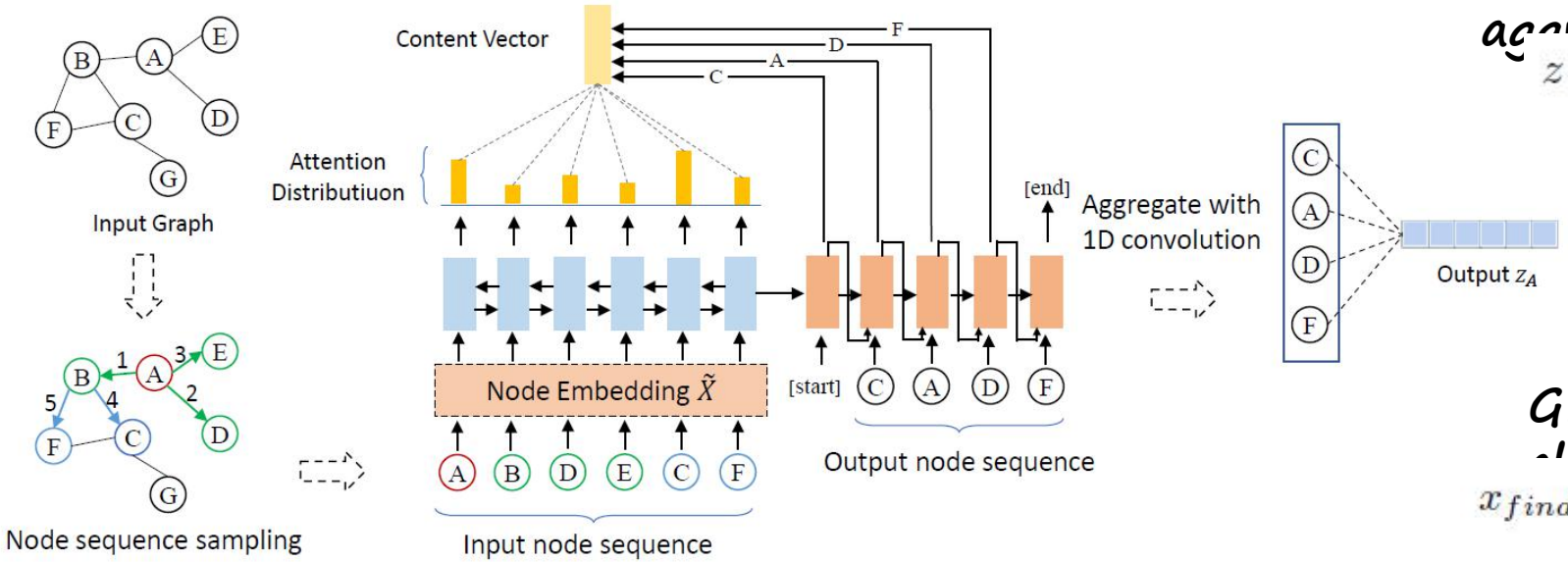


Figure 2: An illustration of the graph pointer generator layer. With the central node A and a sampling depth k=2, the neighbors within two hops are assembled after the node sequence sampling. The pointer network then selects the most relevant nodes to A, followed by a 1D-convolution layer to extract high-level and non-local features at the end.

Non-local aggregation

$$z = \text{aggr}((\text{conv}(o)) \qquad \in \mathbb{R}^{1\times d'} \tag{15}$$

1D-convolution layer

aggr(·) function can be a pool operation

GPNN for node classification

$$x_{final} = \text{concat}(xW, \hat{x}, z) \qquad \in \mathbb{R}^{1\times(d_0+d+d')} \tag{16}$$

$$y_{pred} = softmax(FFN(x_{final})) \tag{17}$$

$$\mathcal{L} = \sum_{i=1}^{N_l} y_i \log y_{pred\ i} \tag{18}$$

Experiments

Datasets	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin
#Nodes	2277	5201	7600	183	183	251
#Edges	36101	217073	33544	295	309	499
#Features	2325	2089	931	1703	1703	1703
#Classes	5	5	5	5	5	5
#Homophily ratio $H(\mathcal{G})$	0.25	0.22	0.24	0.11	0.06	0.16

Table 1: Statistics and properties of benchmark datasets with heterophily.

Experiments

Methods	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin	*Average*
MLP	47.36±2.37	29.82±1.99	35.79±1.09	82.16±7.45	81.08±3.82	85.49±4.99	60.28
GCN (Kipf and Welling 2016)	65.92±2.58	49.78±2.06	30.16±1.27	58.91±8.33	59.73±3.24	58.82±6.06	53.89
GAT (Veličković et al. 2017)	65.32±2.00	46.79±2.08	29.74±1.46	56.76±5.70	59.45±6.37	57.06±7.07	52.52
GraphSage (Ying et al. 2018)	58.73±1.68	41.61±0.74	34.23±0.99	75.95±5.01	82.43±6.14	81.18±5.56	62.36
MixHop (Abu-El-Haija et al. 2019)	60.50±2.53	43.80±1.48	32.22±2:34	73.51±6.34	77.84±7.73	75.88±4.90	60.58
Geom-GCN (Pei et al. 2020)	60.90	38.14	31.63	60.81	67.57	64.12	53.86
H2GCN (Zhu et al. 2020)	59.39±1.98	37.90±2.02	35.86±1.03	82.16±4.80	84.86±6.77	86.67±4.69	64.47
Node2Seq (Yuan and Ji 2021)	69.4±1.6	58.8±1.4	31.4±1.0	58.7±6.8	63.7±6.1	60.3±7.0	57.05
GPNN (ours)	**71.27**±1.88	**59.11**±1.13	**37.08**±1.41	**85.14**±6.00	**85.23**±6.40	**86.86**±2.62	**70.78**

Table 2: Mean accuracy±stdev over different data splits on the six real-world heterophilic graph datasets. The best result is highlighted.

Experiments

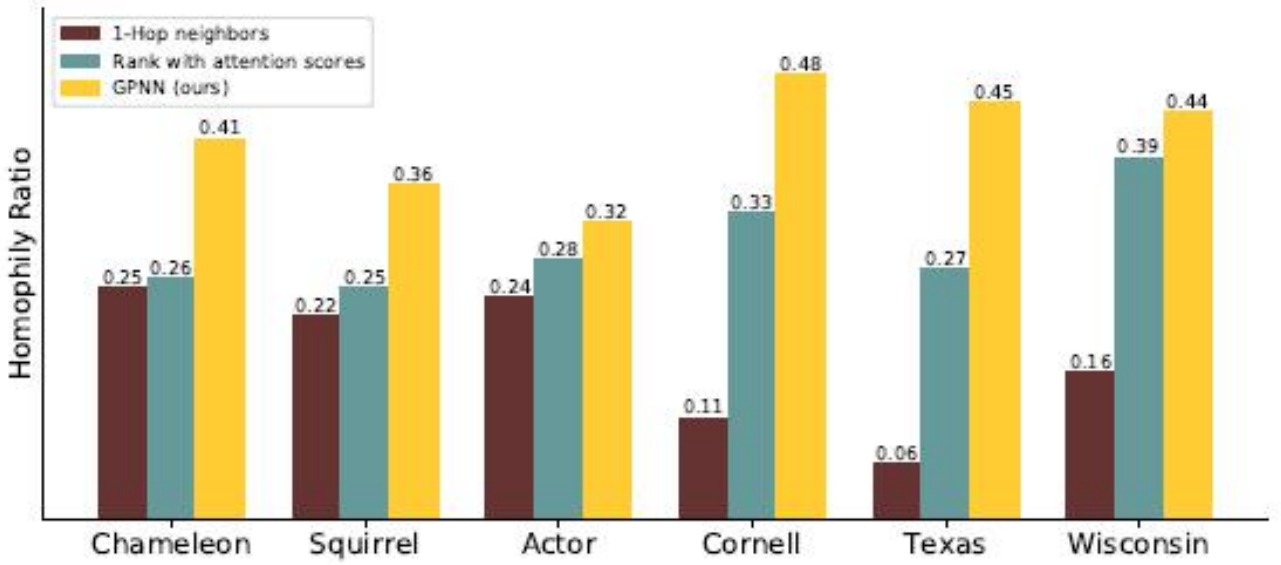
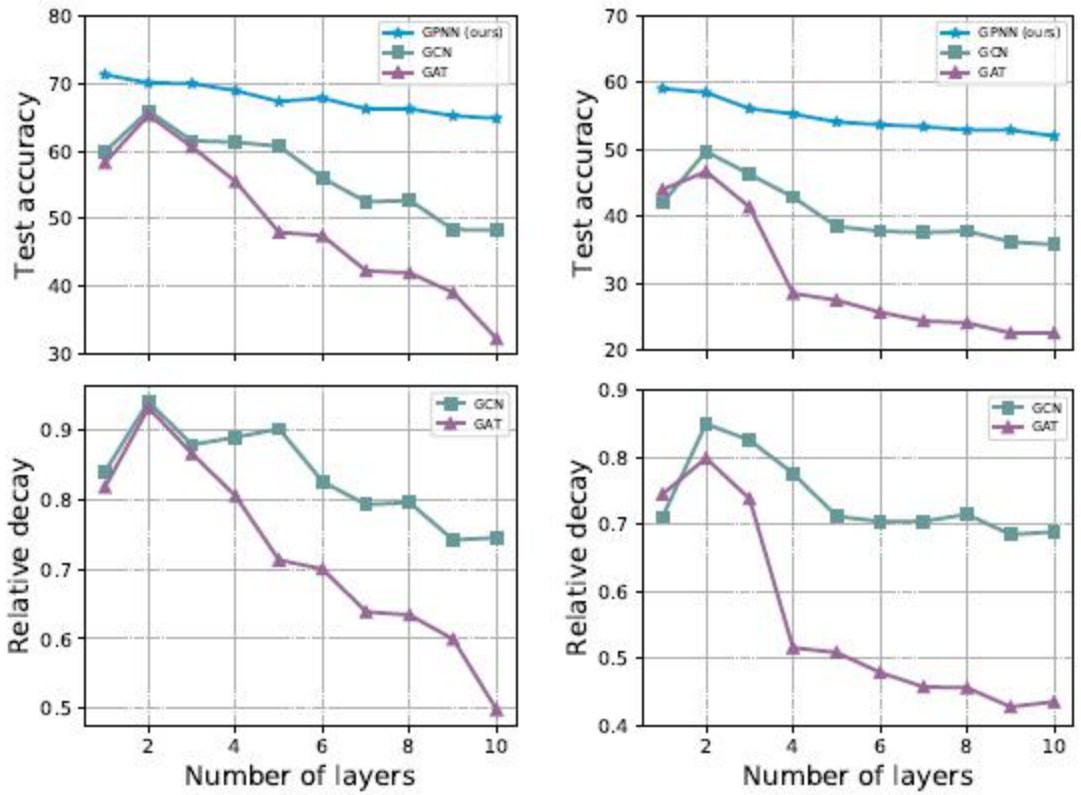


Figure 3: Comparision of homophily ratios between 1-hop neighbors, nodes ranked with attention scores and nodes selected with pointer network in GPNN.

Experiments



(a) Chameleon (b) Squirrel

Figure 4: Over-Smoothing on Chameleon and Squirrel datasets. The top figures show the test accuracy while the bottom ones are relative decays of GCN and GAT compared to GPNN.

Thank you!