UNIVERSITY OF CAMBRIDGE

Understanding Attention Patterns in Graph Neural Networks

Batu El
be301@cam.ac.uk

Deepro Choudhury
dc755@cam.ac.uk

Supervisor: Chaitanya Joshi

Motivation & Research Question

Petar Veličković
@PetarV_93

You are right that, provided sufficient scale, **and** assuming your test data is in the same distribution as your training data, graph structures can be learnt from scratch. When it comes to smaller scale, or OOD, I strongly disagree that Transformers as they are now can work.

1:03 PM · Apr 26, 2024 · **520** Views

Petar Veličković
@PetarV_93

FWIW, I fully agree with you that the graph should be malleable (I teach two full hours at my Master's course discussing rewiring alone!), but this does not mean that the best way of making it malleable is a 'carte blanche' where the GNN decides everything with zero priors.

1:11 PM · Apr 26, 2024 · **796** Views

<u>Question 1:</u> When attention is not restricted to the neighbors, does the learned attention pattern recover the underlying graph structure?

<u>Question 2:</u> If the learned attention pattern does not recover the underlying graph structure, what does the attention mechanism learn?

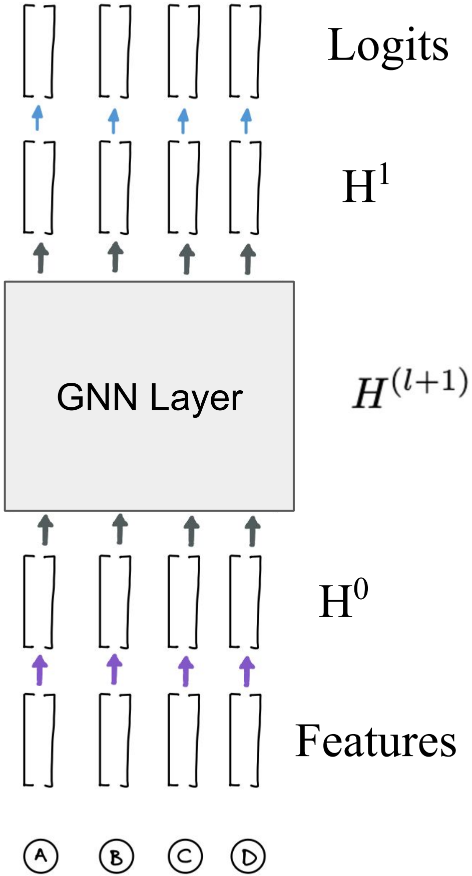
Models

fixed: GNN attention is a constant function of the graph structure.

1. Sparse and Constant: GCN Attention
   GCN (Kipf & Welling, 2016)
2. Sparse and Learnt: Sparse Transformer Attention
   GAT (Veličković et. al, 2017)
3. Dense, Learnt, and Biased: Dense Transformer Attention with **Spatial Bias**
   Graphormer (Ying et. al, 2021)
4. Dense and Learnt: Dense Transformer Attention with **Positional Encoding**
   Graph Transformer Networks (Yun et. al, 2020)

carte blanche: GNN learns attention from scratch.

Models



$$H^{(l+1)} = H^{(l)} + \mathrm{ReLU}(\mathbb{A}V^{(l)}) \text{ , where } V^{(l)} = H^{(l)}W^{(l)}$$

1 Layer 1 Head Example

Node Classification Datasets

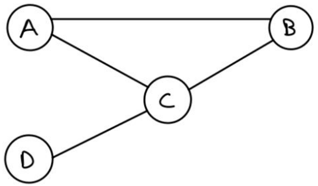
- What is a good measure of homophily? (Platonov et al. 2023)
- Calculated and found errors in homophily calculations are propagated throughout papers.
- Adjusted Homophily:

$$h_{adj} = \frac{h_{edge} - \sum_{k=1}^{C} \bar{p}(k)^2}{1 - \sum_{k=1}^{C} \bar{p}(k)^2}.$$

Metric	Cora	Citeseer	Chameleon	Squirrel	Cornell	Texas	Wisconsin
Node Homophily	82.5	70.6	10.4	8.9	10.6	6.5	17.2
Edge Homophily	81.0	73.6	23.5	22.4	13.1	10.8	19.6
Adjusted Homophily	77.1	67.1	3.3	0.7	-21.1	-25.9	-15.2
Number of Nodes	2708	3327	2277	5201	183	183	251
Number of Edges	10556	9104	36101	217073	298	325	515

Table 1: Summary of Homophily and Network Structure in Various Datasets.

A Simple Example



ADJACENCY

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & 1 & 1 & 0 \\ B & 1 & 1 & 1 & 0 \\ C & 1 & 1 & 1 & 1 \\ D & 0 & 0 & 1 & 1 \end{array}$$

SHORTEST PATHS

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 2 \\ B & 1 & 0 & 1 & 2 \\ C & 1 & 1 & 0 & 1 \\ D & 2 & 2 & 1 & 0 \end{array}$$

A GCN

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & a_{11} & a_{12} & a_{13} & 0 \\ B & a_{21} & a_{22} & a_{23} & 0 \\ C & a_{31} & a_{32} & a_{33} & a_{34} \\ D & 0 & 0 & a_{43} & a_{44} \end{array}$$

A SPARSE TRANSFORMER

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & a_{11} & a_{12} & a_{13} & 0 \\ B & a_{21} & a_{22} & a_{23} & 0 \\ C & a_{31} & a_{32} & a_{33} & a_{34} \\ D & 0 & 0 & a_{43} & a_{44} \end{array}$$

A DENSE TRANSFORMER w BIAS

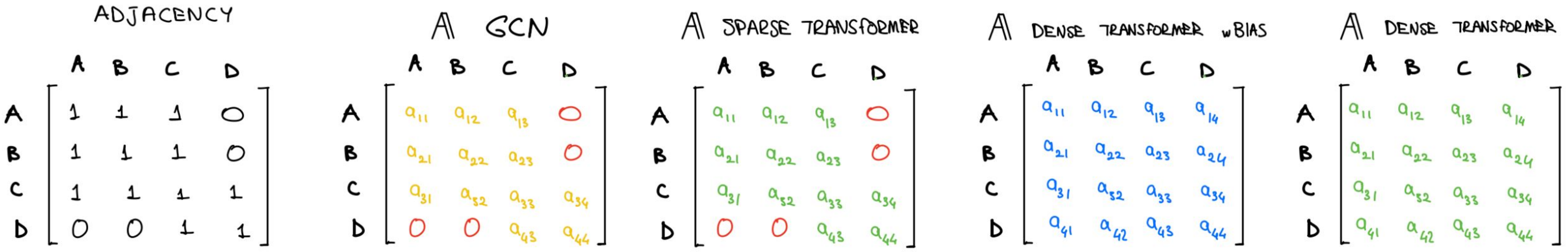
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & a_{11} & a_{12} & a_{13} & a_{14} \\ B & a_{21} & a_{22} & a_{23} & a_{24} \\ C & a_{31} & a_{32} & a_{33} & a_{34} \\ D & a_{41} & a_{42} & a_{43} & a_{44} \end{array}$$

A DENSE TRANSFORMER

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & a_{11} & a_{12} & a_{13} & a_{14} \\ B & a_{21} & a_{22} & a_{23} & a_{24} \\ C & a_{31} & a_{32} & a_{33} & a_{34} \\ D & a_{41} & a_{42} & a_{43} & a_{44} \end{array}$$

Masked   Fixed   Learned from Scratch   Learned and Biased

Does the attention matrix recover the graph structure?



	1L1H	
	DTwB	DenseT
Cora	46.75	0.13
Citeseer	28.24	0.22
Chameleon	51.01	0.76
Squirrel	84.11	0.07
Cornell	57.44	0.21
Texas	58.37	0.61
Wisconsin	58.06	1.48

Treating the Problem as Binary Classification:
We threshold the attention matrix so that the number of edges recovered to matches the number of edges in the adjacency matrix.

The Table demonstrates the F-1 Scores.

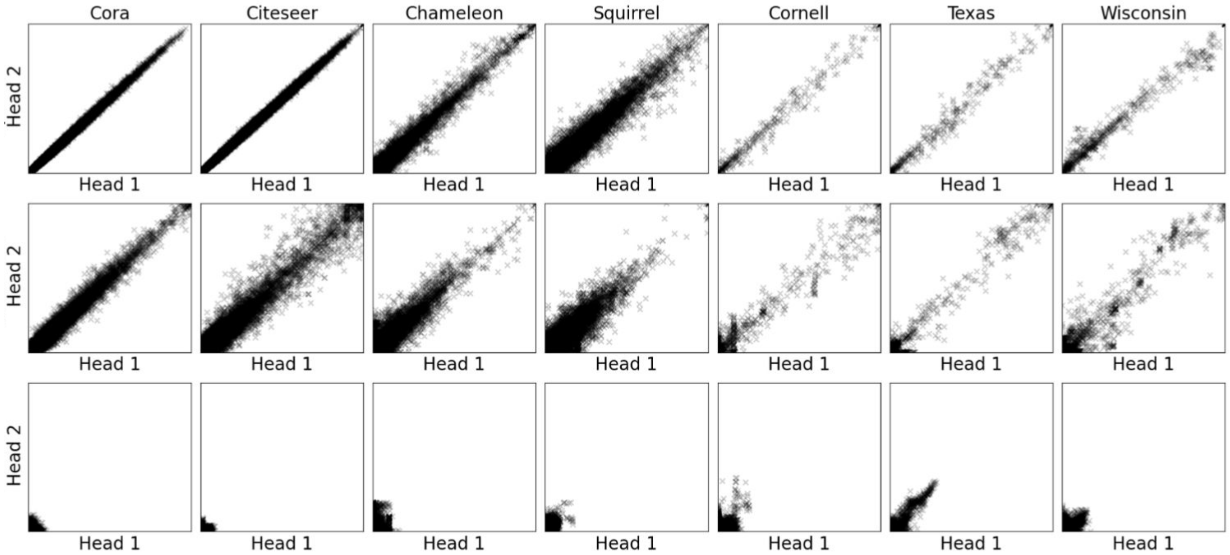
Combining Attention Patterns: *Attention Patterns in Different Heads*

To combine attention patterns, we ***average across heads:*** $\mathbb{A}_{\text{Agg.}} = \dfrac{\mathbb{A}_{H1} + \mathbb{A}_{H2}}{2}$

Sparse Transformer Attention

Dense Transformer Attention with Spatial Bias

Dense Transformer Attention



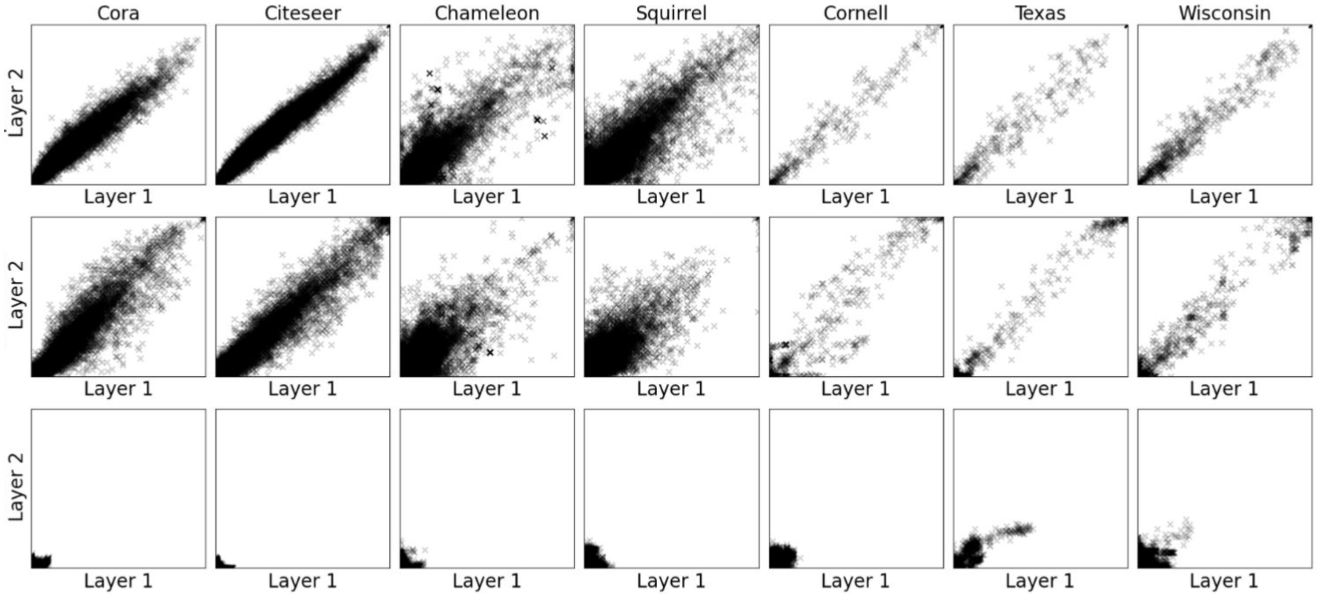
Combining Attention Patterns: *Attention Patterns in Different Layers*

To combine attention patterns, we ***matrix multiply across layers:*** $\mathbb{A}_{\text{Agg.}} = \mathbb{A}_{L2}\mathbb{A}_{L1}$

Sparse Transformer Attention

Dense Transformer Attention with Spatial Bias

Dense Transformer Attention



Capturing the Information Flow Across Layers

Problem:

ATTENTION LAYER 1 | ATTENTION LAYER 2 | AGGREGATED ATTENTION.

$$\begin{array}{c} A \\ B \\ C \\ D \end{array} \begin{bmatrix} aa & ab & ac & ad \\ ba & bb & bc & bd \\ ca & cb & cc & cd \\ da & db & dc & dd \end{bmatrix} \quad \begin{bmatrix} aa & ab & ac & ad \\ ba & bb & bc & bd \\ ca & cb & cc & cd \\ da & db & dc & dd \end{bmatrix} \quad \begin{bmatrix} aa & ab & ac & ad \\ ba & bb & bc & bd \\ ca & cb & cc & cd \\ da & db & dc & dd \end{bmatrix} \begin{array}{c} A \\ B \\ C \\ D \end{array}$$

Our Solution:

$$\mathbb{A}_{\text{Agg.}} = \mathbb{A}_{L2}\mathbb{A}_{L1}$$

ATTENTION LAYER 2 | ATTENTION LAYER 1 | AGGREGATED ATTENTION.

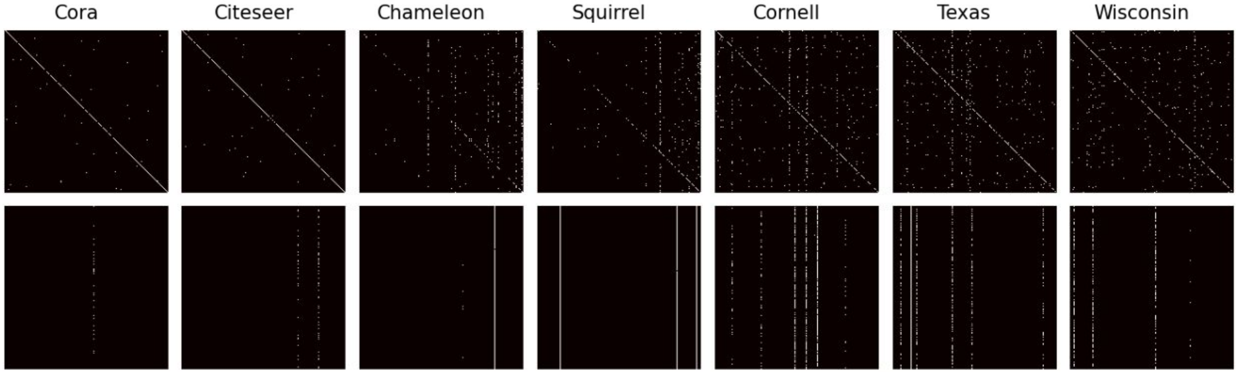
$$\begin{bmatrix} aa & ab & ac & ad \\ ba & bb & bc & bd \\ ca & cb & cc & cd \\ da & db & dc & dd \end{bmatrix} \quad \begin{array}{c} A \\ B \\ C \\ D \end{array} \begin{bmatrix} aa & ab & ac & ad \\ ba & bb & bc & bd \\ ca & cb & cc & cd \\ da & db & dc & dd \end{bmatrix} = \begin{bmatrix} aa\,A + ab\,B + ac\,C + ad\,D \\ ba\,A + bb\,B + bc\,C + bd\,D \\ ca\,A + cb\,B + cc\,C + cd\,D \\ da\,A + db\,B + dc\,C + dd\,D \end{bmatrix} \begin{array}{c} A \\ B \\ C \\ D \end{array}$$

Analyzing Aggregated Attentions

	1L1H		1L2H		2L1H		2L2H	
	DTwB	DenseT	DTwB	DenseT	DTwB	DenseT	DTwB	DenseT
Cora	46.75	0.13	61.58	0.38	36.58	0.21	37.94	0.32
Citeseer	28.24	0.22	41.51	0.69	33.70	0.23	39.02	0.13
Chameleon	51.01	0.76	58.63	0.72	28.32	1.24	33.98	0.39
Squirrel	84.11	0.07	86.31	0.07	49.22	0.43	50.10	0.51
Cornell	57.44	0.21	61.38	0.85	43.51	1.25	48.95	1.04
Texas	58.37	0.61	58.31	1.22	44.83	3.69	47.14	2.95
Wisconsin	58.06	1.48	58.78	0.95	41.94	1.06	41.12	1.47

Treating the Problem as Binary Classification:
We threshold Attention matrix so that the number of edges recovered to matches the number of edges in the adjacency matrix. *F-1 Scores are displayed on the Table.*

The Algorithms Learned by the Models



Dense Transformer Attention with Spatial Bias

Dense Transformer Attention

Model	Cora	Citeseer	Chameleon	Squirrel	Cornell	Texas	Wisconsin
2L2H							
SparseT	**0.87** (0.01)	**0.74** (0.01)	0.61 (0.02)	**0.44** (0.01)	0.61 (0.07)	**0.77** (0.07)	0.75 (0.02)
DenseTwB	0.86 (0.01)	**0.74** (0.01)	**0.63** (0.02)	**0.44** (0.01)	0.62 (0.09)	0.75 (0.07)	0.78 (0.04)
DenseT	0.69 (0.02)	0.69 (0.01)	0.50 (0.02)	0.36 (0.01)	**0.70** (0.06)	**0.77** (0.08)	**0.81** (0.03)

Main Contributions

Theoretical Contributions

1. **Attention and Message Passing (Batu):** A framework to understand message passing operations in GNNs as a generalized attention mechanism.
2. **Combining Attention Patterns (Joint):** A mathematically principled method to combine the attention across multiple heads and layers that enables our analysis of larger models.

Empirical Analysis

1. **Recovering Graph Structure (Joint):** Dense Transformer attention mechanism does not necessarily recover the underlying graph structure when it is not explicitly biased to focus on local neighbours while still demonstrating competitive performance compared to our other models in certain tasks
2. **Insights into Algorithm Learned by the Model (Joint):** Our Dense Transformer with Spatial Bias and Dense Transformer models achieve similar performance on small heterophilous graphs, such as Texas (2). Remarkably, our analysis of the models' attention patterns suggest that they do so by implementing completely different algorithms.
3. **A Toolkit for Attention Analysis (Deepro)**

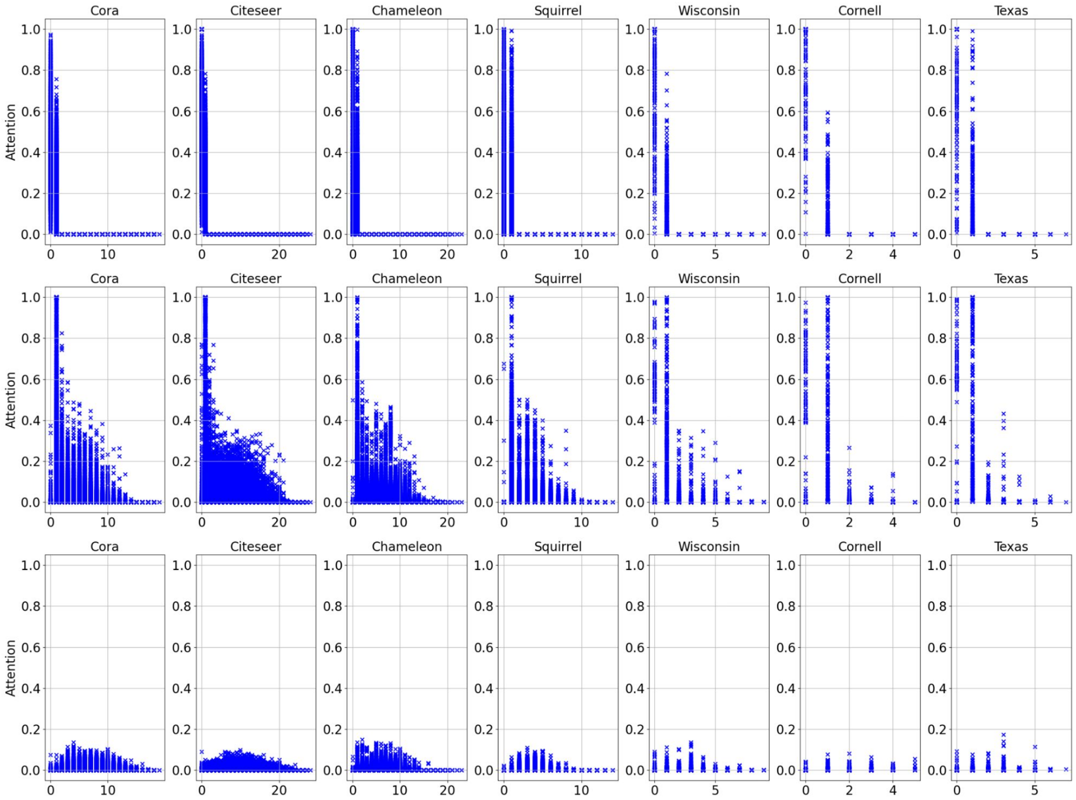
Appendix

Attention to N-hop neighborhood

Sparse Transformer Attention

Dense Transformer Attention
with Spatial Bias

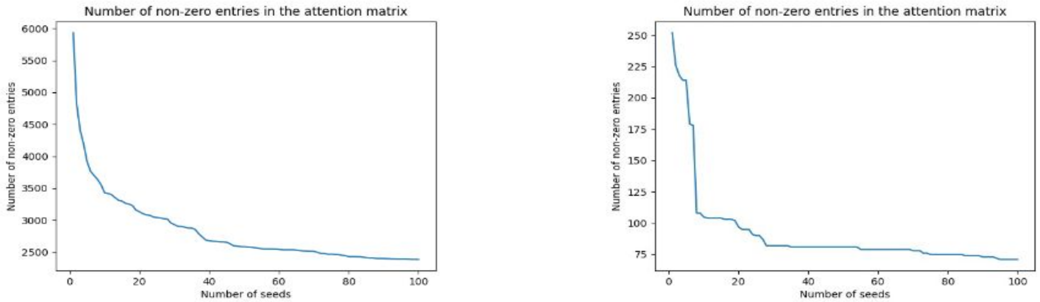
Dense Transformer Attention
(with positional encoding)



Randomness in learned attention

Attention is Random

Threshold the learned attention matrices and perform a logical AND operation across 100 different seeds. The number of significant attention values rapidly decreases towards a core component.



(a) Trained on Cora. (b) Trained on Texas.

Figure 6: Number of non-zero elements in the pseudo-adjacency matrix. We train 100 1 layer 1 head dense transformer models using different random seeds. When we take the logical AND of the pseudo-adjacency matrices from 100 runs we recover a core component of the pseudo-adjacency matrix that is learned independent of the random seed in all of our runs.