Sparse Self-Attention on Graph Convolutional Networks for Low Homophily Graphs

Anton Chen
contact@antonchen.ca

Xiang Xi Chen
xiangxi.chen.ca@gmail.com

Abstract

Graph convolutional networks (GCNs) are the dominant architectures for representation learning on graph-structured data. Despite this, modern GCNs face difficulty learning *low-homophily* graphs, an important class of graphs in application. GCNs with self-attention (GCN-SA) are a recent breakthrough offering exceptional performance on low-homophily graphs: SOTA classification accuracy on 8 graph datasets of varying homophily, outperforming other competitive models by a wide margin on low-homophily graphs. Responsible for this improvement is self-attention (SA), allowing the model to capture long-range dependencies between nodes. However, GCN-SA is limited by poor runtime scaling of its SA mechanisms, restricting the model to small graphs. Advancements in the sister field of graph transformers (GTs) leverage *sparse* SA mechanisms on graphs (e.g. EXPHORMER) to address this poor scaling. In this paper we bridge the gap between advancements in sparse SA with the exceptional performance of GCN-SA on low homophily graphs. We propose that sparse SA mechanisms replace GCN-SA's attention mechanisms to not only reduce runtime complexity from quadratic to linear, but to also maintain similar classification accuracy. We perform experiments on 8 graph datasets of varying degrees of homophily with 2 modified GCN-SA-based models, vanilla GCN-SA, and a GCN baseline.

1 Introduction

It's no secret that real-world problems involve graph-structured data. An important characteristic of graphs are their degree of *homophily*, referring to the tendency for nodes with similar features to be closely connected (Li et al. 2023). For instance, in a professional network (a high-homophily graph), people tend to be connected with similar individuals (e.g. industry, position, years of experience). Low-homophily graphs are an important subset of graphs in practice, including device IOT graphs and specific social media communities. As we explore the literature, it becomes evident that learning low-homophily graphs has presented challenges in performance and computational complexity.

1.1 Related Works

1.1.1 Graph Convolutional Networks

Graph convolutional networks (GCNs) are considered the most promising architecture for graph learning in practice (Hamilton, Ying, and Leskovec 2017). Despite this, modern GCNs perform poorly on low-homophily graphs, including GEOM-GCN (Pei et al. 2020) and H_2GCN (Zhu et al. 2020). This is due to a sole reliance on feature representations that only consider adjacent nodes (*message passing*), failing to capture long-range correlations (Jiang et al. 2024).

1.1.2 GCNs with Self-Attention

Jiang et al. (2024) account for long-range correlations by incorporating SA with GCNs in GCN-SA, resulting in exceptional performance on low-homophily graphs. On 3 low-homophily WebKB graph datasets, GCN-SA's node classification accuracy ($90.8 \pm 4.5, 89.7 \pm 2.6, 91.5 \pm 2.5$) significantly outperforms baseline GCN ($60.8 \pm 5.2, 61.1 \pm 5.7, 62.3 \pm 6.4$) and 8 other GNNs (Jiang et al. 2024).

Despite their success, GCN-SA suffers from poor runtime complexity caused by its "dense" full SA mechanism. Full SA necessitates $\mathcal{O}(n^2)$ operations for n nodes. This is empirically evident, where running time for GCN-SA is significantly longer than GCN ($10^{2\pm0.5}$ log-seconds v.s. $10^{0.2\pm0.1}$ log-seconds for 500 epochs on 5 datasets) (Jiang et al. 2024). It is worth mentioning that the size of these datasets are relatively small ($n = 2708, 3327, 19717, 2277, 5201$), a far cry from larger low-homophily graphs seen in practice (Lim et al. 2021).

1.1.3 Graph Transformers

There has been concurrent effort to explore self-attention on graphs in graph transformers (GTs). Notably, Rampášek et al. (2022) propose GRAPHGPS, a recipe that involves combining local message passing and a global attention mechanism — very similar to the approach Jiang et al. (2024) take.

In particular, we'd like to focus on the efforts made to improve the poor quadratic scaling of graph transformers. The central approach is sparse SA mechanisms, which compute a sparse subset of attention scores in less than quadratic time (usually linear). BIGBIRD (Zaheer et al. 2020) and PERFORMER (Choromanski et al. 2020) are general-purpose sparse SA mechanisms that have been applied in a graph context. Graph-specific sparse SA mechanisms have also been developed, most notably EXPHORMER (Shirzad et al. 2023), leveraging *expander graphs* to build graph transformers that approximate full SA efficiently.

Kong et al. (2023)'s GT-based GOAT architecture illustrate SA to improve performance on low-homophily graphs, but do not explore sparse SA approaches. They implement their own k-means-based dimensionality reduction approach to achieve linear SA score computation.

Unlike GCNs, the success of attention in GTs has been less clear. Despite the purported advantages of GT networks over GNNs (e.g. oversmoothing, limited expressivity), the accuracy of GTs have often lagged behind their GNN counterparts (Shirzad et al. 2023; Kong et al. 2023). Shirzad et al. (2023) even suggest that for some datasets, it is better to avoid using attention at all.

1.2 Problem Statement

In this paper we address the problem of balancing high performance and computational efficiency for low-homophily graph learning. We are particularly interested in bridging the gap between the literature of sparse SA for GTs (e.g. EXPHORMER), and the bleeding edge performance of GCN-SA.

1.3 Our Contributions

We offer a focused study of sparse SA techniques applied to the GCN-SA architecture. We perform experiments to determine whether GCN-SA, modified with various sparse SA mechanisms, improves runtime while maintaining similar classification accuracy. We compare 2 baselines GCN and GCN-SA against 2 GCN-SA models modified with sparse SA (BIGBIRD, EXPHORMER). Analysis is performed on 8 datasets of varying homophily.

2 Methodology

This section first introduces notation, definitions, and the classification task. We then describe the relevant technical details of sparse SA prevalent in the GT literature, and review the GCN-SA architecture. Finally, we describe how sparse SA is integrated with GCN-SA.

2.1 Problem Formulation

Notation will mostly follow Jiang et al. (2024) and should be rather unsurprising. Let $G = (V, E)$ be an undirected graph, where V and E are the sets of nodes and edges respectively. We define $n = |V|$

to be the number of nodes in the graph. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the (symmetric) adjacency matrix of G, where $A_{ij} = \mathbb{1}\{e_{ij} \in E\}$. Denote $\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ as our feature matrix, where $\mathbf{x}_i \in \mathbb{R}^d$ corresponds to the features of node $v_i \in V$. Finally, for a set of classes $[c] = \{1, \cdots, c\}$ and a subset of labelled nodes $V_{\text{lab}} \subset V$, $\mathbf{y}_{\text{lab}} \in [c]^{|V_{\text{lab}}|}$ are the corresponding labels.

Our inference task is to predict the labels of the unlabeled nodes given positional and semantic information ($\mathbf{A}$ and $\mathbf{X}$ respectively). More formally, find the posterior distribution $\mathbf{y} \mid \mathbf{X}, \mathbf{A}, \mathbf{y}_{\text{lab}}$ for completed label vector $\mathbf{y} \subset [c]^n$.

Another important definition to consider is the *homophily ratio* of a graph:

Definition 2.1 (Homophily Ratio)**.** For graph $G = (V, E)$, the homophily ratio h is

$$h = \frac{|\{(v_i, v_j) \in E \colon y_i = y_j\}|}{|E|}.$$

As $h \to 1$, G becomes completely homophilous (high-homophily) and as $h \to 0$, G becomes completely heterophilous (low-homophily) (Zhu et al. 2020). For our experiments, we note the homophily ratio corresponding to each graph.

2.2 (Sparse) Self-Attention

Per Zaheer et al. (2020), recall the definition for a *generalized attention mechanism* and a *sparse attention mechanism*.

Definition 2.2 (Generalized Attention Mechanism)**.** $\text{ATTN}_D \colon \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times n}$ is a generalized attention mechanism if it can be characterized by a digraph $D = ([n] = \{1, \cdots, n\}, E_D)$ where $e_{ij} \in E_D$ if v_i attends v_j. The attention scores for $\mathbf{x}_i$ with H heads is

$$\text{ATTN}_D(\mathbf{X})_i = \mathbf{x}_i + \sum_{h=1}^{H} \text{softmax}\left(Q_h(\mathbf{x}_i) K_h(\mathbf{X}_{N(i)})^\top\right) \cdot V_h(\mathbf{X}_{N(i)})$$

where $N(i) = \{j \colon e_{ij} \in E_D\}$ denotes the nodes that i attends to, $\mathbf{X}_{N(i)}$ the submatrix of $\mathbf{X}$ with columns in $N(i)$, $Q_h, K_h \colon \mathbb{R}^d \to \mathbb{R}^m$ the query and key functions, and $V_h \colon \mathbb{R}^d \to \mathbb{R}^d$ the value function.

Definition 2.3 (Sparse Attention Mechanism)**.** We say the attention mechanism ATTN_D is *sparse* if $|E_D| \in o(n^2)$, and more specifically ATTN_D is linear if $|E_D| \in \mathcal{O}(n)$.

Approaches to find D satisfying sparsity typically follow graph-theoretic approaches (e.g. expander graphs for expander attention in EXPHORMER), yielding desirable properties. Most importantly, sparse SA mechanisms can be universal approximators, preserving the properties of full quadratic models (Zaheer et al. 2020). In practice, sparse SA has performed comparably or even better than full SA on a wide array of tasks (Shirzad et al. 2023).

With this in mind, we suspect that sparse attention may improve the overall runtime of GCN-SA with minimal performance hit.

2.3 GCN-SA

The high-level motivation of GCN-SA is to combine three sources of information for learning by the downstream GCN: node feature information, short-range, and long range structural information We focus on the long-range structure learning by GCN-SA's utilization of SA with *reconnected adjacency matrix generation*.

The motivation for learning a reconnected adjacency matrix is to represent long-range connections between nodes. An attention score matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is formed to generate a sparse reconnected adjacency matrix $\mathbf{A}^* \in \mathbb{R}^{n \times n}$ via H-headed multi-head self-attention (MHSA):

$$\mathbf{S} = \frac{1}{H} \sum_{h=1}^{H} \text{cosine}(\mathbf{X}\mathbf{W}_h^Q), \tag{1}$$

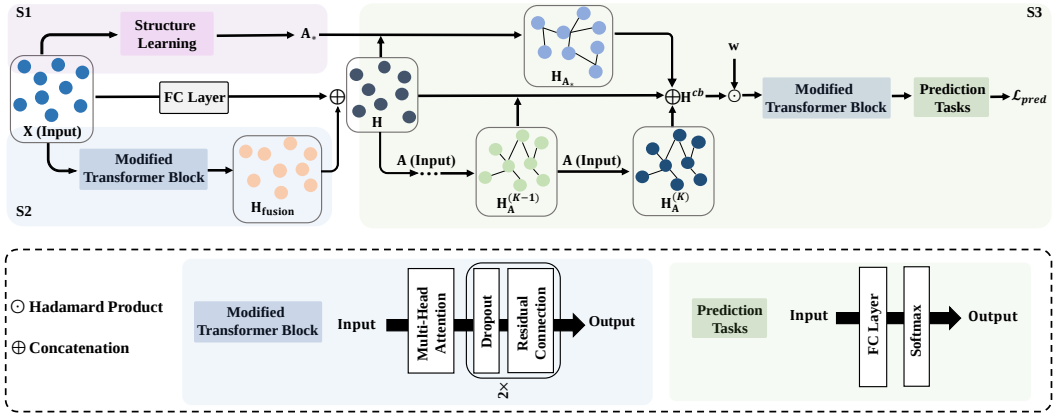

Figure 1: GCN-SA architecture (Jiang et al. 2024).

where $\mathbf{W}_h^Q \in \mathbb{R}^{d\times p}$ are learnable parameters corresponding to head $h = 1, \cdots, H$, cosine: $\mathbb{R}^{n\times p} \to [-1,1]^{n\times n}$ the pairwise cosine similarity function for matrix columns, and $\mathbf{S} \in [-1,1]^{n\times n}$ the attention score matrix. Once $\mathbf{S}$ is obtained, we construct the reconnected adjacency matrix $\mathbf{A}^* \in \{0,1\}^{n\times n}$ with a combined kNN and minimum-threshold approach:

$$A^*_{ij} = \mathbb{1}\{S_{ij} > \varepsilon \text{ or } S_{ij} \in R_i\}$$

for hyperparameters ε, r and R_i denotes the r largest elements of row S_i.. A minor point is afterwards we set $A^*_{ij} = \max\{A^*_{ij}, A^*_{ji}\}$ to enforce symmetry. The intention of this procedure is to ensure $\mathbf{A}^*$ is *sparse* for better downstream performance (Jiang et al. 2024). More specifically, each node attends to $\mathcal{O}(1)$ other nodes. We touch on this in subsection 2.4. For simplicity the authors denote

$$\mathbf{A}^* = \text{StructureLearning}(\mathbf{X}; \mathbf{W}^Q).$$

parameterized by query parameters $\mathbf{W}^Q = (\mathbf{W}_1^Q, \cdots, \mathbf{W}_H^Q)$. Once $\mathbf{A}^*$ is computed, it is used downstream to compute $\mathbf{H}_{\mathbf{A}^*}$, representing aggregated long-range information:

$$\mathbf{H}_{\mathbf{A}^*} = \mathbf{D}_*^{-\frac{1}{2}}\mathbf{A}^*\mathbf{D}_*^{-\frac{1}{2}}\mathbf{H}$$

for $\mathbf{D}_* = \text{diag}(\mathbf{A}^*\mathbf{1}_n)$ the degree matrix of $\mathbf{A}^*$ and $\mathbf{H} \in \mathbb{R}^{n\times q}$ representing node embeddings with hidden dimension q.

2.4 Modifying GCN-SA with Sparse SA

This section describes the central contribution of the paper: replacing the expensive SA computations in GCN-SA with sparse equivalents.

Observe that the runtime of StructureLearning is $\mathcal{O}(n^2 d)$ due to the full SA mechanism. We can reduce the runtime from quadratic (in n) to linear by replacing StructureLearning with a sparse SA mechanism.

StructureLearning will have better runtime complexity. For instance, EXPHORMER's expander attention and global attention yield a runtime of $\mathcal{O}(nd)$ (we exclude EXPHORMER's local attention as we are only concerned with long-range dependencies).

Downstream in the model, $\mathbf{A}^*$ is only used once more to compute $\mathbf{H}_{A^*}$ As $\mathbf{A}^*$ computed with the full SA and kNN + minimum-threshold approach is sparse ($\mathcal{O}(1)$ non-zero entries per row), the computation of $\mathbf{H}_{A^*}$ already occurs in $\mathcal{O}(nq)$ and remains unchanged with our modified architecture.

3 Experiments

The purpose of this section is to evaluate the empirical performance and runtime of 2 baseline models (GCN, GCN-SA) and 2 GCN-SA-based architectures modified with sparse SA mechanisms

Table 1: Node classification accuracy

Method/Dataset	Cora	Cites.	Pubmed	Chame.	Squir.	Corne.	Texas	Wiscon.
GCN	87.5±1.2	78.3±2.1	88.2±0.4	60.7±2.1	42.4±1.2	62.5±3.6	60.9±5.7	60.3±4.9
GCN-SA	90.6±0.4	79.2±0.5	91.0±0.6	63.8±2.0	46.8±1.5	91.5±4.9	90.0±2.4	91.0±2.4
GCN-SA+BBIRD	90.6±0.4	79.0±0.6	90.8±0.6	64.8±1.8	46.7±1.7	91.9±4.2	89.4±2.4	91.9±2.3
GCN-SA+EXPH	90.4±0.4	79.0±0.6	91.0±0.5	63.9±1.9	46.8±1.7	90.1±4.1	90.1±2.8	91.8±2.3

(BIGBIRD, and EXPHORMER) to determine whether GCN-SA fitted with sparse SA mechanisms maintain similar classification performance whilst reducing runtime. We present results on 8 graph datasets of varying degrees of homophily. More specifically, we are interested in 2 main quantities:

1. Classification accuracy for each model evaluated on a $60/20/20$ train/validation/test split for each dataset.
2. Empirical runtime measurements for each model and each dataset on 500 epochs.

3.1 Models and Baselines

We introduce BIGBIRD and EXPHORMER to GCN-SA's architecture:

- GCN-SA+EXPHORMER: introduce EXPHORMER's sparse SA patterns to GCN-SA per subsection 2.4. For this experiment we use default hyperparameters: 5-degree expander graphs, `Random-d` expander algorithm, and 1 virtual global node.
- GCN-SA+BIGBIRD: Employ BIGBIRD's sparse attention mechanism to GCN-SA per subsection 2.4. Again we employ default settings.

We compare 2 modified GCN-SA with sparse SA mechanisms to 2 baselines:

- GCN: The graph convolutional nework is the base to our modified models. Predictions are made by averaging over short-range neighbouring nodes.
- GCN-SA: The base model of our sparse SA mechanisms as described in subsection 2.3.

All parameters and hyperparameters pertaining to GCN-SA will be identical across models, allowing fair testing to clearly visualize the discrepancies in results.

3.2 Results

Table 1 illustrates the node classification accuracy of GCN-SA+BIGBIRD and GCN-SA+EXPHORMER against the accuracy of GCN and GCN-SA. As expected, GCN performs the worst out of the 4 models. There are no clear implications that sparse SA mechanisms reduce the accuracy of GCN-SA.

We expected the models equipped with sparse SA mechanisms to have reduced epoch runtime. However, when comparing with GNC-SA, no clear improvements are seen, expecially some models performed worse on specific datasets like GCN-SA+EXPHORMER on citerseer.

Considering the accuracy of GCN-SA against GCN-SA+BIGBIRD and GCN-SA+EXPHORMER, limited differences are seen. On the Cora dataset, GCN-SA and GCN-SA+BIGBIRD performed exactly the same, (90.6 ± 0.4). Runtime comparisons also dictate no discrepancies between full SA and sparse SA mechanisms with GCN-SA + BIDBIRD and GCN-SA+EXPHORMER performing practically the same as GCN-SA.

4 Discussion and Limitations

Interpreting our results as-is, while we can conclude that sparse SA mechanisms on GCN-SA preserve classification accuracy, it cannot be said that runtime has improved dramatically. We suspect that these results are a consequence of limitations in our experiment design.

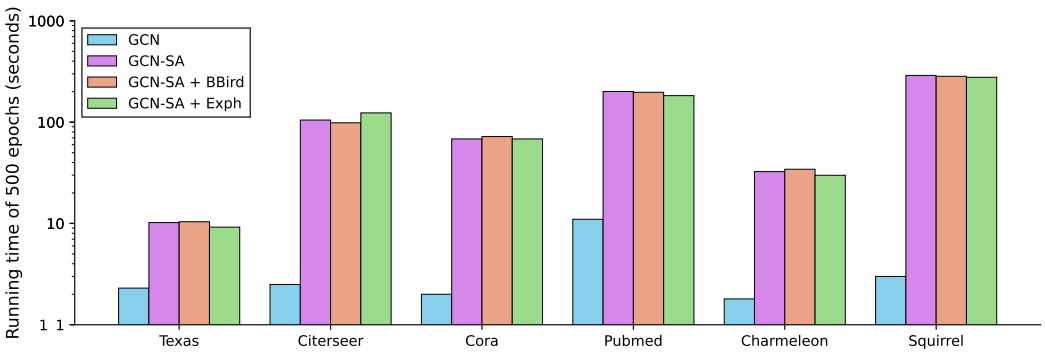

Figure 2: Runtime Comparison.

Graphs explored in the experiment are relatively small (< 5000 nodes), mostly due to constraints in computational resources. With more allocated time, exploration on the Long Range Graph Benchmark datasets (Dwivedi et al. 2022) will illustrate the performance of our model for larger low-homophily graphs, ultimately validating our models' ability to capture low homophily for large datasets at a substantial decrease in runtime. Of course this implies the need for access to improved resources (e.g. GPUs).

While sparse SA mechanisms like BIGBIRD, and PERFORMER introduce linear attention, they sustain computation overhead that dominates the per-epoch computation time for certain graphs (Shirzad et al. 2023). In practice, better computational complexity does not directly entail runtime improvements, as a full-attention transformer can be empirically faster than many sparse SA mechanisms for graphs up to 5000 nodes (Shirzad et al. 2023). Again, experimentation on larger datasets may answer this concern.

Another concern is that our exploration is limited to only one model (GCN-SA) due to its high performing accuracy on low homophily graphs. However, other models with slightly worse performance (IDGL, H2GCN, CPGNN) (Jiang et al. 2024) may respond better to sparse SA mechanisms. However it is worth noting that how SA is applied to these models may be less clean that it is for GCN-SA. Furthermore, additional types of sparse SA, outside of EXPHORMER and BIGBIRD requires investigation (e.g. PERFORMER, LONGFORMER). Different SA algorithms may improve performance on certain models, while performing suboptimally on others. That being said, future studies should consider a wider composite of SA mechanisms being tested on low-homophily datasets.

Finally, learning on larger graphs can propose ethical implications. Social media data and other public service networks have concerns over user privacy, as ensuring legal consent for using individuals' data is essential. For large datasets, it is difficult to guarantee every individuals' consent is obtained, thus raising issues regarding the handling of data when consent is not explicitly given or when it is impractical to obtain.

5 Conclusion

In this study we explore the use of sparse SA mechanisms with the recent breakthrough model GCN-SA for learning low-homophily graphs. We explore related approaches in the literature, review relevant components of GCN-SA and sparse SA, and perform an experiment on 3 modified GCN-SA model with sparse SA against 2 baseline models across 8 datasets. From our results, we observe that although sparse SA mechanisms maintain the classification performance of GCN-SA, it cannot be said that they reduce empirical runtime. Further exploration on larger datasets with more configuration options is required.

References

Choromanski, Krzysztof, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. (2020). "Rethinking attention with performers." *arXiv preprint arXiv:2009.14794*.

Dwivedi, Vijay Prakash, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini (2022). "Long range graph benchmark." *Advances in Neural Information Processing Systems* 35, pp. 22326–22340.

Hamilton, William L, Rex Ying, and Jure Leskovec (2017). "Representation learning on graphs: Methods and applications." *arXiv preprint arXiv:1709.05584*.

Jiang, Mengying, Guizhong Liu, Yuanchao Su, and Xinliang Wu (2024). "Self-Attention Empowered Graph Convolutional Network for Structure Learning and Node Embedding." *arXiv preprint arXiv:2403.03465*.

Kong, Kezhi, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein (2023). "GOAT: A global transformer on large-scale graphs." *International Conference on Machine Learning*. PMLR, pp. 17375–17390.

Li, Wen-Zhi, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai (2023). "Homogcl: Rethinking homophily in graph contrastive learning." *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1341–1352.

Lim, Derek, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim (2021). "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods." *Advances in Neural Information Processing Systems* 34, pp. 20887–20902.

Pei, Hongbin, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang (2020). "Geom-gcn: Geometric graph convolutional networks." *arXiv preprint arXiv:2002.05287*.

Rampášek, Ladislav, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini (2022). "Recipe for a general, powerful, scalable graph transformer." *Advances in Neural Information Processing Systems* 35, pp. 14501–14515.

Shirzad, Hamed, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop (2023). "Exphormer: Sparse transformers for graphs." *International Conference on Machine Learning*. PMLR, pp. 31613–31632.

Zaheer, Manzil, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. (2020). "Big bird: Transformers for longer sequences." *Advances in neural information processing systems* 33, pp. 17283–17297.

Zhu, Jiong, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra (2020). "Beyond homophily in graph neural networks: Current limitations and effective designs." *Advances in neural information processing systems* 33, pp. 7793–7804.

A Dataset Descriptions

This section will discuss the graph datasets used, and define the homophily ratio classifier.

Of the 8 datasets used to validate the proposed sparse SA GCA-SA-based models, 3 are citation networks (Cora, Citeseer, Pubmed), 2 are Wikipedia networks (Chameleon, Squirrel), and the last 3 are WebKB networks (Cornell, Texax, Wisconsin) (Jiang et al. 2024).

The nodes in the citation graphs correspond to papers, each containing an academic topic as a label. Edges are citations between papers with each node. These datasets have a significant amount of does due to the amount of publications, and with certain paper having hundreds of thousands of citations of them, the number of edges are seamingly large as well.

Nodes in the Wikipedia graphs are Wikipedia pages, and edges refer to reciprocal links between pages. The interconnectedness of these graphs are extremely large because of the links between each type of squirrel and chameleon.

Finally, the WebKB datasets were collected from each universitys' respective computer science department. Nodes represent web pages and edges are hyperlinks between web pages. These have the lowest number of nodes as the webpages are relatively small compared to Pubmed or Wikipedia. Here is a summary of all the datasets from (Jiang et al. 2024):

Based on the definition of the homophily ratio, citation networks have the highest degree of homophily while Wikipedia and WebKB networks display low homophily. In the experimentation, we wish to preserve the predictive accuracy of the low homophily networks while improving the runtime through sparse SA mechanisms.

Table 2: Summary of the datasets utilized in our experiment.

Datasets	Cora	Citese.	Pubmed	Chamele.	Squirr.	Cornell.	Texas	Wiscon.
HOM.RATIO h	0.81	0.74	0.8	0.23	0.22	0.3	0.11	0.21
# NODES	2708	3327	19717	2277	5201	183	183	251
# EDGES	5429	4732	44338	31421	198493	295	309	499