Beyond Low-Pass Filters: Adaptive Feature Propagation on Graphs

Shouheng Li[1,3](✉), Dongwoo Kim[2], and Qing Wang[1]

[1] The Australian National University, Australia
[2] GSAI, POSTECH, South Korea
[3] CSIRO Data61, Australia
shouheng.li@anu.edu.au

Abstract. Graph neural networks (GNNs) have been extensively studied for prediction tasks on graphs. As pointed out by recent studies, most GNNs assume local homophily, i.e., strong similarities in local neighborhoods. This assumption however limits the generalizability power of GNNs. To address this limitation, we propose a flexible GNN model, which is capable of handling any graphs without being restricted by their underlying homophily. At its core, this model adopts a node attention mechanism based on multiple learnable spectral filters; therefore, the aggregation scheme is learned adaptively for each graph in the spectral domain. We evaluated the proposed model on node classification tasks over eight benchmark datasets. The proposed model is shown to generalize well to both homophilic and heterophilic graphs. Further, it outperforms all state-of-the-art baselines on heterophilic graphs and performs comparably with them on homophilic graphs.

Keywords: Graph neural network · Representation learning · Spectral methods.

1 Introduction

Graph neural networks (GNNs) have recently demonstrated great power in graph-related learning tasks, such as node classification [12], link prediction [43] and graph classification [15]. Most GNNs follow a message-passing architecture where, in each GNN layer, a node aggregates information from its direct neighbors indifferently. In this architecture, information from long-distance nodes is propagated and aggregated by stacking multiple GNN layers together [4, 12, 38]. However, this architecture underlies the assumption of local homophily, i.e. proximity of similar nodes. While this assumption seems reasonable and helpful to achieve good prediction results on homophilic graphs such as citation networks [26], it limits GNNs' generalizability to heterophilic graphs. Heterophilic graphs commonly exist in the real-world, for instance, people tend to connect to opposite gender in dating networks, and different amino acid types are more likely to form connections in protein structures [45]. Moreover, determining whether a graph is homophilic or not is a challenge by itself. In fact, strong and weak homophily

can both exhibit in different parts of a graph, which makes a learning task more challenging.

Pei et al. [26] proposed a metric to measure local node homophily based on how many neighbors of a node are from the same class. Using this metric, they categorized graphs as homophilic (strong homophily) or heterophilic (weak homophily), and showed that classical GNNs such as GCN [12] and GAT [38] perform poorly on heterophilic graphs. Liu et al. [20] further showed that GCN and GAT are outperformed by a simple multi-layer perceptron (MLP) in node classification tasks on heterophilic graphs. This is because the naive local aggregation of homophilic models brings in more noise than useful information for such graphs. These findings indicate that these GNN models perform sub-optimally when the fundamental assumption of homophily does not hold.

Based on the above observation, we argue that a well-generalized GNN should perform well on graphs regardless of homophily. Furthermore, since a real-world graph can exhibit both strong and weak homophily in different node neighborhoods, a powerful GNN model should be able to aggregate node features using different strategies accordingly. For instance, in heterophilic graphs where a node shares no similarity with any of its direct neighbors, such a GNN model should be able to ignore direct neighbors and reach farther to find similar nodes, or at least, resort to the node's attributes to make a prediction. Since the validity of the assumption about homophily is often unknown, such aggregation strategies should be learned from data rather than decided upfront.

To circumvent this issue, in this paper, we propose a novel GNN model with attention-based adaptive aggregation, called ASGAT. Most existing attention-based aggregation architectures perform self-attention to the local neighborhood of a node [38]. Unlike these approaches, we aim to design an aggregation method that can gather informative features from both close and far-distant nodes. To achieve this, we employ graph wavelets under a relaxed condition of localization, which enables us to learn attention weights for nodes in the spectral domain. In doing so, the model can effectively capture information from frequency components and thus aggregate both local information and global structure into node representations.

To further improve the generalizability of our model, instead of using pre-defined spectral kernels, we propose to use multi-layer perceptrons (MLP) to learn desired spectral filters without limiting their shapes. Existing works on graph wavelet transform choose wavelet filters heuristically, such as heat kernel, wave kernel and personalized page rank kernel [13, 14, 41]. They are mostly low-pass filters, which means that these models implicitly treat high-frequency components as "noises" and have them discarded [2, 9, 25, 34]. However, this may hinder the generalizability of models since high-frequency components can carry meaningful information about local discontinuities, as analyzed in [34]. Our model overcomes these limitations using node attentions derived from fully learnable spectral filters.

To summarize, the main contributions of this work are as follows:

1. We empirically show that high-frequency components carry important information on heterophilic graphs which can be used to improve prediction performance.
2. We propose a generalized GNN model which performs well on both homophilic and heterophilic graphs, regardless of graph homophily.
3. We exhibit that multi-headed attention produced by multiple spectral filters work better than attention obtained from a single filter, as it enables flexibility to aggregate features from different frequency components.

We conduct extensive experiments to compare ASGAT with well-known baselines on node classification tasks. The experimental results show that ASGAT significantly outperforms the state-of-the-art methods on heterophilic graphs where local node homophily is weak, and performs comparably with the state-of-the-art methods on homophilic graphs where local node homophily is strong. This empirically verifies that ASGAT is a general model for learning on different types of graphs.[4]

2 Preliminaries

Let $G = (V; E; A; x)$ be an undirected graph with N nodes, where V, E, and A are the node set, edge set, and adjacency matrix of G, respectively, and $x : V \mathrel{?} \mathbb{R}^m$ is a graph signal function that associates each node with a feature vector. The normalized Laplacian matrix of G is defined as $L = I \quad D^{-1/2} A D^{-1/2}$, where $D \; 2 \; \mathbb{R}^{N \quad N}$ is the diagonal degree matrix of G. In spectral graph theory, the eigenvalues $\quad = \mathrm{diag}(\;_1; \ldots; \;_N)$ and eigenvectors U of $L = U \quad U^H$ are known as the graph's spectrum and spectral basis, respectively, where U^H is the Hermitian transpose of U. The graph Fourier transform of x is $\hat{x} = U^H x$ and its inverse is $x = U\hat{x}$.

The spectrum and spectral basis carry important information on the connectivity of a graph [34]. Intuitively, lower frequencies correspond to global and smooth information on the graph, while higher frequencies correspond to local information, discontinuities and possible noise [34]. One can apply a spectral filter and use graph Fourier transform to manipulate signals on a graph in various ways, such as smoothing and denoising [32], abnormally detection [22] and clustering [39]. Spectral convolution on graphs is defined as the multiplication of a signal x with a filter $g(\;)$ in the Fourier domain, i.e.

$$g(L)x = g(U \quad U^H)x = Ug(\;)U^H x = Ug(\;)\hat{x}: \tag{1}$$

When a spectral filter is parameterized by a scale factor, which controls the radius of neighbourhood aggregation, Equation 1 is also known as the Spectral Graph Wavelet Transform (SGWT) [9, 34]. For example, Xu et al. [41] uses a small scale parameter $s < 2$ for a heat kernel, $g(s\;) = e^{\;\lambda s}$, to localize the wavelet at a node.

[4] The extended version of this work is available on arXiv [18]. Our open-sourced code is available at `https://github.com/seanli3/asgat`.

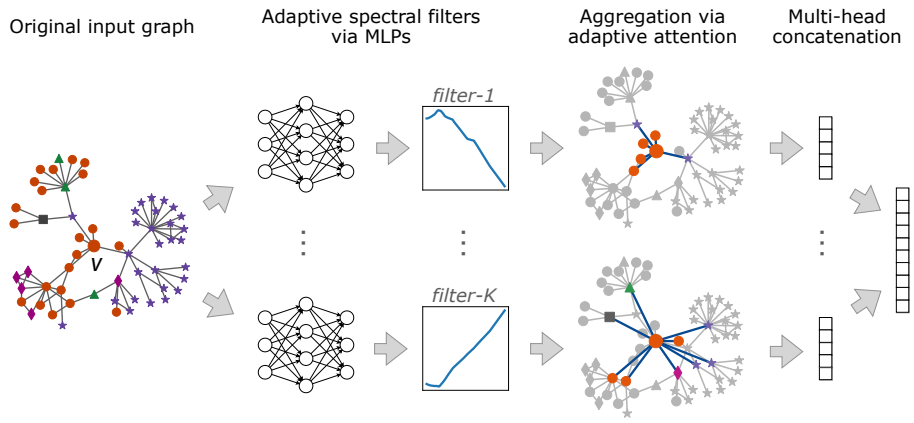

Fig. 1: Illustration of a spectral node attention layer on a three-hop ego network of the central node v from the CiteSeer dataset. Shape and color indicate node classes. Passing the graph through two learned spectral filters place attention scores on nodes, including node v itself. Nodes with positive attention scores are presented in color. Node features are aggregated for node v according to attention scores. The low-pass filter attends to local neighbors (filter 1), while the high-pass filter skips the first hop and attends the nodes in the second hop (filter K). The resulting embeddings from multiple heads are then concatenated before being sent to the next layer (multi-head concatenation). Note that we have visualized learned filters from experiments.

3 Proposed Approach

Graph neural networks (GNNs) learn lower-dimensional embeddings of nodes from graph structured data. In general, given a node, GNNs iteratively aggregate information from its neighbor nodes, and then combine the aggregated information with its own information. An embedding of node v at the lth layer of GNN is typically formulated as

$$m_v = \text{aggregate}(f h_u^{(l-1)} | u \in N_v g)$$
$$h_v^{(l)} = \text{combine}(h_v^{(l-1)}; m_v);$$

where N_v is the set of neighbor nodes of node v, m_v is the aggregated information from the neighbors, and $h_v^{(l)}$ is the embedding of node v at the lth layer ($h_v^{(0)} = x_v$). The embedding $h_v^{(L)}$ of node v at the final layer is then used for some prediction tasks. In most GNNs, N_v is restricted to a set of one-hop neighbors of node v. Therefore, one needs to stack multiple aggregation layers in order to collect the information from more than one-hop neighborhood within this architecture.

Adaptive spectral filters. Instead of stacking multiple aggregation layers, we introduce a spectral attention layer that rewires a graph based on spectral graph

wavelets. A spectral graph wavelet $_v$ at node v is a modulation in the spectral domain of signals centered around the node v, given by an N-dimensional vector

$$_v = Ug( )U^H \ _v; \tag{2}$$

where $g( )$ is a spectral filter and $_v$ is a one-hot vector for node v.

The common choice of a spectral filter is a heat kernel. A wavelet coefficient $_{vu}$ computed from a heat kernel can be interpreted as the amount of energy that node v has received from node u in its local neighborhood. In this work, instead of using pre-defined localized kernels, we use multi-layer perceptrons (MLP) to learn spectral filters. With learnable spectral kernels, we obtain the inverse graph wavelet transform

$$_v = U\mathrm{diag}(\mathrm{MLP}( ))U^H \ _v. \tag{3}$$

Unlike a low-pass heat kernel, where the wavelet coefficients can be understood as the amount of energy after heat diffusion, the learned coefficients $_{vu}$ do not always correspond to energy diffusion. In spectral imaging processing, lower frequency components preserve an image's background, while higher frequency components are useful to detect object edges or outlines. Similarly, in spectral graph theory, lower-frequency components carry smoothly changing signals. Therefore a low-pass filter is a reasonable choice to extract features and denoise a homophilic graph. On the contrary, higher-frequency components carry abruptly changing signals, corresponding to the discontinuities and "opposite attraction" characteristics of heterophilic graphs. In our experiments, the trained MLP resembles a low-pass filter, working as a diffusion operator, with homophilic graphs. In contrast, with heterophilic graphs, the trained MLP reaches a high-pass filter at most times (section 4).

Note that we use the terminology wavelet and spectral filter interchangeably as we have relaxed the wavelet definition from [9] so that learnable spectral filters in our work are not necessarily localized in the spectral and spatial domains.

Remark 1. Equation 3 requires the eigen-decomposition of a Laplacian matrix, which is expensive and infeasible for large graphs. To address this computational issue, one may use well-studied methods such as Chebyshev [9, 14, 41] and Auto-Regressive Moving-Average (ARMA) [11, 19] to efficiently compute an approximate the graph filtering of MLP in Equation 3.

Attention mechanism. Unlike the previous work [41] where the output of inverse graph wavelet transform are directly used to compute node embeddings, we normalize the output through a softmax layer

$$a_v = \mathrm{softmax}(\ _v); \tag{4}$$

where $a_v \ 2 \ \mathrm{R}^N$ is an attention weight vector. With attention weights, an update layer is then formalized as

$$h_v^{(l)} = \left(\sum_{u=1}^{N} a_{vu} h_u^{(l \ \ 1)} W^{(l)} \right); \tag{5}$$

where $W^{(l)}$ is a weight matrix shared across all nodes at the lth layer and is ELU nonlinear activation.

Note that the update layer is not divided into aggregation and combine steps in our work. Instead, we compute the attention a_{vv} directly from a spectral filter. Unlike heat kernel and other spectral filters, the output of inverse graph wavelet transform with a learnable spectral kernel are not always localized. Hence, the model can adaptively aggregate information from both close and far-distant nodes, depending on their attention weights.

Sparsified node attentions. With predefined localized spectral filters such as a heat kernel, most wavelet coefficients are zero due to their locality. In our work, spectral filters are fully determined by data. Consequently, attention weights obtained from learnable spectral filters do not impose any sparsity. This means we need to retrieve all possible nodes in a graph, which is inefficient, to perform an aggregation operation. From our experiments, we observe that most attention weights are negligible after softmax. Thus, we consider a sparsification technique to keep only the largest k entries of Equation 3 for each node, i.e.

$$_{vu} = \begin{cases} _{vu} & \text{if } _{vu} \; 2 \; \text{topK}(f \; _{v0}; :::; \; _{vN} g; k) \\ 1 & \text{otherwise}; \end{cases} \tag{6}$$

where topK is a partial sorting function that returns the largest k entries from a set of wavelet bases $f \; _{v0}; :::; \; _{vN} g$. This technique guarantees attention sparsity such that the embedding of each node can be aggregated from at most k other nodes with a time complexity trade-off of $O(N + k \log N)$. The resulting is then fed into the softmax layer to compute attention weights.

We adopt multi-head attention to model multiple spectral filters. Each attention head aggregates node information with a different spectral filter, and the aggregated embedding is concatenated before sent to the next layer. To reduce redundancy, we adopt a single MLP: $\mathbb{R}^N \; ! \; \mathbb{R}^{N \; M}$, where M is the number of attention heads, and each column of the output corresponds to one adaptive spectral filter.

We name the multi-head spectral attention architecture as a *adaptive spectral graph attention network* (ASGAT). The design of ASGAT is easily generalizable, and many existing GNNs can be expressed as special cases of ASGAT (see Appendix [18]). Figure 1 illustrates how ASGAT works with two attention heads learned from the CiteSeer dataset. As shown in the illustration, the MLP learns adaptive filters such as low-pass and high-pass filters. A low-pass filter assigns high attention weights in local neighborhoods, while a high-pass filter assigns high attention weights on far-distant but similar nodes, which a traditional hop-by-hop aggregation scheme cannot capture.

4 Experiments

To evaluate the performance of our proposed model, we conduct experiments on node classification tasks with homophilic graph datasets, and heterophilic graph

Table 1: Micro-F1 results for node classification. The proposed model consistently outperforms the GNN methods on heterophilic graphs and performs comparably on homophilic graphs. Results marked with y are obtained from Pei et al. [26]. Results marked with z are obtained from Zhu et al. [45].

	Homophily ⟸					⟹ Heterophily		
	Cora	Pubmed	CiteSeer	Chameleon	Squirrel	Wisconsin	Cornell	Texas
β	0.83	0.79	0.71	0.25	0.22	0.16	0.11	0.06
#Nodes	2,708	19,717	3,327	2,277	5,201	251	183	183
#Edges	5,429	44,338	4,732	36,101	217,073	515	298	325
#Features	1,433	500	3,703	2,325	2,089	1,703	1,703	1,703
#Classes	7	3	6	5	5	5	5	5
GCN	87.4 ± 0.2	87.8 ± 0.2	78.5 ± 0.5	$59.8 \pm 2.6^{\ddagger}$	$36.9 \pm 1.3^{\ddagger}$	64.1 ± 6.3	59.2 ± 3.2	64.1 ± 4.9
ChevNet	88.2 ± 0.2	89.3 ± 0.3	79.4 ± 0.4	66.0 ± 2.3	39.6 ± 3.0	82.5 ± 2.8	76.5 ± 9.4	79.7 ± 5.0
ARMANet	85.2 ± 2.5	86.3 ± 5.7	76.7 ± 0.5	62.1 ± 3.6	47.8 ± 3.5	78.4 ± 4.6	74.9 ± 2.9	82.2 ± 5.1
GAT	87.6 ± 0.3	83.0 ± 0.1	77.7 ± 0.3	$54.7 \pm 2.0^{\ddagger}$	$30.6 \pm 2.1^{\ddagger}$	62.0 ± 5.2	58.9 ± 3.3	60.0 ± 5.7
SGC	87.2 ± 0.3	81.1 ± 0.3	78.8 ± 0.4	33.7 ± 3.5	46.9 ± 1.7	51.8 ± 5.9	58.1 ± 4.6	58.9 ± 6.1
GraphSAGE	86.3 ± 0.6	89.2 ± 0.5	77.4 ± 0.5	51.1 ± 0.5	$41.6 \pm 0.7^{\ddagger}$	77.6 ± 4.6	67.3 ± 6.9	82.7 ± 4.8
APPNP	$\mathbf{88.4} \pm 0.3$	86.0 ± 0.3	77.6 ± 0.6	45.3 ± 1.6	31.0 ± 1.6	81.2 ± 2.5	70.3 ± 9.3	79.5 ± 4.6
Geom-GCN	86.3 ± 0.3	89.1 ± 0.1	$\mathbf{81.4} \pm 0.3$	$60.9^{\dagger}$	$38.1^{\dagger}$	$64.1^{\dagger}$	$60.8^{\dagger}$	$67.6^{\dagger}$
H_2GCN	88.3 ± 0.3	89.1 ± 0.4	78.4 ± 0.5	59.4 ± 2.0	37.9 ± 2.0	86.5 ± 4.4	82.2 ± 6.0	82.7 ± 5.7
MLP	72.1 ± 1.3	88.6 ± 0.2	74.9 ± 1.8	45.7 ± 2.7	28.1 ± 2.0	82.7 ± 4.5	81.4 ± 6.3	79.2 ± 6.1
Vanilla ASGAT	–	–	–	–	–	$\mathbf{86.9} \pm 4.2$	$\mathbf{84.6} \pm 5.8$	82.2 ± 3.2
ASGAT-Cheb	87.5 ± 0.5	$\mathbf{89.9} \pm 0.9$	79.3 ± 0.6	$\mathbf{66.5} \pm 2.8$	$\mathbf{55.8} \pm 3.2$	86.3 ± 3.7	82.7 ± 8.3	$\mathbf{85.1} \pm 5.7$
ASGAT-ARMA	87.4 ± 1.1	88.3 ± 1.0	79.2 ± 1.4	65.8 ± 2.2	51.4 ± 3.2	84.7 ± 4.4	83.2 ± 5.5	79.5 ± 7.7

datasets. Further ablation study highlights the importance of considering the entire spectral frequency.

4.1 Experimental Setup

Baseline methods. An exact computation of Equation 3 requires to compute the eigenvectors of the Laplacian matrix, which is often infeasible due to a large graph size. To overcome this issue, we approximate graph wavelet transform response of MLP with Chebyshev polynomial, dubbed as ASGAT-Cheb, and ARMA rational function, dubbed as ASGAT-ARMA. We also report the results from the exact computation of eigenvectors whenever possible, which is dubbed as vanilla ASGAT.

We compare all variants against 10 benchmark methods, they are vanilla GCN [12] and its simplified version SGC [40]; two spectral methods: ChevNet [4] and ARMANet [1]; the graph attention model GAT [38][5]; APPNP, which also adopts adaptive aggregation [13]; the neighbourhood-sampling method GraphSage [8]; Geom-GCN [26] and H_2GCN [45], both also target prediction on het-

[5] It was reported in Velickovic et al. [38] that GAT does not always outperform GCN when using different data splittings, and similar results have been reported by Zhu et al. [45]

erophilic graphs. We also include MLP in the baselines since it performs better than many GNN methods on some heterophilic graphs [20].

Datasets. We evaluate our model and the baseline methods on node classification tasks over three citation networks: **Cora**, **CiteSeer** and **Pubmed** [33], three webgraphs from the WebKB dataset[6]: **Wisconsin**, **Texas** and **Cornell**, and webgraphs from Wikipedia called **Chameleon** and **Squirrel** [30].

To quantify the homophily of graphs, we use the metric introduced by Pei et al. [26],

$$= \frac{1}{N} \sum_{v 2 V} \quad {}_{v} \quad \text{and} \quad {}_{v} = \frac{jfu\ 2\ N_v j`(u) = `(v) gj}{jN_v j}; \tag{7}$$

where $`(v)$ refers to the label of node v. measures the degree of homophily of a graph, and ${}_{v}$ measures the homophily of node v in the graph. A graph has strong local homophily if is large and vice versa. Details of these datasets are summarized in Table 1.

Experimental settings. For citation networks, we follow the experimental setup for node classification from [3, 8, 10] and report the results averaged on 10 runs. For webgraphs, we run each model on the 10 splits provided by [26] and take the average, where each split uses 60%, 20%, and 20% nodes of each class for training, validation and testing, respectively. The results we report on GCN and GAT are better than Pei et al. [26] as a result of converting the graphs to undirected before training [7]. Geom-GCN uses node embeddings pre-trained from different embedding methods such as Isomap [37], Poincare [24] and struc2vec [29]. We report the best micro-F1 results among all three variants for Geom-GCN.

We use the best-performing hyperparameters specified in the original papers of baseline methods. For hyperparameters not specified in the original papers, we use the parameters from Fey and Lenssen [6]. We report the test accuracy results from epochs with both the smallest validation loss and highest validation accuracy. Early termination is adopted for both validation loss and accuracy, thus training is stopped when neither validation loss or accuracy improve for 100 consecutive epochs. For ASGAT, we use a two-layer architecture where multi-headed filters are learned using a MLP of 2 hidden layers. Each layer of the MLP consists of a linear function and a ReLU activation. To avoid overfitting, dropout is applied in each ASGAT layer on both attention weights and inputs equally. Results for vanilla ASGAT are only reported for small datasets where eigen-decomposition is feasible. Other hyperparameters are obtained by grid search, where details are given in Appendix [18].

4.2 Results and Discussion

We use two evaluation metrics to evaluate the performance of node classification tasks: micro-F1 and macro-F1. The results with micro-F1 are summarized in

[6] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

[7] https://openreview.net/forum?id=S1e2agrFvS

Table 1. Overall, on homophilic citation networks, ASGAT performs comparably with the state-of-the-art methods, ranking first on Pubmed and second on Cora and CiteSeer in terms of micro-F1 scores. On heterophilic graphs, ASGAT outperforms all other methods by a margin of at least 2.4% on 3 out of 4 datasets. These results indicate that ASGAT generalizes well on different types of graphs. The results with macro-F1 are summarized in Appendix [18]. Macro-F1 scores have not been reported widely in the literature yet. Here, we report the macro-F1 since the heterophilic graphs have imbalanced class distributions than the homophilic graphs. As the results show, ASGAT outperforms all other methods across all heterophilic graphs in macro-F1. The difference between the two approximation methods is not significant. Except for a few cases, the difference comes from hyperparameters selection. The vanilla ASGAT gives more consistent results than the approximations, although the difference seems marginal.

Although ASGAT performs well on both homophilic and heterophilic graphs, it is unclear how ASGAT performs on heterophilic neighbourhoods of an homophilic graph where nodes are mostly of different classes. Thus, we report an average classification accuracy on nodes at varying levels of $_v$ in Figure 2 on the homophilic graphs CiteSeer and Pubmed. The nodes are binned into five groups based on $_v$. For example, all nodes with $0.3 < {}_v \quad 0.4$ belong to the bin at 0.4. We have excluded Cora from the report since it has very few heterophilic neighbourhoods.

The results in Figure 2 show that all models except ASGAT perform poorly when $_v$ is low. One may argue that the performance on heterophilic graphs might improve by stacking multiple GNN layers together to obtain information from far-distant nodes. However, it turns out that this approach introduces an oversmoothing problem [17] that degrades performance. On the other hand, the better performance of ASGAT on heterophilic nodes suggests the adaptive spectral filters reduce noise aggregated locally while allowing far-distant nodes to participate.

Attention sparsification. The restriction on top k entries in Equation 6 guarantees a certain level of sparsification. Nonetheless, ASGAT requires a partial sorting which adds an overhead of $O(n + k \log N)$. To further analyze the impact of attention sparsity on run-time, we plot the density of an attention matrix with varying k in Figure 3 along with its running time. The results are drawn from two datasets: the heterophilic dataset Chameleon and the homophilic dataset Cora. As expected, ASGAT shows a stable growth in the attention density as the value of k increases. It also shows that ASGAT runs much faster when attention weights are well-sparsified. In our experiments, we find the best results are achieved on $k < 20$. The impact of k on classification performance is further analyzed in Appendix [18].

Frequency range ablation. To understand how adaptive spectral filters contribute to ASGAT's performance on heterophilic graphs, we conduct an ablation study on spectral frequency ranges. We first divide the entire frequency range $(0 \quad 2)$ into a set of predefined sub-ranges exclusively. Then we manually set the filter frequency responses to zero for each sub-range to check its impact

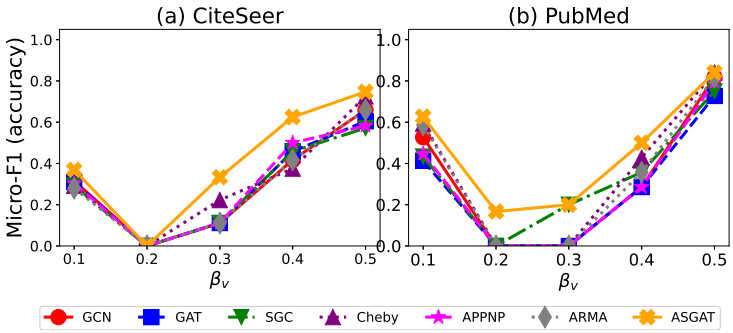

Fig. 2: Micro-F1 results for classification accuracy on heterophilic nodes ($_v$ 0:5). ASGAT shows better accuracy on classifying heterophilic nodes than the other methods.

on the classification. The frequencies within a selected sub-range contribute to neither node attention nor feature aggregation, therefore helping to reveal the importance of the sub-range. We consider three different lengths of sub-ranges, i.e., step=1:0, step=0:5, and step=0:25. The results of frequency ablation on the three homophilic graphs are summarized in Figure 4.

The results for step=1:0 reveal the importance of high-frequency range (1 2) on node classification of heterophilic graphs. The performance is significantly dropped by ablating high-frequency range on all datasets. Further investigation at the finer-level sub-ranges (step=0:5) shows that ablating sub-range 0:5 1:5 has the most negative impact on performance, whereas the most important sub-range varies across different datasets at the finest level (step=0:25). This finding matches our intuition that low-pass filters used in GNNs underlie a homophily assumption similar to naive local aggregation. We suspect the choice of low-pass filters also relates to oversmoothing issues in spectral methods [17], but we leave it for future work.

Attention head ablation. In ASGAT, each head uses a spectral filter to produce attention weights. To delve into the importance of a spectral filter, we further follow the ablation method used by Michel et al. [21]. Specifically, we ablate one or more filters by manually setting their attention weights to zeros. We then measure the impact on performance using micro-F1. If the ablation results in a large decrease in performance, the ablated filters are considered important. We observe that all attention heads (spectral filters) in ASGAT are of similar importance, and only all attention heads combined produce the best performance. Please check Appendix [18] for the detailed results.

Time complexity. In vanilla ASGAT, eigen-decomposition is required for Equation 3 which has a time complexity of $O(N^3)$. ASGAT-Cheb and ASGAT-ARMA avoid eigen-decomposition and are able to scale to large graphs as their time complexities are $O(R \quad jEj)$ and $O((P \quad T + Q) \quad jEj)$ respectively, where R, P and Q are polynomial orders that are normally less than 30, T is the number

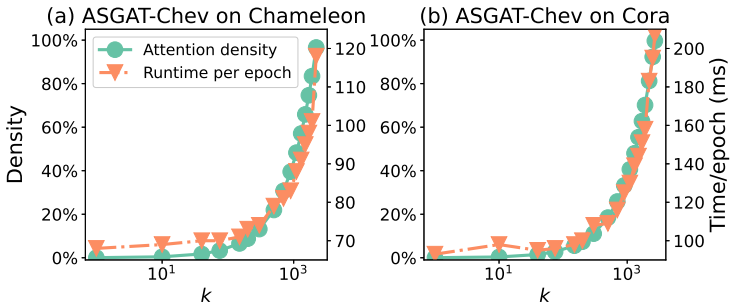

Fig. 3: Attention matrix density and training runtime with respect to k. Attention matrix sparsified by keeping the top k elements at each row, which effectively improves runtime efficiency.

of iterations that is normally less than 50. Therefore, both ASGAT-Cheb and ASGAT-ARMA scale linearly with the number of edges $|E|$. Readers can refer to Appendix [18] for a more detailed introduction of these two methods. Secondly, partial sorting used in the attention sparsification of Equation 6 requires $O(N + k \log N)$. Lastly, Equation 4 is performed on a length-k vector for N rows; therefore, a time complexity of $O(k \cdot N)$ is needed. In practice, we have $R \cdot P \cdot T \cdot Q \cdot k \ll N \ll |E|$ for most graphs, therefore, for a model with M heads, the overall time complexity is $O(M \cdot R \cdot |E|)$ for ASGAT-Cheb and $O(M \cdot (P \cdot T + Q) \cdot |E|)$ for ASGAT-ARMA.

5 Related Work

Graph neural networks have been extensively studied recently. We categorize work relevant to ours into three perspectives and summarize the key ideas.

Attention on graphs. Graph attention networks (GAT) [38] was the first to introduce attention mechanisms on graphs. GAT assigns different importance scores to local neighbors via an attention mechanism. Similar to other GNN variants, long-distance information propagation in GAT is realized by stacking multiple layers together. Therefore, GAT suffers from the oversmoothing issue [44]. Zhang et al. [42] improve GAT by incorporating both structural and feature similarities while computing attention scores.

Spectral graph filters and wavelets. Some GNNs also use graph wavelets to extract information from graphs. Xu et al. [41] applied graph wavelet transform defined by Shuman et al. [34] to GNNs. Klicpera et al. [14] proposed a general GNN argumentation using graph diffusion kernels to rewire the nodes. Donnat et al. [5] used heat wavelet to learn node embeddings in unsupervised ways and showed that the learned embeddings closely capture structural similarities between nodes. Other spectral filters used in GNNs can also be viewed as special forms of graph wavelets [1, 4, 12]. Coincidentally, Chang et al. [2] also noticed

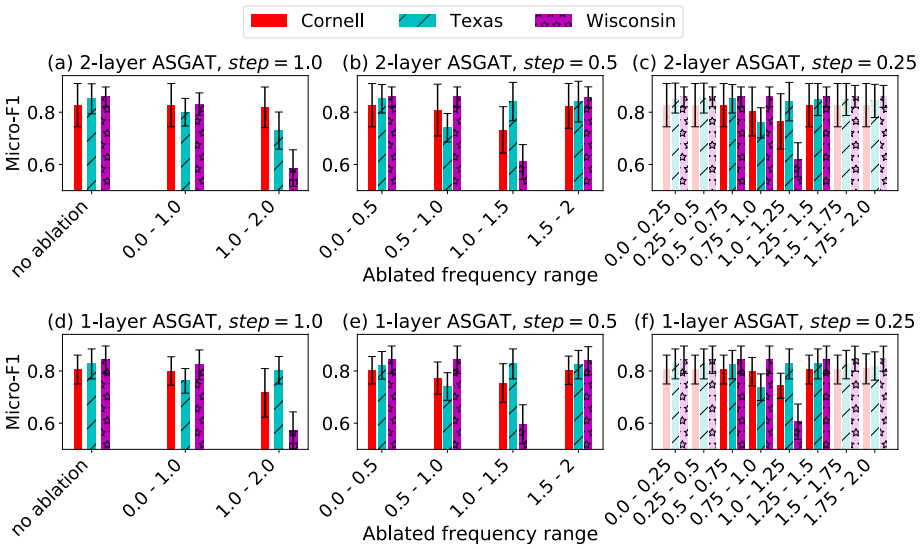

Fig. 4: Micro-F1 with respect to ablated frequency sub-ranges on heterophilic graphs. We divide the frequency range into a set of sub-ranges with different lengths. The results (a) and (d) reveal the importance of high-frequency range $(1 \quad 2)$. Further experiments show that there is a subtle difference in the most important range across datasets, but it ranges between $(0.75 \quad 1.25)$.

useful information carried by high-frequency components from a graph Laplacian. Similarly, they attempted to utilize such components using node attentions. However, they resorted to the traditional choice of heat kernels and applied such kernels separately to low-frequency and high-frequency components divided by a hyperparameter. In addition to this, their work did not link high-frequency components to heterophilic graphs.

Prediction on heterophilic graphs. Pei et al. [26] have drawn attention to GCN and GAT's poor performance on heterophilic graphs very recently. They try to address the issue by essentially pivoting feature aggregation to structural neighborhoods from a continuous latent space learned by unsupervised methods. Liu et al. [20] proposed another attempt to address the issue. They proposed to sort locally aggregated node embeddings along with a one-dimensional space and used a one-dimensional convolution layer to aggregate embeddings a second time. By doing so, non-local but similar nodes can attend to the aggregation. Very recently, Zhu et al. [45] showed that a heuristic combination of ego-, neighbor and higher-order embedding improves GNN performance on heterophilic graphs. Coincidentally, they also briefly mentioned the importance of higher-frequency components on heterophilic graphs, but they did not provide an empirical analysis.

Although our method shares some similarities in motivation with the work above, it is fundamentally different in several aspects. To the best of our knowledge,

our method is the first architecture we know that computes multi-headed node attention weights purely from learned spectral filters. As a result, in contrast to the commonly used heat kernel, our method utilizes higher-frequency components of a graph, which helps predict heterophilic graphs and neighbourhoods.

6 Conclusion

In this paper, we study the node classification tasks on graphs where local homophily is weak. We argue that the assumption of homophily is the cause of poor performance on heterophilic graphs. In order to design more generalizable GNNs, we suggest that a more flexible and adaptive feature aggregation scheme is needed. To demonstrate, we have introduced the adaptive spectral graph attention network (ASGAT), which achieves flexible feature aggregation using learnable spectral graph filters. By utilizing the full graph spectrum adaptively via the learned filters, ASGAT can aggregate features from close and far nodes. For node classification tasks, ASGAT outperforms all benchmarks on heterophilic graphs and performs comparably on homophilic graphs. On homophilic graphs, ASGAT also performs better for nodes with weak local homophily. We find that the performance gain is closely linked to the higher end of the frequency spectrum through our analysis.

Acknowledgement

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1F1A1061667).

References

[1] Bianchi, F.M., Grattarola, D., Livi, L., Alippi, C.: Graph neural networks with convolutional ARMA filters. CoRR abs/1901.01343 (2019)
[2] Chang, H., Rong, Y., Xu, T., Huang, W., Sojoudi, S., Huang, J., Zhu, W.: Spectral graph attention network. CoRR abs/2003.07450 (2020)
[3] Chen, J., Ma, T., Xiao, C.: Fastgcn: Fast learning with graph convolutional networks via importance sampling. In: 6th International Conference on Learning Representations, ICLR, OpenReview.net (2018)
[4] Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 3837–3845 (2016)
[5] Donnat, C., Zitnik, M., Hallac, D., Leskovec, J.: Learning structural node embeddings via diffusion wavelets. In: Proceedings of the 24th ACM International Conference of Knowledge Discovery & Data Mining (KDD), pp. 1320–1329 (2018)
[6] Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. CoRR abs/1903.02428 (2019)

[7] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864, ACM (2016)

[8] Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Annual Conference on Neural Information Processing Systems, pp. 1024–1034 (2017)

[9] Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis (2011), ISSN 10635203

[10] Huang, W., Zhang, T., Rong, Y., Huang, J.: Adaptive sampling towards fast graph representation learning. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Annual Conference on Neural Information Processing Systems, pp. 4563–4572 (2018)

[11] Isufi, E., Loukas, A., Simonetto, A., Leus, G.: Autoregressive Moving Average Graph Filtering. IEEE Transactions on Signal Processing 65(2), 274–288 (2017), ISSN 1053587X

[12] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) (2017)

[13] Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)

[14] Klicpera, J., Weißenberger, S., Günnemann, S.: Diffusion improves graph learning. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 13333–13345 (2019)

[15] Lee, J.B., Rossi, R.A., Kong, X.: Graph classification using structural attention. In: Proceedings of the 24th ACM International Conference on Knowledge Discovery & Data Mining (KDD), pp. 1666–1674 (2018)

[16] Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Annual Conference on Neural Information Processing Systems, pp. 2177–2185 (2014)

[17] Li, Q., Han, Z., Wu, X.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the 32nd Conference on Artificial Intelligence (AAAI), pp. 3538–3545 (2018)

[18] Li, S., Kim, D., Wang, Q.: Beyond low-pass filters: Adaptive feature propagation on graphs. Preprint abs/2103.14187 (2021)

[19] Liu, J., Isufi, E., Leus, G.: Filter Design for Autoregressive Moving Average Graph Filters. IEEE Transactions on Signal and Information Processing over Networks 5(1), 47–60 (2019), ISSN 2373776X

[20] Liu, M., Wang, Z., Ji, S.: Non-local graph neural networks. CoRR abs/2005.14612 (2020)

[21] Michel, P., Levy, O., Neubig, G.: Are sixteen heads really better than one? In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Annual Conference on Neural Information Processing Systems 2019, pp. 14014–14024 (2019)
[22] Miller, B.A., Beard, M.S., Bliss, N.T.: Matched filtering for subgraph detection in dynamic networks. In: 2011 IEEE Statistical Signal Processing Workshop (SSP), pp. 509–512 (2011)
[23] Napoli, E.D., Polizzi, E., Saad, Y.: Efficient estimation of eigenvalue counts in an interval. Numer. Linear Algebra Appl. 23(4), 674–692 (2016)
[24] Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Advances in Neural Information Processing Systems (2017), ISSN 10495258
[25] NT, H., Maehara, T.: Revisiting graph neural networks: All we have is low-pass filters. CoRR abs/1905.09550 (2019)
[26] Pei, H., Wei, B., Chang, K.C., Lei, Y., Yang, B.: Geom-gcn: Geometric graph convolutional networks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net (2020)
[27] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Macskassy, S.A., Perlich, C., Leskovec, J., Wang, W., Ghani, R. (eds.) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pp. 701–710, ACM (2014)
[28] Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J.: Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (eds.) Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, pp. 459–467, ACM (2018)
[29] Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: Struc2vec: Learning node representations from structural identity. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017), ISBN 9781450348874
[30] Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale attributed node embedding. CoRR abs/1909.13021 (2019)
[31] Sakiyama, A., Watanabe, K., Tanaka, Y.: Spectral Graph Wavelets and Filter Banks with Low Approximation Error. IEEE Transactions on Signal and Information Processing over Networks 2(3), 230–245 (2016), ISSN 2373776X
[32] Schaub, M.T., Segarra, S.: Flow smoothing and denoising: Graph signal processing in the edge-space. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 735–739 (2018)
[33] Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29(3), 93–106 (2008)
[34] Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30(3), 83–98 (2013)

[35] Tang, J., Qu, M., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: Cao, L., Zhang, C., Joachims, T., Webb, G.I., Margineantu, D.D., Williams, G. (eds.) Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165–1174, ACM (2015)
[36] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: Gangemi, A., Leonardi, S., Panconesi, A. (eds.) Proceedings of the 24th International Conference on World Wide Web, WWW, pp. 1067–1077, ACM (2015)
[37] Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science (2000), ISSN 00368075
[38] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations (ICLR) (2018)
[39] Wai, H., Segarra, S., Ozdaglar, A.E., Scaglione, A., Jadbabaie, A.: Community detection from low-rank excitations of a graph filter. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4044–4048 (2018)
[40] Wu, F., Jr., A.H.S., Zhang, T., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML), vol. 97, pp. 6861–6871 (2019)
[41] Xu, B., Shen, H., Cao, Q., Qiu, Y., Cheng, X.: Graph wavelet neural network. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
[42] Zhang, K., Zhu, Y., Wang, J., Zhang, J.: Adaptive structural fingerprints for graph attention networks. In: Proceedings of the 8th International Conference on Learning Representations (ICLR) (2020)
[43] Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, pp. 5171–5181 (2018)
[44] Zhao, L., Akoglu, L.: Pairnorm: Tackling oversmoothing in gnns. In: Proceedings of the 8th International Conference on Learning Representations (ICLR) (2020)
[45] Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)