Node Classification Beyond Homophily: Towards a General Solution

Zhe Xu
University of Illinois Urbana-Champaign
USA
zhexu3@illinois.edu

Yuzhong Chen
Visa Research
USA
yuzchen@visa.com

Qinghai Zhou
University of Illinois Urbana-Champaign
USA
qinghai2@illinois.edu

Yuhang Wu
Visa Research
USA
yuhawu@visa.com

Menghai Pan
Visa Research
USA
menpan@visa.com

Hao Yang
Visa Research
USA
haoyang@visa.com

Hanghang Tong
University of Illinois Urbana-Champaign
USA
htong@illinois.edu

ABSTRACT

Graph neural networks (GNNs) have become core building blocks behind a myriad of graph learning tasks. The vast majority of the existing GNNs are built upon, either implicitly or explicitly, the homophily assumption, which is not always true and could heavily degrade the performance of learning tasks. In response, GNNs tailored for heterophilic graphs have been developed. However, most of the existing works are designed for the specific GNN models to address heterophily, which lacks generality. In this paper, we study the problem from the structure learning perspective and propose a family of general solutions named ALT. It can work hand in hand with most of the existing GNNs to handle graphs with either low or high homophily. At the core of our method is learning to (1) decompose a given graph into two components, (2) extract complementary graph signals from these two components, and (3) adaptively integrate the graph signals for node classification. Moreover, analysis based on graph signal processing shows that our framework can empower a broad range of existing GNNs to have adaptive filter characteristics and further modulate the input graph signals, which is critical for handling complex homophilic/heterophilic patterns. The proposed ALT brings significant and consistent performance improvement in node classification for a wide range of GNNs over a variety of real-world datasets.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → **Data mining**; • **Theory of computation** → **Graph algorithms analysis**.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599446

KEYWORDS

graph machine learning; graph data augmentation; graph neural network; node classification

ACM Reference Format:
Zhe Xu, Yuzhong Chen, Qinghai Zhou, Yuhang Wu, Menghai Pan, Hao Yang, and Hanghang Tong. 2023. Node Classification Beyond Homophily: Towards a General Solution. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599446

1 INTRODUCTION

Graph neural networks (GNNs) have demonstrated great power as building blocks for a variety of graph learning tasks, such as node classification [15], graph classification [36], link prediction [42], clustering [2], and many more. Most of the existing GNNs follow the homophily assumption, i.e., edges tend to connect nodes with the same labels and similar node features. Such an assumption holds true for networks such as citation networks [4, 40] where a paper tends to cite related literature. However, in many other cases, the *heterophilic* settings arise. For instance, to form a protein structure, different types of amino acids are more likely to be linked together [48]. On such heterophilic networks, the performance of classic GNN models [13, 16, 30] could degrade greatly and might be even worse than a multilayer perceptron (MLP) which does not utilize any topology information at all [48].

In response, researchers have analyzed the limitations of the existing GNNs in the presence of node heterophily and further proposed specific models to address it from both the spatial and spectral perspectives. For instance, an important design by H2GCN [48] is that high-order neighbors should be considered during message aggregation. GPRGNN [6] also aggregates messages from multi-hop neighbors but it emphasizes that messages can also be negative via a set of learnable aggregation weights. From the spectral perspective, FAGCN [3] points out that low-pass filter-based GNNs smooth the node representations between connected nodes, which is not desirable for the heterophilic settings where connected nodes are more likely to have different labels. Hence, FAGCN [3] adaptively mixes the low-pass graph filter with the high-pass graph filter via

an attention mechanism to tackle this problem. A more detailed review of related work can be found in Section 5.

Despite the theoretic insights and empirical performance gain, most of the existing works focus on the model level, i.e., they aim to propose better GNNs models to handle the heterophilic graphs. In other words, the success of their methods relies on specific designs of GNN models. In this paper, we take a step further and ask: *how to develop a generic method to benefit a broad range of GNNs for node classification beyond homophily, even if they are not originally tailored for the heterophilic graphs?* To this end, we address this problem from a structure learning [49] perspective, that is, we optimize the given graph structure to benefit downstream tasks (e.g., node classification). Different from the existing approaches that refine the specific GNNs models, our approach focuses on the data level by optimizing the graph topology to tackle heterophily.

Challenges. In pursuing such a data-centric general solution, here are the key challenges. First (*model diversity*), our goal is to strengthen a broad range of established GNNs so that they can handle graphs with arbitrary homophily. However, the aggregation mechanism and the graph convolution kernels are different between various GNN models. It is unknown how to accommodate diverse GNNs seamlessly. Second (*theoretical foundation*), analyses on the success of some specific GNNs for heterophilic graphs have recently emerged (e.g., from the graph signal processing perspective [27]). However, few works focus on the theoretical foundation of structure learning and its connection to dealing with graphs with low homophily. Our main contributions are listed as follows:

- **General framework.** We propose a general graph structure learning framework named duAL sTructure learning (ALT), which can accommodate a variety of GNN models. Specifically, after removing the activation function from the last layer, a large variety of GNNs can be plugged into our framework and be trained end-to-end with common optimizers.
- **Proof and Analysis.** We provide an in-depth analysis from the graph signal processing perspective. Our analysis guides the design of ALT and validates its effectiveness theoretically
- **Empirical evaluations.** Experiments show that with the help of the ALT, the average accuracy boosting of existing GNNs is from 7%-15% on 8 heterophilic datasets, and from 1.6%-4% on 8 homophilic graphs. Moreover, results show that a classic low-pass filter-based GNN working together with our proposed ALT can be a strong competitor against state-of-the-art baseline methods.

The rest of this paper is organized as follows. In Section 2, we introduce the notations and the semi-supervised node classification task. We present the proposed ALT framework in Section 3 with a detailed analysis. In Section 4, experimental settings and empirical results are provided. In Section 5, we review related works and after that, we conclude this paper in Section 6.

2 PRELIMINARIES

Notations. We use bold uppercase letters for matrices (e.g., $\mathbf{A}$), bold lowercase letters for column vectors (e.g., $\mathbf{u}$), lowercase and uppercase letters in regular font for scalars (e.g., d, K), and calligraphic letters for sets (e.g., $\mathcal{T}$). We use $\mathbf{A}[i,j]$ to represent the entry of matrix $\mathbf{A}$ at the i-th row and the j-th column, $\mathbf{A}[i,:]$ to represent the i-th row of matrix $\mathbf{A}$, and $\mathbf{A}[:,j]$ to represent the j-th column of matrix $\mathbf{A}$. Similarly, $\mathbf{u}[i]$ denotes the i-th entry of vector $\mathbf{u}$. Superscript $\top$ denotes the transpose of matrices and vectors. $\odot$ denotes the Hadamard product.

An attributed graph can be represented as $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ which is composed of an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and an attribute matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n is the number of nodes and d is the node feature dimension. In total, nodes can be categorized into a set of classes C. The normalized Laplacian matrix is $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ where $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{A}$. It can be decomposed as $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U} \in \mathbb{R}^{n\times n}$ is the eigenvector matrix and $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ is the diagonal eigenvalue matrix. In graph signal processing [27], the diagonal entry of $\mathbf{\Lambda}$ represents frequency and $\mathbf{\Lambda}[i,i] = \lambda_i$. Given a signal $\mathbf{x} \in \mathbb{R}^n$, its graph Fourier transform [27] is represented as $\hat{\mathbf{x}} = \mathbf{U}\mathbf{x}$, and its inverse graph Fourier transform is defined as $\mathbf{x} = \mathbf{U}^\top\hat{\mathbf{x}}$. For a diffusion matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$, its frequency response (or profile [1]) is defined as $\mathbf{\Phi}_{\mathtt{fp}} = \mathtt{diag}(\mathbf{U}^\top\mathbf{C}\mathbf{U})$ where $\mathtt{diag}(\cdot)$ returns the diagonal entries. This frequency response is also known as the filter and the convolution kernel.

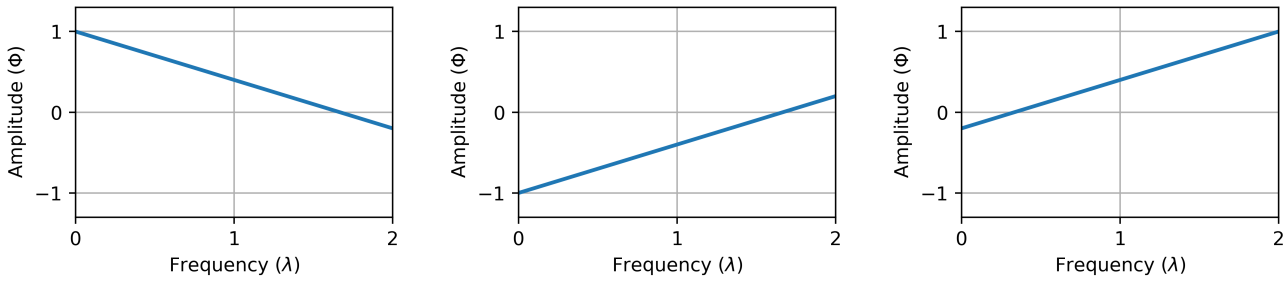
Semi-supervised Node Classification. In this paper, we study *semi-supervised node classification* [15, 40] where the graph topology $\mathbf{A}$, all node features $\mathbf{X}$, and a part of node labels are given and our goal is to predict the labels of unlabelled nodes. Numerous works [15, 16, 30] achieve impressive performance on this problem. However, recent studies show that their successes heavily rely upon the homophily assumption of the given graphs [46, 48]. In general, homophily describes to what extent edges tend to link nodes with the same labels and similar features. Following previous works [24, 48], this paper focuses on the node label homophily. There are various homophily metrics and we introduce one of them named edge homophily [48] as: $h(\mathcal{G}) = \frac{\sum_{i,j,\mathbf{A}[i,j]=1}[\![\mathbf{y}[i]=\mathbf{y}[j]]\!]}{\sum_{i,j}\mathbf{A}[i,j]} \in [0,1]$, where $[\![x]\!] = 1$ if x is true and 0 otherwise. The more homophilic a given graph is, the closer its $h(\mathcal{G})$ is to 1.

3 PROPOSED METHODS

In this section, we first propose a flexible method named ALT-global which empowers a wide range of GNN with an **adaptive filter** characteristics. Next, we carefully analyze the expressiveness of ALT-global from the graph signal processing perspective [27]. This analysis guides the design of another more advanced method named ALT-local which enhances the spectral expressiveness of a broad range of GNNs to be local adaptive filters by **modulating** the input graph signals.

3.1 ALT-global: A Global Adaptive Method

Intuitively, nodes with different labels should be located as far as possible in the embedding space and nodes with the same labels should be assigned closely. This intuition is aligned well with the utility of many classic GNNs (e.g., GCN [15]) on homophilic graphs. That is because, on homophilic graphs, many same-label nodes are connected, whose embeddings will be smoothed by those classic low-pass filter GNNs [1, 3]. In contrast, the low-pass filter GNNs' performance degrades significantly on heterophilic graphs since the connected nodes' embeddings should not be smoothed. Many efforts [3, 6] point out that a key design to deal with graphs with unknown homophily is to equip GNNs with an adaptive filter.



(a) A given filter (b) Reflection (c) Reflection + offset

Figure 1: The Illustration of obtaining a filter with complementary filter characteristics. Given a filter (a), its reflected frequency response (b) with offset (c) has complementary filter characteristics.

We aim to propose a data-centric solution such that minimal modification on the given GNNs (e.g., a low-pass filter GNN) is needed. As we do not make any assumption about the filter characteristic of the given GNN, its filter can be either low-pass, high-pass, band-pass, or others. To equip the given GNN with an adaptive filter, our core idea is to *adaptively combine signals from two filters with the complementary filter characteristics*. For example, if a low-pass filter GNN is given, it should be adaptively combined with another high-pass filter. To find such a complementary filter, a two-step modification of the frequency response is needed. Figure 1 shows that we can first *reflect the frequency response curve over the frequency axis* and then *set an appropriate offset to the reflected frequency response*. Guided by this idea, the mathematical details of the proposed ALT-global are as follows,

$$\mathbf{H}_1 = \texttt{GNN}(w\mathbf{A}, \mathbf{X}, \theta_1), \tag{1a}$$

$$\mathbf{H}_2 = \texttt{GNN}((1-w)\mathbf{A}, \mathbf{X}, \theta_2), \tag{1b}$$

$$\mathbf{H}_{\texttt{offset}} = \texttt{MLP}(\mathbf{X}, \theta_3), \tag{1c}$$

$$\mathbf{Z} = \texttt{softmax}(\mathbf{H}_1 - \mathbf{H}_2 + \eta\mathbf{H}_{\texttt{offset}}), \tag{1d}$$

where θ_1 and θ_2 are the parameters of the **backbone dual GNNs** (i.e., GNNs from Eq. 1a and Eq. 1b), θ_3 is the parameter of a multi-layer perceptron (MLP), $\eta \in \mathbb{R}$ and $w \in [0, 1]$ are learnable parameters, and $\mathbf{Z} \in \mathbb{R}^{n \times C}$ is the prediction matrix. Here the $\texttt{softmax}$ is applied row-wise. For models using the normalized adjacency matrix (e.g., $\tilde{\mathbf{A}} = (\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}$) as the diffusion matrix (e.g., GCN [15]), the re-weighting can be set over the normalized adjacency matrix (i.e., $w\tilde{\mathbf{A}}$ and $(1-w)\tilde{\mathbf{A}}$).

We elaborate more on the design of ALT-global. First, all the insights we obtained from Figure 1 are applicable to the convolution kernel directly. Nonetheless, since our method works in a plug-and-play fashion that does not modify the backbone GNNs, it uses a well-designed aggregation (i.e., Eq. 1d) to achieve an equivalent effect. Specifically, (1) $\mathbf{H}_1$ is the signals from a backbone GNN with positive re-scaling; (2) $-\mathbf{H}_2$ is the negative signals that correspond to the signals from a reflected filter; (3) $\eta\mathbf{H}_{\texttt{offset}}$ is the offset term which is equivalent to signals from an all-pass filter. Second, the adaptive mixture of the above three sets of graph signals is controlled by the learnable parameters w and η. Other aggregation functions are also applicable. One of the options is an MLP whose input is the concatenation of $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_{\texttt{offset}}$. However, it is not used in this paper because (1) it increases the analysis difficulties dramatically and (2) empirically, no performance advantage is observed in the ablation study (Section 4.4). Analysis in the following section shows that ALT-global bears strong flexibility in filter characteristics.

3.2 Analysis of ALT-global

For brevity, in the analysis, we assume that the backbone GNNs are graph-augmented MLPs (GA-MLPs) as defined below. This is because many GNNs fall into the GA-MLP family if part of the nonlinear functions is removed; also, GA-MLPs have shown strong empirical performance while enjoying provable expressiveness [5].

Definition 1. ***Graph-Augmented Multi-Layer Perceptron (GA-MLP)*** *[5] is a family of GNNs that first conduct feature transformation via an MLP and then diffuse the features. Mathematically they compute node embeddings as* $\mathbf{H} = \mathbf{C} \cdot \texttt{MLP}(\mathbf{X})$ *where* $\mathbf{C}$ *is the diffusion matrix.*

The (full) frequency profile [1] is used for analysis as follows.

Definition 2. ***Frequency profile*** *[1] is defined as* $\mathbf{\Phi}_{\texttt{fp}} = \texttt{diag}(\mathbf{U}^\top\mathbf{C}\mathbf{U})$ *where* $\texttt{diag}(\cdot)$ *returns the diagonal entries if* $\mathbf{U}^\top\mathbf{C}\mathbf{U}$ *is a diagonal matrix. In case* $\mathbf{U}^\top\mathbf{C}\mathbf{U}$ *is not a diagonal matrix,* ***full frequency profile*** *[1] is defined as* $\mathbf{\Phi} = \mathbf{U}^\top\mathbf{C}\mathbf{U}$.

It is well-known that the frequency profile of a diffusion matrix (if diagonal) is a filter/convolution kernel for the input graph signal. Next, we show that ALT is indeed equipped with an adaptive filter.

Lemma 1. *The filter characteristic of the proposed ALT-global (Eq. 1d) is adaptive regardless of the frequency filtering functionality of the backbone GNNs (Eq. 1a and Eq. 1b).*

Proof. For analysis convenience, we assume (1) the learnable weight w is multiplied with the diffusion matrix $\mathbf{C}$, and (2) the backbone GNNs are GA-MLPs whose MLP modules (from Eq. 1a and Eq. 1b) share common parameters with the offset MLP (from Eq. 1c). We start from the case where backbone GNNs are fixed low-pass filters. Without loss of generality, their corresponding full frequency profiles can be presented as $\mathbf{\Phi} = \mathbf{I} - \xi(\mathbf{\Lambda})$ where ξ is a monotonically increasing function. Then, in this case, the diffusion matrices from two GNNs are re-weighted as $w\mathbf{C}$ and $(1-w)\mathbf{C}$ respectively. Considering the offset MLP as a special GA-MLP whose diffusion matrix is $\mathbf{I}$, the aggregated graph signals are $w\mathbf{C} \cdot \texttt{MLP}(\mathbf{X}) - (1-w)\mathbf{C} \cdot \texttt{MLP}(\mathbf{X}) + \eta\mathbf{I} \cdot \texttt{MLP}(\mathbf{X}) = \tilde{\mathbf{C}} \cdot \texttt{MLP}(\mathbf{X})$ where the aggregated diffusion matrix is $\tilde{\mathbf{C}} = w\mathbf{C} - (1-w)\mathbf{C} + \eta\mathbf{I}$. Hence the diagonal entry of the corresponding full frequency profile is

$$\mathbf{\Phi}[i,i] = \Phi(\lambda_i) = (2w-1)(1-\xi(\lambda_i)) + \eta.$$

When $w > 0.5$, i.e., $2w - 1 > 0$, $\Phi(\lambda_i)$ is a monotonically decreasing function. The proposed method is a low-pass filter when $\eta > 0$. Similarly, it is a high-pass filter when w is close to 0 and $\eta > 1$. The above conditions are sufficient and in fact, there are many other combinations of w and η which lead to low-pass/high-pass filters. Similar results can be obtained when the backbone GNNs are fixed high-pass filters and we omit that part for brevity. □

Remarks. The filter characteristics of the ALT-global can also be interpreted from the Graph Diffusion Equation (GDE) [23] perspective and we provide the GDE-related analysis in Appendix.

3.3 Global Filters vs. Local Filters

ALT-global is proved to be equipped with adaptive filter characteristics. However, ALT-global fundamentally applies a global filter to every node, which could lead to suboptimal performance. Recent studies [32, 47] reveal that heterophilic connection patterns differ between different nodes. Take gender classification on a dating network as an example. While node pairs are often of different labels (i.e., genders), homosexuality also exists between some node pairs. Therefore, simply applying a global low-pass or high-pass filter over all the nodes can degrade the overall classification performance.

Next, we study how to generalize our proposed ALT-global to a local (i.e., node-specific) and adaptive filter. Before that, let us take a closer look at the full frequency profile [1]: $\mathbf{\Phi} = \mathbf{U}^\top \mathbf{C}\mathbf{U}$. In the following proposition, we point out that $\mathbf{\Phi}$ can describe both the filter and modulator characteristics of a given diffusion matrix $\mathbf{C}$.

Proposition 1. *The diagonal entries of the full frequency profile $\mathbf{\Phi}$ of the diffusion matrix serve as the **filter** and the non-zero off-diagonal entries are the **frequency modulator**.*

Proof. The diffusion of the input graph signal $\mathbf{X}_{\texttt{in}} = \texttt{MLP}(\mathbf{X})$ can be represented as $\mathbf{C}\mathbf{X}_{\texttt{in}} = \mathbf{U}\mathbf{\Phi}\mathbf{U}^\top\mathbf{X}_{\texttt{in}} = \mathbf{U}(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})$, where $\hat{\mathbf{X}}_{\texttt{in}}$ is the input graph signal in spectral domain. From the perspective of graph signal processing [27], $(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})[i\,:]$ represents the amplitude of output graph signal whose frequency is λ_i. We further expand the computation and obtain

$$(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})[i\,:] = \sum_j \mathbf{\Phi}[i,j] \cdot \mathbf{X}_{in}[j,:].$$

In the summation, the diagonal terms of $\mathbf{\Phi}$ represent the filter/convolution kernel adopted by many spectral GNNs [1]. If non-zero off-diagonal entries of $\mathbf{\Phi}$ exist, it shows that the λ_i-component of the output graph signal is merged with scaled (by $\mathbf{\Phi}[i,j]$) λ_j-component of the input signal which is essentially the modulation [27]. □

Based on the above property of $\mathbf{\Phi}$, the following proposition points out the key design for local filter characteristics.

Proposition 2. *Modulation of the input graph signal (i.e., **non-zero off-diagonal entries** of $\mathbf{\Phi}$) is necessary for **local** filters.*

Proof. We follow the terminology used in the proof of Proposition 1. If the full frequency profile $\mathbf{\Phi}$ only contains non-zero diagonal entries, we can obtain

$$(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})[i,:] = (\texttt{diag}(\mathbf{\Phi}))^\top \odot \hat{\mathbf{X}}_{\texttt{in}}[i,:], \tag{2}$$

where diag extracts the diagonal entries into a vector from the input square matrix. Hence, if we define the scaling of the λ_i-frequency signal over node p after and before the operator $\mathbf{\Phi}$ as $\texttt{SCALING}(i,p,\mathbf{\Phi}) = \frac{(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})[i,p]}{\hat{\mathbf{X}}_{\texttt{in}}[i,p]}$, from Eq. (2) we have

$$\forall i,p,q, \quad \texttt{SCALING}(i,p,\mathbf{\Phi}) = \texttt{SCALING}(i,q,\mathbf{\Phi}).$$

I.e., for any specific frequency (e.g., λ_i), its scaling over any two nodes (p and q) are equal. In other words, the filter $\mathbf{\Phi}$ works globally over every node. If we expect the filter $\mathbf{\Phi}$ to not work globally, i.e.

$$\exists i,p,q, \quad \texttt{SCALING}(i,p,\mathbf{\Phi}) \neq \texttt{SCALING}(i,q,\mathbf{\Phi}).$$

The above inequality is equivalent to

$$\frac{\sum_{k,k\neq i}\mathbf{\Phi}[i,k]\cdot\hat{\mathbf{X}}_{\texttt{in}}[k,p]}{\hat{\mathbf{X}}_{\texttt{in}}[i,p]} \neq \frac{\sum_{k,k\neq i}\mathbf{\Phi}[i,k]\cdot\hat{\mathbf{X}}_{\texttt{in}}[k,q]}{\hat{\mathbf{X}}_{\texttt{in}}[i,q]}.$$

Assume that $\forall k$, if $k \neq i$, $\mathbf{\Phi}[i,k] = 0$, and then the left-hand side is equal to the right-hand side which leads to a contradiction. Hence, non-zero off-diagonal entries of the full frequency profile $\mathbf{\Phi}$ must exist if we expect the filter to not work globally. Notice that the above definition of scaling (e.g., $\frac{(\mathbf{\Phi}\hat{\mathbf{X}}_{\texttt{in}})[i,p]}{\hat{\mathbf{X}}_{\texttt{in}}[i,p]}$) is not fully aligned with the classic graph filtering [27] but a combination of filtering and modulation as we mentioned in Proposition 1. □

Next, we present a family of GA-MLPs whose spectral expressiveness is limited to a global filter.

Proposition 3. *A family of GA-MLPs are global filters if their full frequency profiles are in the form of $\mathbf{C} = \sum_k a_k\tilde{\mathbf{A}}^k + b\mathbf{I}$ which only contains non-zero diagonal entries.*

Proof. Since $\{\tilde{\mathbf{A}}^k\}$ and $\mathbf{I}$ share the same eigenvectors, the diffusion matrix can be decomposed as

$$\mathbf{C} = \sum_k a_k\tilde{\mathbf{A}}^k + b\mathbf{I} = \mathbf{U}(\sum_k a_k(\mathbf{I}-\tilde{\mathbf{\Lambda}})^k + b\mathbf{I})\mathbf{U}^\top.$$

Hence, the frequency profile is $\mathbf{\Phi} = \sum_k a_k(\mathbf{I}-\tilde{\mathbf{\Lambda}})^k + b\mathbf{I}$ whose off-diagonal entries are zero. □

A wide range of GA-MLPs (e.g., SGC [34], APPNP [16]) follow the above form and therefore cannot modulate graph signal. Unfortunately, even when they are equipped with our proposed ALT-global, they are still *global* filters because ALT-global assigns the same weight to every edge (i.e., $w\tilde{\mathbf{A}}$ and $(1-w)\tilde{\mathbf{A}}$).

3.4 ALT-local: A Local Adaptive Method

In this subsection, we propose a more flexible method based on ALT-global. Our goal is to empower the backbone GNNs with local adaptive signal filtering capabilities, which is an essential property for capturing complex heterophilic connection patterns [32, 47]. According to Proposition 3, we know that if all the edges are assigned with the same weight (e.g., $w\tilde{\mathbf{A}}$) the corresponding full frequency profile will only contain diagonal non-zero entries. Lemma 2 provides a critical clue on how to bring non-zero off-diagonal entries in full frequency profiles.

Lemma 2. *By re-weighting the edge weights non-uniformly (i.e., if re-weighting by $\mathbf{W}\odot\tilde{\mathbf{A}}$, $\exists i,j,k,l, \mathbf{W}[i,j] \neq \mathbf{W}[k,l]$), the off-diagonal entries of $\mathbf{\Phi}$ can be non-zero.*

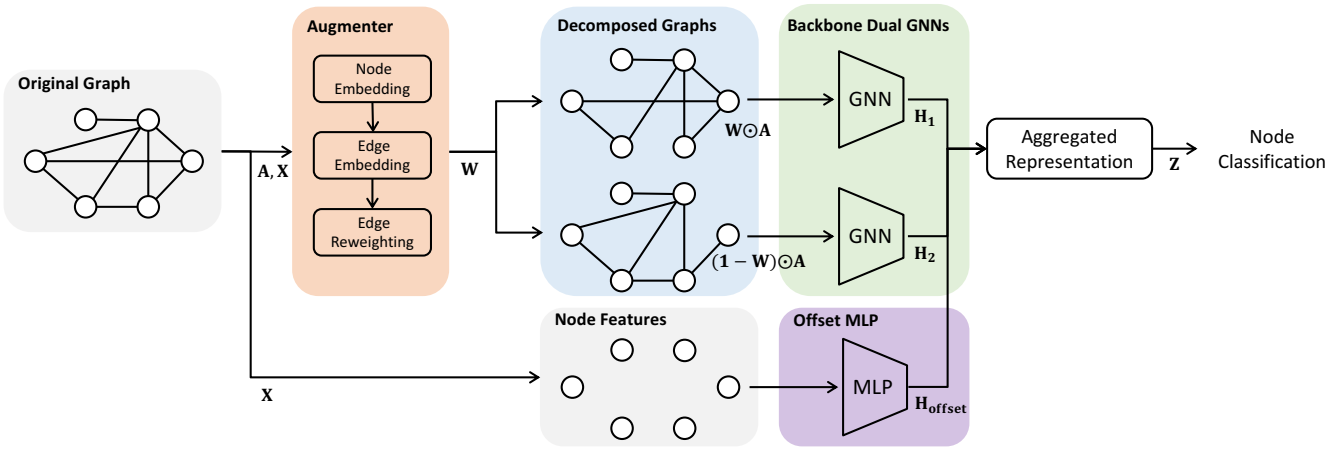


Figure 2: The proposed ALT-local.

Proof. We follow the assumption mentioned in the proof of Lemma 1. The diffusion matrix $\mathbf{C}$ can be decomposed as $\mathbf{C} = \mathbf{U}\mathbf{\Phi}\mathbf{U}^\top$. For the full frequency profile $\mathbf{\Phi}$, its off-diagonal entry $\mathbf{\Phi}[i,j] = \sum_{l,k} \mathbf{U}[l,i]\mathbf{C}[l,k]\mathbf{U}[k,j] = 0, \forall i \neq j$. If we re-weight the diffusion matrix by $\mathbf{W} \odot \mathbf{C}$ such that $\mathbf{W}[l,k] = w_{lk}$ and $\mathbf{W}[i,j] = w \neq w_{lk}, \forall i \neq l \text{ and } j \neq k$. In other words, we start from the most basic case where only one edge (l,k) ($\mathbf{C}[l,k] \neq 0$) is re-weighted by w_{lk} and all the remaining edges are re-weighted as w. Given the zero off-diagonal entries of $\mathbf{\Phi}$ we have

$$\begin{aligned}\mathbf{\Phi}_{\texttt{re-weighted}}[i,j] &= (\mathbf{U}^\top(\mathbf{W}\odot\mathbf{C})\mathbf{U})[i,j] \\ &= (\mathbf{U}^\top(\mathbf{W}\odot\mathbf{C})\mathbf{U})[i,j] - w\mathbf{\Phi}[i,j] \\ &= \mathbf{U}[l,i]\mathbf{C}[l,k]\mathbf{U}[k,j](w_{lk} - w).\end{aligned}$$

It is common that $\mathbf{U}[l,i]\mathbf{U}[k,j] \neq 0$, and thus, as long as $w_{lk} \neq w$, we have $\mathbf{\Phi}_{\texttt{re-weighted}}[i,j] \neq 0$. Therefore, we proved that if the edge weights are re-weighted non-uniformly, the off-diagonal entries of $\mathbf{\Phi}$ can be non-zero, i.e., the GNN can be a local filter. □

Guided by Lemma 2 we modify ALT-global as follows so that the edge weights are different:

$$\mathbf{H}_1 = \texttt{GNN}(\mathbf{W}\odot\mathbf{A}, \mathbf{X}, \theta_1), \tag{3a}$$
$$\mathbf{H}_2 = \texttt{GNN}((\mathbf{1}-\mathbf{W})\odot\mathbf{A}, \mathbf{X}, \theta_2), \tag{3b}$$
$$\mathbf{H}_{\texttt{offset}} = \texttt{MLP}(\mathbf{X}, \theta_3), \tag{3c}$$
$$\mathbf{Z} = \texttt{softmax}(\mathbf{H}_1 - \mathbf{H}_2 + \eta\mathbf{H}_{\texttt{offset}}), \tag{3d}$$

One option is to set $\mathbf{W}$ as a learnable parameter which is prune to overfitting as the number of parameters is equal to the number of edges. Therefore, we parameterize the edge weight $\mathbf{W}$ by an edge augmenter as follows,

$$\mathbf{H} = \texttt{GNN}_{\texttt{aug}}(\mathbf{A}, \mathbf{X}, \phi_1), \tag{4a}$$
$$\mathbf{W}[i,j] = w_{ij} = \texttt{sigmoid}(\texttt{MLP}(\mathbf{H}[i,:]||\mathbf{H}[j,:], \phi_2)) \tag{4b}$$

where ϕ_1 and ϕ_2 are the parameters of the augmenter GNN and a multi-layer perceptron (MLP) respectively. Here we first obtain the node embedding matrix via the augmenter GNN (i.e., $\texttt{GNN}_{\texttt{aug}}$) in Eq. 4a. Then we concatenate node embeddings into edge embeddings (i.e., $\mathbf{H}[i,:]||\mathbf{H}[j,:]$). The edge weight (i.e., w_{ij}) is computed via an MLP with `sigmoid` activation. Naturally, the node embeddings from the augmenter GNN (Eq. 4a) should be as discriminative as possible so that the edge importance can be better measured. Thus, we use a two-layer high-pass filter GNN as the $\texttt{GNN}_{\texttt{aug}}$ whose mathematical formulation is as follows,

$$\texttt{GNN}_{\texttt{aug}}(\mathbf{A}, \mathbf{X}, \phi_1) = \tilde{\mathbf{A}}^2_{\texttt{high}}\texttt{MLP}(\mathbf{X}, \phi_1), \tag{5a}$$
$$\tilde{\mathbf{A}}_{\texttt{high}} = \epsilon\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{5b}$$

where ϵ is a scaling hyper-parameter to adjust the amplitude of the high-pass filter. We name the above model (i.e., Eqs.3a-5b) as ALT-local which is summarized in Figure 2.

Remarks. Our method is partly inspired by FAGCN [3] and we clarify the uniqueness and advantages of our work compared with FAGCN as follows. From the method perspective, FAGCN explicitly mixes high-frequency and low-frequency signals. ALT generalizes this idea to the 'mixture of complementary filters'; thus, even though the backbone GNN's convolution kernel is unknown, ALT can still boost its performance decently, which provides great generality. For the theoretical analysis, [3] analyze the spatial effects of signals with different frequencies. Our analysis takes a solid step forward to reveal the intrinsic connections between (i) the full frequency profile, (ii) graph signal modulation, and (iii) local adaptive filters.

3.5 Training Objective

The optimization objective of ALT is as follows.

$$\phi^*, \theta^* = \arg\min_{\phi} \mathcal{L}_{\texttt{cla}}(g(\mathcal{G}, \phi), \theta, \mathcal{Y}) \tag{6}$$

where the augmenter is denoted as $g(\cdot)$ whose parameter is ϕ and the dual backbone GNNs are parameterized as θ for brevity. Specifically, for the ALT-global, $\theta = \{\theta_1, \theta_2, \theta_3\}$ and $\phi = w$ are from Eq.1a, Eq.1b, and Eq.1c. For ALT-local, $\theta = \{\theta_1, \theta_2, \theta_3\}$ is from Eq. 3a, 3b, and Eq. 3c; $\phi = \{\phi_1, \phi_2\}$ is from Eq. 4a and 4b. $\mathcal{L}_{cla}$ is cross-entropy loss between the classification results (Eq. 1d for ALT-global and Eq. 3d for ALT-local) and the labeled nodes.

If all the feature dimensions of different layers (including the input layers) from different backbone GNNs and MLPs are denoted

Table 1: Performance comparison (mean±std accuracy) on heterophilic graphs. The last column indicates the average performance boosting for a specific backbone GNN over all the datasets.

Backbone	ALT?	Chameleon	Squirrel	Texas	Wisconsin	Cornell	Film	Cornell5	Penn94	Avg. Δ
GCN	No	58.4±1.1	35.4±0.6	57.6±3.5	51.2±1.6	55.9±1.6	28.1±0.3	72.1±0.1	74.7±0.3	+12.4
	Yes	65.8±0.9	52.4±0.8	70.9±4.3	76.4±3.9	73.9±5.1	35.5±1.2	77.5±0.1	80.1±0.4	
SGC	No	58.4±0.6	37.1±0.4	58.6±1.9	48.3±1.8	57.0±3.4	27.3±0.1	72.6±0.4	75.0±0.4	+11.9
	Yes	65.6±2.0	53.2±0.6	71.5±2.8	72.8±1.6	72.1±9.0	34.9±0.8	79.0±0.3	80.2±0.1	
APPNP	No	48.0±1.2	33.8±0.4	59.5±1.1	48.8±2.0	56.3±1.4	28.7±0.3	70.6±0.5	73.4±0.4	+15.1
	Yes	65.4±1.1	53.2±0.9	71.2±2.9	76.6±2.7	78.4±3.4	34.0±0.3	79.7±0.1	81.7±0.4	
GPRGNN	No	59.2±2.5	38.4±0.8	69.1±1.0	72.4±1.6	69.6±2.5	32.1±1.1	74.3±1.3	78.7±0.3	+7.1
	Yes	66.7±0.9	53.0±1.0	75.4±2.4	79.7±0.5	70.6±1.5	32.8±1.0	80.2±0.8	82.3±0.4	
FAGCN	No	54.3±1.9	32.5±1.4	61.5±1.3	56.6±5.2	66.0±1.7	33.8±0.7	69.1±0.2	72.8±0.3	+11.1
	Yes	64.5±1.0	52.8±1.4	69.4±0.7	76.4±5.7	75.1±6.8	35.7±0.5	79.9±0.1	81.9±0.4	
H2GCN	No	49.9±1.4	31.5±0.8	67.6±2.1	70.4±2.1	69.4±3.3	34.5±0.3	69.5±0.4	73.5±0.1	+8.1
	Yes	61.5±0.7	51.6±0.5	76.0±4.7	77.7±4.4	78.4±3.4	35.7±0.3	71.4±0.2	78.7±0.8	

as d and all the models (GNNs and MLPs) contain 2 feature transformation matrices, the number of trainable parameters of ALT-local is composed of three parts: (1) $\texttt{GNN}_{\texttt{aug}}$ ($2d^2$), (2) MLP from Eq. 4b ($2d^2 + d$), (3) $\texttt{GNN}_1$, $\texttt{GNN}_2$, and offset MLP ($3d^2 + 3dc$) where c is the number of classes. In practice, the parameter number is much smaller than the estimated number. For example for datasets whose $d > 500$, empirically, setting the hidden dimension as 32 is enough. However, compared with vanilla backbone GNNs (e.g., a simple GCN [15]), ALT-local inevitably contains more parameters as ALT-local is composed of 3 GNNs and 2 MLPs in total. Even for ALT-global, it is still composed of 2 GNNs and 1 MLP. Hence, the increased number of parameters is a potential limitation of ALT-local and ALT-global.

4 EXPERIMENTS

Experiments in this section aim to answer the following questions.

- How applicable is the proposed ALT to improve the backbone GNNs with arbitrary filtering characteristics?
- How effective can an existing backbone classifier be, after equipping the proposed ALT?

4.1 Experiment setup

Datasets. 16 datasets are used including Cora [40], Citeseer [40], Pubmed [40], DBLP [4], Computers [26], Photos [26], CS [26], Physics [26], Cornell [24], Texas [24], Wisconsin [24], Chameleon [25], Squirrel [25], Film [24], Cornell5 [19], and Penn94 [19]. We obtain all the datasets from pytorch-geometric[1] which are publicly available. In Section 4.3, in order to compare with the state-of-the-art methods, we adopt the dataset split 48/32/20% (training/validation/test) from a recent work ACM-GCN [20]. In the other subsections, to fully test the applicability of ALT, we use the following challenging dataset split: (1) we follow the given dataset split for Cora (8.5/30.5/61.0%), Citeseer (7.4/30.9/61.7%), and Pubmed (3.8/32.1/64.1%); (2) for the remaining datasets, we randomly split them into 20/20/60% (training/validation/test). Detailed statistics of the datasets are presented in Table 5 and Table 6 in Appendix.

Accuracy (ACC) is adopted as the metric and we report the average accuracy with the standard deviation in 10 runs. Detailed settings and reproducibility are provided in Appendix[2].

4.2 Applicability of ALT

As the main goal of this paper is to propose a general solution to handle graphs with arbitrary homophily, this section studies the applicability of the proposed approach, ALT. Specifically, we select 6 representative backbone node classifiers including 3 classic GNNs: GCN [15], SGC [34], and APPNP [16], and 3 adaptive GNNs: GPRGNN [6], FAGCN [3], and H2GCN [48] which use specific designs to tackle graphs with low homophily. We aim to compare the performance improvement of the above backbone classifiers after being equipped with ALT. As ALT-local is more powerful than ALT-global, we mainly show the performance improvement after being equipped with ALT-local (short as ALT). The comparison between ALT-local and ALT-global will be presented in Section 4.4.

We present the performance comparison on heterophilic graphs in Table 1 and have the following observations. First, on the heterophilic graphs, in general, our method ALT can significantly improve the performance of most of the existing GNNs, especially for methods originally not designed for the heterophilic graphs (e.g., GCN, SGC, and APPNP), whose performance, on average, is improved over 10%. Second, over the heterophilic graphs, for adaptive GNNs (e.g., GPRGNN, FAGCN, and H2GCN), their performance improvement is not as significant as low-pass filter GNNs. This is expected since these methods have already dealt with heterophily to some extent. Nonetheless, we still gain 7-11% performance improvements averaged over all 8 heterophilic datasets.

The performance comparison on homophilic graphs is presented in Table 2. We test 48 graph-GNN combinations, out of which, 29 cases show accuracy improvements ≥ 0.5%. It is worth noting

[1]https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html

[2]The code is available at https://github.com/pricexu/ALT .

Table 2: Performance comparison (mean±std accuracy (%)) on homophilic graphs. The last column indicates the average performance boosting for a specific backbone GNN over all the datasets.

Backbone	ALT?	Cora	Citeseer	Pubmed	DBLP	Computers	Photos	CS	Physics	Avg. Δ
GCN	No	81.1±0.3	71.2±0.7	79.0±0.4	83.7±0.1	66.2±1.0	84.1±0.5	88.2±0.2	95.3±0.1	+3.4
	Yes	81.2±0.5	71.4±0.4	79.1±0.9	83.7±0.4	84.1±0.1	88.9±0.6	92.3±0.4	95.6±0.9	
SGC	No	80.8±0.1	71.0±0.2	79.5±0.5	83.8±0.0	69.1±0.4	86.2±0.4	89.7±0.1	95.3±0.0	+2.6
	Yes	80.7±0.4	71.2±0.6	79.5±0.7	83.7±0.0	84.0±0.4	88.8±1.4	92.5±0.3	96.0±0.0	
APPNP	No	82.1±0.1	71.8±0.1	79.8±0.5	83.8±0.2	66.7±1.1	83.4±1.2	87.8±0.1	94.9±0.0	+4.0
	Yes	82.7±0.3	72.1±0.3	79.3±0.2	84.6±0.1	84.6±0.4	88.7±0.3	93.8±0.1	96.4±0.1	
GPRGNN	No	78.6±1.5	68.9±0.9	77.6±0.9	84.4±0.2	85.0±0.5	92.4±0.2	92.3±0.1	95.5±0.4	+1.6
	Yes	83.0±0.4	71.0±0.4	80.3±0.2	85.1±0.2	85.8±0.2	92.9±0.2	93.4±0.2	96.2±0.1	
FAGCN	No	79.0±0.6	72.1±0.5	78.0±1.1	81.1±1.1	74.8±3.4	91.2±0.3	93.0±1.4	95.7±0.3	+1.8
	Yes	79.0±0.5	71.7±0.5	78.3±1.2	82.5±0.3	86.0±0.8	91.5±0.4	93.6±1.1	96.3±0.2	
H2GCN	No	78.9±0.6	70.3±1.0	78.2±1.0	82.4±0.0	75.8±0.3	89.7±0.2	92.5±0.2	96.2±0.1	+2.0
	Yes	79.0±0.4	70.9±0.8	78.0±1.3	82.0±0.4	87.0±0.3	92.0±0.6	94.1±0.2	96.6±0.1	

Table 3: Performance comparison (mean±std accuracy (%)) with the state-of-the-art methods. The best and the second best are bold and underlined, respectively. Results marked "*" are reported from [20] with the same dataset split.

Dataset	*ACM-GCN	BernNet	*LINKX	*ACMII-GCN++	*GloGNN++	ALT-APPNP	ALT-APPNP+
Cornell	85.1±6.1	81.1±8.4	77.8±5.8	86.5±6.7	86.0±5.1	86.8±4.3	**90.4±4.5**
Wisconsin	88.4±3.2	87.3±4.6	75.5±5.7	88.4±3.7	88.0±3.2	**88.9±2.5**	88.6±3.3
Texas	87.8±4.4	82.6±4.9	74.6±8.4	88.4±3.4	84.1±4.9	88.7±3.3	**89.5±2.2**
Film	36.6±0.8	34.2±1.5	36.1±1.6	37.1±1.3	**37.7±1.4**	37.6±0.7	37.3±1.2
Chameleon	69.1±1.9	45.4±1.9	68.4±1.4	74.8±2.2	71.2±1.8	66.7±2.0	**77.0±1.9**
Squirrel	55.2±1.5	33.1±1.4	61.8±1.8	67.4±2.2	57.9±1.8	54.3±1.2	**69.4±1.5**
Cora	87.9±1.0	87.6±0.6	84.6±1.1	88.3±1.0	88.3±1.1	88.1±0.5	**89.6±1.3**
Citeseer	77.3±1.7	76.1±0.3	73.2±1.0	77.1±1.6	77.2±1.8	77.6±1.5	**79.9±1.2**
PubMed	90.0±0.5	86.2±0.3	87.9±0.8	89.7±0.5	89.2±0.4	89.9±0.6	**90.3±0.5**

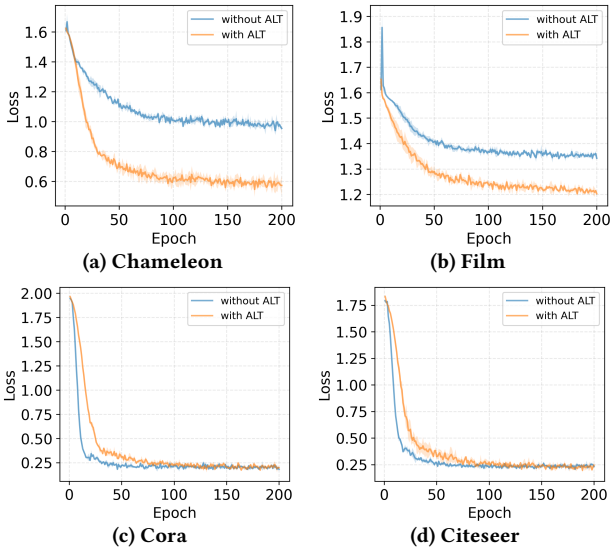


Figure 3: Training losses of APPNP (with/without ALT).

that even though GCN, SGC, and APPNP are designed mainly for homophilic graphs, the proposed ALT is still able to significantly boost their performance on Computers by nearly 18%. Moreover, for each backbone GNN, the average gain of applying the proposed ALT over all 8 homophilic graphs is always positive. Thus, we conclude that ALT can retain or even boost the performance of given backbone GNNs on homophilic graphs.

As we mentioned at the end of Section 3.5, the model equipped with ALT will have more model parameters compared with a vanilla backbone GNN classifier. Thus, we further study the training stability of a backbone classifier when working with ALT. To be specific, we select 2 homophilic datasets Cora and Citeseer, and 2 heterophilic datasets Chameleon and Film. We select the backbone classifier as APPNP. The training loss (negative log-likelihood loss) with respect to the number of epochs is reported in Figure 3a-3d from which we clearly observe that (1) the APPNP's training stability is not significantly affected after equipping ALT, (2) ALT-APPNP can fit the homophilic graphs as good as vanilla APPNP and, importantly, it can fit the heterophilic graphs much better (with lower training loss) than the vanilla APPNP. Observation (2) aligns well with our performance comparison reported in Table 1 and Table 2.

Table 4: Ablation study with different backbone models.

(a) Backbone model: GCN

Backbone	Version	Chameleon	Squirrel	Film	Computers	Photos	CS
GCN	None	58.4±1.1	35.4±0.6	28.1±0.3	66.2±1.0	84.1±0.5	88.2±0.2
	Global	61.3±1.0	44.1±0.3	30.6±0.1	72.7±0.8	85.2±1.4	89.9±0.3
	Local-low	63.3±0.8	48.8±1.2	32.5±0.2	81.1±0.3	86.5±0.9	91.0±0.2
	Local-concat	47.1±2.6	31.3±1.4	34.4±1.1	76.4±5.8	85.3±3.7	87.1±1.2
	Local	**65.8±0.9**	**52.4±0.8**	**35.5±1.2**	**84.1±0.1**	**88.9±0.6**	**92.3±0.4**

(b) Backbone model: SGC

Backbone	Version	Chameleon	Squirrel	Film	Computers	Photos	CS
SGC	None	58.4±0.6	37.1±0.4	27.3±0.1	69.1±0.4	86.2±0.4	89.7±0.1
	Global	59.7±0.8	41.6±0.2	31.4±0.5	71.6±0.4	86.6±0.7	91.1±0.2
	Local-low	61.6±2.3	44.6±0.3	33.3±0.2	79.3±0.6	87.4±0.6	91.5±0.1
	Local-concat	44.0±5.9	36.4±1.7	34.0±1.9	79.6±2.1	88.1±3.0	90.2±0.7
	Local	**65.6±2.0**	**53.2±0.6**	**34.9±0.8**	**84.0±0.4**	**88.8±1.4**	**92.5±0.3**

(c) Backbone model: APPNP

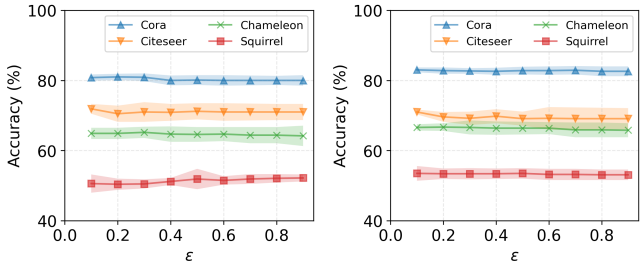
Backbone	Version	Chameleon	Squirrel	Film	Computers	Photos	CS
APPNP	None	48.0±1.2	33.8±0.4	28.7±0.3	66.7±1.1	83.4±1.2	87.8±0.1
	Global	50.8±0.4	36.1±0.7	31.7±0.2	71.5±0.8	85.3±0.9	90.9±0.4
	Local-low	58.8±1.1	48.2±0.8	33.2±1.0	80.1±0.9	87.0±0.6	92.7±0.2
	Local-concat	51.9±0.7	40.0±1.0	33.6±0.7	75.5±2.3	81.8±2.5	89.9±0.4
	Local	**65.4±1.1**	**53.2±0.9**	**34.0±0.3**	**84.6±0.4**	**88.7±0.3**	**93.8±0.1**

4.3 Effectiveness of ALT

In this section, we show that our proposed approach ALT can also be a strong competitor against state-of-the-art methods. We select APPNP [16] as our backbone method which is not designed for graphs with high heterophily. Recent efforts to handle graphs with arbitrary heterophily are selected, which include LINKX [19], BernNet [14], ACM-GCN [20] and GloGNN [18]. For a fair comparison, we adopt the same dataset split as the recent work ACM-GCN [20]. The performance comparison is in Table 3. We observe that ALT-APPNP shows comparable performance against state-of-the-art methods on most of the datasets (except Chameleon and Squirrel). We notice that methods LINKX, ACM-GCN++, and GloGNN++ all use a technique to encode adjacency matrix by an MLP (i.e., $\texttt{MLP}(\mathbf{A})$) as a supplementary of node embeddings. Since this technique is independent of the model design, once it is applied to our framework, the model ALT-APPNP+ achieves very strong performance on Chameleon and Squirrel datasets. In conclusion, our proposed ALT can be comparable to, or stronger than state-of-the-art methods even if it works with a fixed-filter backbone GNN, APPNP.

4.4 Ablation Study

We present an ablation study on datasets: Chameleon [25], Squirrel [25], Film [24], Computers [26], Photos [26], and CS [26]. Specifically, we have the following ablated versions: (1) ALT-local, (2) ALT-local with a low-pass filter augmenter (i.e., change Eq. 5b as a two-layer SGC) named ALT-local-low, (3) ALT-local-concat whose aggregation step (Eq. 3d) is instantiated by 'concatenation' followed by an MLP, (4) ALT-global, and (5) vanilla backbone GNNs without our methods (named as None). Results are presented in Table 4, from which we observe that (1) the ALT-local has consistent advantages over all ablated versions, (2) the variant ALT-local-concat's performance is highly unstable which may be due to its large number of parameters in aggregating representations.

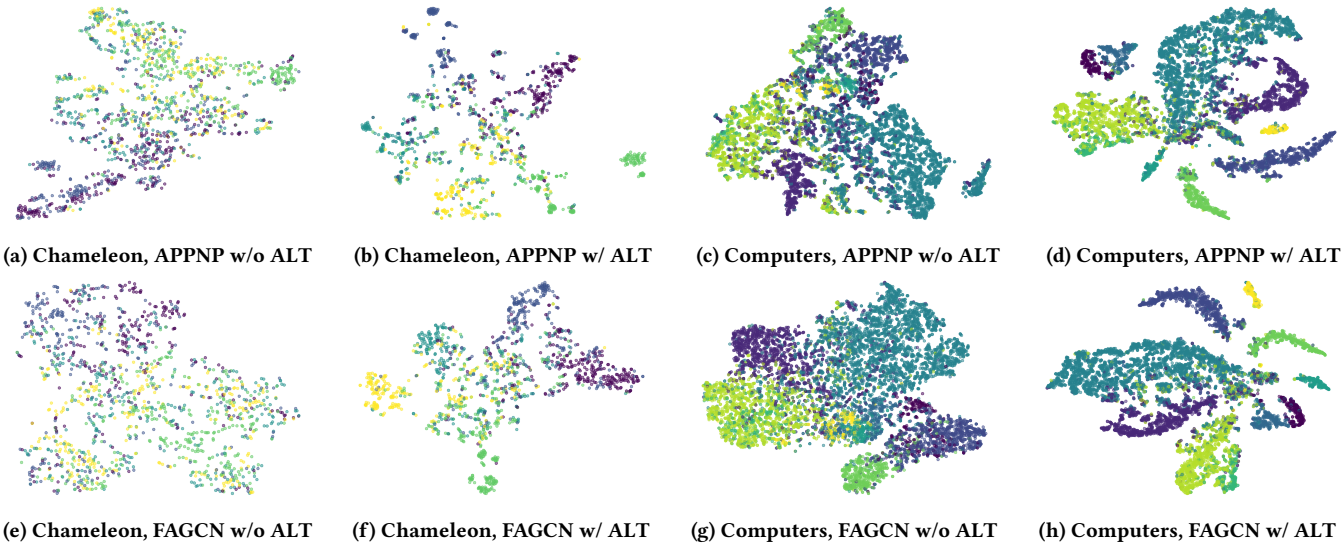


(a) GCN **(b) GPRGNN**

Figure 4: Hyperparameter sensitivity of ALT with backbone GNN as (a) GCN and (b) GPRGNN.

4.5 Hyperparameter Sensitivity Study

We study the sensitivity of ALT-local regarding the amplitude of the augmenter GNN (i.e, ϵ from Eq. 5b). We select GCN [15] and GPRGNN [6] as backbone GNNs and conduct experiments over Cora [40], Citeseer [40], Chameleon [25], Squirrel [25] datasets. Results are presented in Figure 4 from which we observe that the



(a) Chameleon, APPNP w/o ALT (b) Chameleon, APPNP w/ ALT (c) Computers, APPNP w/o ALT (d) Computers, APPNP w/ ALT

(e) Chameleon, FAGCN w/o ALT (f) Chameleon, FAGCN w/ ALT (g) Computers, FAGCN w/o ALT (h) Computers, FAGCN w/ ALT

Figure 5: Visualization of backbone models with/without ALT on datasets Chameleon and Computers.

model performance is stable for the selection of ϵ over four datasets and both backbone GNNs (i.e., GCN and GPRGNN).

4.6 Visualization

As a supplementary study of the model effectiveness, we visualize the node representations from the backbone models APPNP and FAGCN with/without our proposed ALT. To be specific, we use t-SNE [29] to map the representations of test nodes into two-dimensional vectors for visualization. We select a heterophilic graph Chameleon and a homophilic graph Computers. Figure 5a-5h show that after equipping with our proposed ALT (1) clusters of nodes with the same class (i.e., color in our visualization) are more cohesive in the embedding space and (2) backbone GNN's node representations from different classes are more discriminative.

5 RELATED WORK

This section reviews two topics that are closely related to our paper: graph structure learning and learning on heterophilic graphs.

Graph Structure Learning. Graph structure learning aims to modify the given graph structure to improve the performance of downstream tasks [7, 10, 11, 44, 45, 49]. For instance, to boost message propagation, inserting virtual nodes is an effective approach [12, 17]. For topology denoising, dropping some existing edges can improve the model robustness [21, 35] and eliminate redundant information from the input [41]. Another line of research views the given graph as the optimization variable and updates them according to the performance of downstream node classifiers (e.g., LDS [9] and Gasoline [37]). Other works which formulate the given graph as a random variable and infer its optimal parameters include Bayesian GCNN [43], GEN [31], and many more.

Learning on Heterophilic Graphs. Heterophilic graphs are also known as disassortative graphs. Many message-passing-based GNNs suffer from performance degradation on the heterophilic graphs and several approaches have been developed for that. For example, Geom-GCN [24] and H2GCN [48] expand the message-passing mechanism beyond the first-order neighbors. GPRGNN [6] and BernNet [14] reweight the propagation results so that their proposed models are adaptive graph filters. Besides, there are many other works that carefully design the propagation of GNN for heterophilic graphs including CPGNN [47], HOG-GCN [33], GloGNN [18], and WRGAT [28]. Another popular line is to increase the spectral expressiveness of GNN including FAGCN [3], GBK [8], ACM-GCN [20], and DMP [39]; our work falls into this line by proposing a general approach. Interestingly, Luan et al. [20] and Ma et al. [22] both report some cases where high heterophily will not hurt the performance of low-pass filter GNN which reveals under-explored space for this problem. Learning on heterophilic graphs is also related to other topics. For example, [38] reveals the connections between oversmoothing and network heterophily. A comprehensive survey [46] is provided by Zheng et al.

6 CONCLUSION

In this paper, we propose a general framework ALT for the semi-supervised node classification problem on graphs beyond homophily. Our method introduces a novel structure learning-based augmenter to decompose the given graph. After that, most of the existing GNNs can be plugged into our framework. In-depth theoretical analysis shows that our proposed method can adaptively filter and modulate the graph signals which is critical to address complex heterophilic connection patterns. Comprehensive empirical evaluations demonstrate that the proposed ALT obtains significant performance improvement for a wide range of GNN models, on a variety of graph datasets with arbitrary homophily.

ACKNOWLEDGMENTS

ZX, QZ, and HT are partially supported by NSF (1947135, 2134079 and 1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799), DHS (17STQAC00001-06-00), and ARO (W911NF2110088).

REFERENCES

[1] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. 2021. Analyzing the expressive power of graph neural networks in a spectral perspective. In *ICLR*.

[2] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral clustering with graph neural networks for graph pooling. In *ICML*.

[3] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI*.

[4] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *ICLR*.

[5] Lei Chen, Zhengdao Chen, and Joan Bruna. 2021. On Graph Neural Networks versus Graph-Augmented MLPs. In *ICLR*.

[6] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*.

[7] Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. 2022. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter* 24, 2 (2022), 61–77.

[8] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. 2022. GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily. In *TheWebConf*.

[9] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. Learning discrete structures for graph neural networks. In *ICML*.

[10] Dongqi Fu and Jingrui He. 2022. Natural and Artificial Dynamics in Graphs: Concept, Progress, and Future. *Frontiers Big Data* 5 (2022). https://doi.org/10.3389/fdata.2022.1062637

[11] Dongqi Fu, Zhe Xu, Hanghang Tong, and Jingrui He. 2023. Natural and Artificial Dynamics in GNNs: A Tutorial. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, Tat-Seng Chua, Hady W. Lauw, Luo Si, Evimaria Terzi, and Panayiotis Tsaparas (Eds.). ACM, 1252–1255. https://doi.org/10.1145/3539597.3572726

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NeurIPS* (2017).

[14] Mingguo He, Zhewei Wei, Hongteng Xu, et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *NeurIPS* (2021).

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[16] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*.

[17] Junying Li, Deng Cai, and Xiaofei He. 2017. Learning graph-level representation for drug discovery. *arXiv preprint arXiv:1709.03741* (2017).

[18] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. 2022. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. *arXiv preprint arXiv:2205.07308* (2022).

[19] Derek Lim, 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. *NeurIPS* (2021).

[20] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2021. Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification? *arXiv preprint arXiv:2109.05641* (2021).

[21] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. 2021. Learning to drop: Robust graph neural network via topological denoising. In *WSDM*.

[22] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2021. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134* (2021).

[23] Mark Newman. 2018. *Networks*. Oxford university press.

[24] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2019. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.

[25] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

[26] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[27] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[28] Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. 2021. Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns. In *KDD*.

[29] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research* 15, 1 (2014), 3221–3245.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[31] Ruijia Wang, Shuai Mou, Xiao Wang, Wanpeng Xiao, Qi Ju, Chuan Shi, and Xing Xie. 2021. Graph Structure Estimation Neural Networks. In *TheWebConf*.

[32] Tao Wang, Di Jin, Rui Wang, Dongxiao He, and Yuxiao Huang. 2022. Powerful Graph Convolutional Networks with Adaptive Propagation Mechanism for Homophily and Heterophily. In *AAAI*.

[33] Tao Wang, Rui Wang, Di Jin, Dongxiao He, and Yuxiao Huang. 2022. Powerful Graph Convolutioal Networks with Adaptive Propagation Mechanism for Homophily and Heterophily. In *AAAI*.

[34] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *ICML*.

[35] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph Information Bottleneck. *NeurIPS* (2020).

[36] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *ICLR*.

[37] Zhe Xu, Boxin Du, and Hanghang Tong. 2022. Graph sanitation with application to node classification. In *Proceedings of the ACM Web Conference 2022*. 1136–1147.

[38] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2021. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462* (2021).

[39] Liang Yang, Mengzhe Li, Liyang Liu, Chuan Wang, Xiaochun Cao, Yuanfang Guo, et al. 2021. Diverse Message Passing for Attribute with Heterophily. *NeurIPS* (2021).

[40] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *ICML*.

[41] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2020. Graph Information Bottleneck for Subgraph Recognition. In *ICLR*.

[42] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *NeurIPS* (2018).

[43] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. 2019. Bayesian graph convolutional neural networks for semi-supervised classification. In *AAAI*.

[44] Beidi Zhao, Boxin Du, Zhe Xu, Liangyue Li, and Hanghang Tong. 2022. Learning Optimal Propagation for Graph Neural Networks. *arXiv preprint arXiv:2205.02998* (2022).

[45] Tong Zhao, Gang Liu, Stephan Günnemann, and Meng Jiang. 2022. Graph Data Augmentation for Graph Machine Learning: A Survey. *arXiv preprint arXiv:2202.08871* (2022).

[46] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph Neural Networks for Graphs with Heterophily: A Survey. *arXiv preprint arXiv:2202.07082* (2022).

[47] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2021. Graph Neural Networks with Heterophily. In *AAAI*.

[48] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *NeurIPS* (2020).

[49] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. 2021. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036* (2021).

A ANALYSIS OF ALT-GLOBAL FROM THE GRAPH DIFFUSION EQUATION (GDE) PERSPECTIVE

As we claimed in Lemma 1, our proposed ALT-global can be an adaptive filter even if the given backbone GNNs only have fixed filters. Here, we prove this from the Graph Diffusion Equation (GDE) [23] perspective. Our proof will focus on the case where the diffusion matrix is the normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ whose convolution kernel is fixed. Other cases can be proved in similar ways.

Given graph signals $\mathbf{H}$, its diffusion process can be presented as $\mathbf{H}^{(t+1)} = \tilde{\mathbf{A}}\mathbf{H}^{(t)}$. Thus, we have

$$\mathbf{H}^{(t+1)} - \mathbf{H}^{(t)} = \frac{\mathbf{H}^{(t+1)} - \mathbf{H}^{(t)}}{(t+1) - t} = \tilde{\mathbf{A}}\mathbf{H}^{(t)} - \mathbf{H}^{(t)}. \tag{7a}$$

In the GNN case, $t > 0$ denotes the GNN depth and in the GDE context, it denotes the diffusion time. Thus, if we set the time interval as Δt, the graph diffusion dynamics can be presented as follows,

$$\frac{\mathbf{H}^{(t+1)} - \mathbf{H}^{(t)}}{\Delta t} = \tilde{\mathbf{A}}\mathbf{H}^{(t)} - \mathbf{H}^{(t)}, \quad \frac{d\mathbf{H}^{(t)}}{dt} = -\mathbf{L}\mathbf{H}^{(t)}, \tag{8a}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized Laplacian matrix. As ALT-global re-weights all the edges into $w\tilde{\mathbf{A}}$ and $(1-w)\tilde{\mathbf{A}}$, we have

$$\frac{d\mathbf{H}_1^{(t)}}{dt} = w\tilde{\mathbf{A}}\mathbf{H}_1^{(t)} - \mathbf{H}_1^{(t)} = (-w\mathbf{L} - (1-w)\mathbf{I})\mathbf{H}_1^{(t)}, \tag{9a}$$

$$\frac{d\mathbf{H}_2^{(t)}}{dt} = (1-w)\tilde{\mathbf{A}}\mathbf{H}_2^{(t)} - \mathbf{H}_2^{(t)} = (-(1-w)\mathbf{L} - w\mathbf{I})\mathbf{H}_2^{(t)}, \tag{9b}$$

Recap that the prediction matrix of ALT-global is by combining signals from dual backbone GNNs and an offset MLP as $\mathbf{Z} = \texttt{softmax}(\mathbf{H}_1 - \mathbf{H}_2 + \eta\mathbf{H}_{\texttt{offset}})$. We keep the assumption that the dual backbone GNNs are both GA-MLPs [5] which shares parameters with our offset MLP. Thus, we have $\mathbf{H}_1^{(0)} = \mathbf{H}_2^{(0)} = \mathbf{H}_{\texttt{offset}} = \mathbf{H} = \texttt{MLP}(\mathbf{X})$

As we are analyzing its diffusion dynamics, there is no interaction between any two columns of the feature matrix $\mathbf{H}_1^{(t)}$ (and $\mathbf{H}_2^{(t)}$). Hence, for brevity, we only show analysis of a single feature $\mathbf{h}_1^{(t)} = \mathbf{H}_1^{(t)}[:, m]$, $\mathbf{h}_2^{(t)} = \mathbf{H}_2^{(t)}[:, m]$, $\mathbf{h} = \mathbf{h}_{\texttt{offset}} = \mathbf{H}_{\texttt{offset}}[:, m]$, $\mathbf{z}^{(t)} = \mathbf{Z}^{(t)}[:, m]$, $\forall m \in \{1, \ldots, n\}$. The dual GNNs' GDEs can be presented as follows,

$$\frac{d\mathbf{h}_1^{(t)}}{dt} = (-w\mathbf{L} - (1-w)\mathbf{I})\mathbf{h}_1^{(t)}, \tag{10a}$$

$$\frac{d\mathbf{h}_2^{(t)}}{dt} = (-(1-w)\mathbf{L} - w\mathbf{I})\mathbf{h}_2^{(t)}, \tag{10b}$$

Proposition 4. *The solutions of Eq. 10a and Eq. 10b can be presented as* $\mathbf{h}_1^{(t)} = \sum_{i=0}^{n}\left(a_i^{(0)}e^{-(w\lambda_i+(1-w))t}\right)\mathbf{u}_i$ *and* $\mathbf{h}_2^{(t)} = \sum_{i=0}^{n}\left(a_i^{(0)}e^{-((1-w)\lambda_i+w)t}\right)\mathbf{u}_i$*, where* $\mathbf{u}_i$ *and* λ_i *refers to the* i*-th eigenvector and eigenvalue of* $\mathbf{L}$*; initial state* $a_i^{(0)}$ *is determined by* $\mathbf{h}_1^{(0)} = \mathbf{h}_2^{(0)} = \sum_i a_i^{(0)}\mathbf{u}_i$.

Proof. Here we prove the solution of Eq. 10a and for Eq. 10b its solution can be obtained in a similar way. For Eq. 10a, by decomposing the graph signal with the eigenvectors ($\{\mathbf{u}_i\}$) of the normalized Laplacian $\mathbf{L}$ we have:

$$\mathbf{h}_1^{(t)} = \sum_i a_i^{(t)}\mathbf{u}_i. \tag{11}$$

As only $\mathbf{h}$ and a_i are the functions of t, based on the fact that $\mathbf{L}\mathbf{u}_i = \lambda_i\mathbf{u}_i$ and $\mathbf{I}\mathbf{u}_i = \mathbf{u}_i$ we have:

$$\sum_i \left(\frac{da_i^{(t)}}{dt} + w\lambda_i a_i^{(t)} + (1-w)a_i^{(t)}\right)\mathbf{u}_i = 0. \tag{12}$$

As all the eigenvectors are orthogonal with each other, by multiplying both sides of the above equation with $\mathbf{u}_i^\top$ we have

$$\left(\frac{da_i^{(t)}}{dt} + w\lambda_i a_i^{(t)} + (1-w)a_i^{(t)}\right)\mathbf{u}_i = 0. \tag{13}$$

$$\frac{da_i^{(t)}}{dt} + w\lambda_i a_i^{(t)} + (1-w)a_i^{(t)} = 0. \tag{14}$$

Hence, the graph signal $\mathbf{h}_1^{(t)}$ can be represented as

$$\mathbf{h}_1^{(t)} = \sum_{i=0}^{n}\left(a_i^{(0)}e^{-(w\lambda_i+(1-w))t}\right)\mathbf{u}_i. \tag{15a}$$

Similarly, the graph signal $\mathbf{h}_2^{(t)}$ can be presented as

$$\mathbf{h}_2^{(t)} = \sum_{i=0}^{n}\left(a_i^{(0)}e^{-((1-w)\lambda_i+w)t}\right)\mathbf{u}_i. \tag{16a}$$

□

Thus, the aggregated signal can be presented as

$$\mathbf{z}^{(t)} = \mathbf{h}_1^{(t)} - \mathbf{h}_2^{(t)} + \eta\mathbf{h}_{\texttt{offset}}, \tag{17a}$$

$$= \sum_{i=0}^{n} a_i^{(0)}\left(e^{-(w\lambda_i+(1-w))t} - e^{-((1-w)\lambda_i+w)t} + \eta\right)\mathbf{u}_i, \tag{17b}$$

where we use $\mathbf{h}_{\texttt{offset}} = \mathbf{h}^{(0)} = \sum_{i=1}^{n} a_i^{(0)}\mathbf{u}_i$.

According to the graph signal processing [27], $\mathbf{u}_i$ denotes the graph signal with λ_i frequency. Hence, the λ_i-frequency signal amplitude is denoted as $a_i^{(0)}\left(e^{-(w\lambda_i+(1-w))t} - e^{-((1-w)\lambda_i+w)t} + \eta\right)$ after filtered by ALT-global. We know the signal before filtering (i.e., diffusion) is

$$\mathbf{h}^{(0)} = \mathbf{h}_1^{(0)} = \mathbf{h}_2^{(0)} = \mathbf{h}_{\texttt{offset}}^{(0)} = \sum_{i=0}^{n} a_i^{(0)}\mathbf{u}_i, \tag{18}$$

and the amplitude of the the λ_i-frequency signal before filtering is a_i^0. Hence, the filter response to λ_i frequency is

$$\Phi(\lambda_i) = \frac{a_i^{(0)}\left(e^{-(w\lambda_i+(1-w))t} - e^{-((1-w)\lambda_i+w)t} + \eta\right)}{a_i^{(0)}} \tag{19a}$$

$$= e^{-(w\lambda_i+(1-w))t} - e^{-((1-w)\lambda_i+w)t} + \eta \tag{19b}$$

It is clear when $w > 0$,, $\Phi(\lambda_i)$ is a monotonically decreasing function and when $w < 0$, $\Phi(\lambda_i)$ is a monotonically increasing function. With appropriate η and different w, ALT-global can be instantiated as either a low-pass filter or a high-pass filter.

Table 5: Dataset statistics of heterophilic graphs.

	Chameleon	Squirrel	Texas	Wisconsin	Cornell	Film	Cornell5	Penn94
# Nodes	2,277	5,201	183	251	183	7,600	18,660	41,554
# Edges	62,792	396,846	325	515	298	30,019	1,581,554	2,724,458
# Features	2,325	2,089	1,703	1,703	1,703	932	4,735	4,814
# Classes	5	5	5	5	5	5	2	2
$h(\mathcal{G})$	0.231	0.222	0.108	0.196	0.305	0.219	0.479	0.470

Table 6: Dataset Statistics of homophilic graphs.

	Cora	Citeseer	Pubmed	DBLP	Computers	Photos	CS	Physics
# Nodes	2,708	3,327	19,717	17,716	13,752	7,650	18,333	34,493
# Edges	10,556	9,104	88,648	105,734	491,722	238,162	163,788	495,924
# Features	1,433	3,703	500	1,639	767	745	6,805	8,415
# Classes	7	6	3	4	10	8	15	5
$h(\mathcal{G})$	0.810	0.736	0.802	0.828	0.777	0.827	0.808	0.931

Table 7: Updating time (seconds per iteration) with and without ALT.

Backbone	ALT?	Cora	Citeseer	Chameleon	Film
GCN	No	0.0063	0.0034	0.0032	0.0029
	Yes	0.0107	0.0096	0.0094	0.0093
APPNP	No	0.0035	0.0042	0.0034	0.0031
	Yes	0.0090	0.0103	0.0089	0.0092
GPRGNN	No	0.0053	0.0045	0.0054	0.0048
	Yes	0.0121	0.0120	0.0141	0.0136
FAGCN	No	0.0044	0.0040	0.0045	0.0042
	Yes	0.0109	0.0105	0.0124	0.0115

B DATASET STATISTICS

Detailed dataset statistics are presented in Table 5 and Table 6.

C TRAINING TIME STUDY

An analysis of the model complexity is provided at the end of Section 3. Also, ALT does not significantly increase the training epochs, which is illustrated in Figure 3. In this section, we study the training time of a backbone GNN with and without being plugged into ALT. We select 4 datasets (Cora, Citeseer, Chameleon, Film) and 4 backbone GNNs (GCN, APPNP, GPRGNN, FAGCN) to show the updating time comparison per iteration in Table 7, which shows that ALT will increase the time of every training iteration. That is because, from Figure 2, we know the output of the augmenter GNN is the input of the backbone dual GNNs. Thus, according to the chain rule, the update of the augmenter GNN requires a more complex 'gradient computational graph' (and more computations) compared with the update of a vanilla backbone GNN.

D HYPERPARAMETER SETTINGS AND REPRODUCIBILITY

Hardware. We implement ALT in pytorch[3] and pytorch-geometric [4] using one NVIDIA Tesla V100 SXM2-32GB GPU.

Hyperparameters for ALT's augmenter and backbone GNNs. We search the hidden dimension in $\{16, 32\}$, and set the learning rate as 0.05. For all the backbone GNNs, their weight decay is set as 0.0005. For the augmenter GNN, its weight decay is searched in $\{0.005, 0.0005, 0.00005\}$, and its ϵ is searched in {0.2, 0.4, 0.6, 0.8}. Both the augmenter GNN and the offset MLP use ReLU as the activation function. We will release the code upon the publication of the paper.

Hyperparameters for baseline GNNs. For baseline GNNs, their hidden dimension is searched in $\{16, 32\}$, their weight decay is searched in $\{0.005, 0.0005, 0.00005\}$, and their learning rate is set as 0.05. In addition, the propagation step of APPNP is searched between 5 and 10 and the receptive fields' initialization of GPRGNN is searched between the uniform distribution and personalized PageRank distributions as the original paper did.

E LIMITATIONS AND FUTURE WORK

One limitation of backbone GNNs equipped with ALT, as we mentioned in Section 3.5 and Section C, is the increased number of parameters and training time compared with a vanilla backbone GNN. Fortunately, it provides significant performance boosting and does not influence the training stability, as presented in Section 4.2. Besides, our theoretical analysis relies on the assumption that the backbone GNNs are GA-MLPs. Generalizing our theoretical results to a broader range of GNNs is our future work.

[3] https://pytorch.org/

[4] https://pytorch-geometric.readthedocs.io/en/latest/