ASWT-SGNN: Adaptive Spectral Wavelet Transform-Based Self-Supervised Graph Neural Network

Ruyue Liu¹², Rong Yin¹²*, Yong Liu¹, Weiping Wang¹

¹Institute of Information Engineering, Chinese Academy of Sciences
²University of Chinese Academy of Sciences
³Renmin University of China
{liruyue, yinrong, wangweiping}@iie.ac.cn, liuyonggsai@ruc.edu.cn

Abstract
Graph Comparative Learning (GCL) is a self-supervised method that combines the advantages of Graph Convolutional Networks (GCNs) and comparative learning, making it promising for learning node representations. However, the GCN encoders used in these methods rely on the Fourier transform to learn fixed graph representations, which is inherently limited by the uncertainty principle involving spatial and spectral localization trade-offs. To overcome the inflexibility of existing methods and the computationally expensive eigen-decomposition and dense matrix multiplication, this paper proposes an Adaptive Spectral Wavelet Transform-based Self-Supervised Graph Neural Network (ASWT-SGNN). The proposed method employs spectral adaptive polynomials to approximate the filter function and optimize the wavelet using contrast loss. This design enables the creation of local filters in both spectral and spatial domains, allowing flexible aggregation of neighborhood information at various scales and facilitating controlled transformation between local and global information. Compared to existing methods, the proposed approach reduces computational complexity and addresses the limitation of graph convolutional neural networks, which are constrained by graph size and lack flexible control over the neighborhood aspect. Extensive experiments on eight benchmark datasets demonstrate that ASWT-SGNN accurately approximates the filter function in high-density spectral regions, avoiding costly eigen-decomposition. Furthermore, ASWT-SGNN achieves comparable performance to state-of-the-art models in node classification tasks.

Introduction
Graphs are essential in various real-world domains such as social networks, brain networks, transportation networks, and citation networks (Wang et al. 2016; Yü, Lee, and Sohn 2020). Recently, the emergence of graph neural networks (GNNs) (He et al. 2022; Wang et al. 2022b) has attracted much attention due to their success in applications involving graph-structured data such as node classification (Kipf and Welling 2016) and edge prediction (Hasanzadeh et al. 2019). Graph Comparative Learning (GCL) (Velickovic et al. 2019; Sun et al. 2019) combines the capabilities of GNN and comparative learning techniques, making it a promising paradigm in the field of graph analysis (He et al. 2020). Typically, GCL methods generate multiple views by randomly augmenting input data and optimize the GNN encoder by learning consistency across views. GCL reduces the dependence of graph representation learning on human annotations and achieves state-of-the-art performance in tasks such as node classification.

Most GCL methods utilize graph convolutional neural networks (GCNs) as encoders (Xie et al. 2022; Wang et al. 2022b). Similar to convolutional neural networks (CNNs) in computer vision, spectral GCNs use Fourier bases in the design of graph-based operators (Bruna et al. 2013). However, these operators are localized in the frequency rather than the spatial domain. Additionally, they require costly multiplications between eigen-decomposition and dense matrices, leading to high computational expenses. In order to address this issue and achieve spatial localization, methods such as ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) and GCN (Kipf and Welling 2016) employ polynomial approximation. While GCN is widely adopted for graph problems due to its impressive performance and computational efficiency (Shi et al. 2020), it encounters limitations and challenges when applied to large graphs, especially in mini-batch settings (Zeng et al. 2019b). To overcome the scaling challenges of GCN on large graphs, researchers have proposed layer sampling methods (Ying et al. 2018; Kaler et al. 2022) and subgraph sampling methods (Zeng et al. 2019b,a). However, the filter size is determined by the size of the entire graph or the sampled subgraph, which restricts flexibility for inputs of different sizes. Although some flexible spatial methods have been proposed, their aggregators lack learnability and convolutional properties. Consequently, the problem of designing a flexible filter that combines the learnability of spatial methods and the convolutional properties of spectral methods still needs to be solved.

To address the issues of inflexible and unlearnable filters, as well as the limited applicability caused by high computational complexity in existing methods, this paper proposes a novel graph comparative learning paradigm based on the adaptive spectral wavelet transform. More specifically, a fast spectral adaptive approximation method is utilized to estimate the wavelet filter, and contrast loss is employed to optimize the wavelet scale directly. Additionally, the intro-
duction of residual links mitigates over-smoothing during information aggregation. By avoiding the expensive eigen-decomposition of the graph Laplacian operator and enabling localization in both the spectral and spatial domains, this approach effectively overcomes the limitations of graph convolutional neural networks that are constrained by graph size and lack flexibility in controlling neighborhood aspects. In comparison to state-of-the-art methods, this work introduces the following innovations:

- We propose a novel self-supervised graph representation learning method based on sparse graph wavelets that creates localized filters in both the spectral and spatial domains. It reduces the computational complexity and addresses the limitation that graph convolutional neural networks cannot flexibly control the neighborhood aspect.

- Theoretically, we demonstrate that nodes with similar network neighborhoods and features exhibit similar ASWT-SGNN embeddings, providing a performance guarantee for the proposed method.

- Extensive experiments on eight benchmark datasets show that the proposed method reduces approximation errors in high-density spectral components without requiring expensive eigen-decomposition and achieves competitive performance with state-of-the-art models in node classification tasks.

**Related Works**

**Graph Convolutional Neural Network**

Following the success of CNNs in computer vision and natural language processing, researchers have sought to extend CNNs to the graph domain. The key challenge lies in defining the convolution operator for graphs. Graph convolutional neural networks can be broadly categorized into two main approaches: spectral and spatial. Spectral methods employ the graph Fourier transform to transfer signals from the spatial domain to the spectral domain, where convolution operations are performed. Spectral GNN (Bruna et al. 2013) is the first attempt to implement CNNs on graphs. ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) introduces a parameterization method using Chebyshev polynomials for the graph convolution operator for graphs. Graph convolutional neural networks incorporate node diffusion into the graph comparison framework. GCA (Zhu et al. 2021) learns node representations by considering other nodes as negative samples, while BGRL (Thakoor et al. 2021) proposes a no-negative-sample model. CCA-SSG (Zhang et al. 2021) optimizes feature-level objectives in addition to instance-level differences. GRADE (Wang et al. 2022b) investigates fairness differences in comparative learning and proposes a novel approach to graph enhancement. Several surveys (Xie et al. 2022; Ding et al. 2022; Kumar, Rawat, and Chauhan 2022) summarize recent advancements in graph contrastive learning. Despite these methods’ notable achievements, most rely on GCN and its variants as the base models, inheriting the limitations of GCN. These limitations restrict the performance of these graph contrastive learning methods in tasks that require preserving fine-grained node features.

**Graph Wavelets**

The wavelet transform exhibits favorable structural properties by utilizing finite length and attenuation basis functions. This approach effectively localizes signals within both the spatial and spectral domains. Additionally, it is noteworthy that the basis and its inverse in the wavelet transform often showcase sparsity, contributing to its utility and efficiency. To construct the wavelet transform on graphs, Hammond et al. (Hammond, Vandergheynst, and Gribonval 2011) propose a method that approximates the wavelet using Chebyshev polynomials. This approach effectively avoids the need for eigen-decomposition of the Laplace matrix. Building upon this, GWNN (Xu et al. 2019) redefines graph convolution based on graph wavelets, resulting in high efficiency and sparsity. M-GWNC (Behmanesh et al. 2022) applies the multi-scale graph wavelet transform to learn representations of multimodal data. Collectively, these works showcase the value of graph wavelets in signal processing on graphs. However, these methods primarily employ wavelet transforms in supervised or semi-supervised tasks and heavily rely on labeled data.

**Preliminary**

**Graph Fourier Transform**

Let $G = (\mathcal{V}, \mathcal{E})$ denotes a graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ represents the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. $G$ is associated with a feature matrix $X \in \mathbb{R}^{N \times p}$, $X = [x_1, \ldots, x_n]$ and an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $A_{ij} = 0$ otherwise. We define the Laplacian matrix of the graph as $L = D - A$, where $D = \text{Diag}(d_1, \ldots, d_N)$, $d_i = \sum_j A_{ij}$. The symmetric normalized Laplacian is defined as $L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = U \Lambda U^\top$. $U = [u_1, \ldots, u_N]$, where $u_i \in \mathbb{R}^N$ denotes the $i$-th eigenvector of $L_{sym}$ and $\Lambda = \text{Diag}(\lambda_1, \ldots, \lambda_N)$ is the eigenvalue matrix.

For a discrete graph signal $X$, its Fourier transform has the following form (Shuman et al. 2013):

$$\hat{f}(\lambda) = \sum_{i=1}^{N} x_i U_i^\top(i) = \sum_{i=1}^{N} x_i U_i^\top(i),$$

(1)
where \( U^* \) denotes the conjugate transposition of the eigenvalues \( \lambda \) corresponding to the eigenvectors. Since the complex number is not involved in the scope of this work, it can be regarded as a common transposition, that is, \( U^* = U^T \).

Graph Fourier transform provides a means to define the graph convolution operator using the convolution theorem. The filter signal can be mathematically expressed as:

\[
f_o = U g(\Lambda) U^T X,
\]

where \( g(\Lambda) \) denotes the filter function on the eigenvalues, and \( U g(\Lambda) U^T \) is the graph filtering matrix.

**Uncertainty Principle**

The implementation of graph convolution using the Fourier transform lacks spatial localization despite its ability to achieve localization in the spectral domain. Additionally, its localization is significantly influenced by computational efficiency. We employ the spatial and spectral concentration metric proposed by Tsitsvero et al. (Tsitsvero, Barbarossa, and Di Lorenzo 2016) to establish a more comprehensive and precise notion of localization.

\[
\left\| Bx \right\|_2^2 = a^2, \quad \left\| Cx \right\|_2^2 = b^2,
\]

where the square of the Euclidean paradigm \( \| x \|_2^2 \) denotes the total energy of the signal \( x \). \( B \) is the diagonal matrix representing the node restriction operator. Given a node subset \( S \subseteq V \), \( B = \text{Diag} \{I(\ell \in S)\} \), where \( I(\cdot) \) is an indicator function. \( C = U \Sigma_F U^{-1} \) is the band-limiting operator. Given a matrix \( U \) and a subset of frequency indices \( F \subseteq V^* \), where \( V^* = \{1, \cdots, N\} \) denotes the set of all frequency indices. \( \Sigma_F \) is a diagonal matrix defined as \( \Sigma_F = \text{Diag} \{I(\ell \in F)\} \).

More specifically, considering a pair of vertex sets \( S \) and frequency sets \( F \), where \( a^2 \) and \( b^2 \) represent the percentage of energy contained within the sets \( S \) and \( F \) respectively, our objective is to determine the balance between \( a \) and \( b \) and identify signals that can achieve all feasible pairs. The resulting uncertainty principle is formulated and presented in the theorem (Tsitsvero, Barbarossa, and Di Lorenzo 2016).

\[
\cos^{-1} a + \cos^{-1} b \geq \cos^{-1} \lambda_{\text{max}}(BC),
\]

where \( \lambda_{\text{max}}(BC) \) is the maximum eigenvalue of \( BC \).

Considering the uncertainty principle, we retain the entire frequency band if we follow the graph localization method in Eq. (2). However, real graph signals show a non-uniform distribution in the frequency domain, which suggests that \( C \) can be chosen more efficiently. Moreover, a deeper network leads to a wider propagation of the signal in the graph domain, which limits the width of the frequency bands and leads to the dominance of low-frequency signals (smoothing signals), which produce over-smoothing.

**Graph Wavelet Transform**

As the graph Fourier transform, the graph wavelet transform also necessitates a set of suitable bases to map the graph signal to the spectral domain. In this case, we denote the wavelet operator as \( \Psi_s = U g_s(\Lambda) U^T \), where \( s \) is the scaling parameter. The wavelet transform breaks down a function \( f \) into a linear combination of basis functions localized in spatial and spectral. This paper employs the Heat Kernel wavelet as the low-pass filter (denoted as \( g^l \)). In contrast, the Mexican-Hat wavelet is the band-pass filter (denoted as \( g^b \)). These filters are defined as follows:

\[
g^l_s(\lambda) = e^{-s\lambda},\]

\[
g^b_s(\lambda) = \frac{2}{\sqrt{3}\pi} \left( 1 - (\lambda s)^2 \right) e^{-\lambda s^2}.
\]

In this work, we integrate filter functions to achieve the combined effect of a low-pass filter and a wide band-pass filter. Figure 1 exemplifies how the combination of low-pass and band-pass filters can generate a more intricate wavelet filter capable of capturing signal components at various scales.
scales. This design enables us to obtain a richer representation of the signal, encompassing its frequency information more nuancely.

\[ g_0(\lambda) = g_{s_0}^l(\lambda) + \sum_{l=1}^L g_{s_l}^l(\lambda), \]  

(7)

where \( \theta = \{ s_0, s_1, \ldots, s_l \} \) is the set of scale parameters.

Using the graph wavelet transform instead of the graph Fourier transform in Eq. (2), we get the graph convolution as follows:

\[ f_o = \Psi G \Psi^T X, \]  

(8)

where \( G \) is a diagonal matrix, which acts as a filter, the scale set \( \theta \) of wavelet coefficients is omitted for simplification.

**Methodology**

**Wavelet Coefficients Approximation**

The model formulation discussed in the preceding sections necessitates an eigen-decomposition of the Laplace operator of the input graph \( \mathcal{G} \). However, this process presents a computational complexity of \( O(N^3) \), rendering it infeasible for larger graphs. In order to overcome this constraint, we employ the polynomial approximation method previously introduced by Hammond et al. (Hammond, Vandergheynst, and Gribonval 2011). This method involves expressing the wavelet filter as 

\[ g_0(\lambda) \approx p_0(\lambda) = \gamma_0 + \gamma_1 \lambda + \cdots + \gamma_m \lambda^m. \]  

This allows rewriting the wavelet operator as \( \Psi_\theta = U p_0(\Lambda) U^T = p_0(\mathbf{L}_{\text{sym}}) \).

While existing methods rely on Chebyshev polynomial approximations, we aim to optimize scales. Therefore, we utilize a least squares approximation that can parameterize the set of wavelet scales \( \theta \), which can be expressed as follows:

\[ \gamma_\theta = (V_\Lambda^T V_\Lambda)^{-1} V_\Lambda^T g_0(\Lambda), \]  

(9)

where \( V_\Lambda \in \mathbb{R}^{N \times (m+1)} \) is the Vandermonde matrix of \( \Lambda \) from order 0 to order \( m \), \( N \) is the number of eigenvalues.

Accurate calculation of the eigenvalues requires an expensive eigen-decomposition of the graph Laplace operator, which is not feasible in our case. As an alternative, we transform the set of eigenvalues into a sequence of linearly spaced points \( \xi = \{ \xi_i \}_{i=1}^K \) within the interval \([0, 2]\) on the spectral domain. However, in graphs that exhibit multiscale features, the eigenvalues do not follow a uniform distribution on the spectral domain. Instead, they display spectral gaps corresponding to different scales within the data. To address this non-uniform distribution, we incorporate the estimated spectral density \( \omega \) as the weight for each of the \( K \) sample points \( \xi \) on the spectral domain (Fan et al. 2020). Consequently, we can compute the weighted least squares coefficients:

\[ \gamma_\theta = (V_\xi^T \text{Diag}(\omega) V_\xi)^{-1} V_\xi^T \text{Diag}(\omega) g_0(\xi), \]  

(10)

where the spectral density \( \omega = \{ \omega_j \}_{j=1}^K \), and \( \omega_i = \frac{1}{N} \sum_{j=1}^N \{ \mathbb{I}(\lambda_j = \xi_i) \} \).

The goal of spectral density estimation is to approximate the density function without expensive graph Laplacian eigen-decomposition. To achieve this, we determine the number of eigenvalues less than or equal to each \( \xi_i \) in the set \( \xi \). It can be achieved by computing an approximation to the trace of the eigen-projection matrix \( P \) (Di Napoli, Polizzi, and Saad 2016). In practice, directly obtaining the projector \( P \) is often not feasible. However, it can be approximated efficiently using polynomials or rational functions of the Laplacian operator \( L \). In this approximation, we interpret \( P \) as a step function of \( L \), which can be expressed as follows:

\[ P_{\xi_i} = h(L), \text{ where } h(\lambda) = \begin{cases} 1, & \text{if } \lambda \leq \xi_i \\ 0, & \text{otherwise}. \end{cases} \]  

(11)

Although it is impossible to compute \( h(\lambda) \) exactly, it can be approximated using a finite sum of Jackson-Chebyshev polynomials, denoted as \( \phi(\lambda) \). Please refer to Appendix A (Liu et al. 2023) for detailed information on the approximation method. In this form, it becomes possible to estimate the trace of \( P \) by an estimator developed by Hutchinson (Hutchinson 1989) and further improved more recently (Tang and Saad 2012). Hutchinson’s stochastic estimator relies solely on matrix-vector products to approximate the matrix trajectory. The key idea is to utilize Rademacher random variables, where each entry of a randomly generated vector \( R \in \mathbb{R}^N \) takes on the values \(-1 \) and \( 1 \) with equal probability of \( \frac{1}{2} \). Thus, an estimate of the trace \( tr(P) \) can be obtained by generating \( n_r \) samples of random vectors \( R_k, k = 1, \ldots, n_r \), and computing the average over these samples. The estimator can be expressed as follows:

\[ tr(P) \approx \frac{1}{n_r} \sum_{k=1}^{n_r} R_k^T P R_k \approx \frac{1}{n_r} \sum_{k=1}^{n_r} R_k^T \phi(\mathbf{L}_{\text{sym}}) R_k. \]  

(12)

We obtain the approximation \( \Omega \) to the cumulative spectral density function.

\[ \Omega = \left\{ \left( \xi_i, \frac{1}{N} \sum_{k=1}^{n_r} R_k^T \phi(\mathbf{L}_{\text{sym}}) R_k \right) \right\}_{i=1}^K. \]  

(13)

Finally, the cumulative spectral density \( \Omega \) is differentiated to obtain an approximation of the spectral density \( \omega \), i.e. \( \omega = \frac{d}{d\xi} \Omega \). Using the estimated spectral density \( \omega \) as a weight for each sample point \( \xi_i \) on the spectral domain, the weighted least squares coefficient \( \gamma_\theta \) can be calculated by substituting it into Eq. (10).

These coefficients are used to approximate the wavelet filter matrix \( \Psi_\theta \), which can be expressed as follows:

\[ \Psi_\theta = \gamma_0 I + \gamma_1 \mathbf{L}_{\text{sym}} + \cdots + \gamma_m \mathbf{L}_{\text{sym}}^m. \]  

(14)

**Encoder**

In this paper, we construct a graph wavelet multilayer convolutional network (ASWT-SGNN). Based on the above, we define the \( l \)-th layer of ASWT-SGNN as

\[ H^{l+1} = \sigma(H^{lT} W^l), \]  

(15)

where \( \sigma \) is the activation function, \( H^{lT} \in \mathbb{R}^{N \times p} \) is the results of graph convolution of the \( l \)-th layer, \( W^l \in \mathbb{R}^{p \times q} \) is
the weight of the $l$-th layer, $p$ is the number of features in current layer, and $q$ is the number of features in next layer. $H^l$ in Eq. (15) is described as follows:

$$H^l = \alpha F^l H^l + (1 - \alpha) H^l,$$

where $1 - \alpha$ represents the proportion of the original features $H^l$ in $l$-th layer, and $0 \leq \alpha \leq 1$. By incorporating this residual connection, we guarantee that regardless of the number of layers we add, the resulting representation will always retain a portion of the initial features. To enhance the encoder’s capability to aggregate both local and global information effectively, we define the diffusion operator $F^l$ of $l$-th layer in Eq. (16) as

$$F^l = \beta (\Psi_\theta G^l \Psi_\theta^T) + (1 - \beta) (D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}}),$$

where $\beta$ stands for the ratio of the graph wavelet term and $0 \leq \beta \leq 1$, $\theta$ is the learnable multi-scale parameters. $G^l$ is the learnable diagonal filter matrix of $l$-th layer. $\tilde{A} = A + I$ represents the adjacency matrix of the self-loop graph of $\mathcal{G}$, where $I$ is the identity matrix.

Optimization Objective

Typical GCL methods involve generating augmented views and subsequently optimizing the congruence between their encoded representations. In this paper, we generate two augmented graphs, $z$ and $o$, by using feature augmentation. We randomly sample the mask vector $m_f \in \{0, 1\}^B$ to hide part of the dimensions in the node feature. Each element in mask $m_f$ is sampled from Bernoulli distribution $Ber(1 - f_d)$, where the hyperparameter $f_d$ is the feature descent rate. Therefore, the augmented node feature $X$ calculated by the following formula:

$$X = [x_1 \circ m_f, x_2 \circ m_f, \ldots, x_N \circ m_f].$$

We use the comparison objective for the node representation of the two graph augmentation obtained. For node $v_a$, the node representations from different graph augmentation $z_a$ and $o_a$ form a positive pair. In contrast, the node representations of other nodes in the two graph augmentation are considered negative pairs. Therefore, we define the paired objective of each positive pair $(z_a, o_a)$ as

$$L_a(z, o) = \log \frac{e^{\theta(z_a, o_a)}}{e^{\theta(z_a, o_a)} + \sum_{b \neq a} e^{\theta(z_a, o_b)} + e^{\theta(z_a, z_b)}},$$

where the critic $\theta(z, o)$ is defined as $\text{sim}(H_z, H_o)$, and $\text{sim}(\cdot, \cdot)$ refers to cosine similarity function. $H_z$ is the graph embedding generated by graph augmentation through the proposed method ASWT-SGNN. The overall objective to be maximized is the average of all positive pairs, i.e., $L = -\frac{1}{N^2} \sum_{a=1}^N \left[ L_a(z, o) + L_a(o, z) \right]$. 

Complexity Analysis

The first step of ASWT-SGNN is wavelet coefficients approximation. Eq. (8) shows that the immediate solution requires eigen-decomposition of the Laplace matrix to obtain the eigenvalues and eigenvectors of the matrix. However, the complexity of the direct solution is very high. For example, the time complexity of the quick response (QR) algorithm is $O(N^3)$, and the space complexity is $O(N^2)$. Therefore, we use the least squares approximation to approximate the solution in this step. If the $m$-order Chebyshev polynomial approximation is used, the sum of the complexity of each item in the polynomial is calculated as $O(m \times |E|)$.

The second step of ASWT-SGNN is to use the wavelet transform for graph convolution. Spectral CNN (Bruna et al. 2013) has high parameter complexity $O(N \times p \times q)$. ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) approximates the convolution kernel by the polynomial function of the diagonal matrix of the graph wavelet term, reducing the parameter complexity to $O(m \times p \times q)$, where $m$ is the order of the polynomial function. GCN (Kipf and Welling 2016) simplifies ChebyNet by setting $m=1$. In this paper, the feature transformation is performed first, and the parameter complexity is $O(p \times q)$. Then the graph convolution is performed, and the parameter complexity is $O(N)$.

Theoretical Analysis

**Lemma 1.** Consider nodes $v_a$ and $v_b$ within a graph, characterized by their similarity in features or labels. Consequently, their $k$-hop neighbors exhibit a one-to-one mapping, specifically $N_k(b) = \pi(N_k(a))$. In this context, it holds true that $|\Psi_{am} - \Psi_{bm}| \leq 2\epsilon$, where $\Psi_{am}$ signifies the wavelet coefficient between nodes $v_a$ and $v_b$.

**Theorem 1.** If the graph following the above assumption and Lemma 1, the expectation of embedding is given by:

$$\mathbb{E} \left[ F_i \right] = W \mathbb{E}_{y \sim P(y_i), x \sim P_x(a)} [\Gamma_i x],$$

where $\Gamma = \Psi G \Psi^T$, $\Gamma_i$ is the $i$-th row of $\Gamma$. In this context, we simplify the model by excluding the residual connection component. With probability at least $1 - \delta$ over the distribution for the graph, we have:

$$\|H_i - \mathbb{E}[H_i]\|_2 \leq \frac{\sigma^2_{\max}(W)p \log(2p/\delta)}{2N\|\Gamma_i x\|_{\psi_2}},$$

where the sub-gaussian $\|\Gamma_i x\|_{\psi_2} = \min \|\Gamma_i x_i, d\|_{\psi_2}$, $p$ is the dimension of features, $d \in [1, p]$ and $\sigma^2_{\max}(W)$ is the largest singular value of $W$.

**Proof.** The proof is shown in Appendix A (Liu et al. 2023).

The theorem stated above indicates that the proposed method can map nodes with the same label to an area centered around the expectation in the embedding space. This holds true for any graph in which each node’s feature and neighborhood pattern are sampled from distributions that depend on the node label.

**Experiments**

**Experimental Setting**

**Datasets** We evaluate the approach on eight benchmark datasets, which have been widely used in GCL methods. Specifically, citation datasets include Cora, CiteSeer
and PubMed (Yang, Cohen, and Salakhudinov 2016), co-purchase and co-author datasets include Photo, Computers, CS and Physics (Suresh et al. 2021). Wikipedia dataset includes WikiCS (Mernyei and Cangea 2020).

**Baselines** We consider several baseline methods for the node classification task. These include semi-supervised learning methods like GCN (Kipf and Welling 2016), GAT (Velickovic et al. 2017) and GCNII (Chen et al. 2020) and wavelet neural network GWNN (Xu et al. 2019). Furthermore, we evaluate six GCL methods and two graph generation learning methods, which are GMI (Peng et al. 2020), MVGRL (Hassani and Khasahmadi 2020), GCA-SSG (Zhang et al. 2021), GRADE (Wang et al. 2022b), AF-GCL (Wang et al. 2022a), MA-GCL (Gong, Yang, and Shi 2023), GraphMAE (Hou et al. 2022), and MaskGAE (Li et al. 2023). These methods represent state-of-the-art approaches in the field of node classification tasks.

**Evaluation protocol.** For the ASWT-SGNN model, node representations are learned unsupervised using a 2-layer model. Following that, a linear classifier is applied as a post-processing step for assessment. The dataset is randomly partitioned, with 20% of nodes allocated to the training set, another 20% to the validation set, and the remaining 60% to the test set. To ensure the robustness of our findings, we conducted the experiments five times for each dataset, each time with different random seeds. The results include both the average accuracy and the corresponding standard deviation. All experiments use PyTorch on a server with four NVIDIA A40 GPUs. ASWT-SGNN utilizes the Adam Optimizer with a learning rate of 0.001. The specific hyperparameters are as follows: the number of sampling points in the spectral domain, $K$, is set to 20, the feature update ratio, $\alpha$, is set to 0.8, and the wavelet terms ratio, $\beta$, is set to 0.4.

**Wavelet Operator Approximation Experiments**

The proposed method of approximating wavelet operators is applied to real-world graph data. We evaluate its performance using two distinct metrics: the similarity between the approximated wavelet filter $g_\theta$ and the precisely computed wavelet filter, and the Mean Absolute Error (MAE) between the approximated wavelet operator $\Psi_\theta$ and the actual wavelet operator. Figure 2 presents a specific example showing the accurate and approximate multiscale filters, computed precisely and approximately, demonstrating a significant overlap between them. The MAEs for different scaled training sets are depicted in Figure 3. The experimental results indicate that the proposed wavelet operator approximation method reduces the approximation error in the high-density spectral domain, while avoiding the need for computationally expensive eigen-decomposition.

**Node Classification**

Table 1 displays the results of node classification accuracy. Notably, our proposed ASWT-SGNN achieves state-of-the-art (SOTA) performance on six out of the eight graphical benchmarks. Specifically, compared to other self-supervised methods across the eight datasets, ASWT-SGNN outperforms the best self-supervised method, MaskGAE, by an average of 0.8%, and it outperforms the worst self-supervised...
The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Figure 3: MAE between the approximated and the actual wavelet operator at the eigenvalues (left), MAE between the predicted labels and the actual labels of the nodes on the Cora dataset (right).

Figure 4: Ablation studies. The degree of over-smoothing under various parameter settings.

Method, GML, by 3.2%. Furthermore, experiments are carried out within a semi-supervised setting, revealing that the proposed method consistently outperforms the best semi-supervised benchmark, GCNII, by an average margin of 3.1%. Additionally, it surpasses the best self-supervised benchmark, MaskGAE, by an average of 2.0%. These results further demonstrate the effectiveness of the proposed method ASWT-SGNN in node classification tasks.

Ablation Studies

The proposed method incorporates two pivotal hyperparameters: $\alpha$ and $\beta$. By adjusting these parameters, ASWT-SGNN can be simplified to its core forms: when $\alpha$ is set to 1 and $\beta$ is set to 0, ASWT-SGNN aligns with GCN; setting both $\alpha$ and $\beta$ to 1 makes ASWT-SGNN behave similarly to GWNN (Xu et al. 2019); and when $\alpha \neq 1$ and $\beta$ is set to 0, ASWT-SGNN exhibits similarities to GCNII (Chen et al. 2020). We comprehensively compare ASWT-SGNN and transformation models, including GCN, GWNN, and GCNII. Clear observations can be made from Figure 4: as the number of layers increases, the performance of GCN and GWNN significantly declines. In contrast, the performance of GCNII and ASWT-SGNN remains relatively stable even with more layers stacked. Notably, due to the incorporation of graph wavelet bases, ASWT-SGNN model outperforms GCNII in semi-supervised node classification tasks.

Other Experiments

To comprehensively investigate the impact of parameters $\alpha$ and $\beta$ on model performance, we systematically vary their values from 0 to 1. The results are presented in Figure 5, illustrating several noticeable trends. When $\alpha$ is set to 0, indicating the exclusion of residual connections, accuracy is significantly decreased. On the other hand, larger values of $\alpha$ result in fixed node representations, leading to lower performance. $\beta$ represents the proportion of the graph wavelet base; if it is tiny, it may not effectively extract local information. Conversely, excessively large values of $\beta$ could result in the neglect of global information, ultimately reducing performance. These findings highlight the complex relationship between parameters $\alpha$ and $\beta$ in shaping the model’s performance. This delicate balance allows our model to synthesize local and global information, achieving optimal performance effectively.

Furthermore, we use t-SNE (Van der Maaten and Hinton 2008) to visualize the node embeddings. Figure 6 illustrates that compared to MA-GCL, ASWT-SGNN exhibits more pronounced gaps between different classes. This suggests that ASWL-SGNN captures more detailed class information and clarifies the boundaries between samples from different classes. Further extensive experimental analyses, including sparse, robustness, and uncertainty analyses, are exhaustively presented in Appendix B (Liu et al. 2023).

Conclusion

This paper introduces an adaptive graph wavelet self-supervised neural network called ASWT-SGNN. By utilizing multiple wavelet scales, the model integrates different levels of localization on the graph, enabling the capture of elements beyond the low-frequency ones. To avoid the expensive eigen-decomposition in the spectral domain, the model employs a polynomial approximation of the wavelet operator. Comprehensive experimental results demonstrate the competitiveness of the proposed method against state-of-the-art GCL models on real graph datasets. As our framework applies to all message-passing GNNs and polynomial graph filters, we plan to extend its application to more intricate graph neural architectures. Moreover, we will also consider larger datasets with the increase in GPU resources.
Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No.62106259, No.62076234), Beijing Outstanding Young Scientist Program (NO.BJWZYJH012019100020098), and Beijing Natural Science Foundation (No. 4222029).

References


