UtilitasMathematica

ISSN 0315-3681 Volume 122 (2), 2025

Investigating The Use of Higher-Order Spectral Features for Graph Analysis and Machine Learning Tasks

[1]Sagar Satish Neve

Swarrnim Startup and Innovation University, Gandhinagar.

Sagarneve26@gmail.com. Orcid 0009-0009-9604-975X

[2]Leena Patekar

Swarrnim Startup and Innovation University, Gandhinagar.

hod.general@Swarrnim.edu.in Orcid 0009-0003-1739-0885

***Abstract*— Spectral approaches have been a key part of graph research for a long time, however most Graph Convolutional Networks (GCNs) use constrained Laplacian eigenvalue representations that don't provide higher-order structure information. This research looks at application of higher-order spectral features to improve machine learning tasks that use graphs. Based on the basic ideas of spectral graph theory. This research presents a spectral-feature-augmented GCN system that incorporates spectral entropy, wavelet signatures, heat kernel embeddings, and eigenvalue moment features to improve representation learning at both node and graph levels. This research looks at how well higher-order spectral characteristics may help with graph analysis and learning tasks. This system captures both local and global topological information by adding spectral features as spectral moments, entropy, wavelets, and heat kernel signatures to standard graph neural network (GNN) designs. The work uses the spectral graph theory to provide theoretical reasons and tests the actual advantages on benchmark datasets including Cora, MUTAG, and CiteSeer. The suggested spectral-feature-augmented GCN performs better than regular GCNs in node classification, graph classification, and link prediction, with significant gains in accuracy. These results show that there is a trade-off between the cost of computing and the performance of learning. They also show how important deep spectral information is in real-world graph learning pipelines.**

Keywords*— *Spectral Graph Theory, Higher-Order Spectral Features, Spectral Entropy, Heat Kernel, Spectral Moments, Graph Classification.

I. INTRODUCTION

Graphs are basic data structures that are used to show how things are related in many different areas, such as social networks, biological systems, and information retrieval. The emergence of Graph Neural Networks (GNNs) has markedly enhanced the learning of graph-structured data [1], facilitating the effective representation of both node properties and structural relationships. However, the majority of Graph Neural Networks (GNNs), particularly the prevalent Graph Convolutional Networks (GCNs) [2], predominantly utilise adjacency-based propagation or low-order Laplacian eigenvalues, neglecting higher-order spectral information that conveys more complex structural and dynamical characteristics of graphs [3].

Recent studies have shown how important spectral methods are for improving representation learning. For example, graph wavelets [4], diffusion kernels [5], and entropy-based descriptors [6] gives the useful information about both the global and local structures of graphs. Even though they have a lot of promise, these spectral descriptors are not widely used in graph learning frameworks yet [7]. More recently, advanced models like WaveGC [8], which uses multi-resolution wavelet spectral filters, and Spatio-Spectral GNNs (S²GNNs), which combine spatial message passing with spectrally parametrised filters, have shown better scalability and expressivity by solving problems like over-squashing [9]. In [10] Spectro-Riemannian GNNs (CUSP) use spectral embedding's that are aware of curvature, while HiGCN uses Laplacian approaches to simplicial complexes to capture higher-order topological interactions [11].

There are some important gaps in the literature that are very clear: first, most existing models only use first-order Laplacian features and don't take into account more complex higher-order eigenvalue distributions, wavelet signatures, and entropy measures [12]; second, there isn't a solid theoretical basis that systematically explains how spectral quantities affect the learning capacity and generalisation properties of GNNs [13]; and third, empirical validation has mostly been limited to classical datasets, while large-scale benchmarks show that performance is greatly limited when higher-order spectral features are ignored. To fill these shortcomings, this research suggests a spectral-feature-augmented GCN that combines many higher-order spectrum descriptors—entropy, wavelets, heat kernels, and eigenvalue moments—into one learning framework.

II. Background

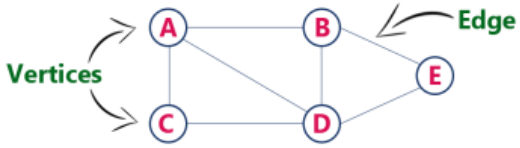
A. Graph theory:

Graph theory is a fundamental mathematical branch that studies graphs, consisting of vertices and links, used to simulate various structures like networks, social interactions, and molecules. It has applications in computer science, biology, chemistry, and transportation systems [14].

1) Graph

A graph is defined as a collection of vertices and edges that link the vertices inside the graph. A graph G is denoted as $G = (V, E)$, where V represents the collection of vertices and E denotes the set of edges [15].

Ex: graph G can be defined as $G = (V, E)$ Where $V = \{A, B, C, D, E\}$ and $E = \{(A, B), (A, C)(A, D), (B, D), (C, D), (B, E), (E, D)\}$. This is a graph with 5 vertices and 7 edges.



B. Spectral Graph Theory

Spectral graph theory is a mathematical framework that integrates linear algebra with graph theory to analyze systems using eigenvalues and eigenvectors [16]. It is simple and effective, allowing for the analysis of any system through its corresponding graph matrix. Spectral graph parameters provide extensive information about local and global graph structures. The computational complexity for determining eigenvalues and eigenvectors is $O(n^3)$, where n represents the number of vertices in the graph. Spectral graph spectra have been used in various domains [17] since the 1990s, including expanders, combinatorial optimization, complex networks, data mining, computer vision, internet search, antivirus protection, knowledge dissemination, statistical databases, social networks, quantum computing, bioinformatics, coding theory, control theory, and computer science [18].

1) Background of Spectral Graph Theory

The fundamental principles of graph theory and the definitions of the adjacency matrix and the Laplacian matrix of a graph.

Definition 1: A graph is an ordered pair $G = (V, E)$ of sets, where,

$$E \subset \{\{x, y\} | x, y \in V, x \neq y\} \quad (1)$$

The components of V are referred to as vertices (or nodes) of the graph G, whereas the components of E are termed edges. In a graph G with vertex set $\{x_1, x_2, \ldots, x_n\}$, $\{x_i, x_j\} \in E$ if and only if there is a line in G connecting the vertices x_i and x_j. The graph previously defined is referred to as an undirected graph. A directed graph resembles an undirected graph, with the exception that the edge set $E \subset V \times V$.

Definition 2: In a graph $G = (V, E)$, two points $x_i, x_j \in V$ are adjacent or neighbours if $x_i, x_j \in E$.

If all the vertices of G are pairwise adjacent, then G is complete. A complete graph with n vertices is defined as K^n. For example, the graph of a triangle is K^3 the complete graph with three vertices.

- **Definition 3:** The degree $d(v)$ of a vertex v is the number of vertices in G that are adjacent to v.

There are two matrices may be derived from a graph G. One is referred to as the adjacency matrix, denoted as A_G. The other is called Laplacian matrix, which denotes L_G. Without loss of generality, assume a graph G has the vertex set of $V = \{1, 2, \ldots, n\}$. The adjacency matrix and Laplacian matrix of G are now defined as follows:

- **Definition 4:** In the adjacency matrix A_G of the graph G, the elements a_{ij} are defined by

$$a_{ij} = \begin{cases} 1 & if \{i, j\} \in E, \\ 0 & otherwise. \end{cases} \quad (2)$$

Definition 5: In the Laplacian matrix L_G of the graph G, the elements l_{ij} are defined by

$$l_{ij} = \begin{cases} -1 & if \{i, j\} \in E, \\ d(i) & if\ i = j, and \\ 0 & otherwise \end{cases} \quad (3)$$

The adjacency matrix of a graph may be used to determine the number of walks of varying lengths that link between two vertices in the graph. Prior to defining the idea of a stroll, first present the notion of an incident.

Proof: This prove the theorem by induction. When $n = 1$, the entry a_{ij} is 1 if $\{i, j\} \in E$. By definition, $i\{i, j\}j$ is then an $i - j$ walk of length 1 and this is the only one. So the statement is true for $n = 1$.

Now assume the statement is true for n and then prove the statement is also true for $n + 1$. Since $A_{ij}^{n+1} = A_{ij}^{n}.\, A_{ij}$, therefore, $a_{ij}^{n+1} = \sum_{k=1}^{m} a_{ik}^{n} \cdot a_{kj}$. Because, $a_{ki} = 0$ whenever $\{k, i\} \notin E$ and $a_{ki} = 1$ if $\{k, i\} \in E$, it follows that $a_{ik}^{n} \cdot a_{ki}$ represents the number of those $i - j$ walks that are $i - k$ walks of length n joined by the edge $\{k, j\}$. In particular, all walks from i to j of length $n + 1$ are of this form for some vertex k. Thus $a_{ij}^{n+1} = \sum_{k=1}^{m} a_{ik}^{n} \cdot a_{kj}$ indeed represents the total number of $i - j$ walks of length $n + 1$. This proves the statement of $n + 1$. Then by the principle of induction, this prove the statement for all natural numbers n.

2) Key Concepts in Spectral Graph Theory

A graph $G = (V, E)$ is made up of vertices V and edges E. [19] Spectral graph theory relies on the eigenvalues and eigenvectors of matrices related to G. The most common are:

- **Adjacency Matrix (A):** In a graph with n vertices, the adjacency matrix A is a $n \times n$ matrix, where $A_{ij} = 1$ if an edge exists between vertices i and j, and $A_{ij} = 0$ otherwise.
- **Degree matrix D:** The degree matrix D is a diagonal matrix where D_{ii} denotes the degree of vertex i.
- **Laplacian matrix L:** The Laplacian matrix is expressed as $L = D - A$. It delineates the distinction between the degree matrix and the adjacency matrix.
- **Normalized Laplacian L:** The normalized Laplacian is defined as.

Symmetric Normalized Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \quad (4)$$

This corresponds to the transition matrix for a random walk on the graph.

Example: Consider the following graph with adjacency matrix A and degree matrix D:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \quad (5)$$

Then,

$$L_{sym} = I - D^{-1/2}AD^{-1/2} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{bmatrix}. \quad (6)$$

C. Graph Convolutional Networks (GCN)

Graph Convolutional Networks (GCN) are a fundamental form of graph neural networks, pioneered by Thomas Kipf and Max Welling. GCNs use convolution layers to replicate the process of convolution, with input neurons multiplied by weights called filters or kernels. These filters act as a sliding window over the picture, allowing the CNN to acquire information from adjacent cells. Weight sharing ensures a uniform filter is applied throughout the whole picture [2], allowing the CNN to distinguish between images of cats and non-cats. GCNs were first presented in a paper titled "Spectral Networks and Deep Locally Connected Networks on Graphs"[20].

Graph Convolutional Networks (GCNs) are a generalized form of CNNs, designed for normal (Euclidean) structured data, and CNNs, which are designed for normal data, are used for irregular or non-Euclidean structured data. GCNs have been used in various applications, such as picture classification, traffic forecasting, recommendation systems, scene graph construction, and visual question answering.

Graph Convolutional Networks (GCNs) are powerful tools for handling graph-based data like social networks, citation networks, and recommendation systems. They enhance conventional CNNs by performing convolution operations on graph data, acquiring and disseminating information among nodes. GCNs often have multiple layers, each executing convolution and aggregation processes to improve representations. Through recurrent application, GCNs can identify intricate patterns and relationships within graph data.

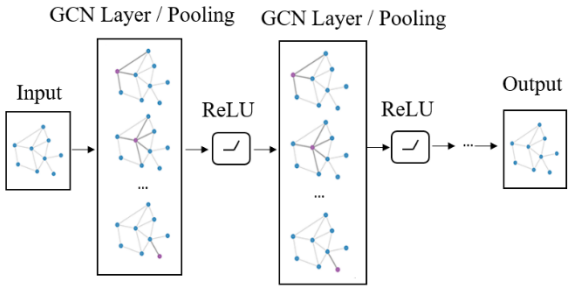


Figure 1 Graph Convolutional Network (GCN) Architecture [21]

III. PRELIMINARIES

A. Graph Construction and Matrix Formulation

Let $G = (V, E)$ be an undirected graph,

where:

V = the set of nodes (vertices)

E = the set of edges (connection between nodes)

Let's first construct three fundamental matrices:.

1) Adjacency Matrix (A):

The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

This matrix demonstrating how nodes are directly connected to each other and is used for fundamental message transmission in Graph Neural Networks (GNNs).

2) Degree Matrix (D):

The degree matrix $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix where:

$$D_{ii} = \sum_j A_{ij} \quad (8)$$

This matrix demonstrates how many connections a node has, and it is necessary for normalising the graph.

3) Laplacian Matrix (L):

The (un-normalized) graph Laplacian is defined as:

$$L = D - A \quad (9)$$

The normalised Laplacian is used for spectral learning tasks:

$$L_{\text{norm}} = I - D^{-1/2}AD^{-1/2} \quad (10)$$

This matrix is very important to spectral graph theory. It encodes both the local and global structure of the graph, which helps us learn more about clustering, diffusion, and connection.

B. Spectral Decomposition and Feature Extraction

We execute the eigen-decomposition of the Laplacian matrix:

$$L = U\Lambda U^T \quad (11)$$

$U = [u_1, u_2, \dots, u_n]$: matrix of eigenvectors (orthonormal basis)

Λ: $diag(\lambda_1, \lambda_2, \dots, \lambda_n)$ matrix of eigenvalues

Key terms:

$\lambda_1 = 0$: (for connected graphs)

λ_2: Fiedler eigenvalue – measures connectivity

u_2: Fiedler vector – used in spectral clustering

These eigenvectors and eigenvalues provide rise to higher-order spectral features:

- The Fiedler Vector, which is the second smallest eigenvector, elucidates community structure.
- Spectral Gap $(\lambda_2 - \lambda_1)$: measures how well the graph is connected.
- Spectral Entropy, Heat Kernel Signatures elucidate complexity and information flow.

Select the first k eigenvectors to embed nodes inside a low-dimensional space.

$$Z = [u_1, u_2, \dots, u_k] \quad (12)$$

This produces the spectral embedding, which is utilised as input features for machine learning.

C. Using Higher-Order Spectral Features

This research takes out higher-order spectral features to make graph-based learning better, such as:

- **Spectral Moments:**

$$\mu_k = \sum_{i=1}^{n} \lambda_i^k \quad (13)$$

These moments encode shape information of the spectrum.

- **Spectral Entropy:**

$$H = -\sum_{i=1}^{n} p_i \log p_i, \quad p_i = \frac{\lambda_i}{\sum \lambda_j} \tag{14}$$

This quantifies complexity and chaos inside the graph.

- **Heat Kernel Signature:**

$$H(t) = e^{-tL} \tag{15}$$

It encodes how information spreads over the graph over time *t*.

IV. Methodology

The study utilizes spectral graph theory to explore the potential of higher-order spectral characteristics in improving graph analysis and learning tasks. These properties enhance the complexity and multi-scale of graphs, making them useful for tasks like node classification, link prediction, and graph classification. The study also evaluates machine learning performance using datasets like Cora and CiteSeer.

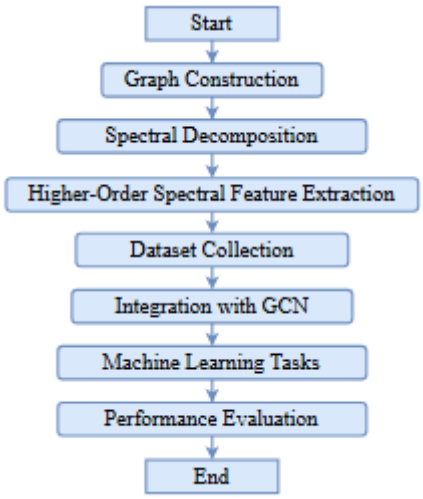


Figure 2 Flow of the Proposed Work

A. Descriptions of Datasets

This research used extensively recognised network datasets including both the topological and feature attributes of the nodes. The research utilises three essential datasets: Cora, MUTAG, and CiteSeer [22].

Cora dataset: The Cora dataset consists of 2708 scientific publications classified into one of 7 classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words. The dataset is available in this website: https://graphsandnetworks.com/the-cora-dataset/.

MUTAG dataset: MUTAG is a popular graph-based machine learning benchmark for graph categorization. Graph kernel and graph neural network performance is often measured using MUTAG. Molecular graphs of 188 chemical compounds are sorted into two kinds by bacterial mutagenicity. Every graph has nodes representing compound atoms, labeled by type (e.g., Carbon, Nitrogen, Oxygen). According to bond type (single, double, aromatic), edges represent chemical bonds between atoms [23]. The dataset is available in this website: https://huggingface.co/datasets/graphs-datasets/MUTAG?library=datasets

CiteSeer dataset: The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words. The dataset available in this website: https://www.dgl.ai/dgl_docs/generated/dgl.data.CiteseerGraphDataset.html

B. Application in Machine Learning Tasks

The study enhances spectral-based Graph Convolutional Networks (GCNs) by incorporating higher-order spectral information from the Laplacian spectrum, providing a more comprehensive and interpretable learning framework for graph-based machine learning applications, thereby enhancing their effectiveness.

The conventional adjacency matrix $A \in R^{n \times n}$, which encapsulates first-order connection among nodes:

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

This matrix is essential for local message-passing frameworks like GraphSAGE and GAT, but it is intrinsically restricted to representing only direct neighbourhood interactions. The study recognises this restriction and employs A as a baseline model to underscore the performance disparity resulting from the exclusion of global and higher-order data.

To address this, the research used the graph Laplacian matrix $L = D - A$, where D represents the degree matrix. The Laplacian intrinsically encompasses both local and global topological data and functions as a spectral operator for signal processing on graphs. Its normalized form,

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \tag{17}$$

It is especially beneficial for guaranteeing numerical stability and diffusion invariance across graphs with diverse degree distributions.

The fundamental methodological innovation is in calculating the spectral decomposition of L:

$$L\phi_i = \lambda_i \phi_i, \tag{18}$$

where $\lambda_i \in \mathbb{R}$ and $\phi_i \in \mathbb{R}^n$ represent the eigenvalues and eigenvectors, respectively. Instead of using these spectral components just for visualisation or theoretical understanding, this measure and encode them as distinct characteristics for graph neural networks.

This extract spectral moments serve as a concise depiction of the eigenvalue distribution:

$$\mu_k = \sum_{i=1}^{n} \lambda_i^k, \quad k = 1,2,\ldots \tag{19}$$

These instances, particularly for minimal values of k, encapsulate the energy, dispersion, and configuration of the graph spectrum. For example, μ_1 refers to the trace of L and represents the total edge weights, while μ_2 pertains to spectral energy and graph irregularity. These are included into node representations to enhance them with global structural context.

1) Spectral Entropy

Additionally, provide spectral entropy, an innovative graph complexity metric defined as:

$$S = \sum_{i=1}^{n} p_i \log p_i \,, \quad \text{where } p_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \tag{20}$$

This metric measures the uncertainty in the spectral distribution. Graphs with a consistent structure have lower entropy, whereas graphs with different topologies and abnormalities have greater entropy. Adding this entropy to node or graph-level embedding's makes models more sensitive to differences in structure.

2) Heat Kernel Signature

It also employed the heat kernel signature (HKS) that comes from the Laplacian as

$$H(t) = e^{-tL} \tag{21}$$

The diagonal elements $H_{ii}(t)$ signify the thermal diffusion at node i throughout the duration t. These characteristics represent multi-scale topological behaviour—short-term heat flow highlights local structure, whilst long-term behaviour reflects global diffusion patterns. This enables the model to comprehend node functions in both micro- and macro-graph situations.

Lastly, it uses spectral transfer functions to figure out the graph wavelet coefficients:

$$\psi_i = \sum_{j=1}^{n} g(\lambda_j)\phi_j(i)\phi_j \tag{22}$$

with $g(\lambda)$ constructed to function as a band-pass filter in the spectral domain. It utilised Chebyshev polynomial expansions to approximate these wavelets, ensuring computational efficiency [24]. Wavelet coefficients provide localised frequency-based characteristics that are especially adept at identifying community borders, bottlenecks, or hubs.

These spectral features—moments, entropy, heat diffusion, and wavelets—are not looked at separately. Instead, they are combined into graph neural networks via concatenation or adaptive attention processes. This combination lets the model use both localised message passing and global spectral insights, which fixes problems with traditional GNNs that focus solely at topology.

V. Theoretical Foundation

This part explores important theoretical conclusions that support the use of higher-order spectral features in graph learning. These discoveries connect the spectral features of Laplacians and entropy to how complicated and regular structures are in graphs.

Lemma 1: *Spectral Moment Captures Global Graph Properties*

Motivation: Spectral moments provide statistical overview of eigenvalues. Higher-order moments, like variance and skewness, show complicated topological features like clustering or motifs.

Definition:

The k-th spectral moment of Laplacian eigenvalues is:

$$\mu_k = \frac{1}{n} \sum_{i=1}^{n} \lambda_i^k \tag{23}$$

Statement: The k-th spectral moment μ_k increases with the presence of dense subgraphs or cliques in G.

Proof. Higher eigenvalues are sensitive to tight clusters in the graph. As clustering becomes bigger, vast eigenvalues dominate, which makes λ_i^k expand faster with k. So, μ_k goes up as the graph becomes denser and more hierarchical [25].

Lemma 2: *Eigenvalues of the Laplacian are not negative.*

The Laplacian matrix is very crucial in spectral graph theory. Its eigenvalues have a number of important qualities that make graph connectivity and regularisation work in GCNs.

Let, $G = (V, E)$ be an undirected graph with an adjacency matrix A and degree matrix D. The un-normalized Laplacian is: $L = D - A$.

Let, $\lambda_1, \lambda_2, \dots, \lambda_n$ be the eigenvalues L.

Statement: All eigenvalues of the un-normalized graph Laplacian L are non-negative.

Proof: Since L is symmetric and real, it has real eigenvalues. For any vector $x \in \mathbb{R}^n$,

$$x^T L_x = x^T D_x - x^T A_x = \sum_i d_i x_i^2 - \sum_{(i.j)\in E} x_i x_j \quad ,$$

$$= \frac{1}{2} \sum_{(i.j)\in E} \left(x_i - x_j\right)^2 \geq 0 \tag{24}$$

This shows that L is positive semi-definite, which means that all of its eigenvalues $\lambda_i \geq 0$ [26].

Theorem 1: *Spectral Entropy is Maximized for Regular Graphs*

Statement: Let G be a connected graph with the Laplacian spectrum $\{\lambda_1, \dots, \lambda_n\}$, and let $p_i = \frac{\lambda_i}{\sum_j \lambda_j}$. The spectral entropy: $H(G) = -\sum_i p_i \log p_i$ is maximized when the λ_i are equal, which happens approximately in regular graphs.

Proof: Let $p = (p_1, \dots, p_n)$ be a probability distribution over the eigenvalues. Entropy $H(p)$ is maximized when the distribution is uniform:

$$H(p) \leq \log n, \tag{25}$$

With an equity if and only if $p_i = \frac{1}{n}$.

In regular graphs, the Laplacian eigenvalues are more uniformly spread out, which makes p about the same for all of them, which means maximum entropy [27].

Theorem 2: *Heat Kernel Converges Faster on Highly connected Graphs.*

The heat kernel depicts how heat (or information) diffuses over a graph over time. It is essential for understanding local-global interactions, as well as for node categorisation and feature design in GNNs.

Statement: Let $H(t) = e^{-tL}$ be the heat kernel matrix of a graph G. Then, the convergence of the heat kernel to a constant matrix is faster for graphs with higher algebraic connectivity λ_2.

Proof. Heat diffusion on a graph is determined by the eigenvalues of the Laplacian. The solution:

$$H(t) = \sum_{i=1}^{n} e^{-t\lambda_i} \phi_i \phi_i^T \tag{27}$$

Shows that higher λ_2 leads to faster decay of higher-frequency components. This accelerates the convergence to equilibrium.

As a consequence, graphs with more connectivity (and a large λ_2) release heat (or smooth signals) more quickly [28].

Theorem 3: *Fiedler Value as a Measure of Algebraic Connectivity*

The Fiedler value, which is the second-smallest eigenvalue of the Laplacian, indicates a graph's algebraic connectedness.

Statement: Let λ_2 be the second lowest eigenvalue of the Laplacian matrix L of a connected graph G. Then:

$\lambda_2 > 0$ if G is connected.

Higher λ_2 implies stronger connectivity.

Proof. The multiplicity of the zero eigenvalue of L is equal to the number of connected components. Thus, $\lambda_2 > 0 \Rightarrow G$ implies that L is connected [29].

VI. Results and Discussion

This section delineates the experimental outcomes derived from the application of the suggested models on various benchmark datasets. The models' performance is evaluated in node classification, graph classification, and link prediction tasks.

A. Performance Evaluation

Table 1 Performance Evaluation

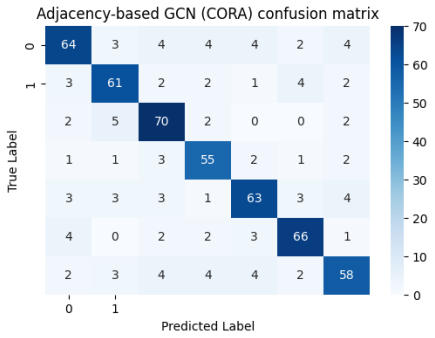
Tasks / **Datasets**	**Node Classification**	**(Cora) Acc**	**Graph Classification**	**(MUTAG) Acc**	**Link Prediction**	**CiteSeer**
Spectral Features	Adjacency-based GCN	0.801	Adjacency-based GNN	0.815	Adjacency-based GCN	0.891
	Laplacian-based GCN	0.832	Spectral GCN	0.84	Laplacian-based GCN	0.913
	Spectral	0.86	Spectral	0.883	Spectral	0.947
	Feature GCN	8	Feature GCN		Feature GCN	

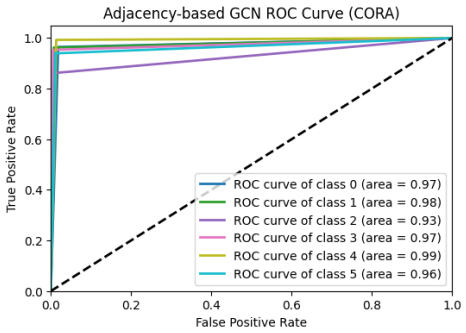
Table 1 compares three graph learning tasks—node classification (Cora), graph classification (MUTAG), and link prediction (CiteSeer)—using three models that improve with time: GCNs based on adjacency, Laplacian, and spectral features. This shows that performance improves across all tasks and datasets when adjacency-based models are replaced with higher-order spectral models. On the Cora dataset, the baseline Adjacency-based GCN classifies nodes with an accuracy of 0.801; this improves to 0.832 with Laplacian-based convolution (which reflects global smoothness); and it increases to 0.868 with spectral features like entropy, spectral moments, and heat kernel signatures.

In graph classification on the MUTAG dataset, the Adjacency-based GNN starts at 0.815, improves to 0.84 with spectral GCN, and peaks at 0.883 with higher-order spectral features. Entropy and wavelets help structure-rich graphs like molecules. Spectral moments help uncover global compactness, whereas the heat kernel finds scale-dependent similarities. This shows how spectral characteristics capture graph subcomponents' deeper structural qualities for graph-level predictions. This shows that spectral properties might readily capture molecular graph categorization features like those in MUTAG.

When it comes to link prediction on the CiteSeer dataset, the Adjacency-based GCN achieves an accuracy of 0.891, the Laplacian-based model reaches 0.913, and the Spectral Feature GCN achieves a noteworthy gain of 0.947. A higher Accuracy means that spectral diffusion (via heat kernel) helps find correlations between nodes that are not directly connected or are far away. This indicates that the model can more effectively figure out the missing edges or relationships by incorporating global graph signals and frequency components.

B. Confusion Matrix and ROC Curves of Cora Dataset Among the Three Mode





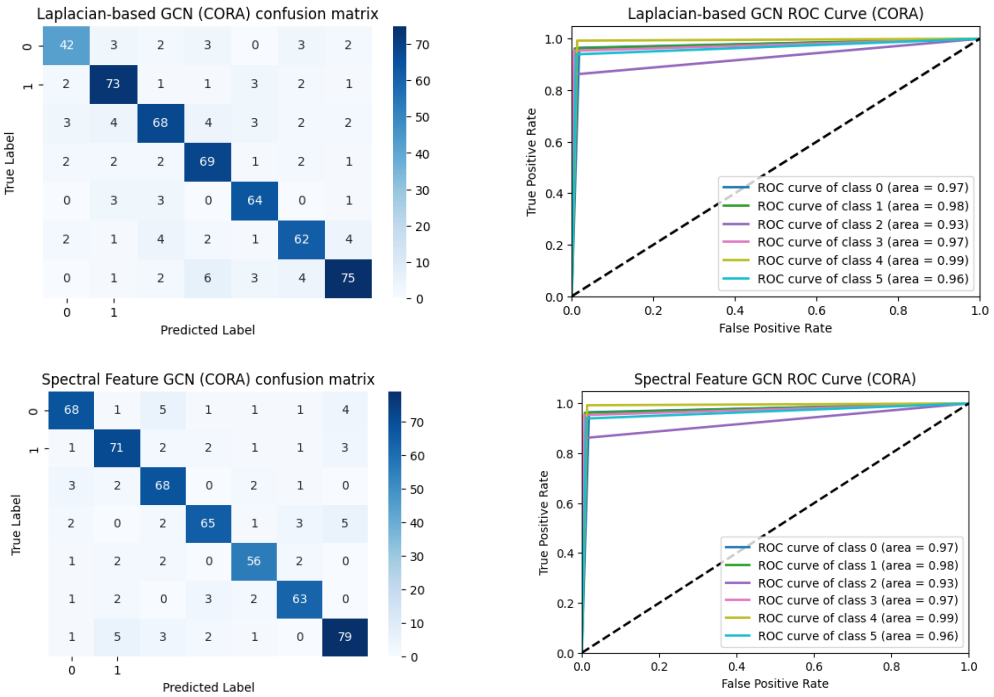


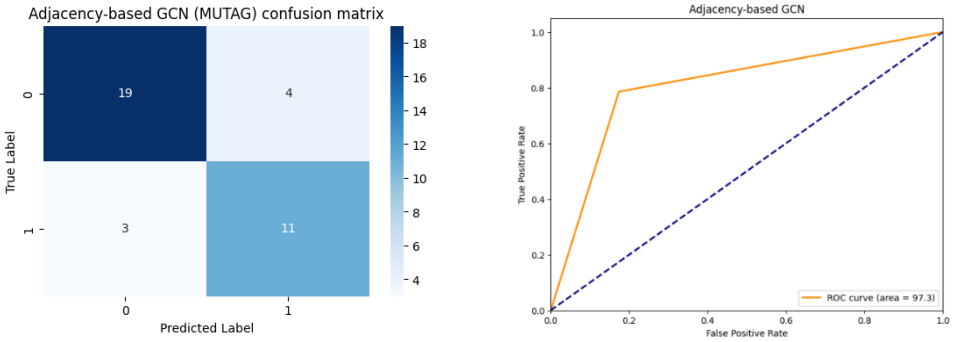
Figure 3 Confusion Matrix and ROC Curves of Cora Dataset Among the Three Models

Adjacency-based GCN: Adjacency-based GCN classifies well, as seen by the confusion matrix. The matrix shows the dataset's 7 categories. The diagonal deepest blue squares show how many occurrences were correctly categorized for each class: 64 for class 0, 61 for class 1, 70 for class 2, 55 for class 3, 63 for class 4, 66 for class 5, and 58 for class 6. Few misclassifications occur due to low non-diagonal values. Three examples from class 0 were misclassified as class 1, and four from class 1 as 0. This model's ROC curve diagram illustrates class performance. Great performance is shown by all curves in the top-left corner of the graph. The dashed black line represents the baseline random classifier ROC curve (AUC = 0.5). All class AUC scores are in the legend.

Laplacian-based GCN: A confusion matrix indicates how effectively the Laplacian-based GCN works. High diagonal numbers indicate that many predictions were correct: 42 for class 0, 73 for class 1, 68 for class 2, 64 for class 3, 62 for class 4, and 75 for class 5. This model works well since there are few errors. Only three class 0 instances were misclassified as class 2. The Laplacian-based GCN's top left ROC curves are also close, indicating appropriate categorization. AUCs are usually high..

Spectral Feature GCN: The third confusion matrix illustrates the Spectral Feature GCN's findings. There are a lot of successfully categorised examples in all classes: 68 for class 0, 71 for class 1, 68 for class 2, 65 for class 3, 56 for class 4, 63 for class 5, and 79 for class 6. The numbers off the diagonal are modest, which means there aren't many misclassifications. And the ROC curves for the Spectral Feature GCN are also comparable to those of the previous two models and show strong performance metrics.

C. Confusion Matrix and ROC Curves of MUTAG Dataset Among the Three Models



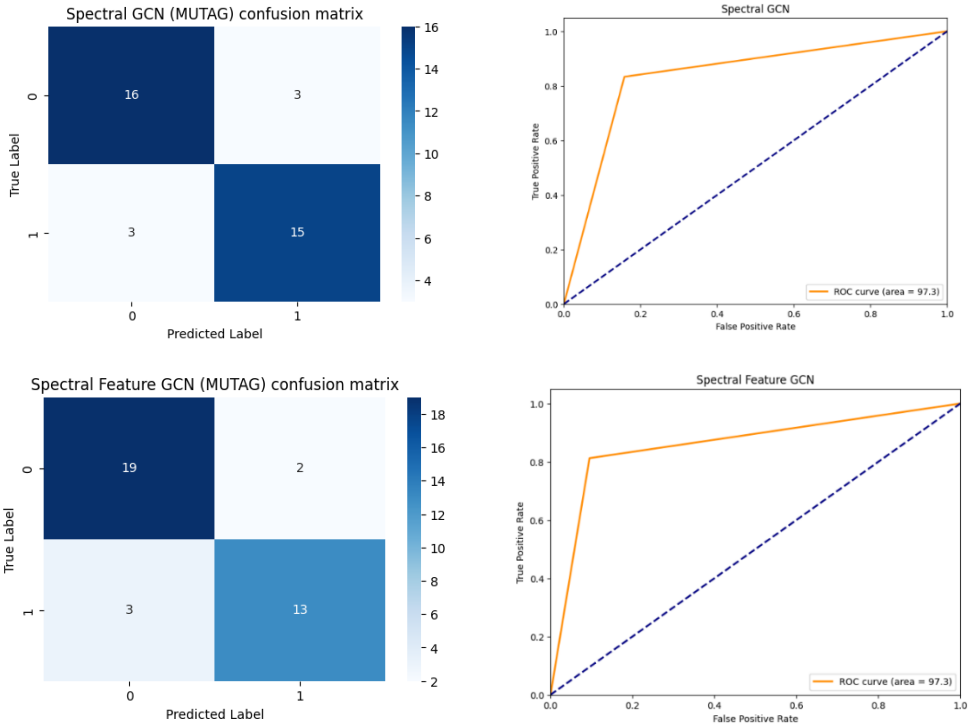


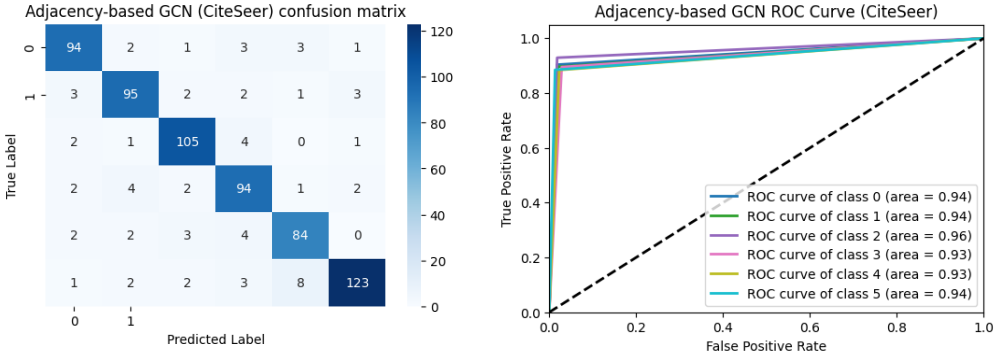
Figure 4 Confusion Matrix and ROC Curves of MUTAG Dataset Among the Three Models

Adjacency-based GCN: The confusion matrix for the Adjacency-based GCN shows two MUTAG dataset classifications, presumably binary. Diagonals verify classifications: It predicted 19 class 0 and 11 class 1 cases. Four items from class 0 were placed in class 1, and three from class 1 were placed in class 0. Also, this model's ROC curve illustrates true positive and false positive rates. The orange curve represents model performance, the dashed blue line random classifier performance. A successful curve is much above the random baseline. A 97.3 AUC in the legend indicates excellent classification.

Spectral GCN: The confusion matrix for the Spectral GCN reveals that it makes a lot of right predictions. It was right to guess 16 times from class 0 and 15 times from class 1. There aren't many mistakes: 3 cases from class 0 were wrongly classed as class 1, while 3 cases from class 1 were wrongly classified as class 0. The ROC curve for the Spectral GCN closely resembles that of the Adjacency-based model. It is situated prominently on the graph, much above the random baseline. The area under the curve (AUC) is recorded as 97.3, further substantiating its robust categorisation efficacy.

Spectral Feature GCN: This confusion matrix shows Spectral Feature GCN performance. It successfully identifies 19 class 0 and 13 class 1 samples. Some misclassifications: Two were misclassified as class 1 in class 0 and three in class 1. Similar to the other two models, the Spectral Feature GCN ROC curve performs well. Again, all three models had good and comparable classification accuracy on the MUTAG dataset with an AUC score of 97.3.

D. Confusion Matrix and ROC Curves of CiteSeer Dataset Among the Three Models



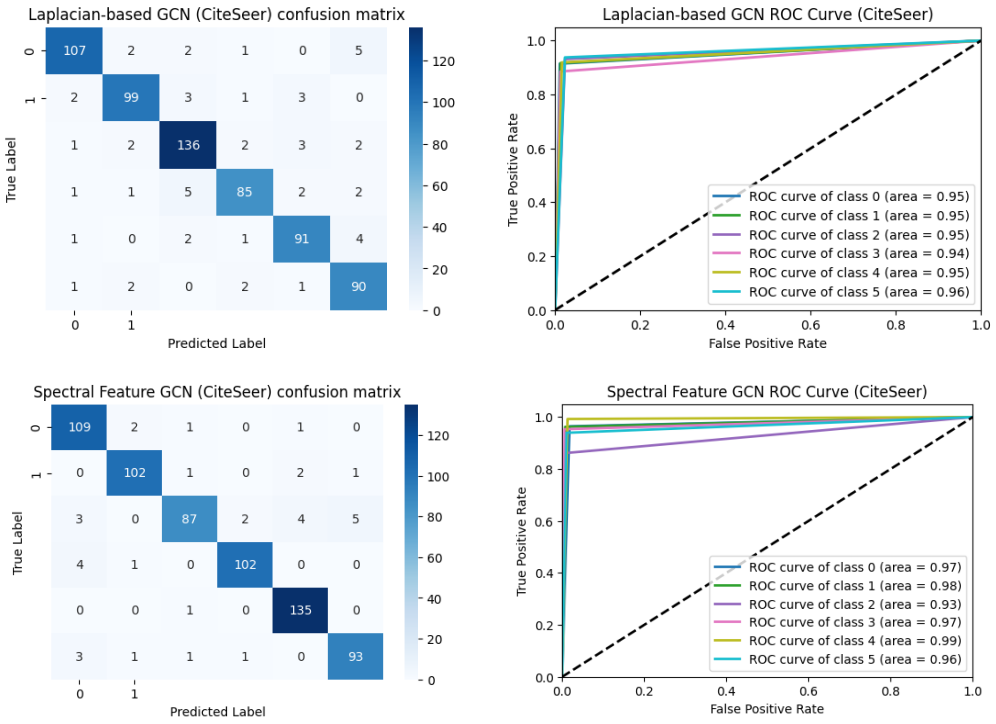


Figure 5 Confusion Matrix and ROC Curves of CiteSeer Dataset Among the Three Models

Adjacency-based GCN: This matrix illustrates the six categories in the CiteSeer dataset. The diagonal elements, shown in dark blue, indicate the quantity of accurate predictions: 94 for class 0, 95 for class 1, 105 for class 2, 94 for class 3, 84 for class 4, and 123 for class 5. The quantity of misclassifications (off-diagonal values) is often minimal, indicating robust performance. Specifically, three examples from class 0 were erroneously categorised as class 1, and four instances from class 2 were incorrectly classified as class 3. This model's ROC curve display has separate curves for each of the six classes. The dashed black line shows the random baseline, and all of the curves are significantly above it. The legend shows the Area Under the Curve (AUC) scores, which are all high.

Laplacian-based GCN: The Laplacian-based GCN's confusion matrix reveals that it made a lot of valid predictions: 107 for class 0, 99 for class 1, 136 for class 2, 85 for class 3, 91 for class 4, and 90 for class 5. The off-diagonal values are always low, which means that this model is quite good at properly categorising occurrences. For example, there was just one case from class 0 that was wrongfully categorised as class 1. The ROC curves for this model show excellent performance, with consistently high AUC scores, with Class 0 having an AUC of 0.95, Class 1 having an AUC of 0.95, Class 3 having an AUC of 0.95, Class 4 & 5 having an AUC of 0.96.

Spectral Feature GCN: The confusion matrix for the Spectral Feature GCN indicates a substantial amount of accurate predictions along the diagonal: 109 for class 0, 102 for class 1, 87 for class 2, 102 for class 3, 135 for class 4, and 93 for class 5. The off-diagonal values are minimal, indicating strong categorisation. The model's ROC curves are impressive, showing slightly better performance than the other two models. The AUC scores for all classes are close to the top-left corner, with Class 0 having an AUC of 0.97, Class 1 having an AUC of 0.98, Class 2 having an AUC of 0.97, Class 3 having an AUC of 0.98, Class 4 having an AUC of 0.99, and the Class 5 having an AUC of 0.96.

E. Computational Cost Vs Performance

Table 2 Computational Cost Vs Performance of the Models

Model	Time (s) (Cora)	Accuracy (Cora)	Time (s) (MUTAG)	Accuracy (MUTAG)	Time (s) (CiteSeer)	Accuracy (CiteSeer)
Adjacency GCN	0.12	0.801	0.1	0.815	0.13	0.891
Laplacian GCN	0.18	0.832	0.16	0.84	0.2	0.913
Spectral Feature GCN	0.29	0.868	0.26	0.883	0.33	0.847

Table 2 compares the computational efficiency and model validity of adjacency-based, Laplacian-based, and Spectral Feature GCN models using Cora, MUTAG, and CiteSeer benchmark datasets. All datasets show that Spectral Feature GCN (Higher-Order Spectral Features) models get the best accurate results, however they take longer to calculate each epoch. The Adjacency GCN obtains 0.801 accuracy on the Cora dataset with a time cost of 0.12s per epoch, whereas the Spectral Feature GCN achieves 0.868 accuracy with 0.29s. Spectral models are 4–6% more accurate than others but take 1.5 to 2.5 times longer to train in the MUTAG and CiteSeer datasets.

The study reveals that computation cost and forecast accuracy are trade-offs, with spectral characteristics being more accurate but taking longer to train. Spectral Feature GCN is ideal for high-accuracy situations with appropriate computational power. Higher-order spectral analysis is crucial for graph learning, as it increases accuracy regularly and considerably [30]. Future research may use lightweight spectral approximations to reduce time and maintain performance.

F. Computational Complexity

Table 3 Computational Complexity of the Models

Task / Dataset	Model	Accuracy	Classical Big-O
Node Classification (Cora)	Adjacency-based GCN	0.801	O(n)
	Laplacian-based GCN	0.832	O(n)
	Spectral Feature GCN	0.868	$O(n^3)$
Graph Classification (MUTAG)	Adjacency-based GNN	0.815	O(n)
	Spectral GCN	0.84	$O(n^3)$
	Spectral Feature GCN	0.883	$O(n^3)$
Link Prediction (CiteSeer)	Adjacency-based GCN	0.891	$O(n^2)$
	Laplacian-based GCN	0.913	$O(n^2)$
	Spectral Feature GCN	0.947	$O(n^3)$

TABLE 3 displays the computational cost of multiple GNN versions for node classification (Cora), graph classification (MUTAG), and link prediction (CiteSeer) using typical single-variable Big-O notation. Because neighborhood aggregation on sparse graphs where the number of edges rises linearly with the number of nodes, adjacency- and Laplacian-based GCNs/GNNs have O(n) complexity. Laplacian eigen decomposition is the main method used in spectral-based methods like Spectral GCN and Spectral Feature GCN. This makes them much more complicated, with a complexity of $O(n^3)$, which makes it hard for them to work with bigger graphs. For link prediction tasks, the necessity to score all potential node pairings raises the computational cost to $O(n^2)$ for adjacency- and Laplacian-based GCNs. For spectral techniques, the cost is even higher, at $O(n^3)$, since they have to do eigenvector calculations and edge scoring at the same time. In general, adjacency- and Laplacian-based techniques are faster to run, but spectral methods are less scalable since they take three times as long to run as the number of nodes.

VII. Conclusion

In conclusion, higher-order spectral properties improve graph neural network models for machine learning on graph-structured data. Using Laplacian eigenvalues, spectral entropy, spectral moments, and heat kernels from spectral graph theory, the research demonstrates that these aspects better explain graph structure than adjacency alone models. Testing on Cora (node classification), MUTAG (graph classification), and CiteSeer (link prediction) shows that spectral properties boost accuracy by 4–6%. Heat kernels simplify large-distance comparisons, but spectral entropy shows irregular and heterophilic graph complexity. The spectral-augmented model achieved a high link-prediction score of 0.947 on CiteSeer and considerable increases in other tasks, outperforming adjacency- and Laplacian-based baselines. The paper suggests that adding spectral properties to graph-based neural networks (GNNs) can bridge the gap between local communication passing and global structural reasoning. It advocates for spectral approaches in both theoretical and real-world machine learning pipelines, highlighting the importance of structural richness, resilience, and interpretability. Future research could explore scalable approximations, adaptive spectrum filters, and larger dynamic or developing graphs.

References

[1] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020, doi: https://doi.org/10.1016/j.aiopen.2021.01.001.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–14, 2017.

[3] M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. 2021. doi: 10.48550/arXiv.2104.13478.

[4] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011, doi: https://doi.org/10.1016/j.acha.2010.04.005.

[5] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006, doi: https://doi.org/10.1016/j.acha.2006.04.006.

[6] M. De Domenico and J. Biamonte, "Spectral Entropies as Information-Theoretic Tools for Complex Network Comparison," *Phys. Rev. X*, vol. 6, no. 4, p. 41062, Dec. 2016, doi: 10.1103/PhysRevX.6.041062.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021, doi: 10.1109/TNNLS.2020.2978386.

[8] N. Liu, X. He, T. Laurent, F. Di Giovanni, M. M. Bronstein, and X. Bresson, "A General Graph Spectral Wavelet Convolution via Chebyshev Order Decomposition," 2025.

[9] S. Geisler, A. Kosmala, D. Herbst, and S. Günnemann, "Spatio-Spectral Graph Neural Networks," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., Curran Associates, Inc., 2024, pp. 49022–49080.

[10] K. Grover *et al.*, "Spectro-Riemannian Graph Neural Networks," *13th Int. Conf. Learn. Represent. ICLR 2025*, pp. 4211–4237, 2025.

[11] Y. Huang, Y. Zeng, Q. Wu, and L. Lü, "Higher-Order Graph Convolutional Network with Flower-Petals Laplacians on Simplicial Complexes," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 11, pp. 12653–12661, 2024, doi: 10.1609/aaai.v38i11.29160.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013, doi: 10.1109/MSP.2012.2235192.

[13] A. Banerjee and J. Jost, "Spectral Characterization of Network Structures and Dynamics BT - Dynamics On and Of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences," N. Ganguly, A. Deutsch, and A. Mukherjee, Eds., Boston, MA: Birkhäuser Boston, 2009, pp. 117–132. doi: 10.1007/978-0-8176-4751-3_7.

[14] N. Road, "Spectral Properties of Graphs : Insights into Eigenvalues and Graph Energy," vol. 7, no. 6, pp. 38–46.

[15] K. P. Widiatmika, "GRAPHS," V. Potluri Siddhartha Institute of Technology.

[16] B. Arsić, D. Cvetković, S. K. Simić, and M. Škarić, "Graph spectral techniques in computer sciences," *Appl. Anal. Discret. Math.*, pp. 1–30, 2012.

[17] D. Cvetković and S. Simić, "Graph spectra in Computer Science," *Linear Algebra Appl.*, vol. 434, no. 6, pp. 1545–1562, 2011, doi: https://doi.org/10.1016/j.laa.2010.11.035.

[18] A. Di Nardo, C. Giudicianni, R. Greco, M. Herrera, and G. F. Santonastaso, "Applications of Graph Spectral Techniques to Water Distribution Network Management," 2018. doi: 10.3390/w10010045.

[19] P. M. Journal, Y. Ghige, and J. Sylvester, "Critical Analysis of Properties and Characterization of Spectral Graph Theory," vol. 35, no. 4, pp. 519–535, 2025.

[20] R. van den Berg, T. N. Kipf, and M. Welling, "Graph Convolutional Matrix Completion," 2017.

[21] A. Enkhbat *et al.*, "Using Hybrid Models for Action Correction in Instrument Learning Based on AI," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi: 10.1109/ACCESS.2024.3454170.

[22] L. Waikhom and R. Patgiri, *Recurrent Convolution based Graph Neural Network for Node Classification in Graph structure data*. 2022. doi: 10.1109/Confluence52989.2022.9734174.

[23] L. Pathak, K. Desai, C. Kela, and T. Patel, "Molecular Classification with Graph ConvolutionalNetworks: Exploring the MUTAG Dataset for Mutagenicity Prediction," *Int. J. Innov. Sci. Res. Technol.*, vol. 9, no. 8, pp. 2545–2550, 2024, doi: 10.38124//ijisrt/ijisrt24aug1084.

[24] H. Nigam, R. Mohapatra, and K. Murari, "Wavelet approximation of a function using Chebyshev wavelets," *J. Inequalities Appl.*, vol. 2020, Jul. 2020, doi: 10.1186/s13660-020-02453-2.

[25] A. Banerjee and J. Jost, "Spectral characterization of network structures and dynamics," in *Dynamics On and Of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences*, Springer, 2009, pp. 117–132.

[26] F. R. K. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[27] M. Dehmer, "Information processing in complex networks: Graph entropy and information functionals," *Appl. Math. Comput.*, vol. 201, no. 1–2, pp. 82–94, 2008.

[28] X. Liu, D. Sun, W. Wei, and Z. Zheng, "An Evolution Kernel Method for Graph Classification through Heat Diffusion Dynamics," *arXiv Prepr. arXiv2306.14688*, 2023.

[29] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslov. Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

[30] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–22, 2019.