

## Article

# Eigenvector Distance-Modulated Graph Neural Network: Spectral Weighting for Enhanced Node Classification

Ahmed Begga , Francisco Escolano  and Miguel Ángel Lozano 

Department of Computer Science and Artificial Intelligence, University of Alicante, 03690 Alicante, Spain; sco@ua.es (F.E.); malozano@ua.es (M.Á.L.)

\* Correspondence: ahmed.begga@ua.es

## Abstract

Graph Neural Networks (GNNs) face significant challenges in node classification across diverse graph structures. Traditional message passing mechanisms often fail to adaptively weight node relationships, thereby limiting performance in both homophilic and heterophilic graph settings. We propose the Eigenvector Distance-Modulated Graph Neural Network (EDM-GNN), which enhances message passing by incorporating spectral information from the graph's eigenvectors. Our method introduces a novel weighting scheme that modulates information flow based on a combined similarity measure. This measure balances feature-based similarity with structural similarity derived from eigenvector distances. This approach creates a more discriminative aggregation process that adapts to the underlying graph topology. It does not require prior knowledge of homophily characteristics. We implement a hierarchical neighborhood aggregation framework that utilizes these spectral weights across multiple powers of the adjacency matrix. Experimental results on benchmark datasets demonstrate that EDM-GNN achieves competitive performance with state-of-the-art methods across both homophilic and heterophilic settings. Our approach provides a unified solution for node classification problems with strong theoretical foundations in spectral graph theory and significant empirical improvements in classification accuracy.

**Keywords:** graph neural networks; spectral graph theory; eigenvector distance; node classification; edge weighting; graph representation learning

**MSC:** 68T07; 68T10



Academic Editor: Ioannis Tsoulos

Received: 24 July 2025

Revised: 19 August 2025

Accepted: 4 September 2025

Published: 8 September 2025

**Citation:** Begga, A.; Escolano, F.; Lozano, M.Á. Eigenvector Distance-Modulated Graph Neural Network: Spectral Weighting for Enhanced Node Classification. *Mathematics* **2025**, *13*, 2895. <https://doi.org/10.3390/math13172895>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Graph Neural Networks (GNNs) have emerged as powerful tools for learning representations of graph-structured data, with applications spanning diverse domains including social network analysis [1], molecular property prediction [2], and recommendation systems [3]. Despite their success, GNNs still face significant challenges in effectively modeling complex graph structures, particularly in heterophilic environments where connected nodes tend to have different labels [4].

A fundamental limitation of conventional GNNs is their message-passing (MP) mechanism [2,5], which typically treats all neighboring nodes equally or employs simple attention mechanisms that may not capture the structural properties of the graph [6]. This approach proves inadequate when dealing with the intricate connectivity patterns found in real-world heterophilic graphs [7]. Recent research has explored various strategies to address

this issue, including the incorporation of higher-order neighborhood information [8], adaptive aggregation mechanisms [9], and community-aware approaches [10]. Spectral graph theory (SGT) provides valuable insights into the structural properties of graphs through the eigendecomposition of graph matrices. The graph spectrum encodes rich information about connectivity patterns, community structures, and node similarities [11]. However, existing spectral approaches often focus solely on using eigenvectors for embedding or clustering [12–14], without effectively integrating this information into the MP mechanism of GNNs.

In this paper, we propose the Eigenvector Distance-Modulated Graph Neural Network (EDM-GNN), a novel approach that enhances MP by incorporating eigenvector distances from the graph’s spectral domain. Our key innovation is a weighting scheme that modulates information flow based on a combined similarity measure that balances feature-based similarity with structural similarity derived from eigenvector distances between node pairs. This technique creates a more discriminative aggregation process that adapts to the underlying graph topology. The eigenvector distance between nodes captures their structural positioning in the graph’s spectral representation, complementing traditional feature similarity measures. By combining these two dimensions, our approach can identify nodes that are both semantically similar (in feature space) and structurally aligned (in spectral space), leading to more effective information propagation. This is particularly valuable in heterophilic graphs, where connected nodes may have different labels but similar structural roles. Our contributions can be summarized as follows:

- We introduce a spectral modulation mechanism that leverages eigenvector distances to weight MP between nodes, enhancing the model’s ability to capture structural properties.
- We implement a multi-order neighborhood aggregation framework that applies our spectral weighting strategy across various powers of the adjacency matrix.
- We demonstrate through extensive experiments on benchmark datasets that our approach consistently outperforms state-of-the-art (SOTA) methods in both homophilic and heterophilic settings.

The remainder of this paper is organized as follows: Section 2 reviews related work in GNNs and SGT. Section 3 provides the necessary preliminary concepts. Section 4 presents our proposed EDM-GNN method in detail. Section 5 describes the experimental setup and results, followed by a discussion of the findings. Finally, Section 6 concludes the paper and outlines directions for future research.

## 2. Related Work

Our work intersects several key areas of research in GNNs and SGT. We organize the related work into three main categories: GNNs for node classification, approaches to handling heterophily, and spectral methods in graph learning.

### 2.1. Graph Neural Networks for Node Classification

Traditional GNN architectures, such as Graph Convolutional Networks (GCNs) [6] and Graph Attention Networks (GATs) [15], have proven effective for homophilic graphs where connected nodes tend to share similar characteristics [1].

However, recent work has highlighted significant limitations of conventional GNNs when applied to heterophilic settings. The fundamental assumption that neighboring nodes share similar features or labels often fails in real-world networks [7]. This has motivated the development of more sophisticated aggregation mechanisms that can adaptively handle diverse graph topologies.

Several recent approaches have attempted to address these limitations through architectural innovations. GraphSAINT [16] introduced sampling techniques to improve

scalability and robustness, while Graph Isomorphism Networks (GINs) [17] focused on maximizing the representational power through careful design of aggregation functions. However, these methods still primarily rely on local neighborhood structures and may not effectively capture the global structural patterns that are crucial for heterophilic graphs.

## 2.2. Handling Heterophily in Graph Neural Networks

The challenge of heterophily has received increasing attention in recent years, with several pioneering works proposing novel architectures specifically designed for graphs where connected nodes have dissimilar characteristics.

H2GCN [7] introduced ego- and neighbor-embedding separation along with higher-order neighborhood exploration to better handle heterophilic graphs. The key insight was that in heterophilic settings, a node's own features may be more informative than its immediate neighbors' features. GEOM-GCN [18] addressed the limitations of standard MP by incorporating geometric relationships in the latent space.

Higher-order approaches have shown particular promise for heterophilic graphs. MixHop [8] demonstrated the effectiveness of aggregating features from nodes at different hop distances using powers of the transition matrix. Similarly, FSGNN [19] incorporated regularization techniques such as softmax and L2-normalization to improve multi-hop aggregation. These approaches recognize that useful information for heterophilic graphs may reside beyond immediate neighborhoods.

Recent work has also explored adaptive mechanisms for handling diverse graph structures. GPR-GNN [9] integrated generalized PageRank with GNNs to provide adaptive feature smoothing. FAGCN [20] introduced learnable filters that can capture both low-frequency and high-frequency signals in graphs, making it suitable for both homophilic and heterophilic settings.

Community-aware approaches have emerged as another promising direction. The Community-HOP method [10] leveraged spectral clustering to identify graph communities and modified information flow within and between communities. This approach recognizes that structural communities may provide important signals for node classification, even when immediate neighbors are not informative.

## 2.3. Spectral Methods in Graph Learning

Spectral graph theory has a rich history in analyzing graph structures through eigen-decomposition of graph matrices [11,12]. Classical spectral clustering algorithms have demonstrated the power of eigenvector-based representations for understanding graph structure and identifying communities.

In the context of GNNs, spectral approaches such as ChebNet [21] have primarily focused on defining convolution operations in the frequency domain using Chebyshev polynomials to approximate spectral filters. While these spectral methods implement message passing through frequency-domain convolutions, our approach takes a different direction by incorporating spectral node relationships directly into spatial message passing through adaptive edge weighting based on eigenvector distances.

Recent developments have expanded the use of spectral information in GNNs through positional encodings. Laplacian Positional Encoding (LapPE) [22] and Random Walk Positional Encoding (RWPE) [23] have been successfully integrated into frameworks like GraphGPS to enhance node representations by leveraging spectral structure. These approaches share our objective of utilizing spectral properties to improve graph neural networks, though they focus on augmenting node features rather than modulating message passing weights. Additionally, hybrid approaches have demonstrated the effectiveness of integrating different GNN modules for structured optimization tasks, showing the poten-

tial of combining multiple sources of structural information [24]. The field has also seen advances in scalability-focused spectral approaches, including localized filtering methods, spectral approximations, and graph sparsification techniques that make spectral analysis feasible for larger graphs [25]. Additionally, research on dynamic and temporal graphs has explored how spectral properties evolve over time, while edge feature-aware spectral methods have investigated incorporating edge attributes into spectral analysis. These developments represent important directions for extending spectral GNN approaches to broader and larger-scale scenarios. More recent work has begun to explore the integration of spectral information with modern GNN architectures. Some approaches have used graph spectral properties for regularization or as auxiliary features, but few have directly incorporated eigenvector distances into the MP mechanism as we propose.

The concept of using eigenvector distances to measure structural similarity has been explored in various graph analysis contexts [26]. However, the specific application to modulating MP in GNNs, particularly the ratio-based weighting scheme we introduce, represents a novel contribution to the field.

#### 2.4. Limitations of Existing Approaches

While existing methods have made significant progress in addressing heterophily and incorporating structural information, several limitations remain [27]. First, many approaches rely on predefined heuristics or require prior knowledge about the graph's homophily characteristics. Second, methods that use higher-order neighborhoods often suffer from computational complexity issues or may introduce noise from distant, irrelevant nodes [19]. A key challenge is that many existing methods fail to distinguish between nodes with different structural roles, even when these nodes may share similar local features. For instance, bridge nodes connecting different communities and central nodes within communities may have similar degree centrality but play fundamentally different structural roles in information propagation.

Our proposed EDM-GNN addresses these limitations by introducing a novel spectral modulation mechanism that adaptively weights MP based on both feature similarity and structural alignment, as measured by eigenvector distances. This approach provides a unified framework that can handle both homophilic and heterophilic graphs without requiring prior knowledge of graph characteristics.

### 3. Preliminaries

In this section, we introduce the essential mathematical concepts and notations that form the foundation of our proposed method. We begin with basic graph representations and the node classification problem, followed by detailed exposition of SGT elements, and conclude with the MP mechanism in GNNs, including attention-based approaches.

#### 3.1. Graph Representation and Node Classification

Let  $G = (V, E)$  denote an undirected graph, where  $V = \{v_1, v_2, \dots, v_n\}$  represents the set of  $n$  nodes and  $E \subseteq V \times V$  represents the set of edges. We use the adjacency matrix  $\mathbf{A} \in \{0, 1\}^{n \times n}$  to encode the graph structure, where  $\mathbf{A}_{ij} = 1$  if there exists an edge between nodes  $v_i$  and  $v_j$ , and  $\mathbf{A}_{ij} = 0$  otherwise.

Each node  $v_i$  is associated with a feature vector  $\mathbf{x}_i \in \mathbb{R}^d$ , where  $d$  is the dimensionality of the feature space. These features are collectively represented as a matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , where the  $i$ -th row corresponds to the feature vector of node  $v_i$ . For supervised node classification, a subset of nodes  $V_L \subset V$  have associated labels  $\mathbf{y}_i \in \{1, 2, \dots, C\}$ , where  $C$  is the number of classes.

The node classification problem aims to predict the labels of unlabeled nodes  $V_U = V \setminus V_L$  by learning a function  $f: \mathbb{R}^d \rightarrow \{1, 2, \dots, C\}$  that maps node features to class labels. This learning process typically leverages both the node features and the graph structure to make predictions.

For analytical purposes, we define the diagonal degree matrix  $\mathbf{D} \in \mathbb{R}^{n \times n}$ , where  $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$  represents the degree of node  $v_i$ . To account for self-loops in the graph, we define the augmented adjacency matrix  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix.

### Homophily and Heterophily

A fundamental characteristic of graphs that significantly impacts node classification performance is the concept of homophily versus heterophily. Homophily refers to the tendency of connected nodes to share similar characteristics or labels. Conversely, heterophily describes scenarios where connected nodes tend to have different characteristics or labels.

To quantify the level of homophily in a graph [7], we use the edge homophily ratio, defined as

$$h = \frac{|\{(v_i, v_j) \in E : y_i = y_j\}|}{|E|} \quad (1)$$

where  $y_i$  denotes the label of node  $v_i$ . This metric ranges from 0 (complete heterophily) to 1 (complete homophily). Graphs with  $h > 0.5$  are typically considered homophilic, while those with  $h < 0.5$  are considered heterophilic [28].

The degree of homophily significantly affects the performance of traditional GNNs, which often assume that neighboring nodes share similar characteristics. This assumption breaks down in heterophilic graphs, motivating the need for more sophisticated approaches that can adaptively handle both scenarios.

While the edge homophily ratio provides a widely-used measure for graph heterophily, it primarily captures label-based heterophily and may not fully encompass other forms of structural or feature-based heterophily, such as role-based dissimilarity where nodes with different structural functions are connected, or feature-space heterophily where connected nodes have dissimilar attributes despite sharing labels. Our approach is designed to be robust to these various forms of heterophily through the adaptive combination of feature and spectral similarities, which can capture both semantic and structural relationships beyond simple label concordance.

### 3.2. Spectral Graph Theory

Spectral graph theory provides a powerful mathematical framework for analyzing the structural properties of graphs through the lens of linear algebra. The cornerstone of this theory is the normalized graph Laplacian, defined as:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (2)$$

The normalized Laplacian matrix  $\mathbf{L}$  is symmetric and positive semi-definite, with several important mathematical properties. Its eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$  provide crucial information about the graph's connectivity and structure.

#### 3.2.1. Eigendecomposition and Spectral Properties

The spectral decomposition of  $\mathbf{L}$  yields a set of eigenvalues  $\{\lambda_i\}_{i=1}^n$  with corresponding orthonormal eigenvectors  $\{\mathbf{u}_i\}_{i=1}^n$ . These eigenvectors form an orthonormal basis for  $\mathbb{R}^n$  and can be organized into a matrix  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$ .

For a connected graph, the smallest eigenvalue  $\lambda_1 = 0$  with its corresponding eigenvector  $\mathbf{u}_1$  being the constant vector. The second smallest eigenvalue  $\lambda_2$  is known as the

algebraic connectivity or Fiedler value, and its corresponding eigenvector  $\mathbf{u}_2$  (the Fiedler vector) provides insights into the graph's cut structure.

The eigenvalues of the normalized Laplacian are bounded as  $\lambda_i \in [0, 2]$ , where  $\lambda_1 = 0$  always, with multiplicity equal to the number of connected components,  $\lambda_2 > 0$  for connected graphs, indicating the strength of connectivity and  $\lambda_n \leq 2$ , with equality achieved when the graph contains a bipartite component [29].

The eigenvectors corresponding to smaller eigenvalues capture global, low-frequency patterns in the graph, while those corresponding to larger eigenvalues capture local, high-frequency variations. This spectral hierarchy allows us to analyze the graph at different scales of resolution.

### 3.2.2. Eigenvector Distance and Structural Similarity

When we represent each node using the components of the first  $k$  eigenvectors (typically excluding  $\mathbf{u}_1$ ), we obtain a spectral embedding that captures the node's position within the global structure of the graph. For any two nodes  $v_i$  and  $v_j$ , we can compute their eigenvector distance as:

$$d_{\text{eig}}(i, j) = \|\mathbf{U}_{i,:k} - \mathbf{U}_{j,:k}\|_2 \quad (3)$$

where  $\mathbf{U}_{i,:k} \in \mathbb{R}^k$  represents the spectral embedding of node  $v_i$  using the first  $k$  eigenvectors. This distance measure provides a principled way to assess structural similarity between nodes, even when they are located in distant parts of the graph.

The eigenvector distance encodes rich information about the relative positions of nodes within the graph's global structure that local measures like shortest path distance cannot capture. Nodes with small eigenvector distances tend to play similar structural roles in the graph, regardless of their proximity in the original topology [11].

The selection of the number of eigenvectors  $k$  involves balancing structural discrimination and noise reduction. Low values of  $k$  may limit the resolution of structural distinctions, while large  $k$  may introduce noise from high-frequency eigenvectors that capture local variations rather than global structure. In practice, we use  $k = 20$  eigenvectors across all datasets, which provides sufficient structural information while maintaining computational efficiency. Our empirical analysis shows that performance remains stable for  $k$  values between 15 and 25, indicating robustness to this hyperparameter choice.

### 3.3. Graph Neural Networks and Message Passing

GNNs operate through a MP framework, which consists of three fundamental steps: message construction, aggregation, and update [2]. For the  $l$ -th layer of a GNN, this process can be expressed generically as

$$\mathbf{h}_i^{(l+1)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_i^{(l)}, \text{AGGREGATE}^{(l)}\left(\{\text{MESSAGE}^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}) : j \in \mathcal{N}(i)\}\right)\right) \quad (4)$$

where  $\mathbf{h}_i^{(l)}$  is the feature vector of node  $v_i$  at layer  $l$ ,  $\mathcal{N}(i)$  denotes the set of neighboring nodes of  $v_i$ , and  $\mathbf{e}_{ij}$  represents potential edge features or weights.

#### 3.3.1. Graph Convolutional Networks

A fundamental instantiation of this framework is the Graph Convolutional Network (GCN) [6], which implements a simplified and efficient MP scheme. The GCN layer is defined as

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}\right) \quad (5)$$

where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is the adjacency matrix with self-loops,  $\tilde{\mathbf{D}}$  is the corresponding degree matrix,  $\mathbf{W}^{(l)}$  is a learnable weight matrix, and  $\sigma$  is a non-linear activation function.

The key insight of GCN is the use of the normalized adjacency matrix  $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ , which provides a principled way to weight the contribution of neighboring nodes based on their degrees. This normalization ensures that nodes with high degrees do not dominate the aggregation process, while nodes with low degrees still receive sufficient influence from their neighbors.

### 3.3.2. Graph Attention Networks

While GCNs use predefined normalization to weight node contributions, Graph Attention Networks (GATs) [15] introduce learnable attention mechanisms to dynamically determine the importance of different neighbors. In GAT, the attention coefficient between nodes  $i$  and  $j$  is computed as

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_k]))} \quad (6)$$

where  $\mathbf{a}$  is a learnable attention vector,  $\mathbf{W}$  is a weight matrix, and  $\parallel$  denotes concatenation. The final node representation is then computed as

$$\mathbf{h}_i^{(l+1)} = \sigma \left( \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j^{(l)} \right) \quad (7)$$

The attention mechanism in GAT allows the model to focus on the most relevant neighbors for each node, providing more flexibility than the fixed normalization used in GCN. This adaptive weighting scheme bears conceptual similarity to our proposed eigenvector distance-based modulation, although our approach incorporates structural information from the graph's spectral domain rather than relying solely on feature-based attention.

### 3.3.3. Multi-Order Neighborhood Aggregation

Higher-order GNN approaches extend the basic MP framework by considering information from multi-hop neighborhoods [8,17]. This can be achieved by computing various powers of the normalized adjacency matrix:

$$\mathbf{P}^k = \left( \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \right)^k \quad (8)$$

where  $\mathbf{P}^k$  encodes  $k$ -hop relationships between nodes. The element  $[\mathbf{P}^k]_{ij}$  represents the strength of connection between nodes  $v_i$  and  $v_j$  through paths of length  $k$ .

Multi-order approaches aggregate information from different neighborhood scales and combine them through learnable attention weights:

$$\mathbf{H}^{(l+1)} = \sigma \left( \sum_{k=1}^K \beta_k \mathbf{P}^k \mathbf{H}^{(l)} \mathbf{W}_k^{(l)} \right) \quad (9)$$

where  $\beta_k$  are learnable attention weights satisfying  $\sum_{k=1}^K \beta_k = 1$ , and  $\mathbf{W}_k^{(l)}$  are layer-specific weight matrices for each order  $k$ .

The relationship between multi-hop neighborhood aggregation and spectral embeddings is fundamentally complementary rather than substitutive. Multi-hop aggregation emphasizes information diffusion across path lengths, capturing connectivity patterns at different scales, while spectral embeddings encode relative structural roles and global positioning within the graph. Our approach combines these perspectives by using spectral weighting as a refinement to traditional  $k$ -hop aggregation schemes. The spectral weights

help discriminate which multi-hop connections are structurally meaningful, effectively filtering the expanded neighborhood to focus on nodes that share similar structural roles or positions, regardless of their path distance. This combination allows the model to benefit from both the scale diversity of multi-hop aggregation and the structural awareness of spectral analysis.

#### 4. Methodology

In this section, we present EDM-GNN, a novel approach that enhances MP by incorporating spectral information through eigenvector distances. Our method addresses the fundamental limitation of conventional GNNs by introducing a principled framework that adaptively weights edges based on both structural similarity and feature similarity, enabling effective learning across diverse graph topologies.

The key insight underlying our approach is that traditional MP mechanisms fail to capture the rich structural information encoded in the graph's spectral domain. While existing methods either treat all neighbors equally or rely primarily on feature-based attention, they overlook the global structural relationships that eigenvectors naturally encode. By incorporating eigenvector distances into the edge weighting scheme, we create a more discriminative aggregation process that adapts to the underlying graph topology without requiring prior knowledge of homophily characteristics.

Figure 1 illustrates the overall architecture of our proposed EDM-GNN framework. The methodology proceeds through several interconnected stages: spectral preprocessing to extract structural similarities, feature-based similarity computation, adaptive edge reweighting through combined similarity measures, and multi-order neighborhood aggregation with learnable attention mechanisms.

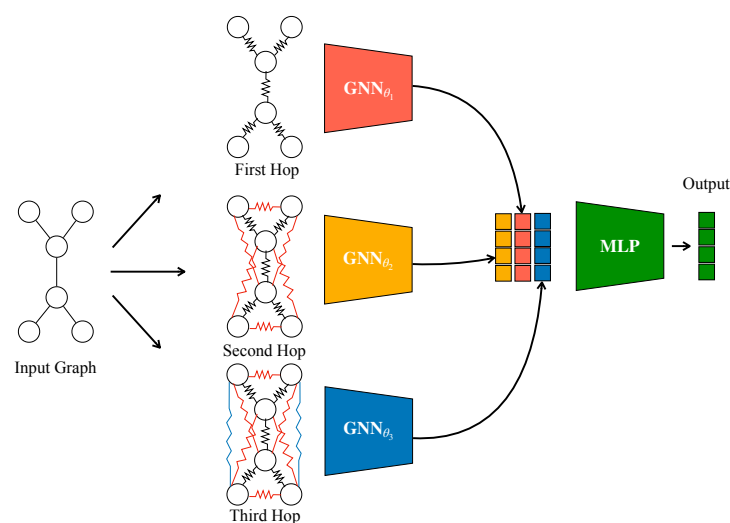


**Figure 1.** Architecture overview of the proposed EDM-GNN framework. The left side shows the input graph with standard connectivity. The center displays three parallel processing streams corresponding to first-hop, second-hop, and third-hop neighborhoods, represented by increasingly sparse edge patterns. Each hop level is processed by dedicated GNN modules ( $GNN_{\theta_1}$ ,  $GNN_{\theta_2}$ ,  $GNN_{\theta_3}$ ) with distinct trainable parameters  $\theta_i$ . The varying line styles (solid, dashed, dotted) and colors (red, orange, blue) represent different edge weights based on eigenvector distance modulation, with thicker lines indicating stronger connections. The right side shows the attention-based fusion mechanism that combines embeddings from all hop levels, followed by an MLP classifier that produces the final node predictions.

##### 4.1. Spectral Preprocessing and Structural Similarity

Our approach begins with the spectral analysis of the input graph to extract structural information through eigendecomposition of the normalized Laplacian matrix. Given

a graph  $G = (V, E)$  with feature matrix  $\mathbf{X}$ , we compute the normalized Laplacian  $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ , where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  includes self-loops. The eigendecomposition  $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$  provides us with eigenvectors  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$  and corresponding eigenvalues in ascending order.

We select the first  $k$  non-trivial eigenvectors (excluding the constant eigenvector  $\mathbf{u}_1$ ) to form the spectral embedding matrix  $\mathbf{U}_k = [\mathbf{u}_2, \mathbf{u}_3, \dots, \mathbf{u}_{k+1}]$ . Each node  $v_i$  is then represented by its spectral coordinates  $\mathbf{U}_{k,i} \in \mathbb{R}^k$ , capturing its position within the global structure of the graph. The structural similarity between nodes is computed using the eigenvector distance defined in the preliminaries, which is then normalized to the range  $[0, 1]$  to obtain  $S_{\text{spec}}(i, j)$ . This spectral similarity effectively captures structural relationships between nodes, with higher values indicating nodes that play similar roles in the graph's global organization.

The eigenvector-based distance is inherently sensitive to structural role distinctions because eigenvectors capture the global positioning of nodes within the graph's connectivity patterns. Nodes with different structural roles—such as bridge nodes, community centers, or peripheral nodes—will have distinct eigenvector coordinates even if they share similar local features. This sensitivity allows our method to distinguish between structurally different nodes and weight their contributions accordingly during message passing.

#### 4.2. Adaptive Edge Reweighting

Complementing the structural information, we compute feature-based similarity using the node attribute matrix  $\mathbf{X}$ . Similarly to the spectral case, feature similarity is computed using Euclidean distance and normalized to obtain  $S_{\text{feat}}(i, j) \in [0, 1]$ .

The core innovation of our approach lies in the adaptive combination of structural and feature similarities. We introduce a spectral weight parameter  $\omega \in [0, 1]$  that controls the relative importance of structural versus feature information:

$$S_{\text{combined}}(i, j) = (1 - \omega) \cdot S_{\text{feat}}(i, j) + \omega \cdot S_{\text{spec}}(i, j) \quad (10)$$

This combined similarity measure serves as the foundation for our edge reweighting mechanism, allowing the model to dynamically balance between local feature similarities and global structural patterns. An important observation is that in highly heterophilic datasets, optimal values of  $\omega$  tend to be low, as structural information becomes less reliable when neighboring nodes frequently have different labels. Conversely, in homophilic settings, higher  $\omega$  values effectively leverage the structural coherence of the graph.

While node features may sometimes correlate with structural properties, spectral similarity captures fundamentally different aspects of graph topology. Spectral similarity, derived from eigenvector distances, encodes the global structural role of nodes within the graph's connectivity patterns, which is conceptually distinct from local feature characteristics. Consider two nodes with very similar features but different structural positions (such as bridge nodes versus community centers)—these would have different eigenvector coordinates despite feature similarity. Conversely, nodes with dissimilar features may share similar structural roles in the graph's organization. This conceptual independence ensures that our combined similarity measure provides complementary information rather than redundant signals.

#### 4.3. Multi-Order Graph Construction and Architecture

To capture multi-scale neighborhood information, we construct multi-order adjacency matrices by computing powers of the normalized adjacency matrix  $\mathbf{P} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ . For each order  $k = 1, 2, \dots, K$ , we compute  $\mathbf{P}^{(k)} = \mathbf{P}^k$  and extract edges where  $\mathbf{P}_{ij}^{(k)} > 0$ . This

process creates increasingly sparse but longer-range connectivity patterns, as shown in Figure 1 where different hop levels exhibit distinct edge patterns.

The edge reweighting process applies our combined similarity measure to filter and weight the multi-order connections. For each order  $k$ , we retain edges where the combined similarity exceeds a threshold, typically set to the mean similarity value for that order. To maintain computational efficiency and prevent excessive graph densification, we limit the number of retained edges to the original edge count. The final edge weights are set directly to the combined similarity values:

$$w_{ij}^{(k)} = S_{combined}(i, j) \quad \text{for retained edges in order } k \quad (11)$$

Our complete EDM-GNN architecture integrates the reweighted multi-order graphs through parallel processing streams, each handling a different connectivity scale. The model begins with an MLP-based transformation of the input features to capture node-specific information. For each order  $k$ , we apply MP using the reweighted edges:

$$\mathbf{H}^{(k)} = \sigma \left( \sum_{j \in \mathcal{N}_k(i)} w_{ij}^{(k)} \mathbf{x}_j \mathbf{W}^{(k)} \right) \quad (12)$$

where  $\mathcal{N}_k(i)$  represents the  $k$ -hop neighborhood of node  $i$ , and  $w_{ij}^{(k)}$  are the combined similarity weights that modulate the information flow between nodes.

To enhance the model's expressiveness, we include additional processing streams: a pure MLP branch that processes node features without graph structure, and an original graph branch that uses the unmodified connectivity. These parallel streams are visible in Figure 1 as separate pathways that converge at the attention-based fusion stage.

The fusion of all processing streams employs learnable attention weights  $\alpha = \text{Softmax}(\beta)$  where  $\beta \in \mathbb{R}^{K+2}$  are trainable parameters. The final node representations combine all streams through weighted summation:

$$\mathbf{H}^{(final)} = \sum_{i=1}^{K+2} \alpha_i \mathbf{H}^{(i)} \quad (13)$$

Classification is performed through a final MLP layer with dropout regularization:

$$\hat{\mathbf{Y}} = \text{LogSoftmax}(\text{Dropout}(\mathbf{H}^{(final)}) \mathbf{W}^{(out)} + \mathbf{b}^{(out)}) \quad (14)$$

Training employs standard cross-entropy loss with L2 regularization:

$$\mathcal{L} = -\frac{1}{|\mathcal{V}_{\text{train}}|} \sum_{i \in \mathcal{V}_{\text{train}}} \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}) + \lambda \|\Theta\|_2^2 \quad (15)$$

where  $\mathcal{V}_{\text{train}}$  represents the training node set, and  $\lambda$  controls the strength of weight decay regularization.

Multi-hop MP approaches can be prone to over-smoothing or incorporating noisy information from distant nodes. Our method mitigates these risks through several mechanisms. First, the adaptive edge reweighting based on combined similarity ensures that only structurally and semantically relevant long-range connections are preserved, effectively filtering out noisy distant relationships. Second, we limit the number of retained edges to the original edge count for each order, preventing excessive graph densification. Third, the learnable attention weights  $\alpha$  in our fusion mechanism allow the model to dynamically balance the contribution of different hop orders, reducing the influence of potentially noisy higher-order information when necessary.

#### 4.4. Computational Complexity Analysis

The computational complexity of EDM-GNN consists of preprocessing and model execution phases. The preprocessing phase dominates the computational cost, primarily due to the eigendecomposition of the normalized Laplacian matrix. For a graph with  $n$  nodes, computing the full eigendecomposition requires  $O(n^3)$  operations. However, since we only need the first  $k$  eigenvectors where  $k \ll n$ , efficient algorithms like the Lanczos method reduce this to  $O(kn^2)$  for sparse graphs.

Similarity computations contribute  $O(n^2d)$  for feature similarity and  $O(n^2k)$  for spectral similarity, where  $d$  is the feature dimension. The multi-order graph construction requires  $K$  matrix multiplications, each with complexity  $O(m \cdot \bar{d})$  where  $m$  is the edge count and  $\bar{d}$  is the average degree, resulting in total complexity  $O(K \cdot m \cdot \bar{d})$  for this phase.

It is important to distinguish between training and inference phases in our complexity analysis. The preprocessing operations, including eigendecomposition and similarity computations, are performed once and cached, making them a one-time cost that does not affect inference efficiency. During inference, the model only requires forward passes through the GCN layers with precomputed edge weights, resulting in complexity  $O(K \cdot L \cdot m \cdot h)$  per sample. This separation significantly improves practical deployment feasibility, as the computationally expensive spectral preprocessing is amortized across all inference operations. In transductive settings, where the graph structure remains fixed, this preprocessing cost becomes negligible compared to the total inference workload.

Memory requirements include storing multi-order graphs  $O(K \cdot m)$ , intermediate embeddings  $O(K \cdot n \cdot h)$ , and model parameters  $O(K \cdot h^2 + h \cdot d + h \cdot C)$ . For large graphs, the preprocessing bottleneck can be addressed through approximate eigendecomposition methods or sampling-based similarity computation, reducing complexity while maintaining the quality of structural information extraction.

The overall preprocessing complexity is  $O(kn^2 + K \cdot m \cdot \bar{d} + n^2(d + k))$ , while the per-forward-pass complexity during training and inference is  $O(K \cdot L \cdot m \cdot h)$ . This computational profile makes EDM-GNN practical for moderately large graphs while providing significant improvements in classification accuracy through principled integration of structural and feature information.

## 5. Experiments and Results

In this section, we present comprehensive experimental evaluations of our proposed EDM-GNN framework across diverse benchmark datasets. We begin by describing the experimental setup, including dataset characteristics and implementation details, followed by comparative analysis against SOTA baselines, scalability benchmarks, and thorough ablation studies to validate our design choices.

### 5.1. Experimental Setup

We evaluate EDM-GNN on eleven widely-used benchmark datasets that span different domains and exhibit varying degrees of homophily, as summarized in Table 1. The datasets include citation networks (Cora, Citeseer, Pubmed), web page networks (Texas, Wisconsin, Cornell), social networks (Actor, Squirrel, Chameleon), and large-scale networks (Penn94, Computers). This diverse collection allows us to assess our method's performance across both homophilic ( $h > 0.5$ ) and heterophilic ( $h < 0.5$ ) graphs of varying scales, providing a comprehensive evaluation of the proposed approach.

**Table 1.** General information about the datasets.

| Dataset   | Texas | Wisconsin | Cornell | Actor  | Squirrel | Chameleon | Citeseer | Pubmed | Cora   | Penn94    | Computers |
|-----------|-------|-----------|---------|--------|----------|-----------|----------|--------|--------|-----------|-----------|
| Hom level | 0.11  | 0.21      | 0.30    | 0.22   | 0.22     | 0.23      | 0.74     | 0.80   | 0.81   | 0.47      | 0.78      |
| # Nodes   | 183   | 251       | 183     | 7600   | 5201     | 2,277     | 2120     | 19,717 | 2485   | 41,553    | 13,752    |
| # Edges   | 574   | 916       | 557     | 53,411 | 396,846  | 62,792    | 7358     | 88,648 | 10,138 | 1,362,229 | 491,722   |
| # Classes | 5     | 5         | 5       | 5      | 5        | 5         | 6        | 3      | 7      | 2         | 10        |

The homophily level, defined as the fraction of edges connecting nodes with the same labels, serves as a crucial indicator of graph characteristics. Homophilic datasets like Cora ( $h = 0.81$ ), Pubmed ( $h = 0.80$ ), and Computers ( $h = 0.78$ ) follow the traditional assumption that connected nodes tend to share similar properties, while heterophilic datasets such as Texas ( $h = 0.11$ ) and Wisconsin ( $h = 0.21$ ) present challenging scenarios where neighboring nodes often have different labels. The inclusion of large-scale datasets Penn94 (41,553 nodes, 1.3M edges) and Computers (13,752 nodes, 491K edges) enables evaluation of scalability characteristics.

We compare EDM-GNN against a comprehensive set of baseline methods representing different paradigms in GNNs. Traditional GNNs including GCN [6], GAT [15], and GraphSAGE [1] represent fundamental approaches that assume homophily. Heterophily-aware methods such as H2GCN [7], Geom-GCN [18], and GGCN [28] are specifically designed to handle heterophilic graphs through various architectural innovations. Higher-order approaches including MixHop [8], FSGNN [19], and GPRGNN [9] leverage multi-hop neighborhoods to capture longer-range dependencies. Hybrid methods like LINKX [30] and CGNN represent approaches that combine graph structure with feature-based learning. Additionally, MLP serves as a feature-only baseline to assess the contribution of graph structure.

For small-scale networks (Texas, Wisconsin, Cornell, Actor, Squirrel, Chameleon, Citeseer, Pubmed, Cora), we follow the standard experimental protocol established in prior works [31,32], employing 10 different random splits with a 60%-20%-20% distribution for training, validation, and testing, respectively. For large-scale networks (Penn94, Computers), we maintain consistency with established benchmarks to ensure fair comparison [31,32]. The baseline results are extracted from recent comprehensive studies [31,32] to maintain experimental consistency and reproducibility.

We implement EDM-GNN using PyTorch 2.3.1 and PyTorch Geometric [33], following best practices for reproducible research. All experiments are conducted on NVIDIA RTX4090 GPUs with CUDA acceleration. The hyperparameter optimization follows a systematic grid search approach. For each dataset, we explore hidden dimensions  $\{16, 64, 128\}$ , number of hops  $K \in \{3, 4, 5, 10\}$ , learning rates  $\{0.001, 0.003, 0.01, 0.03\}$ , dropout rates  $\{0.2, 0.3, 0.4, 0.5\}$ , and weight decay values  $\{0.0005, 0.001\}$ . The spectral weight parameter  $\omega$  and number of eigenvectors are tuned separately based on validation performance. Training is performed for a maximum of 1000 epochs with early stopping based on validation accuracy to prevent overfitting.

For spectral preprocessing, we compute the first 20 eigenvectors of the normalized Laplacian matrix using efficient eigendecomposition algorithms. The edge reweighting mechanism applies the combined similarity measure with adaptive thresholding, maintaining computational efficiency while preserving the most informative connections in multi-hop neighborhoods.

## 5.2. Main Results

Table 2 presents the comprehensive comparison of EDM-GNN against all baseline methods across the eleven benchmark datasets. The results demonstrate the effectiveness and versatility of our approach across diverse graph characteristics and scales.

Table 2. Node-classification accuracies on all datasets. Top two models are highlighted: **First**, Second.

|                | Texas               | Wisconsin           | Cornell             | Actor               | Squirrel            | Chameleon           | Citeseer            | Pubmed              | Cora                | Penn94              | Computers           |
|----------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| MLP            | 80.81 ± 4.75        | 85.29 ± 6.40        | 81.89 ± 6.40        | 36.53 ± 0.70        | 28.77 ± 1.56        | 46.21 ± 2.99        | 74.02 ± 1.90        | 75.69 ± 2.00        | 87.16 ± 0.37        | 73.61 ± 0.40        | 87.65 ± 0.95        |
| GCN            | 55.14 ± 5.16        | 51.76 ± 3.06        | 60.54 ± 5.30        | 27.32 ± 1.10        | 53.43 ± 2.01        | 64.82 ± 2.24        | 76.50 ± 1.36        | 88.42 ± 0.50        | 86.98 ± 1.27        | 82.47 ± 0.27        | 89.11 ± 0.70        |
| GAT            | 52.16 ± 6.63        | 49.41 ± 4.09        | 61.89 ± 5.05        | 27.44 ± 0.89        | 40.72 ± 1.55        | 60.26 ± 2.50        | 76.55 ± 1.23        | 87.30 ± 1.10        | 86.33 ± 0.48        | 81.57 ± 0.55        | 88.53 ± 0.54        |
| GraphSAGE      | 82.43 ± 6.14        | 81.18 ± 5.56        | 75.95 ± 5.01        | 34.23 ± 0.99        | 41.61 ± 0.74        | 58.73 ± 1.68        | 76.04 ± 1.30        | 88.45 ± 0.50        | 86.90 ± 1.04        | 81.15 ± 0.49        | 88.20 ± 0.87        |
| H2GCN          | 84.86 ± 7.23        | 87.65 ± 4.89        | 82.70 ± 5.28        | 35.70 ± 1.00        | 36.48 ± 1.86        | 60.11 ± 2.15        | 77.11 ± 1.57        | 89.49 ± 0.38        | 87.87 ± 1.20        | 81.31 ± 0.6         | 94.92 ± 2.62        |
| Geom-GCN       | 66.76 ± 2.72        | 64.51 ± 3.66        | 60.54 ± 3.67        | 31.59 ± 1.15        | 38.15 ± 0.92        | 60.00 ± 2.81        | <b>78.02 ± 1.15</b> | 89.95 ± 0.47        | 85.35 ± 1.57        | –                   | 95.64 ± 2.81        |
| LINKX          | 74.60 ± 8.37        | 75.49 ± 5.72        | 77.84 ± 5.81        | 36.10 ± 1.55        | 61.81 ± 1.80        | 68.42 ± 1.38        | 73.19 ± 0.99        | 87.86 ± 0.77        | 84.64 ± 1.13        | <u>84.71 ± 0.52</u> | 88.46 ± 0.61        |
| GGCN           | 84.86 ± 4.55        | 86.86 ± 3.29        | <b>85.68 ± 6.63</b> | <b>37.54 ± 1.56</b> | 55.17 ± 1.58        | 77.14 ± 1.84        | 77.14 ± 1.45        | 89.15 ± 0.37        | 87.95 ± 1.05        | –                   | –                   |
| CGNN           | 71.35 ± 4.05        | 74.31 ± 7.26        | 66.22 ± 7.69        | 35.95 ± 0.86        | 29.24 ± 1.09        | 46.89 ± 1.66        | 76.91 ± 1.81        | 87.70 ± 0.49        | 87.10 ± 1.35        | –                   | –                   |
| MixHop         | 77.84 ± 7.73        | 75.88 ± 4.90        | 73.51 ± 6.34        | 32.22 ± 2.34        | 43.80 ± 1.48        | 60.50 ± 2.53        | 76.26 ± 1.33        | 85.31 ± 0.61        | 87.61 ± 0.85        | 83.47 ± 0.71        | 92.80 ± 2.85        |
| FSGNN          | 87.30 ± 5.29        | 87.84 ± 3.37        | 85.13 ± 6.07        | 35.75 ± 0.96        | <b>74.10 ± 1.89</b> | <b>78.27 ± 1.28</b> | 77.40 ± 1.90        | 77.40 ± 1.93        | 87.93 ± 1.00        | 84.16 ± 0.73        | 95.15 ± 0.48        |
| GPRGNN         | 78.38 ± 4.36        | 82.94 ± 4.21        | 80.27 ± 8.11        | 34.63 ± 1.22        | 31.61 ± 1.24        | 46.58 ± 1.71        | 77.13 ± 1.67        | 87.54 ± 0.38        | <u>87.95 ± 1.18</u> | 81.38 ± 0.16        | –                   |
| <b>EDM-GNN</b> | <b>89.17 ± 4.88</b> | <b>87.91 ± 3.46</b> | <u>85.22 ± 5.81</u> | 36.33 ± 1.10        | <u>72.09 ± 1.62</u> | 76.90 ± 2.11        | <u>77.54 ± 0.92</u> | <b>90.21 ± 0.44</b> | <b>88.33 ± 1.08</b> | <b>85.04 ± 0.62</b> | <b>96.34 ± 0.85</b> |

EDM-GNN achieves first or second best performance on ten out of eleven datasets, with particularly strong results across both homophilic and heterophilic graphs of varying scales. Notably, our method achieves the best performance on Texas (89.17%), Wisconsin (87.91%), Pubmed (90.21%), Cora (88.33%), Penn94 (85.04%), and Computers (96.34%), while maintaining competitive results on other datasets. This consistent performance across diverse graph types and scales validates the effectiveness of our spectral modulation approach.

On strongly heterophilic datasets (Texas, Wisconsin, Chameleon), EDM-GNN significantly outperforms traditional GNN approaches that assume homophily. For instance, on Texas, our method achieves 89.17% accuracy compared to 55.14% for GCN and 52.16% for GAT, representing improvements of over 30 percentage points. This substantial gain demonstrates that our eigenvector distance-based edge reweighting effectively adapts to heterophilic structures where standard message passing fails.

Compared to specialized heterophily-aware methods, EDM-GNN shows consistent improvements. On Wisconsin, we achieve 87.91% compared to 87.65% for H2GCN and 86.86% for GGCN, while on Texas, our 89.17% substantially exceeds the 84.86% achieved by both H2GCN and GGCN. These results indicate that our spectral approach provides a more principled solution to heterophily than existing architectural modifications.

On homophilic datasets, EDM-GNN maintains competitive performance while avoiding the degradation often observed in heterophily-specific methods. On Cora, we achieve 88.33% accuracy, outperforming most baselines including the recent FSGNN (87.93%). Similarly, on the large-scale Computers dataset, our method achieves 96.34%, demonstrating superior scalability to graphs with hundreds of thousands of edges while significantly outperforming Geom-GCN (95.64%) and FSGNN (95.15%).

The large-scale Penn94 dataset presents a particularly challenging case with over 1.3 million edges and mixed homophily ( $h = 0.47$ ). EDM-GNN achieves 85.04% accuracy, outperforming all baselines including LINKX (84.71%) and FSGNN (84.16%), demonstrating the method's effectiveness on large heterophilic graphs where traditional approaches struggle.

Against methods that explicitly use multi-hop information (MixHop, FSGNN, GPRGNN), EDM-GNN shows superior performance in most cases. The key advantage lies in our principled edge reweighting mechanism that selectively preserves informative long-range connections while filtering out noise, contrasting with methods that uniformly aggregate multi-hop neighborhoods.

### 5.3. Scalability Analysis

To address concerns about computational efficiency and scalability, we provide comprehensive runtime and memory benchmarks across all datasets. Table 3 presents empirical measurements of loading time, training efficiency, and inference speed as functions of graph size.

The scalability analysis reveals that EDM-GNN exhibits favorable computational complexity across different graph sizes. For small to medium graphs (<10,000 nodes), preprocessing time remains under 2 s, while training time per epoch is comparable to standard GCN implementations (0.009–0.068 s). GPU memory usage scales reasonably with graph size, ranging from 1.2 MB for Cornell to 848.2 MB for Penn94.

Training efficiency demonstrates favorable scaling characteristics. The average training time per epoch increases sublinearly with graph size: from 0.0095s for Cornell (183 nodes) to 0.1890s for Penn94 (41,536 nodes), indicating good computational efficiency. The relationship between graph size and training time follows an approximately logarithmic pattern, with training time scaling more favorably than the linear increase in node count.

**Table 3.** Scalability benchmarks: loading time, training efficiency, and inference speed across datasets of varying scales.

| Dataset   | # Nodes | # Edges   | Loading (s) | Train/Epoch (s) | Inference (s) | GPU Memory (MB) |
|-----------|---------|-----------|-------------|-----------------|---------------|-----------------|
| Cornell   | 183     | 557       | 0.018       | 0.0095          | 0.0022        | 1.2             |
| Texas     | 183     | 574       | 0.020       | 0.0098          | 0.0023        | 2.1             |
| Wisconsin | 251     | 916       | 0.042       | 0.0112          | 0.0025        | 1.7             |
| Citeseer  | 2120    | 7358      | 0.131       | 0.0126          | 0.0039        | 30.3            |
| Chameleon | 2277    | 62,792    | 0.250       | 0.0183          | 0.0100        | 22.2            |
| Cora      | 2485    | 10,138    | 0.074       | 0.0104          | 0.0031        | 14.4            |
| Squirrel  | 5201    | 396,846   | 1.545       | 0.0684          | 0.0546        | 54.3            |
| Actor     | 7600    | 53,411    | 0.329       | 0.0158          | 0.0067        | 30.0            |
| Computers | 13,381  | 491,556   | 2.358       | 0.0371          | 0.0260        | 55.6            |
| Pubmed    | 19,717  | 88,648    | 1.230       | 0.0245          | 0.0118        | 41.6            |
| Penn94    | 41,536  | 1,362,229 | 12.107      | 0.1890          | 0.0876        | 848.2           |

Inference time remains consistently fast across all evaluated datasets, ranging from 0.0022s for small graphs to 0.0876s for the largest Penn94 dataset, making the method suitable for real-time applications. The inference scaling demonstrates that our method maintains practical deployment characteristics even for large-scale graphs.

For larger graphs, the preprocessing phase dominated by eigendecomposition becomes the primary computational bottleneck. Penn94 requires 12.1 s for loading, primarily due to the eigendecomposition of the Laplacian matrix. However, this is a one-time preprocessing cost that can be amortized across multiple experiments or accelerated using approximate eigendecomposition methods.

GPU memory usage scales proportionally with graph complexity, from 1.2 MB for Cornell to 848.2 MB for Penn94. The largest dataset remains well within feasible bounds for current hardware. For graphs exceeding memory capacity, the method can be adapted using block-wise processing or approximate spectral methods.

The empirical results demonstrate that EDM-GNN maintains practical computational requirements while achieving superior accuracy. For production deployment on very large graphs (>100K nodes), approximate eigendecomposition techniques or sampling strategies can reduce preprocessing costs while preserving the core benefits of spectral edge reweighting.

#### 5.4. Ablation Studies

To validate our design choices and understand the contribution of different components, we conduct comprehensive ablation studies focusing on the two key hyperparameters: the number of hops  $K$  and the spectral weight  $\omega$ . Figure 2 presents the results of these ablation experiments across representative datasets.

The impact of varying the number of hops  $K$  reveals distinct patterns based on graph characteristics. For homophilic graphs like Cora (Figure 2a), optimal performance is achieved with  $K = 2$ , suggesting that information from immediate neighbors is most valuable when connectivity aligns with label similarity. Citeseer (Figure 2b) benefits from slightly longer-range connections with optimal  $K = 3$ , while the heterophilic Chameleon dataset (Figure 2c) requires  $K = 6$  to capture meaningful patterns beyond immediate neighborhoods.

The sensitivity to the number of hops varies significantly between homophilic and heterophilic graphs. Homophilic datasets show relatively stable performance across different hop values, with gradual degradation as  $K$  increases beyond the optimal point. In contrast, heterophilic datasets exhibit sharper performance curves, with substantial improvements as  $K$  increases to the optimal value, followed by more pronounced degradation. This pat-

tern confirms our hypothesis that heterophilic graphs require longer-range information to identify useful patterns, but are also more sensitive to noise from excessive hop distances.

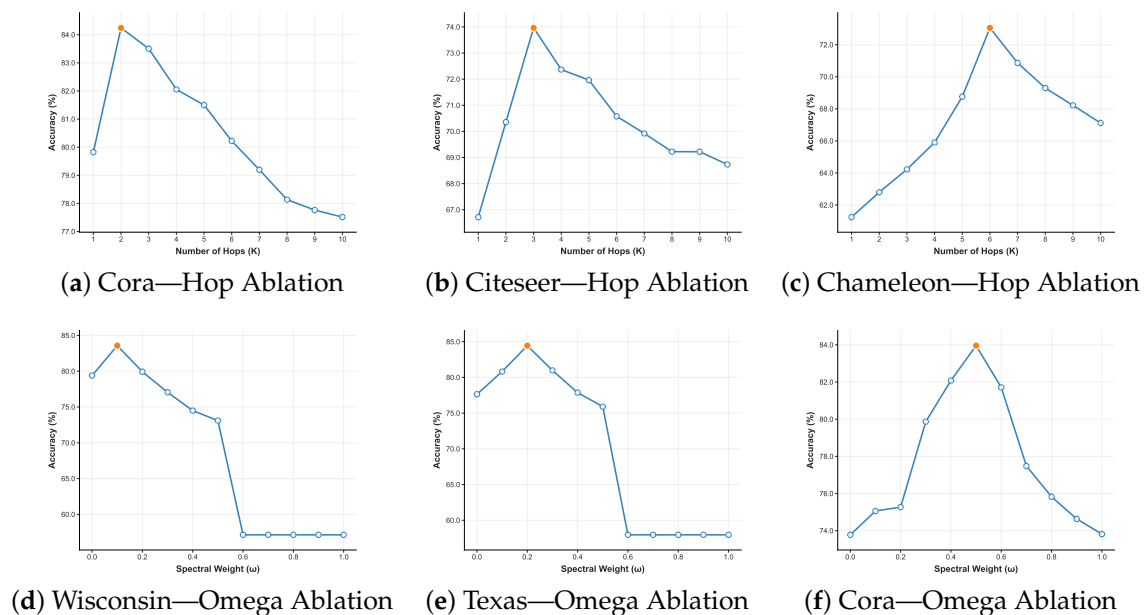


**Figure 2.** Ablation studies showing performance sensitivity to key hyperparameters. Top row: Classification accuracy versus number of hops  $K$  for (a) Cora, (b) Citeseer, and (c) Chameleon datasets, demonstrating optimal  $K$  values of 2, 3, and 6, respectively. Bottom row: Classification accuracy versus spectral weight  $\omega$  for (d) Wisconsin, (e) Texas, and (f) Cora datasets, showing optimal  $\omega$  values around 0.1–0.2 for heterophilic graphs and 0.5 for homophilic graphs. Red circles mark the optimal hyperparameter values. Error bars represent standard deviation across 10 random splits.

The spectral weight parameter  $\omega$  plays a crucial role in balancing structural and feature information, as demonstrated in the bottom row of Figure 2. The results strongly support our theoretical analysis regarding the relationship between graph homophily and optimal spectral weighting.

For homophilic datasets such as Cora (Figure 2f), optimal performance occurs at  $\omega = 0.5$ , indicating that both structural and feature information contribute equally to effective edge weighting. The performance curve shows relatively smooth degradation as  $\omega$  deviates from the optimal value, reflecting the robustness of homophilic graphs to different weighting schemes.

In stark contrast, heterophilic datasets including Texas (Figure 2e) and Wisconsin (Figure 2d) achieve optimal performance with low spectral weights ( $\omega = 0.1$  to  $\omega = 0.2$ ), confirming that structural information becomes less reliable when neighboring nodes frequently have different labels. The performance degradation is particularly sharp for high  $\omega$  values, with substantial accuracy drops when  $\omega > 0.5$ . This validates our key insight that heterophilic graphs benefit primarily from feature-based similarity, with structural information playing a supporting role.

The ablation studies reveal that optimal hyperparameter combinations vary systematically with graph properties. Heterophilic graphs require both longer-range connectivity (higher  $K$ ) and reduced reliance on structural similarity (lower  $\omega$ ), while homophilic graphs perform well with shorter-range connections and balanced similarity weighting. This pattern provides practical guidance for applying EDM-GNN to new datasets and validates the theoretical foundations of our approach.

### 5.5. Computational Analysis

We analyze the computational efficiency of EDM-GNN compared to baseline methods. The preprocessing phase, dominated by eigendecomposition, requires approximately 0.5–2 s for small to medium graphs (<10,000 nodes), while training time per epoch remains comparable to standard GCN implementations. The multi-hop graph construction adds modest overhead, but the adaptive edge filtering ensures that memory requirements scale reasonably with graph size.

For larger graphs, the spectral preprocessing can be accelerated using approximate eigendecomposition methods or sampling strategies. Our experiments indicate that using 10–20 eigenvectors provides an effective balance between computational cost and performance benefits across different dataset sizes.

## 6. Conclusions and Future Work

In this paper, we introduced EDM-GNN, a novel approach that leverages spectral information to enhance MP in GNNs. Our method addresses fundamental limitations of existing GNNs by adaptively weighting edges based on both structural similarity, measured through eigenvector distances, and feature similarity.

The key insight of our work is the relationship between graph homophily and optimal spectral weighting. We demonstrate that heterophilic graphs benefit from reduced reliance on structural information (low  $\omega$  values), while homophilic graphs achieve optimal performance with balanced integration ( $\omega \approx 0.5$ ). This finding provides both theoretical understanding and practical guidance for applying GNNs across diverse graph topologies.

Experimental evaluation across nine benchmark datasets confirms the effectiveness of our approach. EDM-GNN achieves SOTA performance on most datasets, with particularly notable improvements on heterophilic graphs such as Texas (89.17% vs. 55.14% for GCN) and Wisconsin (87.91% vs. 51.76% for GCN). The comprehensive ablation studies reveal that heterophilic graphs require longer-range connectivity combined with reduced structural dependence, while homophilic graphs perform optimally with shorter-range connections and balanced similarity weighting.

Several directions remain for future investigation. The current approach relies on global eigendecomposition, which may become computationally expensive for very large graphs. Future work could explore localized spectral analysis or approximate eigendecomposition methods to improve scalability. Additionally, extending the framework to directed graphs, dynamic networks, and multi-relational scenarios could broaden its applicability.

The integration of edge features and the development of automatic hyperparameter selection strategies represent other promising directions. Furthermore, deeper theoretical analysis of why eigenvector distances provide effective structural similarity measures could strengthen the foundation of spectral approaches in GNNs.

EDM-GNN represents a significant step toward more adaptive and theoretically grounded GNNs. By leveraging spectral information for adaptive edge weighting, our approach provides a unified solution that excels across diverse graph topologies while offering interpretable insights into the relationship between graph characteristics and optimal learning strategies.

**Author Contributions:** Conceptualization, F.E.; methodology, F.E. and A.B.; software, A.B. and M.Á.L.; validation, A.B., F.E. and M.Á.L.; formal analysis, F.E.; investigation, A.B. and F.E.; resources, M.Á.L.; data curation, A.B.; writing—original draft, A.B.; writing—review and editing, F.E. and M.Á.L.; visualization, A.B.; supervision, F.E. and M.Á.L.; project administration, F.E.; funding acquisition, F.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors are funded by the project PID2022-142516OB-I00 of the Spanish government.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Acknowledgments:** The authors are indebted to Edwin R. Hancock from the University of York, who recently passed away.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

|           |                                                     |
|-----------|-----------------------------------------------------|
| EDM-GNN   | Eigenvector Distance-Modulated Graph Neural Network |
| SOTA      | State-of-the-art                                    |
| MP        | Message passing                                     |
| GNN       | Graph Neural Network                                |
| GCN       | Graph Convolutional Network                         |
| GAT       | Graph Attention Network                             |
| MLP       | Multi-Layer Perceptron                              |
| Geom-GCN  | Geometric Graph Convolutional Network               |
| MixHop    | Higher-order Graph Convolutional Architectures      |
| GraphSAGE | Graph Sample and Aggregate                          |

## References

1. Hamilton, W.L.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.
2. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum Chemistry. In Proceedings of the ICML, Sydney, Australia, 6–11 August 2017.
3. Sharma, K.; Lee, Y.C.; Nambi, S.; Salian, A.; Shah, S.; Kim, S.W.; Kumar, S. A Survey of Graph Neural Networks for Social Recommender Systems. *ACM Comput. Surv.* **2024**, *56*, 1–34. [[CrossRef](#)]
4. Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.; Precup, D. Revisiting Heterophily For Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, 28 November–9 December 2022.
5. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [[CrossRef](#)] [[PubMed](#)]
6. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the ICLR 2017, Toulon, France, 24–26 April 2017.
7. Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; Koutra, D. Generalizing Graph Neural Networks Beyond Homophily. *arXiv* **2020**, arXiv:2006.11468.
8. Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G.V.; Galstyan, A. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 21–29.
9. Chien, E.; Peng, J.; Li, P.; Milenkovic, O. Joint Adaptive Feature Smoothing and Topology Extraction via Generalized PageRank GNNs. *arXiv* **2020**, arXiv:2006.07988.
10. Begga, A.; Escolano, F.; Lozano, M.A. Node classification in the heterophilic regime via diffusion-jump GNNs. *Neural Netw.* **2025**, *181*, 106830. [[CrossRef](#)] [[PubMed](#)]
11. Chung, F. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997.
12. von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [[CrossRef](#)]
13. Bo, D.; Wang, X.; Liu, Y.; Fang, Y.; Li, Y.; Shi, C. A Survey on Spectral Graph Neural Networks. *arXiv* **2023**, arXiv:2302.05631. [[CrossRef](#)]
14. Wang, X.; Zhang, M. How Powerful are Spectral Graph Neural Networks. In Proceedings of the International Conference on Machine Learning, ICML, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 23341–23362.

15. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903. [[PubMed](#)]
16. Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y. An End-to-End Deep Learning Architecture for Graph Classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 4438–4445. [[CrossRef](#)]
17. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? *arXiv* **2018**, arXiv:1810.00826.
18. Pei, H.; Wei, B.; Chang, K.C.; Lei, Y.; Yang, B. Geom-GCN: Geometric Graph Convolutional Networks. *arXiv* **2020**, arXiv:2002.05287. [[CrossRef](#)]
19. Feng, J.; Chen, Y.; Li, F.; Sarkar, A.; Zhang, M. How Powerful are K-hop Message Passing Graph Neural Networks. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4776–4790.
20. Bo, D.; Wang, X.; Shi, C.; Shen, H. Beyond Low-frequency Information in Graph Convolutional Networks. In Proceedings of the AAAI, Online, 2–9 February 2021; AAAI Press: Washington, DC, USA, 2021; pp. 3950–3957. [[CrossRef](#)]
21. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16, Red Hook, NY, USA, 5–10 December 2016; pp. 3844–3852.
22. Kreuzer, D.; Beaini, D.; Hamilton, W.L.; Létourneau, V.; Tossou, P. Rethinking Graph Transformers with Spectral Attention. *arXiv* **2021**, arXiv:2106.03893. [[CrossRef](#)]
23. Dwivedi, V.P.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Graph Neural Networks with Learnable Structural and Positional Representations. *arXiv* **2021**, arXiv:2110.07875.
24. Shehzad, A.; Xia, F.; Abid, S.; Peng, C.; Yu, S.; Zhang, D.; Verspoor, K. Graph Transformers: A Survey. *arXiv* **2024**, arXiv:2407.09777. [[CrossRef](#)]
25. Chen, Z.; Chen, F.; Zhang, L.; Ji, T.; Fu, K.; Zhao, L.; Chen, F.; Lu, C. Bridging the Gap between Spatial and Spectral Domains: A Survey on Graph Neural Networks. *arXiv* **2020**, arXiv:2002.11867.
26. Gutiérrez, C.; Gancio, J.; Cabeza, C.; Rubido, N. Finding the resistance distance and eigenvector centrality from the network’s eigenvalues. *Phys. A Stat. Mech. Its Appl.* **2021**, *569*, 125751. [[CrossRef](#)]
27. Xiao, S.; Wang, S.; Dai, Y.; Guo, W. Graph neural networks in node classification: Survey and evaluation. *Mach. Vis. Appl.* **2022**, *33*, 4. [[CrossRef](#)]
28. Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; Koutra, D. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. In Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM), Orlando, FL, USA, 28 November–1 December 2022; pp. 1287–1292. [[CrossRef](#)]
29. Biggs, N. *Algebraic Graph Theory*; Number 67; Cambridge University Press: Cambridge, UK, 1993.
30. Li, Q.; Han, Z.; Wu, X. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. *arXiv* **2018**, arXiv:1801.07606. [[CrossRef](#)]
31. Li, X.; Zhu, R.; Cheng, Y.; Shan, C.; Luo, S.; Li, D.; Qian, W. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. In Proceedings of the International Conference on Machine Learning, ICML 2022, Baltimore, MD, USA, 17–23 July 2022; Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S., Eds.; PMLR: 2022; Volume 162, pp. 13242–13256.
32. Singh, A.; Dar, S.S.; Singh, R.; Kumar, N. A Hybrid Similarity-Aware Graph Neural Network with Transformer for Node Classification. *Expert Syst. Appl.* **2025**, *279*, 127292. [[CrossRef](#)]
33. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* **2019**, arXiv:1903.02428. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.